Syllabus for the Course «Model-Based Systems Engineering&Systems Modeling Language»

Course volume: 2 ECTS (72 hours)

Semester: 3

Contact hours:

- lectures — 16 hours;
- seminars and practical classes — 16 hours.
  Independent study: 40 hours
  Form of interim assessment: pass/fail exam with project system model defence
  Course delivery format: AI-augmented engineering learning (learning with intelligent assistants)

1. General Course Characteristics

The discipline “Model-Based Systems Engineering & Systems Modeling Language” belongs to the professional module of the Master’s educational programme in the field 09.04.02 “Information Systems and Technologies”.

The course is delivered in the third semester of the Master’s programme and focuses on Model-Based Systems Engineering (MBSE) methods.

Within the discipline, system modelling is considered as a key tool for designing complex technical and digital systems.

MBSE enables transition:

- from textual system description → to a formal system model that unites:
    - requirements;
    - architecture;
    - behaviour;
    - component interactions.

Special attention is given to:

- SysML (Systems Modelling Language);
- architectural description of complex systems;
- requirement traceability;

- integration of different system views.

The discipline has a methodological and instrumental character and develops students' skills in building system models of digital and cyber-physical systems.

The practical part is built around a cross-cutting system modelling project, where students create a model of a chosen digital system.

Large Language Models (LLM) are used as tools for:

- domain analysis;
- generating alternative architectural solutions;
- verifying system model consistency.

2. Place in the Educational Programme Structure

The discipline occupies a methodological position between system analysis and digital engineering.

It builds on knowledge acquired in:

- *System Analysis and Foundations of Systems Engineering*, which develops skills in:
    - formulating engineering tasks;
    - analysing stakeholders;
    - defining requirements;
    - building system architectural concepts.

If the system analysis course answers the question:

“What is the system and what requirements apply to it?”

then the MBSE course answers:

“How to formally describe the system as an engineering model?”

Mastering this discipline creates a methodological foundation for subsequent courses:

- Digital Twins;
- Information Systems Architecture;
- Distributed Systems;
- Digital Engineering Projects.

Thus, the discipline serves as a transition from system analysis to model-based system design.

3. Learning Objectives

The objective is to develop students' systemic understanding of MBSE methods and master system modelling tools.

Special attention is given to:

- mastering MBSE principles;
- learning system modelling languages;
- mastering SysML;
- developing architectural system description skills;
- modelling system behaviour;
- integrating different system views;
- fostering systemic engineering thinking;
- building a culture of critical LLM assistant use in system modelling.

4. Course Tasks

Key tasks include:

- learning MBSE principles;
- mastering system model building methods;
- learning SysML;
- mastering requirement modelling methods;
- learning structural system modelling;
- mastering component interaction modelling;
- learning system behaviour modelling;
- mastering system model integration;
- learning requirement traceability principles;
- mastering system model verification;
- developing digital system modelling skills;
- using LLM as an engineering analysis tool.

5. Expected Learning Outcomes

Upon completing the discipline, students should:

Know:

- basic systems engineering principles;
- MBSE methods;

- system model structure;
- SysML (Systems Modelling Language);
- requirement modelling methods;
- architectural system modelling methods;
- component interaction modelling methods;
- system behaviour modelling methods;
- requirement traceability principles;
- LLM capabilities and limitations in system modelling.

Be able to:

- build a system model of a complex system;
- model system requirements;
- develop a system architectural model;
- describe component interactions;
- model system operation scenarios;
- analyse system model consistency;
- identify architectural contradictions;
- use LLM for system model analysis.

Possess skills in:

- system modelling methods;
- architectural system analysis tools;
- building SysML diagrams;
- requirement traceability methods;
- preparing system documentation;
- critical use of LLM assistants.

6. Methodological Concept

6.1 Model-Based Systems Engineering

MBSE is viewed as an approach where the model becomes the central engineering artefact. The system model unites:

- requirements;
- architecture;
- functions;
- component interactions;
- system behaviour.

This approach improves design consistency for complex systems.

6.2 System Model as Integration of Different Views

A system can be described through several interconnected views:

- requirement model;
- structural architecture;
- interaction model;
- behaviour model.

