

**Žilinská univerzita v Žiline**  
**Spoločnosť pre otvorené informačné technológie**  
**Geokomunita, združenie CSTUG**  
**Česká statistická společnost, INO-HUB Energy**

**OTVORENÝ SOFTVÉR VO VZDELÁVANÍ,  
VÝSKUME A V IT RIEŠENIACH**



**Zborník príspevkov medzinárodnej konferencie**  
**OSSConf 2026**

**1. júla – 3. júla 2026**  
**Žilina, Slovensko**

## **Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach**

1. júla – 3. júla 2026, Žilina, Slovensko

### **Vedeckí garanti konferencie:**

doc. Ing. Michal Koháni, PhD., prorektor pre informačné systémy, UNIZA Žilina

doc. Ing. Miroslav Kvaššay, PhD., prodekan pre vedu a výskum, FRI UNIZA Žilina

### **Vedecký a programový výbor:**

doc. Ing. Miroslav Kvaššay, PhD., UNIZA Žilina – **predseda**

doc. Ing. Peter Márton, PhD., UNIZA Žilina

doc. PaedDr. Dalibor Gonda, PhD., UNIZA Žilina

doc. Ing. Jiří Rybička, Dr., MENDELU, Brno (CZ)

doc. Ing. Zdena Dobešová, Ph.D., Univerzita Palackého, Olomouc (CZ)

doc. Mgr. Ondřej Vencálek, Ph.D., PřF UP Olomouc, Česká statistická společnost (CZ)

doc. Ing. Milan Straka, PhD., UNIZA, Žilina

RNDr. Aleš Kozubík, PhD., UNIZA, Žilina

RNDr. Rudolf Blaško, PhD., UNIZA, Žilina

Ing. Tomáš Majer, PhD., UNIZA, SOIT, Žilina

Mgr. Miloslav Ofúkaný, Ministerstvo vnútra SR, Bratislava

Dr inż. Radosław Maciaszczyk, ZUT w Szczecinie (PL)

Dr inż. Mirosław Łazoryszczak, ZUT w Szczecinie (PL)

Ing. Jan Růžicka, Ph.D., GISMentors, Ostrava (CZ)

Mgr. Radim Štampach, Ph.D., Masarykova univerzita, Missing Maps, Brno (CZ)

Ing. Pavel Stříž, Ph.D., Bučovice (CZ)

### **Organizačný výbor:**

Rudolf Blaško, UNIZA, Žilina – **predseda**

Miroslav Kvaššay, UNIZA Žilina

Tomáš Majer, UNIZA, SOIT, Žilina

Miloslav Ofúkaný, Ministerstvo vnútra SR, Bratislava

Pavel Stříž, Bučovice

Peter Márton, UNIZA Žilina

Vydavateľ: Žilinská univerzita v Žiline

ISBN 978-80-554-2211-4

---

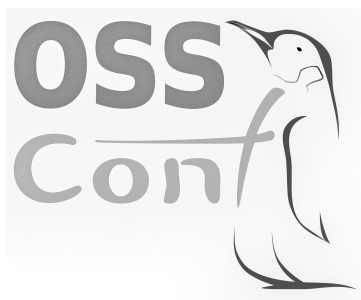
Copyright © 2026 autori príspevkov

Ktokoľvek má dovolenie vyhotoviť alebo distribuovať doslovný opis tohoto dokumentu alebo jeho časti akýmkoľvek médium za predpokladu, že bude zachované oznámenie o copyrighte a o tom, že distribútor príjemcovi poskytuje povolenie na ďalšie šírenie, a to v rovnakej podobe, akú má toto oznámenie.



**Žilinská univerzita v Žiline**  
**Spoločnosť pre otvorené informačné technológie**  
**Geokomunita, združenie CSTUG**  
**Česká statistická společnost, INO-HUB Energy**

**OTVORENÝ SOFTVÉR VO VZDELÁVANÍ,  
VÝSKUME A V IT RIEŠENIACH**



**Zborník príspevkov medzinárodnej konferencie**  
**OSSConf 2026**

**1. júla – 3. júla 2026**  
**Žilina, Slovensko**

**Recenzenti:**

Blaško Rudolf, RNDr. PhD., Borčinová Zuzana, RNDr. PhD.,  
Cézová Eliška, Bc. Ing. Ph.D., Dobešová Zdena, doc. Ing. Ph.D.,  
Drusa Eduard, Ing., Fabo Richard, Ing., Fedorik Slavko, Ing., Hoftich Michal, Mgr.,  
Kaukič Michal, Mgr. CSc., Koháni Michal, doc. Ing. PhD.,  
Koreň Milan, doc. Mgr. PhD., Kozubík Aleš, RNDr. PhD.,  
Kvaššay Miroslav, doc. Ing. PhD., Ledvinka Ondřej, Mgr. Ph.D.,  
Łazoryszczak Mirosław, Dr inż., Maciaszczyk Radosław, Dr inż.,  
Márton Peter, doc. Ing. PhD., Mudra Jiří, Mgr. Ph.D., Nevřiva Petr, Ing.,  
Ofúkaný Miloslav, Mgr., Pavlík Petr, Ing. Ph.D., Peško Štefan, doc. RNDr. PhD.,  
Růžička Jan, Ing. Ph.D., Rybička Jiří, doc. Ing. Dr., Stříž Pavel, Ing. Ph.D.,  
Toth Štefan, Ing. PhD., Velyčko Petr, Bc.

**Editor:**

Rudolf Blaško

Všetky práce, uverejnené v zborníku, boli posúdené dvomi nezávislými recenzentmi.

Za jazykovú úroveň zodpovedajú autori príspevkov.

