SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	**Babes-Bolyai University**
1.2 Faculty	**Mathematics and Computer Science**
1.3 Department	**Computer Science**
1.4 Field of study	**Computer Science**
1.5 Study cycle	**Master**
1.6 Study programme / Qualification	**Component-based programming**

2. Information regarding the discipline

2.1 Name of the discipline	**Petri Nets in Software Modeling and Verification**						
2.2 Course coordinator	**Assist. Prof. Christian Sacarea, PhD**						
2.3 Seminar coordinator	**Assist. Prof. Christian Sacarea, PhD**						
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	3	Of which: 3.2 course	2	3.3 seminar/laboratory	1 sem
3.4 Total hours in the curriculum	42	Of which: 3.5 course	28	3.6 seminar/laboratory	14

Time allotment:	hours
Learning using manual, course support, bibliography, course notes	30
Additional documentation (in libraries, on electronic platforms, field documentation)	30
Preparation for seminars/labs, homework, papers, portfolios and essays	40
Tutorship	20
Evaluations	38
Other activities:	-

3.7 Total individual study hours	158
3.8 Total hours per semester	200
3.9 Number of ECTS credits	8

4. Prerequisites (if necessary)

4.1. curriculum	•
4.2. competencies	• Programming skills

5. Conditions (if necessary)

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

6. Specific competencies acquired

Professional competencies	• Solid knowledge in component-based software engineering • Advanced knowledge of mathematical theories and applications related to decision theory and optimization • Advanced knowledge of theoretical, methodological, and practical developments in computer science and software engineering Abilities of using software tools for all software development activities
Transversal competencies	• modeling and solving real-world problems • use of computer science's conceptual and methodological apparatus to provide solutions for incompletely defined situations, to solve new theoretical and practical problems • advanced knowledge of component-based technologies and languages work in teams, assuming different execution and leading roles, performing professional tasks with autonomy and responsibility

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

7.1 General objective of the discipline	• Abilities to apply and develop advanced information technologies based on suitable formal models, to propose and use such models and theories for automating the design, implementation, and verification of computer-based systems.
7.2 Specific objective of the discipline	• The acquired knowledge and experience will allow the students to actively use Petri nets and the computer-aided tools based on them in modelling, design, verification, and implementation of various classes of systems. Based on the acquired theoretical knowledge, the student is able to transfer approaches of the Petri net theory to the domain of other formal models too.

8. Content

8.1 Course	Teaching methods	Remarks
1. An introduction to Petri nets, their philosophy and applications, the notion of a net and of the derived basic terms	Exposure: description, explanation, examples, discussion of case studies	
2. Condition/Event (C/E) Petri nets, cases and steps, the state space of C/E systems, cyclic and live C/E systems, equivalence of C/E systems.	Exposure: description, explanation, examples, discussion of case studies	
3. Contact-free C/E systems, complementation, case graphs and their application for analysing C/E	Exposure: description, explanation, examples, discussion of case studies	

systems.		
4. Processes of C/E systems, occurrence nets, properties of properties and composition of processes.	Exposure: description, explanation, examples, discussion of case studies	
5. Complementation of C/E systems, the synchronic distance, special synchronic distances, C/E systems and the propositional calculus, facts.	Exposure: description, explanation, examples, discussion of case studies	
6. Place/Transition (P/T) Petri nets, their definition, evolution rules, their state space, basic analytical problems (safety, boundedness, conservativeness, liveness).	Exposure: description, explanation, examples, discussion of case studies	
7. Representing the possibly infinite state space of Petri nets by a reachability tree, computing and using reachability trees for analysing P/T Petri nets.	Exposure: description, explanation, examples, discussion of case studies	
8. P and T invariants of P/T Petri nets, their definition, the ways of computing them and using them for analysing P/T Petri nets.	Exposure: description, explanation, examples, discussion of case studies	
9. Subclasses and extensions of P/T Petri nets, state machines, marked graphs, free-choice Petri nets, Petri nets with inhibitors, timed and stochastic Petri nets.	Exposure: description, explanation, examples, discussion of case studies	
10. The notion of a Petri net language, types of such languages, their closure properties, their relation to the Chomsky hierarchy. Computability and complexity of some selected Petri net-related problems.	Exposure: description, explanation, examples, discussion of case studies	
11. Coloured Petri nets (CPNs), their basic modelling primitives, an inscription language, CPN Design as an example of a tool based on CPNs.	Exposure: description, explanation, examples, discussion of case studies	
12. Analysis of CPNs, occurrence graphs, invariants, and their use in analysing systems.	Exposure: description, explanation, examples, discussion of case studies	
13. Hierarchical and object-oriented Petri nets, basic concepts of a hierarchical design, substitution and invocation, adding object-oriented features on top of Petri nets, PNtalk as a language based on object-oriented Petri nets.	Exposure: description, explanation, examples, discussion of case studies	
14. Hierarchical and object-oriented Petri nets, basic concepts of a hierarchical design, substitution and invocation, adding object-oriented features on top of Petri nets, PNtalk as a language based on object-oriented Petri nets.	Exposure: description, explanation, examples, discussion of case studies	

Bibliography

1. Reisig, W.: Petri Nets, An Introduction, Springer Verlag, 1985. ISBN: 0-387-13723-8

2. Jensen, K.: Coloured Petri Nets, Basic Concepts, Analysis Methods and Practical Use, Springer Verlag, 1993. ISBN: 3-540-60943-1
3. Girault, C., Valk, R.: Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications, Springer Verlag, 2002. ISBN 3-540-41217-4
4. Desel, J., Reisig, W., Rozenberg, G.: Lectures on Concurrency and Petri Nets, Advances in Petri Nets, Lecture Notes in Computer Science, vol. 3098, Springer Verlag, 2004. ISBN 3-540-22261-8

8.2 Seminar / laboratory	Teaching methods	Remarks
1. An application of C/E systems.	Dialogue, team work	The seminar is structured as 2 hours classes every second week
2. An application of P/T Petri nets.	Dialogue, team work	
3. An application of P/T Petri nets.	Dialogue, team work	
4. An application of CPNS.	Dialogue, team work	
5. An application of CPNS.	Dialogue, team work	
6. An application of object-oriented Petri nets.	Dialogue, team work	
7. An application of object-oriented Petri nets.	Dialogue, team work	

Bibliography

1. GUY VIDAL-NAQUET, ANNI CHOQUET-GENIET, Reseaux de Petri et Systemes Paralleles, Armond Colin, 1992.
2. T. JUCAN, F. L. TIPLEA, Retele Petri, Ed. Univ. "Al.I.Cuza", Iasi, 1995.
3. T. JUCAN, F. L. TIPLEA, Retele Petri. Teorie şi Practica, Ed. Academiei, 1999.

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

Petri Nets are a valuable tool for system organization and modelling of several workflows. Hence, graduating this course will provide skills in order to solve real life problems of certain importance in any large organization.

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	Fundamental principles	Exam	60%
	Applying the methods for problem solving		
10.5 Seminar/lab activities	Implementing concepts and algorithms	Homework	20%
	Innovation, initiative, team work	Project, Technical report	20%

10.6 Minimum performance standards

- The final grade is computed as an weighted average of the individual marks obtained at the above activities. The student must obtain at least grade 5 (from a scale of 1 to 10).

Date Signature of course coordinator Signature of seminar coordinator

....

Date of approval Signature of the head of department

....