

UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



# VIZUALIZÁCIA SOFTVÉROVÝCH ARCHITEKTÚR A GENEROVANIE ZDROJOVÉHO KÓDU

Diplomová práca

2022

Bc. Filip Novák

UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



# VIZUALIZÁCIA SOFTVÉROVÝCH ARCHITEKTÚR A GENEROVANIE ZDROJOVÉHO KÓDU

Diplomová práca

Študijný program: Aplikovaná informatika  
Študijný odbor: 2511 Aplikovaná informatika  
Školiace pracovisko: Katedra aplikovanej informatiky  
Školiteľ: doc. Ing. Ivan Polášek, PhD.

Bratislava, 2022

Bc. Filip Novák



## ZADANIE ZÁVEREČNEJ PRÁCE

- Meno a priezvisko študenta:** Bc. Filip Novák  
**Študijný program:** aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)  
**Študijný odbor:** informatika  
**Typ záverečnej práce:** diplomová  
**Jazyk záverečnej práce:** slovenský  
**Sekundárny jazyk:** anglický
- Názov:** Vizualizácia softvérových architektúr a generovanie zdrojového kódu  
*Visualization of software architectures and source code generation*
- Anotácia:** Pre pochopenie rozsiahlych softvérových systémov je užitočné poznať nielen ich štruktúru ale aj dynamickú povahu, scenáre funkcionality a prípadov použitia, ako aj interakcie medzi prvkami softvérovej architektúry.
- Analyzujte vybrané metódy modelovania v softvérovom inžinierstve (napríklad Executable UML a OAL), interaktívnej grafiky v Unity a prototypu animácie UML modelu z roku 2019/2020. Modely je možné vytvárať v CASE nástroji Enterprise Architect a v samotnom prototypu, naprogramovanom v C# v spolupráci s rámcom Unity. Platforma Unity umožňuje prácu v 2D ale aj 3D priestore a migrovať aj do virtuálnej (VR) alebo rozšírenej reality (AR). Práca bude súčasťou rozbiehaného výskumu podpory kolaboratívneho modelovania a vizualizácie vo VR/AR priestore.
- Navrhňte možnosť vytvorenia úložiska scenárov, architektonických štýlov a vzorov. Doplňte vznikajúci prototyp aj o možnosť generovania zdrojového kódu.
- Výstupom DP bude nástroj, obohatený o bázu konkrétnych scenárov, štýlov a vzorov, ktorý by sa dal používať pri modelovaní a vizualizácii softvérovej štruktúry a funkcionality alebo na testovanie funkčnosti a animácie architektonických štýlov a vzorov. Pomohol by aj pri výučbe softvérového inžinierstva na vysvetľovanie modelov, štýlov a vzorov a na podporu experimentovania.
- Cieľ:**
- Metóda fúzie dynamického a štruktúrneho modelu na vizualizáciu scenáru prípadov použitia a funkcionality architektúry
  - Generovanie zdrojového kódu
  - Vytvorenie katalógu štýlov a vzorov pre skúmanú metódu
  - Vykonanie experimentov s predpripraveným katalógom vzorov na verifikáciu vhodnosti navrhovanej metódy
- Literatúra:** JOUAULT, Frédéric, et al. Designing, animating, and verifying partial UML Models. In: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems. 2020. p. 211-217.



Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky

---

Buschmann F. et al.: Pattern-oriented software architecture: a pattern language for distributed computing, Vol. 4. New York : John Wiley & Sons, 2007.

**Vedúci:** doc. Ing. Ivan Polášek, PhD.  
**Katedra:** FMFI.KAI - Katedra aplikovanej informatiky  
**Vedúci katedry:** prof. Ing. Igor Farkaš, Dr.

**Spôsob sprístupnenia elektronickej verzie práce:**  
bez obmedzenia

**Dátum zadania:** 10.12.2020

**Dátum schválenia:** 10.12.2020

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

.....  
študent

.....  
vedúci práce

Čestne prehlasujem, že túto diplomovú prácu som  
vypracoval samostatne len s použitím uvedenej literatúry  
a za pomoci konzultácií u môjho školiteľa.

Bratislava, 2022

.....

Bc. Filip Novák

# Pod'akovanie

# Abstrakt

Klíčové slova:

# Abstract

Keywords:



# Obsah

Úvod	1
Motivácia	2
1 Prehľad a oboznámenie sa s problematikou	3
2 Návrh vylepšenia a pridania funkcionalít	4
3 Implementácia	5
4 Tvorba katalógu štýlov a vzorov	6
5 Výsledky experimentov	7
6 Záver	8

# Úvod

# Motivácia

Na tejto diplomovej práci ma najviac motivuje to, že jej cieľom je pomôcť ľuďom tým, že im jednoducho vysvetlíme zdanlivo možno pre nich ťažkú tému na pochopenie pomocou nášho nástroja AnimArch. Na zložitých diagramoch tried dokážeme ukázať pomocou animácie beh a volanie jednotlivých metód a ich vykonávanie v telách týchto metód, aby ľudia pochopili ako funguje program, ktorý je reprezentovaný diagramom tried alebo aj inými diagramami.

# Kapitola 1

## Prehľad a oboznámenie sa s problematikou

V tejto kapitole si spravíme prehľad problematiky a oboznámime sa s projektom AnimArch. Vysvetlíme si jeho používanie a aj jeho funkcionality.

## Kapitola 2

### Návrh vylepšenia a pridania funkcionalít

# Kapitola 3

## Implementácia

V tejto časti sa zameriame hlavne na implementáciu parsera z jazyka OAL do jazyka Python.

# Kapitola 4

## Tvorba katalógu štýlov a vzorov

Táto kapitola sa zameria na tvorbu a prehľad katalógu rôznych štýlov a vzorov.

# Kapitola 5

## Výsledky experimentov

Táto kapitola sa zameria na vyhodnotenie našich dosiahnutých výsledkov.



## Kapitola 6

## Záver

# Literatúra

- [FPV17] Matej Ferenc, Ivan Polasek, and Juraj Vincúr. Collaborative modeling and visualization of software systems using multidimensional uml. In *2017 IEEE Working Conference on Software Visualization (VISSOFT)*, pages 99–103. IEEE, 2017.
- [GHJV94] Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1. edition, 1994.
- [GS93] David Garlan and Mary Shaw. An introduction to software architecture. In *Advances in software engineering and knowledge engineering*, pages 1–39. World Scientific, 1993.
- [JLSC18] Rodi Jolak, Khan-Duy Le, Kaan Burak Sener, and Michel RV Chaudron. Octobubbles: A multi-view interactive environment for concurrent visualization and synchronization of uml models and code. In *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 482–486. IEEE, 2018.
- [Sel03] Bran Selic. The pragmatics of model-driven development. *IEEE software*, 20(5):19–25, 2003.

## Zoznam obrázkov