# Obsah

|                                                                                                                                                                                  |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Úvod</b>                                                                                                                                                                      | 5  |
| <b>Ivan Doboš</b><br>Kvantové počítače: kde sme dnes a kam smerujeme (vyžiadaná prednáška) ..                                                                                    | 7  |
| <b>Antoni Czołgowski</b><br>Cultural Discrimination in Large Language Models: Detection, Measurement,<br>and Mitigation Through Targeted Fine-Tuning (vyžiadaná prednáška) ..... | 8  |
| <b>Michal Lekýr</b><br>3D Scanning in Practice (vyžiadaná prednáška) .....                                                                                                       | 10 |
| <b>Petr Olšák</b><br>Seznámení s OpT <sub>E</sub> Xem (vyžiadaná prednáška) .....                                                                                                | 12 |
| <b>Nirmal Parmar</b><br>Machine Gnostics: A Step Towards Non-Statistical Machine Learning<br>(vyžiadaná prednáška) .....                                                         | 13 |
| <b>Jiří Šperka</b><br>Fantazie generativní umělé inteligence v technických oborech (vyžiadaná<br>prednáška) .....                                                                | 14 |
| <b>Miroslav Bendík</b><br>Shadertoy – medzi matematikou a umením .....                                                                                                           | 15 |
| <b>Rudolf Blaško</b><br>TikZ – obrázky jednoducho a hravo .....                                                                                                                  | 21 |
| <b>Antoni Czołgowski</b><br>Cultural Misalignment in LLMs .....                                                                                                                  | 37 |
| <b>Eduard Drusa</b><br>CMRX: nový prístup k ochrane pamäte .....                                                                                                                 | 47 |
| <b>Richard Fabo</b><br>L <sup>A</sup> T <sub>E</sub> X, ORG a babylón jazykov .....                                                                                              | 55 |
| <b>Richard Fabo</b><br>Typst – výzva pre L <sup>A</sup> T <sub>E</sub> X alebo len hype? .....                                                                                   | 63 |
| <b>Ondřej Ledvinka</b><br>Paralelizace práce s rastry v R .....                                                                                                                  | 75 |

|                                                                                            |     |
|--------------------------------------------------------------------------------------------|-----|
| <b>Michal Lekýr</b>                                                                        |     |
| 3D Scanning in Practice .....                                                              | 83  |
| <b>Peter Márton, Branislav Blažek</b>                                                      |     |
| Využitie dát Strava Metro v Žiline a okolí .....                                           | 97  |
| <b>Peter Márton, Marián Gogola, Romana Vaščáková</b>                                       |     |
| Využitie dát Strava... ..                                                                  | 107 |
| <b>Tibor Menyhért</b>                                                                      |     |
| Riešenie sudoku pomocou UI .....                                                           | 117 |
| <b>Tibor Menyhért</b>                                                                      |     |
| Tvorba interaktívnych digitálnych pomôcok pomocou UI .....                                 | 125 |
| <b>Petr Olšák</b>                                                                          |     |
| Seznámení s OpTeXem .....                                                                  | 129 |
| <b>Nirmal Parmar</b>                                                                       |     |
| Machine Gnostics – A step towards non-statistical Data Analysis and Machine Learning ..... | 137 |
| <b>Ján Rabčan</b>                                                                          |     |
| Integrating Fuzzy Decision Trees .....                                                     | 147 |
| <b>Jiří Rybička</b>                                                                        |     |
| Laboratoř řízení kolejových vozidel – Škola hrou .....                                     | 161 |
| <b>Tomáš Sobek, Miroslav Kvaššay, Zidong Lin</b>                                           |     |
| NMR Quantum Computer for Teaching .....                                                    | 169 |
| <b>Štefan Toth, Jana Tothová</b>                                                           |     |
| Využitie LLM vo verejnom obstarávaní .....                                                 | 179 |
| <b>Štefan Toth, Jana Tothová</b>                                                           |     |
| Preklad kníh pomocou LLM .....                                                             | 187 |
| <b>Zdeněk Wagner</b>                                                                       |     |
| Kolaborativní příprava sborníku .....                                                      | 197 |
| <b>Zdeněk Wagner</b>                                                                       |     |
| Úprava fotografií v Darktable .....                                                        | 205 |
| <b>Abstrakty nerecenzovaných ukážok, prezentácií a workshopov</b>                          | 215 |

Vážení kolegovia, priatelia a priaznivci otvorených technológií,

do rúk sa Vám dostáva zborník príspevkov zo 14. ročníka konferencie OSSConf 2026 – Otvorený softvér vo vzdelávaní, výskume a v IT riešeníach. Aj tento rok sa začiatkom júla stretávame na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline, kde si OSSConf za roky svojej existencie vybudovala pevné miesto. Štrnásty ročník prichádza po symbolickej trinástke. Štrnástku môžeme s trochou nadhľadu vnímať aj ako dve sedmičky. Ak sedmička znamená šťastie, potom sa v otvorenom softvéri šťastie násobí spoluprácou. Jeden dobrý nápad je dôležitý, no skutočnú silu získa až vtedy, keď ho komunita vyskúša, opraví, rozvinie a použije v praxi.



Práve takouto komunitou je aj OSSConf. Konferencia je pre nás viac než len priestorom na prednášky a odborné príspevky. Je predovšetkým stretnutím ľudí, ktorí veria, že zdieľať poznanie, skúsenosti a zručnosti má zmysel. Otvorený softvér už dávno prerástol rolu obyčajnej alternatívy ku komerčným riešeniam. Používame ho vo výučbe, výskume, pri spracovaní dát, vývoji softvéru, správe systémov, v priemysle aj v komunitných projektoch. Jeho hodnota nie je iba v licencií, ale aj v možnosti porozumieť riešeniu, overiť ho, prispôbiť svojim potrebám a posunúť ho ďalej.

