**CO7217 Domain-Specific Languages**

**Credits:** 15  
**Convenor:** Dr. Artur Boronat  
**Semester:** 1st

**Prerequisites:** Desirable: UML, Java, Eclipse

**Assessment:** Coursework: 40%  
Two hour exam in January: 60%

**Lectures:** 16 hours  
**Surgeries:** 8 hours  
**Laboratories:** 16 hours  
**Private Study:** 72.5 hours

---

**Subject Knowledge**

**Aims** The aim of this module is to give students knowledge and skills in the principles and functions of domain-specific languages (DSLs) and their applications, as well as in their development based on OMG standards (MOF, OCL, QVT) and the Eclipse Modeling Framework (EMF). This shall enable them to use such technologies in practice or to embark on a PhD in this area.

**Learning Outcomes** At the end of this module, successful students will be acquainted with the conceptual and technological foundations of DSLs, i.e., the motivation, basic mechanisms, and open problems of DSLs; model-driven development and its relation to object-oriented development; the realisation of modeling environments based on OMG standards and Eclipse technology.

**Methods** Class sessions together with course notes, recommended textbooks, and worksheets.

**Assessment** Multiple choice and short answer tests, written examinations.

---

**Skills**

**Aims** The module shall provide the basic skills required to use technology for defining DSLs and for using them in practice.

**Learning Outcomes** At the end of this module, successful students will be able to design and develop DSLs using both OMG standards and model-based technology, including: MOF metamodels, describing the abstract syntax of a DSL, their implementation as EMF metamodels, and their realization as Java-based modeling environments; OCL constraints, adding semantics to a MOF metamodel; the definition of the concrete syntax of a DSL, either graphical or textual; automated generative techniques: model transformations and code generation techniques.

**Methods** Worksheets and practical programming experience.

**Assessment** Computer-based exercises, computer programmes.

---

**Explanation of Prerequisites** Basic knowledge of UML Class Diagrams, Eclipse and Java will be helpful.

**Course Description** According to the Standish Group, businesses in the United States spend about $250 billion annually on software development. Only 16% of these projects are completed on schedule and within budget. Another 31% are canceled, primarily because of quality problems, creating losses of about $81 billion annually. Another 53% cost more than planned, exceeding their budgets by an average of 189%, creating losses of about $59 billion annually. Projects that reach completion deliver an average of only 42% of the originally planned features. These losses are primarily due to the complexity of the systems that are developed and to changes in user requirements.

Model-Driven Development (MDD) is intended for increasing the quality and the productivity of a software development process. This is achieved by raising the level of abstraction from code to models and by automating tasks by means of generative techniques, which enhance the reuse of knowledge, patterns and code. A modeling language that permits defining such models in a given domain constitutes a decisive keystone in a MDD process.
The module shall give an introduction to the basic technologies that underly the definition of DSLs and their use in a MDD process. This includes the definition of the syntax of a DSL at different levels: at a conceptual level by means of MOF metamodels, and at a concrete level by means of either textual-based techniques or graphical techniques; their definition in the Eclipse Modeling Framework; the use of OCL constraints to add semantics to the syntax of a DSL; and the use of automated techniques for manipulating models: model transformations by using QVT and code generation techniques.

**Detailed Syllabus**  
In detail, the module will cover the following topics.

- Motivation and concepts of DSLs in MDD Processes;
- Definition of the conceptual layer of DSLs by using MOF and OCL;
- Definition of the concrete syntax of DSLs by using textual and graphical syntax;
- Implementation of DSLs by using the Eclipse Modeling Framework and the Graphical Modeling Framework;
- Automated model transformations by using QVT, OCL and Graph Transformations;
- Automated code generation.

**Reading List**


**Resources**  
Course notes, web page, study guide, worksheets, handouts, lecture rooms with two OHPs.

**Module Evaluation**  
Course questionnaires, course review.