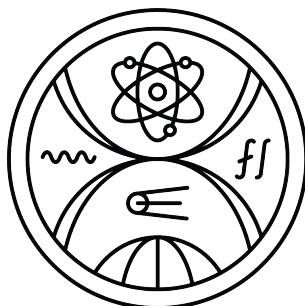


COMENIUS UNIVERSITY IN BRATISLAVA  
FACULTY OF MATHEMATICS PHYSICS AND INFORMATICS

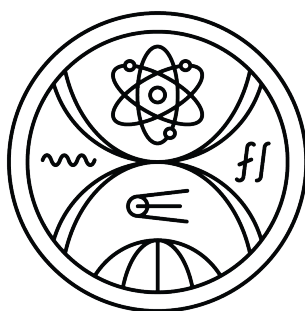


# AI-ASSISTED SOFTWARE MODELLING

Master thesis



COMENIUS UNIVERSITY IN BRATISLAVA  
FACULTY OF MATHEMATICS PHYSICS AND INFORMATICS



# AI-ASSISTED SOFTWARE MODELLING

Master thesis

Study program: Applied informatics  
Branch of study: Applied informatics  
Department: Department of Applied Informatics  
Supervisor: Ing. Lukáš Radoský





Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky

## ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:** Viktor Ovchinnikov  
**Študijný program:** aplikovaná informatika (konverzný program)  
(Jednoodborové štúdium, magisterský II. st., denná forma)  
**Študijný odbor:** informatika  
**Typ záverečnej práce:** diplomová  
**Jazyk záverečnej práce:** anglický  
**Sekundárny jazyk:** slovenský

**Názov:** AI-assisted software modelling  
*Modelovanie softvéru asistované umelou inteligenciou*

**Anotácia:** Rozsiahle jazykové modely rozumejú okrem prirodzeného jazyka aj formálnym jazykom, vrátane programovacích jazykov. Schopnosti týchto modelov im umožňujú modifikovať, generovať, alebo, naopak, interpretovať zdrojový kód v mnohých jazykoch. Komerčné riešenia pre programovanie s asistentom založeným na rozsiahlych jazykových modeloch, resp. modeloch umelej inteligencie sú dnes bežne dostupné.

Tak ako vo fáze implementácie sú využívané programovacie jazyky, vo fáze návrhu sú využívané modely, resp. diagramy softvéru. Takéto diagramy možno zapísať pomocou deklaratívnych jazykov, napr. PlantUML. Analyzujte možnosti využitia rozsiahlych jazykových modelov pri modelovaní softvéru. Implementujte prototyp pre modelovanie softvéru asistované rozsiahlym jazykovým modelom.

**Cieľ:** Analýza existujúcich riešení modelovania softvéru asistovaného rozsiahlym jazykovým modelom  
Implementácia prototypu pre modelovanie softvéru asistované rozsiahlym jazykovým modelom

**Literatúra:** Chunqiu Steven Xia, Yinlin Deng, Soren Dunn, & Lingming Zhang. (2024). Agentless: Demystifying LLM-based Software Engineering Agents.

Radford et al. Better language models and their implications.

A. Fan et al., "Large Language Models for Software Engineering: Survey and Open Problems," 2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE), Melbourne, Australia, 2023, pp. 31-53, doi: 10.1109/ICSE-FoSE59343.2023.00008.

### Kľúčové

**slová:** jazykový model, modelovanie softvéru, virtuálny asistent

**Vedúci:** Ing. Lukáš Radoský  
**Katedra:** FMFI.KAI - Katedra aplikovanej informatiky  
**Vedúci katedry:** doc. RNDr. Tatiana Jajcayová, PhD.



Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky

---

**Dátum zadania:** 18.11.2024

**Dátum schválenia:** 21.11.2024

prof. RNDr. Roman Ďurikovič, PhD.  
garant študijného programu

.....  
š t u d e n t

.....  
v e d ú c i   p r á c e

I hereby declare that I have written this thesis by myself, only with help of referenced literature, under the careful supervision of my thesis advisor.