Aj program tohto ročníka ukazuje, do koľkých oblastí dnes otvorené technológie zasahujú. Nosnou témou je umelá inteligencia – veľké jazykové modely a ich využitie pri výučbe, analýze dát, tvorbe technických špecifikácií aj samotnom programovaní. Popri nej má svoje pevné miesto práca s otvorenými dátami a jazykom R, štatistické spracovanie a vizualizácie, ako aj digitálna sadzba a technická dokumentácia, kde sa stretávajú  $\text{\TeX}$ ,  $\text{\LaTeX}$ ,  $\text{\OpTeX}$ , Typst, Markdown či Pandoc. Samostatnú líniu tvoria otvorené geografické informačné systémy, mapovanie a občianska veda. Výraznú stopu zanecháva aj kyberbezpečnosť a bezpečný vývoj softvéru a nechýbajú ani otvorený hardvér, internet vecí, Linux, výpočtová infraštruktúra či kvantové výpočty. Dôležitejšie než samotné nástroje je však to, ako pomáhajú riešiť skutočné problémy v školách, laboratóriách, firmách, verejných inštitúciách aj dobrovoľníckych komunitách.

Zborník, ktorý držíte v rukách, zachytáva časť tejto pestrej mozaiky. Obsahuje odborné štúdie, praktické návody, skúsenosti z výučby, výsledky výskumu aj opisy projektov, ktoré vznikli z konkrétnej potreby. Texty sú rôznorodé, no spája ich spoločná myšlienka: dobré riešenie má byť možné pochopiť, overiť, použiť, upraviť a ďalej rozvíjať.

Že tento zborník a celá konferencia vôbec vznikli, je zásluhou mnohých ľudí. Spomedzi všetkých, ktorých mená nájdete v jednotlivých výboroch konferencie, by som chcel osobitne poďakovať Pavlovi Střížovi, Rudolfovi Blaškovi, Lukášovi Čechovičovi a Alešovi Kozubíkovi. Alešovi, predchádzajúcemu dlhoročnému predsedovi organizačného výboru, ďakujem za roky práce, ktoré významne prispeli ku kontinuite a dobrému menu konferencie. Lukášovi by som chcel poďakovať za prípravu originálnych 3D konferenčných menoviek pre účely konferencie. Rudovi, ktorý ma naháňa, keď nestíham, ďakujem za to, že dlhodobo riadi organizáciu konferencie na fakulte. Vďaka nemu sa Vám do rúk dostáva aj tento zborník. A napokon jedno obrovské ďakujem patrí Pavlovi, ktorý ešte pred

skončením minuloročnej konferencie začal pripravovať tú tohtoročnú. Aj jeho príčinám je tento ročník mimoriadne bohatý na rôznorodé príspevky súvisiace s informatikou a otvorenými technológiami. Oslovil množstvo ľudí, ktorí prijali pozvanie na konferenciu, pravidelne aktualizuje obsah konferenčného webu a celkovo rieši veľkú časť každodennej organizačnej práce.

Rád by som tiež poďakoval autorom, prednášajúcim, vedúcim seminárov, recenzentom, členom vedeckého, programového a organizačného výboru, ako aj všetkým účastníkom za atmosféru, vďaka ktorej je OSSConf nielen odborným, ale aj priateľským stretnutím. Spomedzi podporovateľov osobitne ďakujem Fakulte riadenia a informatiky Žilinskej univerzity v Žiline za dlhodobé zázemie, spoločnosti INO-HUB Energy j.s.a. a občianskemu združeniu ALUMNI FRI za ich podporu pri finančnom a organizačnom zabezpečení konferencie. K organizácii konferencie, ako aj k príprave a vydaniu tohto zborníka, pomohol taktiež grant KEGA 010ŽU-4/2026 „Matematika ako jazyk: myslenie, diskurz a porozumenie pomocou Open Source nástrojov“ podporený slovenskou Kultúrnou a edukačnou grantovou agentúrou MŠVVaM SR.

Dúfam, že tento zborník bude pre čitateľov zdrojom inšpirácie, praktických nápadov a nových kontaktov. Nech povzbudí ďalšie používanie, vývoj a šírenie otvoreného softvéru vo vzdelávaní, výskume aj v IT riešeniach – a nech je zároveň pozvánkou k ďalšej spolupráci v duchu otvorenosti, zvedavosti a dobrej nálady.

Za organizačný výbor OSSConf 2026

Miroslav Kvaššay

## KVANTOVÉ POČÍTAČE: KDE SME DNES A KAM SMERUJEME

Ivan DOBOŠ (SK)

### VYŽIADANÁ PREDNÁŠKA

**Abstrakt:** Quantum computing patrí medzi najdiskutovanejšie technologické smery súčasnosti. Aký je však skutočný stav tejto technológie a nakoľko sme vzdialení od kvantových počítačov schopných riešiť prakticky významné úlohy? Prednáška poskytne zrozumiteľný úvod do sveta kvantového počítania. Predstavíme si základné princípy fungovania Quantum computingu, vrátane pojmov qubit, superpozícia, prepletenie a meranie. Na jednoduchých príkladoch si ukážeme model založený na kvantových bránach a kvantových obvodoch, ktorý dnes tvorí základ väčšiny vývojových platforiem. Následne sa pozrieme na súčasný stav technológie, hlavné hardvérové modalitty a éru NISQ (Noisy Intermediate-Scale Quantum), v ktorej sa nachádzame dnes. Zhrnieme aktuálne možnosti dostupných kvantových procesorov, roadmapy najvýznamnejších výrobcov a odhady ďalšieho vývoja smerom k Fault-Tolerant Quantum Computingu. Diskutovať budeme aj realistické očakávania, najčastejšie mýty a potenciál praktických aplikácií. Stručne sa dotkneme aj vplyvu Quantum computing na oblasť kryptografie, jeho potenciálu ohroziť súčasné kryptografické mechanizmy a prístupov post-quantum cryptography, ktoré majú zabezpečiť odolnosť digitálnej infraštruktúry v kvantovej ére.



