SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	**Babes-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	**Computational Models for Embedded Systems**						
2.2 Course coordinator	**PhD Lecturer Andreea Vescan**						
2.3 Seminar coordinator	**PhD Lecturer Andreea Vescan**						
2.4. Year of study	**2**	2.5 Semester	**4**	2.6. Type of evaluation	**E**	2.7 Type of discipline	**compulsory**

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	48	Of which: 3.5 course	24	3.6 seminar/laboratory	24
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					24
Additional documentation (in libraries, on electronic platforms, field documentation)					36
Preparation for seminars/labs, homework, papers, portfolios and essays					48
Tutorship					20
Evaluations					24
Other activities:					0

3.7 Total individual study hours	152
3.8 Total hours per semester	200
3.9 Number of ECTS credits	8

4. Prerequisites (if necessary)

4.1. curriculum	•
4.2. competencies	•

5. Conditions (if necessary)

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

6. Specific competencies acquired

Professional competencies	• Understanding and working with basic concepts in software engineering; • Capability of analysis and synthesis; • Modeling and solving real-life problems;
Transversal competencies	• Ability to understand and approach problems of modeling nature from other sciences • Good programming skills in high-level languages; • Abilities to develop and maintain software systems; • Ability to analyze, synthesize and model fenomena and processes from various fields using adequate mathematical, statistical, computational and computer science methods.

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

7.1 General objective of the discipline	Modeling is an essential technique and serves various purposes, so it is important to understand its fundamental concepts. These concepts relate basically to the way a modeling paradigm allows one to deal with the great properties of system: time, computation, data and communication. In general, when modeling a system, we want to be able to formulate particular questions on it: Will a system ever enter a particular state? Will it always be able to avoid a particular dangerous state? Will it always eventually reach some desired state? The classical theory of computation focuses on the function that a program computes. To understand embedded computation, we need to focus on the reactive nature of the interaction of a component with its environement and requirements concerning safety, timeliness, stability, and performance. • The main objectives of this cours are to provide an understanding of the fundamentals concepts of software modeling, more precise for embedded computation; to develop skills in modeling and to describe and verify safety and liveness properties of the system being modeled.
7.2 Specific objective of the discipline	• Students will learn the models of computations for embedded systems. • Students will know to specify, design and verify an embedded system • Students will know how to model a system and specify restrictions on functionalities.

8. Content

8.1 Course	Teaching methods	Remarks
1. Introduction. Model: Why?what? how? Types of systems.	Presentation, Conversation, Didactic	

	demonstration, Problematizations	
2. Requirements and Safety Requirements. Model checking.	Presentation, Conversation, Didactic demonstration, Problematizations	
3. Synchronous models.	Presentation, Conversation, Didactic demonstration, Problematizations	
4. Finite State Machines 5. Finite State Machines (cont.)	Presentation, Conversation, Didactic demonstration, Problematizations	
6. Petri Nets 7. Petri Nets (cont.)	Presentation, Conversation, Didactic demonstration, Problematizations	
8. Asynchronous models	Presentation, Conversation, Didactic demonstration, Problematizations	
9. Dynamical systems	Presentation, Conversation, Didactic demonstration, Problematizations	
10. Timed Models	Presentation, Conversation, Didactic demonstration, Problematizations	
11. Hybrid systems	Presentation, Conversation, Didactic demonstration, Problematizations	
12. Reserved subject		

Bibliography

[1] C. Baier, J.-P. Katoen, Principles of Model Checking, ISBN 978-0-262-02649-9, 2008
[2] M. Ben-Ari, Principles of the Spin Model Checker, ISBN 978-1-84628-769-5
[3] Taylor Booth (1967) Sequential Machines and Automata Theory, John Wiley and Sons, New York. Library of Congress Catalog Card Number: 67-25924.
[4] D. Harel, "Statecharts: A Visual Formalism for Complex Systems", Sci. Comput. Programming 8 (1987), 231-274
[5] Petri, CA (1966) Communication with automata. DTIC Research Report AD0630125
[6] Hoare, CAR (2004) (1985), Communicating Sequential Processes, Prentice Hall International

Optional references

Internet resources and conferences

8.2 Seminar / laboratory	Teaching methods	Remarks
Lab 1 (weeks 1-2) Choose project theme.	Presentation, Conversation, Problematizations, Discovery, Individual study, Exercises	
Lab 2 (theme 1, weeks 3-4) Specifying safety and liveness requirements. Model checking.	Presentation, Conversation, Problematizations, Discovery, Individual study, Exercises	
Lab 3 (theme 2, weeks 5-6) Using Finite State Machines to model an embedded system.	Presentation, Conversation, Problematizations, Discovery, Individual study, Exercises	
Lab 4 (theme 3, weeks 7-8) Using PetriNets to model an embedded system.	Presentation, Conversation, Problematizations, Discovery, Individual study, Exercises	
Lab 5 (theme 2,3, weeks 9-10) Using Finite State Machines and PetriNets to model an embedded system.	Presentation, Conversation, Problematizations, Discovery, Individual study, Exercises	
Lab 6 (weeks 11-12) Project presentation and documentation delivery.	Presentation	
Bibliography		

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

- Students will know how to model, specify, design and test embedded systems.
- Students will learn various computational models for embedded systems.

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	GF=GradeFinal = graded paper at the final exam	Written examination Passing the final exam is conditioned by the grade GF being at least 5.	60
	GP=GradeProjects = documentations and	Specification, design and analysis, presentation of the	40

	projects	project	
10.6 Minimum performance standards			
- Students will know various computational models for embedded systems. - At least grade 5 (from a scale of 1 to 10) at written exam and project work.			

Date | Signature of course coordinator | Signature of seminar coordinator

04.30.2013 | Lect. PhD. Andreea Vescan | Lect.PhD. Andreea Vescan

Date of approval | Signature of the head of department

.... | Prof. PhD. Bazil Parv