Bratislava, 2025

.....  
Bc. Viktor Ovchinnikov





# Acknowledgement

# Abstract

Large language models understand formal languages, including programming languages, in addition to natural language. The capabilities of these models allow them to modify, generate, or, conversely, interpret source code in many languages. Commercial solutions for assistant programming based on large-scale language models or artificial intelligence models are widely available today. Just as programming languages are used in the implementation phase, software models or diagrams are used in the design phase. Such diagrams can be written using declarative languages such as PlantUML. The present study analyzes the potential applications of large language models in the field of software modeling. The implementation of a prototype for software modeling is achieved through the utilization of a large-scale language model. We initiated the development of a copilot system with the objective of enhancing the functionality of the existing AnimArch project. The primary objective of this system is to provide assistance to the user in the domain of software modeling.

**Keywords:** LLM, UML, Class Diagrams, Software prototyping

# Abstrakt

Veľké jazykové modely rozumejú okrem prirodzeného jazyka aj formálnym jazykom vrátane programovacích jazykov. Schopnosti týchto modelov im umožňujú upravovať, generovať alebo naopak interpretovať zdrojový kód v mnohých jazykoch. Komerčné riešenia na asistenčné programovanie založené na rozsiahlych jazykových modeloch alebo modeloch umelej inteligencie sú dnes široko dostupné. Tak ako sa programovacie jazyky používajú vo fáze implementácie, tak sa softvérové modely alebo diagramy používajú vo fáze návrhu. Takéto diagramy možno napísať pomocou deklaratívnych jazykov, ako je PlantUML. Táto štúdia analyzuje potenciálne aplikácie veľkých jazykových modelov v oblasti modelovania softvéru. Implementácia prototypu pre modelovanie softvéru sa dosahuje prostredníctvom využitia rozsiahleho jazykového modelu. Začal sa vývoj kopilotného systému s cieľom rozšíriť funkčnosť existujúceho projektu AnimArch. Primárnym cieľom tohto systému je poskytovať pomoc používateľovi v oblasti modelovania softvéru.

**Kľúčové slová:** LLM, UML, Diagramy tried, Prototypovanie softvéru

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	LLMs . . . . .	2
1.2	Applications of LLMs . . . . .	2
<b>2</b>	<b>Related Work</b>	<b>3</b>
<b>3</b>	<b>System Design Overview</b>	<b>4</b>
<b>4</b>	<b>Implementation of a prototype</b>	<b>5</b>
<b>5</b>	<b>User evaluation</b>	<b>6</b>
<b>6</b>	<b>Results</b>	<b>7</b>

# List of Figures

# List of Tables

# Terminology

## Terms

## Abbreviations





# Motivation

# Chapter 1

## Introduction

### 1.1 LLMs

### 1.2 Applications of LLMs

## Chapter 2

### Related Work

# Chapter 3

## System Design Overview

## Chapter 4

### Implementation of a prototype

# Chapter 5

## User evaluation

# Chapter 6

## Results

# Conclusion



# Bibliography

- [1] Javier Cámara, Javier Troya, Lola Burgueño, and Antonio Vallecillo. On the assessment of generative ai in modeling tasks: an experience report with chatgpt and uml. *Software and Systems Modeling*, 22(3):781–793, 2023.
- [2] Alessio Ferrari, Sallam Abualhaijal, and Chetan Arora. Model generation with llms: From requirements to uml sequence diagrams. In *2024 IEEE 32nd International Requirements Engineering Conference Workshops (REW)*, pages 291–300. IEEE, 2024.
- [3] Oksana Nikiforova, Konstantins Gusarovs, Ludmila Kozacenko, Dace Ahilcenoka, and Dainis Ungurs. An approach to compare uml class diagrams based on semantical features of their elements. In *The tenth international conference on software engineering advances*, pages 147–152, 2015.
- [4] Lukas Radosky and Ivan Polasek. Executable multi-layered software models. In *Proceedings of the 1st International Workshop on Designing Software*, Designing ’24, page 46–51, New York, NY, USA, 2024. Association for Computing Machinery.
- [5] Mina Shehata, Blaire Lepore, Hailey Cummings, and Esteban Parra. Creating uml class diagrams with general-purpose llms. In *2024 IEEE Working Conference on Software Visualization (VISSOFT)*, pages 157–158, 2024.
- [6] Christof Tinnes, Alisa Welter, and Sven Apel. Software model evolution with large language models: Experiments on simulated, public, and industrial datasets. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pages 649–649. IEEE Computer Society, 2025.