**Ivan Doboš.** IT líder s viac ako 30-ročnou skúsenosťou v IT sektore, v spoločnosti Aricoma zastáva pozíciu Technology Evangelist & Head of Quantum. Pracoval vo viacerých popredných IT spoločnostiach na Slovensku aj v zahraničí, má bohaté skúsenosti s vedením tímov, strategickým plánovaním a technologickými inováciami. Implementoval IT riešenia pre významné organizácie a riadil multi-miliónové programy a projekty. V súčasnosti sa sústreďuje na quantum computing, zameriava sa na jeho praktické uplatnenie a popularizáciu.

## CULTURAL DISCRIMINATION IN LARGE LANGUAGE MODELS: DETECTION, MEASUREMENT, AND MITIGATION THROUGH TARGETED FINE-TUNING

Antoni CZOŁGOWSKI (US)

### INVITED LECTURE

**Abstract:** We are all using AI tools. Starting from LLMs embedded in web applications and ending with image and video generators that are more and more often replacing (sadly) real people's content. What we don't know is that AI, especially LLMs, often reflects biases from their training data or biases of their creators. Like most early adopters around me, I began treating language models as advisors while dealing with certain issues. After using them for a while, I realized something troubling. The answers I was getting didn't reflect the cultural context where I am, even after the model collected a vast amount of data about me. I even felt like the model didn't understand my problems at all. That's how the research idea was born.

I started asking questions: Do the creators and their origins implement their own worldview into the model's worldview? Is it the training data diversity, or lack of it, that's responsible for the problem? What are the characteristics of groups of people that models understand the least?

The research evaluates three different models: Gemma 3, reflecting the USA worldview; Bielik, a Polish model; and Qwen 3, to understand the Chinese perspective. All of them are put in the same context. I started by collecting World Values Survey responses to one particular worldview question that differentiates cultural perspectives. This gave me distributions for approximately 500 different demographic profiles, each representing a unique combination of characteristics like country, gender, age, education, and income level. I then asked each LLM to answer the same question while impersonating these profiles. This let me compare how the models' responses matched real human survey data using the Wasserstein distance metric.

I then apply LoRA (Low-Rank Adaptation) fine-tuning specifically targeting "worst-case personas", demographic combinations showing the highest bias, to test whether focused intervention can reduce bias without creating negative spillover effects on other demographic groups.



In my talk I want to demonstrate several things: how we can measure the fairness and bias of AI systems using simple and cost-effective solutions, what we can actually do with it, and finally that meaningful research, and what is more important, reproducible results, can be achieved by using code assistants like PacketLLM. For many parts of my workflow, like data handling, I used R, where my coding was facilitated by my own R library “PacketLLM” available on CRAN [<https://cran.r-project.org/web/packages/PacketLLM/index.html>].



**Antoni Czołgowski.** I am a graduate student in Data Science at the University of Colorado Boulder, working under Prof. Abel Iyasele on my research in LLMs. I work at JILA, a leading research institute in the physical sciences, as Data Engineer for the NSF Q-SEnSE: Quantum Systems through Entangled Science and Engineering Project. I am the developer of PacketLLM, an open-source R package on CRAN that brings LLM-assisted coding directly into RStudio. In my free time, I’m currently discovering new hobbies. In half a year, I have summited 2 of Colorado’s 58 fourteeners, but the ambitions are high, so who knows what will happen before July. This will be my first time at OSSConf.

## 3D SCANNING IN PRACTICE

Michal LEKÝR (SK)

### INVITED LECTURE

**Abstrakt:** This talk focuses on the practical application of 3D scanning for large-scale objects as well as smaller – room size spaces using the laser scanner Leica ScanStation P30. These methods apply also to other brands of laser scanners, which are becoming very popular these days and more affordable. The presentation addresses both methodological and technical aspects of spatial data acquisition, with particular emphasis on measurement planning, scanner positioning, reduction of occluded areas, registration of multiple scans, and subsequent processing of point cloud data.

The talk presents the complete workflow used in the documentation of large and geometrically complex objects, where it is necessary to balance measurement accuracy, point density, acquisition time, and the overall volume of collected data. Attention is also given to common challenges encountered in real-world scanning tasks, such as limited accessibility, shadowed or hidden areas, reflective or otherwise problematic surfaces, and the need to optimize scanning positions to obtain complete and reliable datasets.

An important part of the presentation is devoted to the processing and analysis of acquired data using open-source software tools. Special emphasis is placed on CloudCompare, which provides a robust environment for working with point clouds, including filtering, registration, segmentation, visualization, distance analysis, and comparison of scanned datasets.



In addition, other open-source solutions, such as MeshLab, are briefly introduced as useful tools for further processing, mesh generation, model optimization, and preparation of data for presentation or archiving.

The talk aims to demonstrate that open-source software is not only a supplementary option, but in many cases an effective and fully relevant component of the 3D documentation workflow. At the same time, it highlights that the quality of results in large-scale 3D scanning depends not only on the technical parameters of the scanner itself, but also on the overall measurement methodology, data registration strategy, and the choice of suitable processing tools.

These methods have been tested in real-world conditions, particularly in the scanning and documentation of historical buildings and heritage sites, including Budatín Castle, Bojnice Castle, and many others.



