SYLLABUS

Formal models of concurrent processes

University year 2025-2026

1. Information regarding the programme

1.1. Higher education institution	Babeș-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	Distributed Systems in Internet
1.7. Form of education	Full time

2. Information regarding the discipline

2.1. Name of the discipline	Formal models of concurrent processes					Discipline code	**MME8094**
2.2. Course coordinator				Assoc.prof.phd Adrian Sterca			
2.3. Seminar coordinator				Assoc.prof.phd Adrian Sterca			
2.4. Year of study	2	2.5. Semester	3	2.6. Type of evaluation	E	2.7. Discipline regime	Mandatory

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

3.1. Hours per week	**5**	of which: 3.2 course	**2**	3.3 seminar/laboratory/project	**3**
3.4. Total hours in the curriculum	70	of which: 3.5 course	28	3.6 seminar/laboratory/project	**42**

Time allotment for individual study (ID) and self-study activities (SA)	**hours**
Learning using manual, course support, bibliography, course notes (SA)	20
Additional documentation (in libraries, on electronic platforms, field documentation)	20
Preparation for seminars/labs, homework, papers, portfolios and essays	35
Tutorship	10
Evaluations	20
Other activities:	0

3.7. Total individual study hours	105
3.8. Total hours per semester	175
3.9. Number of ECTS credits	7

4. Prerequisites (if necessary)

4.1. curriculum	• Distributed operating systems, Concurrent and distributed programming
4.2. competencies	• Concurrent and distributed programming

5. Conditions (if necessary)

5.1. for the course	Class room with a video projector device
5.2. for the seminar /lab activities	

6.1. Specific competencies acquired [1]

[1] One can choose either competences 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.

Professional/essential competencies	• Extensive knowledge of theoretical, methodological and practical developments specific to computer science. • Relevant usage of criteria and methods for verifying, validating and evaluating software solutions, the ability to make value judgements and to justify constructive decisions.
Transversal competencies	• Advanced ability to communicate in different professional environments, to use computer vocabulary appropriately in professional communication. • Advanced ability to model specific phenomena and processes in economic, industrial and scientific fields, using fundamental knowledge of mathematics, statistics and computer science.

6.2. Learning outcomes

Knowledge	• The graduate has the necessary knowledge to devise, model and design of complex distributed software applications • The graduate possesses the fundamental knowledge for modelling, being able to analyse real life problems and to translate them in concrete requirements and to design a corresponding software model
Skills	• The graduate can apply advanced distributed systems knowledge starting from a high level of abstraction and being able to offer implementation solutions for complex software system • The graduate proves knowledge related to specifying the requirements of research activities in the domain of computer science in general and distributed systems in particular and he/she understands the role of research in promoting progress
Responsibility and autonomy:	• The graduate proves the capacity to reflect over own learning resources • The graduate proves abilities to work independently in order to obtain knowledge necessary for designing, managing and evaluating research activities in distributed systems domain

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

7.1 General objective of the discipline	• The course offers a strong theoretical approach in modelling concurrent and distributed systems
7.2 Specific objective of the discipline	• The course presents several formalisms for describing concurrent processes

8. Content

8.1 Course	Teaching methods	Remarks
1. Serialization, concurrency and parallelism	Exposure:description, explanation,examples, discussion of case studies	

2. Concurrent programming paradigms: shared memory, message passing, transactional memory	Exposure:description, explanation,examples, discussion of case studies	
3. Concurrency models for transactional systems. The Page Model. The Object model	Exposure:description, explanation,examples, discussion of case studies	
4. Concurrency control – serializability criteria (Page Model)	Exposure:description, explanation,examples, discussion of case studies	
5. Concurrency control – serializability criteria (Page Model)	Exposure:description, explanation,examples, discussion of case studies	
6. Concurrency control algorithms (Page Model)	Exposure:description, explanation,examples, discussion of case studies	
7. Multiversioning data	Exposure:description, explanation,examples, discussion of case studies	
8. Concurrency control – serializability criteria and algorithms (Object Model)	Exposure:description, explanation,examples, discussion of case studies	
9. Failure recovery	Exposure:description, explanation,examples, discussion of case studies	
10. Concurrency control in distributed context	Exposure:description, explanation,examples, discussion of case studies	
11. Process algebra I	Exposure:description, explanation,examples, discussion of case studies	
12. Process algebra II	Exposure:description, explanation,examples, discussion of case studies	
13. CCS – Calculus of Communicating Systems	Exposure:description, explanation,examples, discussion of case studies	
14. Pi-Calculus	Exposure:description, explanation,examples, discussion of case studies	

Bibliography

1. Weikum G. Vossen G. Transactional Information System: Theory, Algorithms, and Practice of Concurrency Control and Recovery. Kaufmann Morgan Publ. 2002.
2. Reichel H. Formal Models of Concurrency, 2003
3. Robin Milner, Communication and Concurrency, Prentice Hall, International Series in Computer Science, ISBN 0-13-115007-3. 1989
4. Bertran Meyer, Concepts of concurrent computations, course, 2015
5. Robert Milner, A Calculus of Communicating Systems, Springer, 1986
6. Luca Aceto, Anna Ingolfsdottir, Kim G. Larsen, Jiri Srba, Reactive Systems: Modelling, Specification and Verification, 2005.

8.2 Seminar / laboratory	Teaching methods	Remarks
1. Discussions related to the project or report's subject	Dialog, debate, case studies, examples	
2. Discussions related to the project or report's subject	Dialog, debate, case studies, examples	
3. Discussions related to the project or report's subject	Dialog, debate, case studies, examples	
4. Report presentations	Dialog, debate, case studies, examples	
5. Report presentations	Dialog, debate, case studies, examples	

6. Project presentations	Dialog, debate, case studies, examples	
7. Project presentations	Dialog, debate, case studies, examples	
Bibliography Recent articles from ACM Digital Library and IEEE Xplore		

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 Curriculla Recommendations for Computer Science studies;
- The course exists in the studying programs of all major universities in Romania and abroad;

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	Knowing the formalisms for describing concurrency presented during the course	Examination	30%
10.5 Seminar/laboratory	Ability to understand recent research and relate it to theoretical concepts presented at the course	Project Research report	40% 30%
10.6 Minimum standard of performance			
• In order to successfully pass this course, students must get at least 5 at each of the 3 examination tasks.			

11. Labels ODD (Sustainable Development Goals)[2]

[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*.".

Not applicable.

Date:
...

Signature of course coordinator

Assoc.prof.phd. Adrian STERCA

Signature of seminar coordinator

Assoc.prof.phd. Adrian STERCA

Date of approval:
...

Signature of the head of department

Assoc.prof.phd. Adrian STERCA