Integrating these views forms a holistic system model.

6.3 Systems Modelling Language (SysML)

SysML is used as a language for describing complex systems. Key model types studied:

- Requirements diagram;
- Block Definition Diagram (BDD);
- Internal Block Diagram;
- Activity Diagram;
- Sequence Diagram.

6.4 Link Between System Modelling and Digital Twins

System models are the basis for building digital models and digital twins. Formal description of system structure and behaviour enables:

- simulating system operation;
- analysing various operation scenarios;
- predicting system behaviour.

7. LLM Use in Education

LLMs support system modelling in several modes:

- LLM as a handbook — explaining systems engineering and modelling language terms;
- LLM as an analytical assistant — helping:
  - analyse domains;
  - identify possible architectural solutions;
  - generate system model variants;
- LLM as an engineering opponent — identifying:
  - requirement contradictions;
  - architectural inconsistencies;
  - incomplete system scenarios.

Mandatory condition: engineering verification of LLM recommendations.

8. Educational Technologies

The course uses AI-augmented learning, combining traditional forms with intelligent assistants:

- lectures;
- seminars;
- practical and digital laboratory work;
- engineering case analysis;
- cross-cutting system modelling project;
- using LLM assistants for analysis and reflection.

9. Independent Study Organisation

Independent study develops system modelling skills. It includes:

- analysing system modelling cases;
- preparing system models;
- building SysML diagrams;
- completing final project stages;
- preparing analytical materials;
- using LLM assistants for model analysis.

10. Final Assessment

Form: pass/fail exam via defence of a digital system's system model developed by the student.

Final project must include:

- problem situation description;
- stakeholder analysis;
- system operation scenarios;
- requirement model;
- system architectural model;
- component interaction model;
- system behaviour model;
- requirement traceability analysis;
- description of digital model or digital twin prospects.

CALENDAR-THEMATIC SCHEDULE

Abbreviations:

- IS — Independent Study;
- LLM — Large Language Model;
- MBSE — Model-Based Systems Engineering;
- TRL — Technology Readiness Level.

Week	Content	Lect. (h)	Sem. (h)	IS (h)
1	**Lecture:** Complexity of modern digital systems and need for modelling. — Growth of complexity in digital and cyber-physical systems (cloud platforms, IoT ecosystems, distributed services); — Limitations of document-based design; — Model as a tool for managing system complexity. **Practical class:** Analysing complex system examples (digital platform, city digital service, IoT system). Students define system boundaries, key components and environment interactions. **LLM use:** Primary domain analysis — identifying possible system components and external actors, followed by critical result verification. **IS:** Essay: “Why modern digital systems require system modelling”.	1	1	2
2	**Lecture:** Model-Based Systems Engineering. — MBSE as an approach with the model as the central artefact; — Transition from document-based to unified system model. **Practical class:** Comparing document-based and model-based development using digital system	1	1	2

	examples. **LLM use:** LLM as an analytical opponent in comparing development approaches. **IS:** Review of MBSE application in a chosen area (IoT, digital platforms, etc.).			
3	**Lecture:** System model as an engineering artefact. — Elements of a system model: needs, requirements, functions, architecture, behaviour; — Role of the model as the basis for engineering decisions. **Practical class:** Building a conceptual model of the chosen system. **LLM use:** Using LLM to generate model structure and verify element completeness. **IS:** Preparing a description of the system's conceptual model.	1	1	2
4	**Lecture:** Modelling languages UML and SysML. — Overview of key modelling languages, their origins and application areas; — Differences between UML and SysML, their roles in systems engineering. **Practical class:** Reviewing main UML and SysML diagrams and their applicability to digital systems. **LLM use:** LLM as an interactive reference for diagram types. **IS:** Comparative analysis of UML and SysML.	1	1	2
5	**Lecture:** Requirements Engineering in MBSE. — Transition from user needs to system requirements; — Requirement traceability and its role in architecture design. **Practical class:** Creating a requirement model and building a Requirements Diagram. **LLM use:**	1	1	2

