Model-driven software development

Academic Year: (2019 / 2020) **Review date: 14/05/2020 13:17:38**

Department assigned to the subject:

Coordinating teacher: GENOVA FUSTER, GONZALO

Type: Electives ECTS Credits : 3.0

Year : 1 Semester : 2

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

A basic knowledge of the UML language is assumed (notation and basic applications) that paves the more technical aspects.

OBJECTIVES

CB6, CB7, CB8, CB9, CB10
CG1, CG2, CG3, CG4
CE1, CE2, CE12

Deepening the OMG initiative known as Model Driven Architecture (MDA), Model Driven Development (MDD) or Model Driven Engineering (MDE), whose central idea is to shift the center of gravity in the software development process from source code to software models: planning, estimating, reuse, maintenance, production, simulation and so on.

This initiative is of great importance for a software development process based on model transformations to be truly iterative: integration and coherence of all phases of development, models synchronized at all levels, etc.

Model-driven systems development has long demonstrated its effectiveness in the development of complex hardware systems, and its application is growing in the development of software systems (or hardware-software combinations), as long as available methodologies and tools arise to manipulate software models.

The main learning outcomes are:

- Mastering the concepts of software modeling at all levels, from the more abstract models used to capture and analyze requirements, to the more detailed design and implementation models, as well as their direct application to software engineering.
- Keeping in mind that modeling is an essential task in the software development process approaches this course to software manufacturing processes in accordance with the latest standards.
- Knowing how to transmit the results of a recent research on the subject, previously agreed with the course teacher.
- Being able to critically analyze research papers.

DESCRIPTION OF CONTENTS: PROGRAMME

A. FUNDAMENTALS
1. The role of models in the development of software.
2. Principles and fundamentals of modeling and metamodeling.
3. Evolution of models and interrelation between models.

B. APPLICATIONS
4. Tools for transformation of models.
5. Executable models: information representation and action semantics.
6. Automatic code generation from design models.

LEARNING ACTIVITIES AND METHODOLOGY

Theoretical and practical lectures
Teamwork to develop and expose theoretical and practical works
Individual student work

Each student will present a recent research article on the subject, previously agreed with the course teacher. The presentation will be followed by a discussion among all participants in the course, moderated by the course teacher, on the main ideas of the article presented. The articles will cover both the principles and foundations of MDA as well as their practical applications for the development of software.

Students will distribute themselves in working teams to develop several exercises in metamodeling and model transformation, learning to use the most common tools in the field.

ASSESSMENT SYSTEM

% end-of-term-examination/test:	40
% of continuous assessment (assigments, laboratory, practicals…):	60

Individual or group work performed during the course
Presentation in class of work done during the course

Each student must submit at the end of the course a Report with a critical remark to the work presented by himself, followed by a brief summary of all the Works presented by the other participants.

The final assessment is broken down into:

CONTINUOUS EVALUATION
40% development of practical exercises (individual or team work)
20% exposition of a theoretical subject (individual or team work)

FINAL EVALUATION
40% final report on the course (individual work)

BASIC BIBLIOGRAPHY

- Chris Raistrick Model Driven Architecture with Executable UML, Cambridge University Press, 2004

- Jack Greenfield, Keith Short, Steve Cook, Stuart Kent Software Factories: Assembling Applications with Patterns, Models, Frameworks and Tools, John Wiley & Sons, 2004

- Jesús García Molina, Félix O. García Rubio, Vicente Pelechano, Antonio Vallecillo, Juan Manuel Vara, Cristina Vicente-Chicote (eds.) Desarrollo de Software Dirigido por Modelos: Conceptos, Métodos y Herramientas, RAMA, 2012

- Jon Holt, Simon Perry, Mike Brownsword Foundations for model-based systems engineering: from patterns to models, Institution of Engineering and Technology, 2016

- OMG Unified Modeling Language Specification 2.0, (www.uml.org).

- OMG Model Driven Architecture Specification , (www.uml.org).

- Varios International Conference on the Unified Modeling Language, serie de congresos UML-1998 hasta 2004 , (www.umlconference.org).

- Varios International Conference on Model Driven Engineering Languages and Systems Series, serie de congresos MoDELS-2005 hasta 2013, (www.umlconference.org).

ADDITIONAL BIBLIOGRAPHY

- Anneke Kleppe MDA Explained: The Model Driven Architecture, Practice and Promise, Addison-Wesley, 2003

- David S. Frankel Model Driven Architecture: Applying MDA to Enterprise Computing, John Wiley and Sons, 2003

- Jim Arlow Enterprise Patterns and MDA: Building Better Software with Archetype Patterns and UML, Addison-Wesley, 2004

- Stephen Mellor MDA Distilled: Principles of Model-Driven Architecture, Addison-Wesley, 2004

- Stephen Mellor Executable UML: A Foundation for Model Driven Architecture, Addison-Wesley, 2002