COURSE UNIT DESCRIPTION

Course unit title	Course unit code
Modelling and verification of software-based systems	

Lecturer(s)	Department where the course unit is delivered
Coordinator: prof. dr. Linas Laibinis **Other lecturers:**	Department of Computer Science Faculty of Mathematics and Informatics Vilnius University

Cycle	Type of the course unit
2nd (MA)	Compulsory

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Face-to-face	2nd semester	English, Lithuanian

Prerequisites
Prerequisites: Informatics fundamentals

Number of credits Allocated	Student‘s workload	Contact hours	Individual work
10	270	80	190

Purpose of the course unit: programme competences to be developed

Purpose of the course unit:
to learn how system modelling and verification methods may help to reason and analyse both system design and its functionality, ensuring that the software-based system under construction adheres to the given requirements and/or exhibits the pre-defined properties.

Generic competences:

- Solve problems *(GK2).*

Specific competences:

- Information management and processing *(SK4).*
- Software engineering (*SK5*).

Learning outcomes of the course unit: students will be able to	Teaching and learning methods	Assessment methods
• Understand and apply the essential concepts behind system modelling and verification (e.g., representing a system model in some formal language, expressing the desired qualitative and quantitative system properties, etc.)	Lectures, problem-oriented teaching, case studies, individual and laboratory work with lecturer consultation.	Laboratory works and result presentations (40%), written exam (60%).
• Learn how to formalize a set of system requirements, its architecture or its given properties in the chosen modelling framework		
• Learn how to formulate and verify the pre-defined system properties (functional, safety, liveness, temporal, etc.)		
• Learn the essential features of two different modelling and verification environments (Event-B, Uppaal)		

Course content: breakdown of the topics	**Contact hours**							**Individual work: time and assignments**	
	Lectures	Tutorials	Seminars	Practice	Laboratory work	Practical training	**Contact hours**	**Individual work**	**Assignments**
Overview of existing modelling languages/ approaches/ frameworks. The role of system models in the current system development processes. System verification methods. Environments for automated modelling and verification.	4				2		**6**	**14**	Individual reading. Laboratory works. Self-control tasks.
The notion of an abstract state machine (state transition system). Static and dynamic machine components. Modelling of reactive systems and system events. System invariant properties.	6				8		**14**	**34**	
Formalisation of the given system requirements into the corresponding elements of a system model or the system properties to be verified. The types of system properties: pre/post conditions, invariants, safety, liveness, temporal.	6				8		**14**	**34**	
Reasoning about machine correctness. Relational/weakest precondition semantics of model elements. Proof obligations: invariant preservation, feasibility, etc.	4				4		**8**	**20**	
Modelling system non-determinism. The notions of model refinement and its verification.	4				4		**8**	**20**	
Modelling of various types of software-based systems: control, service-oriented, communicating, safety-critical, agent-based, etc. System safety, fault tolerance, resilience, dynamic reconfiguration as properties to be verified.	6				8		**14**	**34**	
The notion of model checking. Advantages and disadvantages of model checking. Automated techniques and frameworks for model checking.	4				2		**6**	**14**	
Quantitative system properties. Reasoning about performance and time. Timed automata. Modelling systems as parallel synchronising processes. Validation and verification of such systems. Temporal and timing properties.	6				4		**8**	**20**	
Total	**40**				**40**		**80**	**190**	

Assessment strategy	**Weight %**	**Deadline**	**Assessment criteria**
Exercises and laboratory work	40	During the semester	The students are given 4-5 exercises (tasks) to be solved individually and/or in small groups. The solutions must be presented (defended) during exercise sessions. Each exercise gives the same number of points.
Exam (written)	60	Exam session	During the given time, the students solve 2-3 theoretical and practical tasks.

Author	**Publishing year**	**Title**	**Number or volume**	**Publisher or URL**
Required reading				
J-R. Abrial	2010	Modelling in Event-B: System and Software Engineering		Cambridge University Press
Recommended reading				
S. Schneider	2001	The B-method (Cornerstones of Computing)		Palgrave Macmillan
C. Baier and J.-P. Katoen	2008	Principles of Model Checking		The MIT Press