

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	Inginerie software	

2. Information regarding the discipline

2.1 Name of the discipline	Behaviour Modelling 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		2.6. Type of evaluation	E	2.7 Type of discipline	

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

3.1 Hours per week	4	Of which: 3.2 course	2	3.3 seminar/laboratory	2
3.4 Total hours in the curriculum	56	Of which: 3.5 course	28	3.6 seminar/laboratory	28
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					20
Additional documentation (in libraries, on electronic platforms, field documentation)					10
Preparation for seminars/labs, homework, papers, portfolios and essays					23
Tutorship					7
Evaluations					20
Other activities:					
3.7 Total individual study hours	80				
3.8 Total hours per semester	150				
3.9 Number of ECTS credits	7				

4. Prerequisites (if necessary)

4.1. curriculum	<ul style="list-style-type: none"> • Software Engineering
4.2. competencies	<ul style="list-style-type: none"> • UML basic knowledge

5. Conditions (if necessary)

5.1. for the course	<ul style="list-style-type: none"> • -
5.2. for the seminar /lab activities	<ul style="list-style-type: none"> • Laboratory with computers; high level programming language environment & CASE tools

6. Specific competencies acquired

Professional competencies	<ul style="list-style-type: none"> • Design techniques for behavioral modeling. • Behavioral metamodels and some existing lightweight frameworks
Transversal competencies	<ul style="list-style-type: none"> • Ability to apply software systems modelling techniques to different real life problems • Improved programming & designing abilities

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

7.2 Specific objective of the discipline	<p>To introduce the student in advanced design techniques for behavioral modeling.</p> <p>To present the mappings between UML behavioral metamodels and some existing lightweight frameworks.</p> <p>To offer the student the opportunity to participate at designing a new framework that incorporates mappings between UML models and concrete frameworks.</p> <p>To offer the student the instruments that will allow him/her to enhance the framework to other specific needs.</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 <ul style="list-style-type: none"> • MDA, UML • M2T, M2M 	Exposure: description, explanation, examples, discussion of case studies	
4. Service Component Models <ul style="list-style-type: none"> • Platform specific frameworks: iPOJO, SCA • UML: deployment diagrams, composite structure diagrams 	Exposure: description, explanation, examples, discussion of case studies	
5. Service oriented architecture Modelling Language (SoaML)	Exposure: description.	

<ul style="list-style-type: none"> • SoaML specification • ModelPro/MagicDraw frameworks 	<p>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>Exposure: description, explanation, examples, discussion of case studies</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-Wesley, 2003.</p> <p>[OMG03] OMG MDA Guide Version 1.0.1. Object Management Group, 2003. http://www.omg.org/docs/omg/03-06-01.pdf</p> <p>[OMG06] OMG Business Process Modeling Notation Specification, V1.0. Object Management Group, 2006. http://www.bpmn.org/</p> <p>[WPI06] Workflow Patterns Initiative. Control Flow, Data, Resource, and Exception Handling Patterns</p>		

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	Dialogue, debate, case studies, examples, proofs	
8. Foundational UML	Dialogue, debate, case studies, examples, proofs	
9. Action Language for Foundational UML (Alf)	Dialogue, debate, case studies, examples, proofs	
11. Modeling user interface navigation using state machines	Dialogue, debate, case studies, examples, proofs	
12. Capturing requirements with Business Motivation Model	Dialogue, debate, case studies, examples, proofs	
13. Executable Use Cases	Dialogue, debate, case studies, examples, proofs	
14. Review	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 the software companies as important for a 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

30.04.14

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. Bazil Parv