**Michal Lekýr** is a lecturer at the Faculty of Management and Informatics, University of Žilina, where he teaches computer graphics, 3D computer graphics, and low-level programming. He is also the founder of 3Xcore, with long-term experience in software development, interactive 2D and 3D presentation systems, visualization, and applied programming projects. His work connects academia with real-world development, with a strong focus on practical innovation, analytical thinking, and modern digital technologies.

## SEZNÁMENÍ S OPTEXEM

Petr OLŠÁK (CZ)

### VYŽIADANÁ PREDNÁŠKA

**Abstrakt:** OpTeX je luaTeXový formát, tj. sada maker TeXu podporující veškeré běžné požadavky při tvorbě TeXových dokumentů (podobně jako L<sup>A</sup>T<sub>E</sub>X či ConT<sub>E</sub>Xt). Je volně šířen, je součástí TeXlive, spustíte ho příkazem optex. Byl vytvořen v roce 2020, takže nenese dnes neužitečné nánosy dlouhého historického vývoje (jako např. L<sup>A</sup>T<sub>E</sub>X). Například rovnou pracuje výhradně s Unicode fonty a využívá plně možnosti moderního TeX engine luaTeX. Na rozdíl od L<sup>A</sup>T<sub>E</sub>Xu či ConT<sub>E</sub>Xtu se nesnaží zakrývat TeX-primitivní postupy novou vrstvou uživatelských příkazů s mnoha parametry často v mnoha rozšiřujících balíčcích, ale spíš předpokládá schopnost uživatele používat primárně TeX samotný s tím, že si dokumentovaná makra TeXu dokáže případně přiohnout k obrazu svému. Plno věcí lze v OpTeXu dělat přímočaře a jednoduše. Je to alternativa pro ty, kteří se například už ztratili v tisících rozšiřujících balíčků pro LaTeX sumárně se statisíci stránkami dokumentace a třeba ani netuší, která rada povalující se na internetu je aktuální a která už je zastaralá a dnes spíše k neužitku.

Na přednášce se pokusím ukázat základní vlastnosti OpTeXu na jednoduchých ukázkách: práce s fonty, s bibliografií, tvorba slidů pro zpětný projektor, barvy, OpTeX triky, ...). Také předvedu některé projekty postavené na OpTeXu: šablona pro závěrečné práce studentů, OpBible na tvorbu studijních Biblí.



**Petr Olšák** učí na FIT a FEL ČVUT v Praze matematiku a dále vede tamtéž volitelný kurz Typografie a TeX. TeXu se věnuje intenzivně od roku 1992 a jako popularizátor tohoto nástroje na zpracování sazby napsal o něm několik publikací a mnoho článků. Není příznivcem L<sup>A</sup>T<sub>E</sub>Xu. V mnoha záležitostech přispěl svým softwarovým úsilím k rozšíření všelijakých nástrojů souvisejících s TeXem. Všechny jsou zveřejněny na CTAN. Na tex.stackexchange vložil už skoro dva tisíce odpovědí. Jeho posledním velkým projektem je právě OpTeX. Byl dlouhodobým členem Československého sdružení uživatelů TeXu a v krátkém období na začátku milénia i jeho předsedou.

## MACHINE Gnostics: A STEP TOWARDS NON-STATISTICAL MACHINE LEARNING

Nirmal PARMAR (IN)

### INVITED LECTURE

**Abstract:** For the first time, I am introducing Machine Gnostics to the world – an open-source Python library that redefines the mathematical core of AI.

Under the hood, every model, every distribution function, every metric is built not on probability theory, but on Mathematical Gnostics – a framework grounded in Riemannian geometry, relativistic mechanics, and thermodynamics. Where statistics asks what is the population likely to do, Mathematical Gnostics asks what is this data point actually telling us. Each observation carries its own individual uncertainty, weighted by its own error, treated as a real physical event — not a sample from an imagined distribution.

Move past fragile, assumption-heavy models. Machine Gnostics encodes the laws of nature into algorithms that extract truth from data, even when samples are small, noisy, or corrupted. The familiar workflow remains — regressors, classifiers, clustering, deep learning — but the mathematical engine beneath it is entirely different. Machine Gnostics <https://machinegnostics.com/>.

Laws of nature, encoded – for everyone.



**Nirmal Parmar** is a global leader and Director of AI at Novartis, pioneering the intersection of thermal engineering and non-statistical artificial intelligence. As a distinguished research scientist specializing in Thermal Engineering and non-statistical AI paradigms, he has revolutionized how industries approach machine learning and data science.

Dr. Parmar is the visionary founder of the first machine learning library Machine Gnostics that operates on non-statistical principles, fundamentally challenging traditional approaches to AI. His groundbreaking work focuses on innovation in applying AI across engineering and business processes, delivering transformative solutions that redefine industry standards.

## FANTAZIE GENERATIVNÍ UMĚLÉ INTELIGENCE V TECHNICKÝCH OBORECH

Jiří ŠPERKA (CZ)

### VYŽIADANÁ PREDNÁŠKA

**Abstrakt:** Tato přednáška se bude zabývat generativní umělou inteligencí, zejména pak ve vztahu k výuce technických oborů. V poslední době se setkáváme se spoustou informací o tom, že pokrok v oblasti umělé inteligence způsobuje zásadní revoluci v mnoha oborech lidské činnosti, transformuje pracovní trh, a že umělá inteligence usnadní lidem práci. Mění generativní umělá inteligence svět k lepšímu? Firmy zabývající se umělou inteligencí se v rámci své propagace a reklamy většinou příliš nezabývají nedobrymi výstupy generativní umělé inteligence a jejich negativními dopady. To ale neznamená, že by se jimi nezabýval nikdo. Pojďme si tedy shrnout některé existující kritické pohledy na generativní umělou inteligenci. Zmíníme různé oblasti, ale nejvíce se zastavíme u oborů STEM (Science, Technology, Engineering, Mathematics). Ukážeme si vzdušné zámky a některé výstupy generativní umělé inteligence. Zmíníme ale i to dobré. Je generativní umělá inteligence můj kamarád?



