Description

The following course description is excerpted from the Graduate Course Catalog:

“Engineering practices of formalized models as the basis for analyzing and specifying software artifacts. Topics include key software engineering skills required for formal modeling, techniques for model building and analysis, and applications of formal modeling techniques in the requirements, design, and implementation phases of software development.”

Objectives

Model Driven Development (MDD) is an emerging but controversial topic both in academic software engineering research and in industrial practice. It shifts the focus of software development from writing code to building models, for example, by automatically generating code from models. The key assumption of MDD is the existence of an appropriate model—a representation that is sufficiently general to capture the semantics of many different domains, yet precise enough to support eventual transformation into code. This course will provide an overview of MDD and show how to apply MDD to software development in practice using UML and Java. The students are expected to gain a hands-on experience on building precise models that can be used as a basis of MDD, validating them, and transforming them to implementations by exploring recent advances in MDD and related areas, including:

- Model-Driven Architecture (MDA)
- Unified Modeling Language (UML)
- Object Constraint Language (OCL)
- Model specification, validation and transformation
- Design techniques for MDD (e.g., design patterns)
- Application development using MDD

To this end, the specific learning objectives of this course are:

- Understanding the concepts, principles, and theories of MDD and MDA
- Detailed knowledge of UML and OCL, and ability to apply them to precisely model and specify systems of moderate size
- Understanding correctness properties of models and knowledge of proof techniques which are used to check these properties
- Being able to use various UML models and other design techniques to define designs of moderate-sized systems
- Understanding different kinds of model transformations and being able to apply them to systems of moderate size
- Understanding how to generate Java implementations from models and being able to carry out such implementations for systems of moderate size
- Being able to specify, design and implement medium-sized applications using MDD and Java

Textbooks

The textbook of this course is:

Note that this textbook is out of print by the publisher and its copies may not be available from the UTEP bookstore, but you can still purchase its copies from online stores like Amazon. You are expected to acquire a copy for your use in this course, as reading assignments will be taken from the textbook.

We will also use research papers, reference manuals, and other online documents as course material. The following books are also recommended as supplementary texts:

- Any UML textbook covering UML 2.x, e.g., the following ebooks are available through UTEP library:

**Exams**

There will be one mid-term exam and a final exam. The mid-term exam will take place during the regular class session and will be 80 minutes in length.

**Assignments**

Reading and homework assignments will be handed out or announced in class. If you miss a class, it is your responsibility to find out what you missed. There will be occasional homework assignments, and most may be done in pairs unless otherwise specified. No late submission will be accepted unless arrangements have been made in advance or unless unusual circumstances warrant an exception.

If you are taking CS 5382, there is another type of assignments. You are required to read and present research papers related to the course topics. The number of presentations will be one or two depending on the class size. A list of suggested papers is found at the end of this syllabus, however you are free to choose papers for your presentations. The paper presentation is optional for CS 4390 students but will earn bonus points.

**Project (CS 5382)**

If you are taking CS 5382, you should do a small semester project. The purpose of your project is to apply course topics to your own research/project work and also to explore the current research problems or issues in the areas of the course topics. The instructor may suggest sample project topics. You are expected to write a project proposal, submit a final project report, and present the result in class. The initial proposal should be done individually, however it can be re-scoped and rescaled to make a pair project. The project is optional for CS 4390 students but will earn bonus points.

**Grading**

Your grade is independent of anyone else’s grade; that is, you are not graded on a curve. Everyone can get an A in this course. The purpose of grading is not to rank you, but to uphold a standard of quality and to give you feedback. The final letter grade will be based on a combination of assignments, project, exams, and class participation. The approximate percentages are as follows:

<table>
<thead>
<tr>
<th>Graded work</th>
<th>CS 4390</th>
<th>CS 5382</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>60%</td>
<td>40%</td>
</tr>
<tr>
<td>Exam 1</td>
<td>20%</td>
<td>15%</td>
</tr>
</tbody>
</table>
There are also up to 5% bonus points for class attendance and participation. To earn this, you must arrive at lecture on time and participate in class discussion in a constructive and prepared manner, e.g., by asking or answering questions that demonstrate that you have read and attempted to understand the material.

The nominal percentage-score-to-letter-grade conversion is as follows:

<table>
<thead>
<tr>
<th>Score (%)</th>
<th>Letter grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>90% or higher</td>
<td>A</td>
</tr>
<tr>
<td>80-89%</td>
<td>B</td>
</tr>
<tr>
<td>70-79%</td>
<td>C</td>
</tr>
<tr>
<td>60-69%</td>
<td>D</td>
</tr>
<tr>
<td>below 60%</td>
<td>F</td>
</tr>
</tbody>
</table>

The instructor reserves the right to adjust these criteria downward, e.g., so that 88% or higher represents an A, based on overall class performance. The criteria will not be adjusted upward, however.

**Attendance**
Lecture attendance is required; you should understand that your success in the course will improve greatly by attending classes regularly. The instructor reserves the right to penalize unexcused absences; e.g., your final grade may be lowered by one point for each unexcused absence above three. The following is excerpted from the 2017-2018 Undergraduate/Graduate Catalog.

