

CS 4390/5382: Model-Based Software Development

Fall 2017

CRN: 16096/18031

Lecture: TR 12:00-1:20 pm in CCSB 1.0410

Website: <http://www.cs.utep.edu/cheon/cs5382>

Instructor: Yoonsik Cheon (x-8028, ycheon@utep.edu); office hours: TBA pm in CCS 3.0606

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:

Kevin Lano, *Model-Driven Software Development with UML and Java*, Course Technology, 2009 (ISBN: 978-1844809523).

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 on-line 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:
 - a. Martina Seidl, et al., *UML@Classroom: An Introduction to Object-Oriented Modeling*, Springer, 20015.
 - b. Bernhard Rumpe, *Modeling with UML: Language, Concepts, Methods*, Springer, 2016.
- Jos Warmer and Anneke Kleppe, *The Object Constraint Language*, second edition, Addison-Wesley, 2003.
- Scott W. Ambler. *The Elements of UML 2.0 Style*, Cambridge University Press, 2005.

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:

<i>Graded work</i>	<i>CS 4390</i>	<i>CS 5382</i>
Homework	60%	40%
Exam 1	20%	15%

Final	20%	15%
Paper presentation	N/A	5%
Project	N/A	25%

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:

<i>Score (%)</i>	<i>Letter grade</i>
90% or higher	A
80-89%	B
70-79%	C
60-69%	D
below 60%	F

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.

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

*Martina Seidl, et al., *UML@Classroom: An Introduction to Object-Oriented Modeling*, Springer, 2015. Free ebook through UTEP library.

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.

- [Meservy-Fenstermacher05] Thomas O. Meservy and Kurt D. Fenstermacher. Transforming Software Development: An MDA Road Map. *IEEE Computer*, 38(9): 52-58, September 2005.
- [Warmer-Kleppe99] Jos Warmer and Anneke Kleppe. OCL: The Constraint Language of the UML, *Journal of Object-Oriented Programming*, 2(2):10-13, May 1999.

Paper Presentations

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

UML/MDD

- [France-etal06] Robert B. France, Sudipto Ghosh, Trung Dinh-Trong, and Arnor Solberg. Model-Driven Development Using UML 2.0: Promises and Pitfalls. *IEEE Computer*, 39(2):59-66, February 2006.
- [Fuentes-Vallecillo04] Lidia Fuentes and Antonio Vallecillo. An Introduction to UML Profiles, *UPGRADE*, 5(2):6-13, April, 2004.

OCL/MDA

- [Brown04] Alan W. Brown. Model Driven Architecture: Principles and Practice. *Software and System Modeling*, 3(4):314-327, December, 2004.
- [Hennicker-Hussmann-Bidoit02] Rolf Hennicker, Hinrich Hussmann, and Michel Bidoit. On the Precise Meaning of OCL Constraints. In T. Clark and J. Warmer, editors, *Object Modeling with the OCL, volume 2263 of Lecture Notes in Computer Science*, pages 69-84, Springer, 2002.
- [OCL14] OMG. *Object Constraint Language, version 2.4*, Object Management Group, February 2014. Available from <http://www.omg.org/spec/OCL/2.4/PDF/>.

Transformations

- [Correa-Werner04] A. Correa and C. Werner, *Applying Refactoring Techniques to UML/OCL Models*, UML 2004, Springer-Verlag, LNCS, volume 372, pages 173-187, 2004.
- [Czarnecki-Helsen03] Krzysztof Czarnecki and Simon Helsen, Classification of Model Transformation Approach, *OOPSLA '03 Workshop on Generative Techniques in the Context of Model-Driven Architecture*, 2003.
- [Czarnecki-Helsen06] Krzysztof Czarnecki and Simon Helsen, Feature-based survey of model transformation approaches, *IBM Systems Journal*, 45(3):621-645, July 2006.
- [Griese-Larsen05] M. Griese and D. Larsen. *Simplifying Transformations of OCL Constraints*, MODELS 2005, Springer-Verlag, LNCS, volume 3713, pages 309-323, 2005.
- [Mens-Gorp06] Tom Mens and Pieter Van Gorp, A Taxonomy of Model Transformation, *Electronic Notes in Theoretical Computer Science (ENTCS)*, 152:125-142, March 2006.

Languages, applications and tools

- [Aho-etal09] Pekka Aho, Mati Maki, Daniel Pakkala, and Ela Ovaska, MDA-based tool chain for web services development, *Proceedings of the 4th Workshop on Emerging Web Services Technology (WEWST '09)*, November 9, 2009, Eindhoven, The Netherlands, pages 11-18.
- [Balasubramanian-etal06] K. Balasubramanian, A. Gokhale, G. Karsai, J. Sztipanovits, S. Neema, Developing applications using model-driven design environments, *IEEE Computer*, 39(2): 33-40, February, 2006.

- [Bezivin-etal04] Jean Bezivin, Slimane Hammoudi, Denivaldo Lopes, and Frederic Jouault, Applying MDA approach for Web service platform, Proceedings of the 8th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2004), September 2004, pages 58-70.
- [Dzidek-Briand-Labiche06] W. J. Dzidek, L. C. Briand, and Y. Labiche. Lessons learned from developing a dynamic OCL constraint enforcement tool for Java. In *ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems, Montego Bay, Jamaica, October 2-7, 2005*, volume 3844 of *LNCS*, pages 10–19. Springer-Verlag, 2006.
- [Heitkotter-Majchrzak-Kuchen13] H. Heitkotter, T. A. Majchrzak and H. Kuchen, *Cross-platform model-driven development of mobile applications with md²*, Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra, Portugal, March 18-22, 2013, pages 526-533.
- [Holstein11] Beery D. Holstein, Speed delivery of Android devices and applications with model-driven development, *IBM DeveloperWorks*, June 2011. Available online from <http://www.ibm.com/developerworks/rational/library/model-driven-development-speed-delivery/index.html>.
- [Kleppe05] Anneke Kleppe. Towards General Purpose, High-Level, Software Languages. *Model Driven Architecture – Foundations and Applications*, pages 220-238, Volume 3748 of Lecture Notes in Computer Science, Springer-Verlag, 2005.
- [Langegger-Palkoska-Wagner06] A. Langegger, J. Palkoska, and R. Wagner, DaVinci – A model-driven Web engineering framework, *Journal of Web Information System*, 2(2):119-132, June 2006.