*Motto:*

Plato is my friend,  
Aristotle is my friend,  
but my greatest friend is truth.

*Sir Isaac Newton*

**Jiří Šperka** se věnoval kritice generativní umělé inteligence během dvou přednášek na konferenci OpenAlt 2024 a 2025. Je mu blízké téma otevřené vědy. V současné době se zabývá zejména fyzikou plazmatu.

## Shadertoy – MEDZI MATEMATIKOU A UMENÍM

MIROSLAV BENDÍK (SK)

**Abstrakt.** Cieľom príspevku je predstaviť programovanie shaderov jednoduchým spôsobom pomocou platformy Shadertoy. Stručne opisuje históriu grafických akcelérátorov od jednoduchých jednorúčových zariadení až po moderné programovateľné GPU jednotky. Vysvetlené sú základné postupy pri tvorbe shaderov na jednoduchých príkladoch ako je vykreslenie matematickej funkcie. Článok predstavuje aj pokročilé možnosti, ktoré Shadertoy ponúka. Záver je venovaný potenciálnemu praktickému využitiu.

**Kľúčové slová.** Shadertoy, shader, GPU, grafické programovanie, demo.

## Shadertoy – BETWEEN MATHEMATICS AND ART

**Abstract.** The aim of this paper is to introduce shader programming in a simple way using the Shadertoy platform. It briefly describes the history of graphics accelerators, from simple single-purpose devices to modern programmable GPU units. Basic techniques for creating shaders are explained through simple examples, such as rendering a mathematical function. The article also presents advanced features offered by Shadertoy. The conclusion is devoted to potential practical applications.

**Keywords.** Shadertoy, Shader, GPU, Graphic programming, Demo.

## Úvod

Shadertoy je jednou z platforiem, ktorá umožňuje zdieľať svoje grafické demá. Ide o webovú stránku, ktorá využíva technológiu WebGL. Pre prehliadanie, či prispievanie nie je potrebné inštalovať nič dodatočné okrem relatívne moderného prehliadača.

## 1. Od textúry k shaderom

Aby som podrobnejšie vysvetlil, čo je shader, začnem históriou grafických akcelérátorov. Nebudem sa zaoberať tým, kto bol v čom prvý, pretože pri vývoji bolo veľa odbočiek, slepých uličiek, či dobrých nápadov, ale v nesprávnej dobe, ktoré sa reinkarnovali o 10 až 20 rokov neskôr.

Prvé pre bežného smrteľníka dostupné grafické akcelérátory sa objavovali najskôr na hracích konzolách. Spočiatku bolo podporované len zobrazovanie pohybujúcich sa rastrových obrázkov na obrazovke. Na začiatku 90. rokov začínajú byť pamäte dostatočne lacné na to, aby do nich vošlo omnoho viac dát než len

obrazové dáta aktuálnej scény. Vznikajú zložitejšie čipy umožňujúce okrem posúvania aj zväčšovanie, zmenšovanie, či dokonca rotáciu. Aby tieto operácie mohli fungovať, musel akcelerátor namiesto jednoduchej operácie prístupu do pamäte prepočítavať pre každý pixel adresu v textúre.

Ani nie o 5 rokov neskôr sú už bežne dostupné jednoduché 3D akcelerátory. Namiesto niekoľkých fixných funkcií pre zoom a rotáciu postačuje afinná transformácia realizovaná násobením maticou. Všetky doterajšie transformácie sa mohli vykonať jedinou operáciou a hardvér GPU sa tým paradoxne zjednodušil. Nové akcelerátory vedeli vykresľovať už len trojuholníky, pretože sú to najjednoduchšie rovinné útvary transformované do rastra obrazovky jednoduchým násobením riadiacich bodov (3-zložkový vektor) transformačnou maticou ( $3 \times 3$ ).

Nasledovala dlhá éra pridávania rôznych grafických efektov. Či už to boli rôzne funkcie pre miešanie farieb medzi jednotlivými vrstvami, podpora viacerých textúr na jednom povrchu, výpočet tieňov, ambientného, smerového, bodového osvetlenia, zdrsnenia povrchu, odrazy, alebo tiene, všetko toto komplikovalo grafické karty, ale zároveň limitovalo čo s nimi môžeme robiť.

## 2. Od komplexného k jednoduchému

Okolo roku 2000 máme grafické karty, ktoré vedia kombinovať osvetlenie a textúry pomocou rôznych funkcií sčítania, násobenia, delenia, logaritmov, goniometrických funkcií a to všetko pri použití konfiguračných registrov. Vo výsledku to bolo zložitejšie než keby existoval procesor s vlastnou inštrukčnou sadou, ktorý by mal prístup k zdrojovým a cieľovým dátam a ktorý by všetky tie sčítania, násobenia a odčítania vykonal ako program.

Keď existuje jednoduchší a teda aj lacnejší spôsob, tak sa samozrejme nájdú tí, ktorí ho využijú. Tak sa zrodil shader – malý program bežiaci na GPU. Zložitá konfigurácia a kombinácia registrov bola nahradená nahraním malého programu.

Dnes sa shadery píšu väčšinou vo forme jazyka podobného C. Priamo zdrojový kód sa pošle na kompiláciu ovládaču GPU, pretože neexistuje žiadna univerzálna inštrukčná sada. Skompilovaný program sa nahrá do GPU a spúšťa sa v príslušnom kroku. Najbežnejšie sú:

- Vertex shader – výstupom je pozícia na obrazovke
- Fragment (pixel) shader – výstupom je farba bodu na obrazovke

Tieto 2 programy sú flexibilnou náhradou toho, čo v minulosti robil fixný hardvér.

