

SYLLABUS

Behavior Modeling of Software Systems

University year 2025

1. Information regarding the programme

1.1. Higher education institution	Babeş Bolyai University
1.2. Faculty	Faculty of Mathematics and Computer Science
1.3. Department	Department of Computer Science
1.4. Field of study	Computer Science
1.5. Study cycle	Master
1.6. Study programme/Qualification	Software Engineering
1.7. Form of education	

2. Information regarding the discipline

2.1. Name of the discipline		Behavior Modeling of Software Systems				Discipline code		MME8006
2.2. Course coordinator				Lect. dr. Ioan Lazar				
2.3. Seminar coordinator				Lect. dr. Ioan Lazar				
2.4. Year of study	1	2.5. Semester	2	2.6. Type of evaluation	C	2.7. Discipline regime	Optional	

3. Total estimated time (hours/semester of didactic activities)

3.1. Hours per week	3	of which: 3.2 course	2	3.3 seminar/laboratory/project	1
3.4. Total hours in the curriculum	42	of which: 3.5 course	28	3.6 seminar/laboratory/project	14
Time allotment for individual study (ID) and self-study activities (SA)					hours
Learning using manual, course support, bibliography, course notes (SA)					28
Additional documentation (in libraries, on electronic platforms, field documentation)					14
Preparation for seminars/labs, homework, papers, portfolios and essays					14
Tutorship					7
Evaluations					7
Other activities:					14
3.7. Total individual study hours	22				
3.8. Total hours per semester	120				
3.9. Number of ECTS credits	7				

4. Prerequisites (if necessary)

4.1. curriculum	• Programming Fundamentals
4.2. competencies	• Good programming skills in at least one of the languages Java, C#

5. Conditions (if necessary)

5.1. for the course	
5.2. for the seminar /lab activities	

6.1. Specific competencies acquired ¹

Professional/essential competencies	• C 4.3 Identify models and methods adequate to real life problem solving • C 2.1 Identify adequate software systems development methodologies • C 1.1 Proper description of programming paradigms and language specific mechanisms, and identification of semantical and syntactical differences
Transversal competencies	• CT1 Apply organized and efficient work rules and responsible attitude towards didactical and research field, in order to creatively use work potential; respect professional ethical principles • CT3 Use efficient methods and techniques for: learning, information search, research and development of capacities to adapt to the requirements of a dynamic society and to communicate in an international language

6.2. Learning outcomes

Knowledge	The student knows: development cycle of systems based on services
Skills	The student is able to develop a system based on services
Responsibility and autonomy:	The student has the ability to work independently to build SOA systems

7. Objectives of the discipline (outcome of the acquired competencies)

¹ One can choose either competencies or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

10. Modeling user interface navigation using state machines UML: state machine diagrams 11. Modeling user interface navigation using state machines Platform specific frameworks: Grails and JBoss Seam 12. Capturing requirements with Business Motivation Model BMM 13. Capturing requirements with Business Motivation Model From BMM to SOA 14. Executable Use Cases		
Bibliography [Amblor04] Ambler, S.W. The Object Primer: Agile Model-Driven Development with UML 2.0. Cambridge University Press, 2004. [Fowler99] Fowler, M. Analysis Patterns - Reusable Object Models. Addison-Wesley, 1997. [Evans03] Evans, E. Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley, 2003. [OMG03] OMG. MDA Guide Version 1.0.1. Object Management Group, 2003. http://www.omg.org/docs/omg/03-06-01.pdf [OMG06] OMG. Business Process Modeling Notation Specification, V1.0. Object Management Group, 2006. http://www.bpmn.org/ [WPI06] Workflow Patterns Initiative. Control-Flow, Data, Resource, and Exception Handling Patterns. Workflow Patterns Home Page, 2006. http://www.workflowpatterns.com/		
8.2 Seminar / laboratory	Teaching methods	Remarks
1. Introduction to DSML 2. Model Transformations in the Context of MDA/UML 3. Service Component Models 4. Service oriented architecture Modelling Language (SoaML) 5. Business Processes 6. Workflow Patterns 7. Foundational UML	Dialogue, debate, case studies, examples, proofs	
Bibliography AndroMDA. Business Process Management for Struts Cartridge. 2006. http://galaxy.andromda.org/docs/andromda-bpm4struts-cartridge/index.html Erich Gamma et al. Design Patterns: Elements of Reusable Object Oriented Software. Addison Wesley, 1995. Erich Gamma and Kent Beck. Contributing to Eclipse: Principles, Patterns, and Plug-Ins. Addison Wesley, 2003. Martin Fowler. Patterns of Enterprise Application Architecture. Addison Wesley, 2002. Martin Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition. Addison Wesley, 2003. Rod Johnson et al. Spring 2.0 Reference Documentation. 2006. http://www.springframework.org/ Michael Mahemoff. Ajax Design Patterns. O'Reilly, 2006. Stephen J. Mellor, Kendall Scott, Axel Uhl, and Dirk Weise. MDA Distilled: Principles of Model-Driven Architecture. Addison Wesley, 2004. Object Management Group. MDA Guide Version 1.0.1. 2003. http://www.omg.org/docs/omg/03-06-01.pdf Object Management Group. MOF 2.0 Query/Views/Transformations RFP. 2004. http://www.omg.org/cgi-bin/apps/doc?ad/02-04-10.pdf Object Management Group. UML 2.0 OCL Specification. 2003. http://www.omg.org/cgi-bin/apps/doc?formal/06-05-01.pdf Object Management Group. UML 2.0 Superstructure. 2004. http://www.omg.org/cgi-bin/apps/doc?formal/05-07-04.pdf		

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- The course respects the IEEE and ACM Curricula Recommendations for Computer Science studies;
- The course exists in the studying program of all major universities in Romania and abroad;
- The content of the course is considered by the software companies as important for average programming skills.

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	- know the basic principle of the domain; - apply the course concepts - problem solving	Written exam	50%
10.5 Seminar/lab activities	- be able to implement course concepts and algorithms - apply techniques for different classes of programming languages	-Practical examination -documentation -portfolio -continous observations	50%
10.6 Minimum standard of performance			
• A minimum passing grade is defined by attaining at least 50% (5/10) points for the final project and each of the three lab assignments respectively. • No more than 3 absences are allowed for the seminar/lab activities			

11. Labels ODD (Sustainable Development Goals)²

Not applicable.

² Keep only the labels that, according to the [Procedure for applying ODD labels in the academic process](#), suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable.*”.

Date:
.30.04.2025

Signature of course coordinator

Lect. dr. Ioan Lazar

Signature of seminar coordinator

Lect. dr. Ioan Lazar

Date of approval:

...

Signature of the head of department

Assoc.prof.phd. Adrian STERCA