Study programme(s): Informatics (IM)

Level: master

Course title: Software engineering in critical systems (IB334)

Lecturers: Zoran D. Budimac, Danijela D. Tešendić

Status: elective

ECTS: 7,5

Requirements: None

Learning objectives

To present and critically evaluate critical systems. Requirements for design of critical systems will be presented and the role of formal approaches in life-cycle of critical systems will be evaluated.

Learning outcomes

Minimal: Students should be able to critically evaluate contemporary types of critical systems, including international standards and use of formal methods in the life-cycle of critical systems; to adopt crucial conclusions on time-critical systems in the phases of specification and design.
Optimal: Students should be able to critically evaluate use of temporal logic in engineering and re-engineering of critical systems.

Syllabus

Theoretical instruction

Theoretical foundations of critical systems. Their classifications and analysis including examples and efforts for standardization in the field. Time-critical systems and its technical issues. The role of formal methods and appropriate software in critical systems and real-time software. Formal approaches in life-cycle of critical systems, case studies. Model of a critical system, appropriate calculus, interval-temporal logic, refinement calculus, abstraction calculus and evolution.

Practical instruction

Introduction to formal methods based on models, logic and process algebra; introduction to syntax and semantics of temporal logic with the usage of software tools like executive subset of temporal logic tool “Tempura”; introduction to model of temporal agents and algebraic laws with examples; examples of abstraction rules and tracking evolution wit software tools like “Ana tempura”.

Literature

1. Ian Sommerville, 'Software Engineering, 9th edition', 2010 (chapters 16, 17, 18 and 21)
2. Ben Moszkowski , Executing Temporal Logic Programs, Cambridge Univ. Press (http://www.cse.dmu.ac.uk/~cau/papers/tempura-book.pdf)
3. Michael Huth and Mark Ryan, Logic in Computer Science: Modelling and Reasoning about Systems, Cambridge University Press, 2000
4. Anderson, Ross , Security Engineering, Wiley, 2001
5. Boyd, Colin and Mathuriam, Anish, Protocols for Authentication and Key Establishment, Springer, 2003

Weekly teaching load				Other:
Lectures: 2	Exercises: 3	Other forms of teaching:	Student research:	

Teaching methodology

During lectures classical teaching methods are used, using a video-beam. During exercises, case studies are analyzed theoretically and tried practically by recommended software tools. Students do further research on chosen topics and describe them in seminar papers.

Grading (maximum number of points 100)			
Pre-exam obligations	**points**	**Final exam**	**points**

Active participation in lectures		Written exam	
Practical instruction	**12**	Oral exam	**40**
Colloquia		.....	
Seminar(s)	**48**		