	Generating alternative requirements based on need descriptions. **IS:** Preparing a system requirement list.			
6	**Lecture:** Structural system modelling. — System architecture, components and interfaces; — Role of architectural levels and system modularity. **Practical class:** Building a Block Definition Diagram (BDD). **LLM use:** Generating alternative architectural system variants. **IS:** Preparing system architecture description.	1	1	2
7	**Lecture:** Component interaction within a system. — Interfaces and data flows between subsystems. **Practical class:** Building an Internal Block Diagram. **LLM use:** Analysing possible data flows and component interactions. **IS:** Describing component interactions within the system.	1	1	2
8	**Lecture:** Functional system architecture. — Functional decomposition methods; — Linking system functions to architecture. **Practical class:** Building an Activity Diagram. **LLM use:** Generating alternative functional system scenarios. **IS:** Describing system functions.	1	1	2
9	**Lecture:** System interaction scenarios. — Operational scenarios of system functioning. **Practical class:** Building a Sequence Diagram. **LLM use:** Generating alternative component interaction scenarios. **IS:** Preparing system scenario descriptions.	1	1	2

10	**Lecture:** System states and lifecycle. — System states and transition events between them. **Practical class:** Building a state diagram. **LLM use:** Identifying possible system states and transitions. **IS:** Describing the system lifecycle.	1	1	2
11	**Lecture:** Integrating the system model. — Aligning requirements, architecture and system behaviour. **Practical class:** Integrating developed diagrams into a unified system model. **LLM use:** Verifying model element consistency. **IS:** Refining the system model.	1	1	2
12	**Lecture:** Verifying and validating system models. — Methods for checking model correctness. **Practical class:** Analysing the system model and identifying inconsistencies. **LLM use:** Using the model as an engineering opponent. **IS:** Preparing a model verification report.	1	1	3
13	**Lecture:** MBSE and digital system architecture. — Features of digital platform and distributed service architecture. **Practical class:** Building an architectural model of a digital system. **LLM use:** Analysing typical digital platform architectures. **IS:** Preparing an architectural system description.	1	1	3
14	**Lecture:** MBSE and digital twins. — Link between system models and digital twins. **Practical class:** Analysing possibilities for creating a digital twin of the	1	1	3

	selected system. **LLM use:** Searching and analysing digital twin case studies. **IS:** Preparing the project section on the digital model.			
15	**Practical class:** Working on the final system modelling project. — Discussing students' preliminary models, identifying inconsistencies. **LLM use:** Engineering audit of the system model using LLM. **IS:** Finalising the project.	0	2	4
16	**Practical class:** Consultations on final projects. — Discussing system model structure and logic, fixing errors. **LLM use:** Self-audit of the project before the pass/fail exam. **IS:** Final preparation of the project for defence.	0	2	5

FINAL MBSE PROJECT STRUCTURE (10–12 pages)

System and problem situation description (≈ 1 page):
- brief domain description;
- problem being solved;
- system usage context.
- *Example:* Digital platform for managing city parking spaces.

Stakeholders and needs (≈ 1 page):
- table format: Stakeholder | Need;
- shows how requirements derive from needs.

Operational scenarios (≈ 1–1.5 pages):
- 2–3 scenarios (e.g., parking search, booking);
- bridge between needs and requirements.

Requirement model (≈ 1–1.5 pages):
- requirement table (ID, Requirement);
- Requirements Diagram;
- relations: derive, satisfy, verify.

System architecture (≈ 2 pages):
- Block Definition Diagram (BDD);
- component roles and interfaces;

- *Example architecture:* Parking System with Mobile App, Parking Detection Service, etc.

Component interaction (≈ 1–1.5 pages):
- Internal Block Diagram or Sequence Diagram;
- shows architectural dynamics.

System behaviour (≈ 1–1.5 pages):
- Activity Diagram;
- describes functional logic (e.g., detect free parking → offer options → user selects → reserve).

Requirement traceability (≈ 1 page):
- table: Requirement | Component;
- demonstrates MBSE traceability principle.

Digital twin perspective (≈ 1 page):
- potential uses: load forecasting, parking placement optimisation, traffic simulation.

Final diagram set (4–5 diagrams recommended):

- Requirements Diagram;
- Block Definition Diagram;
- Internal Block Diagram;
- Activity Diagram;
- Sequence Diagram (optional).