

SYLLABUS

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

2. Information regarding the discipline

2.1 Name of the discipline	Behavior Modeling of Software Systems						
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	1	2.6. Type of evaluation	E	2.7 Type of discipline	Mandatory

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	1
3.4 Total hours in the curriculum	36	Of which: 3.5 course	24	3.6 seminar/laboratory	12
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					8
Additional documentation (in libraries, on electronic platforms, field documentation)					7
Preparation for seminars/labs, homework, papers, portfolios and essays					8
Tutorship					2
Evaluations					8
Other activities:					
3.7 Total individual study hours	33				
3.8 Total hours per semester	75				
3.9 Number of ECTS credits	7				

4. Prerequisites (if necessary)

4.1. curriculum	<ul style="list-style-type: none"> • Programming Fundamentals
4.2. competencies	<ul style="list-style-type: none"> • Good programming skills in at least one of the languages Java, C#

5. Conditions (if necessary)

5.1. for the course	<ul style="list-style-type: none"> • Course hall with projector
5.2. for the seminar /lab activities	<ul style="list-style-type: none"> • Laboratory with computers

6. Specific competencies acquired

Professional competencies	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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

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

7.1 General objective of the discipline	<p>Enhance the students understanding of behavior modeling concepts through a practical and pragmatic approach</p> <p>Provide the students with an environment in which they can explore the usage and usefulness of behavior modeling concepts in various business scenarios</p> <p>Induce a realistic and industry driven view of software design concepts such as design patterns and their inherent benefits</p>
7.2 Specific objective of the discipline	<p>Give students the ability to explore various object oriented programming languages</p> <p>Improve the students abilities to tackle business requirements</p> <p>Enhance the students understanding of state machines, activities</p> <p>Provide students with insights into the way of working towards achieving high quality software through skilled trainers from the IT industry</p>

8. Content

8.1 Course	Teaching methods	Remarks
1. Introduction to DSML <ul style="list-style-type: none"> • Textual notations 	Exposure: description, explanation, examples, discussion of case studies	
2. Introduction to DSML <ul style="list-style-type: none"> • Graphical notations 	Exposure: description, explanation, examples, discussion of case studies	
3. Model Transformations in the Context of MDA/UML	Exposure: description,	

<ul style="list-style-type: none"> • MDA, UML • M2T, M2M 	<p>explanation, examples, discussion of case studies</p>	
<p>4. Service Component Models</p> <ul style="list-style-type: none"> • Platform specific frameworks: iPOJO, SCA • UML: deployment diagrams, composite structure diagrams 	<p>Exposure: description, explanation, examples, discussion of case studies</p>	
<p>5. Service oriented architecture Modelling Language (SoaML)</p> <ul style="list-style-type: none"> • SoaML specification • ModelPro/MagicDraw frameworks 	<p>Exposure: description, explanation, examples, discussion of case studies</p>	
<p>6. Business Processes</p> <ul style="list-style-type: none"> • Business Process Modelling Notation (BPMN) Workflow Patterns 	<p>Exposure: description, explanation, examples, discussion of case studies</p>	
<p>7. Business Processes</p> <ul style="list-style-type: none"> • Workflow Patterns 	<p>Exposure: description, explanation, examples, discussion of case studies</p>	
<p>8. Foundational UML</p> <p>Abstract Syntax and Foundational Model Library</p> <p>Java to UML activity mapping</p>	<p>Exposure: description, explanation, examples, discussion of case studies</p>	
<p>9. Action Language for Foundational UML (Alf)</p>	<p>Exposure: description, explanation, examples, discussion of case studies</p>	
<p>11. Modeling user interface navigation using state machines</p> <ul style="list-style-type: none"> • Platform specific frameworks: Grails and JBoss Seam • UML: state machine diagrams 	<p>Exposure: description, explanation, examples, discussion of case studies</p>	
<p>12. Capturing requirements with Business Motivation Model</p> <ul style="list-style-type: none"> • BMM • From BMM to SOA 	<p>Exposure: description, explanation, examples, discussion of case studies</p>	
<p>13. Executable Use Cases</p>	<p>Exposure: description, explanation, examples, discussion of case studies</p>	
<p>14. Review</p> <p>Bibliography</p> <p>[Ambler04] Ambler, S.W. The Object Primer: Agile Model-Driven Development with UML 2.0. Cambridge University Press, 2004.</p> <p>[Fowler99] Fowler, M. Analysis Patterns - Reusable Object Models. Addison-Wesley, 1997.</p> <p>[Evans03] Evans, E. Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-</p>		

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

Dialogue, debate,
case studies,
examples, proofs

2. Introduction to DSML

Dialogue, debate,
case studies,
examples, proofs

3. Model Transformations in the Context of
MDA/UML

Dialogue, debate,
case studies,
examples, proofs

4. Service Component Models

Dialogue, debate,
case studies,
examples, proofs

5. Service oriented architecture Modelling Language
(SoaML)

Dialogue, debate,
case studies,
examples, proofs

6. Business Processes

Dialogue, debate,
case studies,
examples, proofs

7. Workflow Patterns

8. Foundational UML

9. Action Language for Foundational UML (Alf)

11. Modeling user interface navigation using state machines

12. Capturing requirements with Business Motivation Model

13. Executable Use Cases

14. Review

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>

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 the software companies as important for average programming skills.

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the 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 -portofolio -continous observations	50%
10.6 Minimum performance standards			
➤ At least grade 5 (from a scale of 1 to 10) at both written exam and laboratory work.			

Date

30.09.16

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

Prof. dr. Anca Andreica