The student is expected to attend all classes and laboratory sessions. It is the responsibility of the student to inform each instructor of extended absences. When, in the judgment of the instructor, a student has been absent to such a degree as to impair his or her status relative to credit for the course, the instructor can drop the student from the class with a grade of W before the course drop deadline and with a grade of F after the course drop deadline.

**Standards of Conduct**
You are expected to conduct yourself in a professional and courteous manner, as prescribed by the Handbook of Operating Procedures: Student Conduct and Discipline. All graded work (homework, projects, exams) is to be completed independently (unless otherwise specified) and should be unmistakably your own work, although you may discuss your work with others in a general way. You may not represent as your own work material that is transcribed or copied from another source, including persons, books, or Web pages. “Plagiarism” means the appropriation, buying, receiving as a gift, or obtaining by any means another’s work and the unacknowledged submission or incorporation of it in one’s own academic work offered for credit, or using work in a paper or assignment for which the student had received credit in another course without direct permission of all involved instructors. Plagiarism is a serious violation of university policy and will not be tolerated. All cases of suspected plagiarism will be reported to the Dean of Students for further review.

**Disabilities**
If you have a disability and need classroom accommodations, please contact The Center for Accommodations and Support Services (CASS) at 747-5148, or by email to cass@utep.edu, or visit their office located in UTEP Union East, Room 106. For additional information, please visit the CASS website at www.sa.utep.edu/cass.
### Tentative Schedule

The following table shows a tentative schedule of the course; refer to the course website for an up-to-date schedule.

<table>
<thead>
<tr>
<th>Dates</th>
<th>Topics</th>
<th>Readings</th>
<th>Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>Aug. 29: Intro to MDD and MDA, Elaboration vs. translation</td>
<td>Chap 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[Meservy-Fenstermacher05]</td>
<td></td>
</tr>
<tr>
<td>Week 2</td>
<td>Sep. 5: UML - Use case diagram, Class diagram</td>
<td>Chap 2, Chap 3 of UML*</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Chap 4 of UML*</td>
<td></td>
</tr>
<tr>
<td>Week 3</td>
<td>Sep. 12: Class diagram, State machine diagram</td>
<td>Chap 5 of UML*</td>
<td>Homework 1</td>
</tr>
<tr>
<td>Week 4</td>
<td>Sep. 19: Interaction and other diagrams</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Paper presentations: UML/MDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 5</td>
<td>Sep. 26: OCL - Model constraints</td>
<td>Chap 3; [Warmer-Kleppe99]</td>
<td></td>
</tr>
<tr>
<td>Week 6</td>
<td>Oct. 3: Model constraints, Paper presentations: OCL/MDA</td>
<td></td>
<td>Homework 2</td>
</tr>
<tr>
<td>Week 7</td>
<td>Oct. 10: Specification using UML</td>
<td>Chap 4</td>
<td></td>
</tr>
<tr>
<td>Week 8</td>
<td>Oct. 17: Project proposal</td>
<td></td>
<td>Project proposal</td>
</tr>
<tr>
<td></td>
<td>Exam 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 9</td>
<td>Oct. 24: Model validation</td>
<td>Chap 5</td>
<td>Homework 3</td>
</tr>
<tr>
<td>Week 10</td>
<td>Oct. 31: Design techniques, Model transformations</td>
<td>Chap 6, Chap 7</td>
<td></td>
</tr>
<tr>
<td>Week 11</td>
<td>Nov. 7: Model transformations</td>
<td></td>
<td>Homework 4</td>
</tr>
<tr>
<td>Week 12</td>
<td>Nov. 14: Papers: model transformation, Implementation</td>
<td>Chap 8</td>
<td></td>
</tr>
<tr>
<td>Week 13</td>
<td>Nov. 21: System evolution, Thanksgiving</td>
<td>Chap 9</td>
<td></td>
</tr>
<tr>
<td>Week 14</td>
<td>Nov. 28: Papers: S/W lang, apps, and tools, Project work</td>
<td>[Kleppe05]</td>
<td></td>
</tr>
<tr>
<td>Week 15</td>
<td>Dec. 5: Project presentations</td>
<td></td>
<td>Project report</td>
</tr>
<tr>
<td>Week 16</td>
<td>Dec. 12: Final exam at 1:00 – 3:45 pm</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


### Important Dates

- **August 28:** Classes begins
- **September 4:** Labor Day – university closed
- **September 13:** Census day
- **October 19:** Exam 1
- **November 3:** Course drop/withdrawal deadline
- **November 23-24:** Thanksgiving holiday - university closed
- **December 8:** Dead day
- **December 12:** Final on Tuesday at 1:00 – 3:45 pm
**Required Readings**

The following is a tentative list of required readings, and you are welcome to suggest additional readings.


**Paper Presentations**

The following is a list of suggested readings for paper presentations, and you are welcome to suggest additional readings.

**UML/MDD**


**OCL/MDA**


**Transformations**


**Languages, applications and tools**