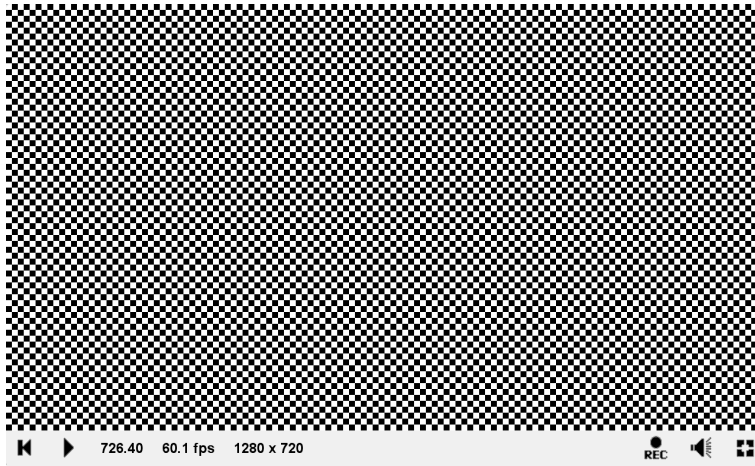
## 3. Jedno plátno, nekonečné možnosti

Na webe <https://www.shadertoy.com/> dostaneme k dispozícii plátno (obdĺžnik), a textové pole, do ktorého sa píše kód pre fragment shader. Úlohou človeka za klávesnicou je premeniť vstup (`in vec2 fragCoord`) na zmysluplne



vyzerajúci výstup (`out vec4 fragColor`). Nasledujúci kód vykreslí šachovnicu s rastrom  $10 \times 10$  pixelov:

```
void mainImage(out vec4 fragColor, in vec2 fragCoord) {
    int x = int(fragCoord.x) / 10;
    int y = int(fragCoord.y) / 10;
    vec3 col = vec3((x + y) % 2);
    fragColor = vec4(col, 0.0);
}
```



Obr. 1. Šachovnica

Tento podprogram sa volá pre každý pixel a jeho volanie je nezávislé na okolí. Preto sa výpočet môže paralelizovať bez akýchkoľvek vedľajších efektov.

Napriek tomu je jazyk OpenGL Shading Language dostatočne flexibilný a čitateľný na to, aby sa v ňom dalo pohodlne napísať napríklad vykresľovanie matematických funkcií.

```
#define M_PI 3.1415926535897932384626433832795

#define MIN_X 0.0
#define MAX_X (M_PI * 2.0)
#define MIN_Y (-1.1)
#define MAX_Y 1.1

float fn(float x) {
    return sin(x);
}

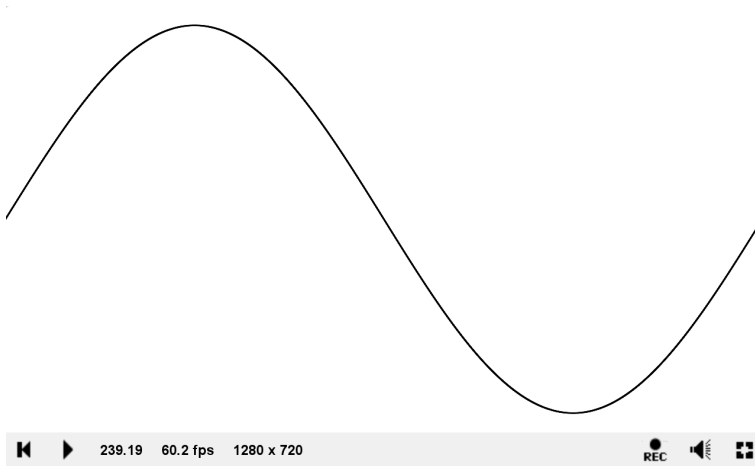
void mainImage(out vec4 fragColor, in vec2 fragCoord)
{
```

```

// Prevod koordinatov na [MIN_X:MAX_X, MIN_Y:MAX_Y]
vec2 uv = vec2(
    MIN_X + (MAX_X - MIN_X) * (fragCoord.x / iResolution.x),
    MIN_Y + (MAX_Y - MIN_Y) * (fragCoord.y / iResolution.y)
);

// Hodnota funkcie
float y = fn(uv.x);
// Vzdialenost koordinatu y od hodnoty funkcie
float dist = abs(uv.y - y);
// Hrubka ciary v smere y (2 pixely)
// Vypocet derivaciou y podla x
float thickness = abs(dFdx(y)) + (2.0 / iResolution.y);
// Vykreslenie hladkej krivky
float color = smoothstep(thickness, thickness + 2.0 /
iResolution.y, dist);
// Vysledna farba
fragColor = vec4(vec3(color), 1.0);
}

```



Obr. 2. Zobrazenie funkcie sínus

Tento príklad si v jednej vlastnosti protirečí s predchádzajúcim odsekom. Nie je možné pristupovať k okolitým pixelom s jednou výnimkou a tou je numerická derivácia (alebo zmena hodnoty) oproti susednému pixelu. Toto je špeciálny prípad, ktorý umožňuje priamo hardvér GPU obmedzený prístup k hodnote susedného pixelu.

Možnosti samozrejme nekončia pri vykreslení jednoduchšej matematickej funkcie. Pri troche snahy sa dá naprogramovať jednoduchý **ray tracing**, alebo rovno rendering algoritmickej generovanej krajiny.



