Beyond Code: An Introduction to Model-Driven Software Development (CISC 836, Fall 2021)

Home page

Teaching staff

Instructor: Juergen Dingel, Goodwin Hall 723, dingel@cs, office hours: tba

Time and place

The course will be held in person. Class times are

- Tuesdays, 10:30am - noon
- Thursdays, 12:30 - 2pm

Location is Jeffrey Hall, Room 110. First meeting is Tuesday, Sept 7, 2021.

Audience

This course is suitable for students with an interest in the theory and practise of software development in general and the use of models in software development in particular.

General description

Models are pervasive in engineering. The impact of models on the practise of software engineering, however, has been relatively limited and pales in comparison to the pivotal roles models play in other engineering disciplines. The idea of making models a more prominent artifact in everyday software development has been enjoying increasing support in academia and industry and many approaches, tools and standards have been proposed. Indeed, in some industries (e.g., automotive) development already revolves around the use of models.

Model-driven software development (MDSD) uses high-level, possibly graphical and domain-specific notations in an attempt to capture relevant artifacts on the most appropriate level of abstraction, facilitate the reliable expression and transfer of design intent, and increase the degree of automation. The course presents key ideas and potential benefits and challenges of software modeling in general and of MDSD in particular. Specific attention will be paid to the use of MDSD for event-based, reactive systems. At the end of the course, students will be familiar with the state of the art in software modeling and have gained some critical understanding of the theory and practice involving the use of models for software development.

Learning outcomes

The goal of the course is to introduce students to the state of the art in software modeling and allow them to gain some critical understanding of the theory and practice involving the use of models for software development.

More specifically, at the end of the course, students will be able to

- appreciate the significance of abstraction and automation in engineering and the sciences
- understand concepts, techniques, and tools used for software modeling and MDSD
- design and develop simple reactive systems using MDSD
- specify and implement simple domain specific languages using MDSD

- understand strengths and limitations of software modeling and MDSD
- evaluate the suitability of software modeling and MDSD for specific software engineering problems

Format

The course will combine lectures, assigned readings, assignments, and a project. The course will cover the topics described on the Content page. The assigned readings will be drawn from a variety of sources and reinforce the lecture material. The projects will serve to, e.g., provide hands-on experience with a specific tool or technology.

Remote students

Although CISC 836 is officially an 'in-person' course, some limited support for remote students will be provided. More concretely, slides will be posted (on 'Schedule' page), lectures will be recorded (but since the available resources to do this will be limited, so quality may not be the best), and office hours will be held online. Also, I would be willing to consider alternative ways to contribute to the class and satisfy the 'Participation' requirement of the marking scheme.

Marking scheme

A student's overall mark will be computed as follows:

- Assignments (4): 40%
- Participation: 10%
- Paper reviews: 10%
- Project and presentation: 40%

For more on projects and how they will be evaluated go to the Projects page.

Prerequisites

Good knowledge of object-oriented programming (preferably in Java or C++).

Material

- Slides: The lecture slides will be made available on the Schedule page.
- Papers: Papers are available on the Schedule page.
- On-line material: Most of the tools discussed in the course have extensive publically accessible web pages.
- Textbooks on the topic **(not mandatory)**
  - On model-driven development
    - Marco Brambilla, Jordi Cabot, Manuel Wimmer. Model-Driven Software Engineering in Practice. 2nd Edition. Morgan and Claypool. 2017. Available as ebook (search Queen's University Library for book title)
  - On UML
    - M. Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition). Addison Wesley. 2004. Available as ebook (search Queen's University Library for book title)
    - G. Booch, J. Rumbaugh, I. Jacobson. UML User Guide. 2nd Edition. Addison Wesley. 2005. Available as ebook (search Queen's University Library for book title)
  - On DSLs, language workbenches, and software language engineering
    - M. Fowler. Domain-specific languages. O'Reilly. 2010. Available as ebook (search Queen's University Library for book title)
    - Lorenzo Bettini. Implementing Domain-Specific Languages with Xtext and Xtend. Packt Publishing. 2013. Available as ebook (search Queen's University Library for book title)
    - Markus Voelter. DSL Engineering: Designing, Implementing and Using Domain-Specific Languages. CreateSpace Independent Publishing Platform. 2013. Available online as

donationware.

Last modified: Thu Jul 08 2021 15:59:42