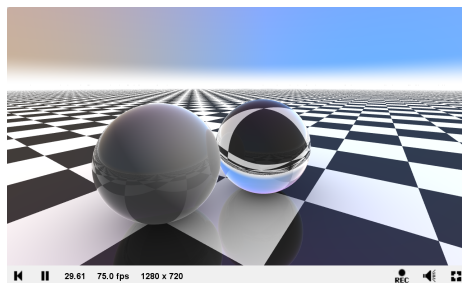
Inigo Quilez



Pablo Andrioli



Miroslav Bendík



Miroslav Bendík

**Obr. 3.** Ďalšie ukážky generovanej grafiky

## 4. Ďalšie možnosti

Okrem vstupov, ktoré som spomenul sú podporované aj ďalšie a to napríklad:

- **Čas.** Namiesto statického obrázka sa môže zobrazíť animácia.
- **Myš.** Umožňuje reagovať na pohyb a klikanie myšou.
- **Systémový čas.** Umožňuje zobrazíť hodiny.
- **Klávesnica.** Shader môže reagovať na stlačenie či pustenie klávesy.
- **Webová kamera.** Umožňuje tvorbu efektov s obrazom z kamery (prístupná ako textúra).
- **Mikrofón.** Zvuk prístupný ako textúra, ktorá obsahuje absolútne PCM hodnoty aj ich dekompozíciu na frekvencie (fft).
- **Hudba.** Zvuk prístupný ako textúra podobne ako pri mikrofóne.
- **Programové buffery A-D, cubemap buffer.** Buffery, ktoré sa dajú používať ako vstup a zároveň aj ako výstup.

- **Textúry.** Zoznam preddefinovaných textúr.
- **Cubemapy.** Zoznam preddefinovaných textúr prostredia.
- **Volumetrické textúry.** Zoznam preddefinovaných priestorových textúr (prístup cez 3 súradnice).
- **Video textúry.** Preddefinované pohyblivé textúry.

Za najzaujímavejšiu možnosť považujem programové buffery, ktoré umožňujú zápis a čítanie. Vedia tak fungovať ako pamäť medzi jednotlivými snímkami. Fantázii sa v tomto prípade medze nekladú a tak existujú príklady použitia, kde je buffer využitý na simuláciu kvapalín, alebo dokonca ako pamäť emulovaného počítača bežiaceho na GPU prakticky bez účasti CPU.

## 5. Praktické využitie

Okrem prezentovania svojich vlastných schopností sa dá projekt využiť napríklad pri tvorbe vlastných grafických algoritmov na fyzikálne presný, alebo naopak kreslený a umelecký rendering, výučbu, alebo na vyskúšanie a vizualizáciu algoritmov, ktoré sa neskôr prepíšu do OpenCL, alebo CUDA.

## Kontaktná adresa

**Ing. Miroslav Bendík**, Kukučínova 480/1, 059 18 Spišské Bystré, Slovensko,  
*E-mailová adresa:* [miroslav.bendik@gmail.com](mailto:miroslav.bendik@gmail.com), <https://linuxos.sk/profil/424/>.

## **TikZ – KAŽDÝ MÔŽE KRESLIŤ KRÁSNE OBRÁZKY JEDNODUCHO A HRAVO**

RUDOLF BLÁŠKO (SK)

**Abstrakt.** Existuje mnoho rôznych programov a aplikácií na vytváranie obrázkov, schém, grafov, diagramov, grafických modelov atď. V tomto príspevku sa budeme venovať populárnemu „kresliacemu“ balíčku TikZ, t. j. „TikZ ist kein Zeichenprogramm“, čo by sme mohli z nemčiny preložiť „TikZ nie je program na kreslenie“, ktorého autorom je Till Tantau. Budeme sa venovať základom, aby aj úplný začiatočník mohol bez problémov kresliť svoje obrázky.

**Kľúčové slová.**  $\text{\TeX}$ , TikZ, obrázok, schéma, diagram, graf, grafický model.

### **TikZ – EVERYONE CAN DRAW BEAUTIFUL PICTURES EASILY AND PLAYFULLY**

**Abstract.** There are many different programs and applications for creating images, diagrams, graphs, diagrams, graphic models, etc. In this article, we will focus on the popular „drawing“ package TikZ, i.e. „TikZ ist kein Zeichenprogramm“, which we could translate from German as „TikZ is not a drawing program“, which was created by Till Tantau. We will cover the basics so that even a complete beginner can draw their images without any problems.

**Keywords.**  $\text{\TeX}$ , TikZ, picture, diagram, graph, graphic model.

## **Úvod**

Existuje mnoho rôznych programov na kreslenie, resp. vytváranie obrázkov, schém, grafov, diagramov, grafických modelov atď. Niektoré sú on-line, niektoré desktopové, niektoré sú lepšie, niektoré horšie, niektoré sú výborné a niektoré na tri body, niektoré sú profesionálne (samozrejme aj drahé) a niektoré sú tiež veľmi dobré a navyše Open Source. Do tej poslednej skupiny patrí TikZ, ktorý je súčasťou systému  $\text{\TeX}$  a umožňuje vytvárať vysoko kvalitnú vektorovú grafiku. K tomu, aby používateľ TikZ-u vytvoril kvalitný a navyše veľmi pekný obrázok, nepotrebuje žiadne veľké schopnosti. Postačí mu ovládať niekoľko jednoduchých príkazov a ich parametrov, ktoré sú v princípe rovnaké a vychádzajú z anglických slov. Je pravda, že oproti klikacím programom, kde mnohokrát sa našinec ukliká, v tomto programe sa objekty vykresľujú pomocou príkazov. Nevýhodou, ktorá mnohých odradí hneď na začiatku, je skutočnosť, že užívateľ musí i keď nie veľký čas venovať počiatočnému štúdiu základom spôsobu vytvárania obrázkov. Nedá sa kresliť bez vedomostí, ale investícia sa vyplatí a výsledok je neporovnateľný

s výtvormi napr. komerčného kancelárskeho balíka MS Office. Je pravda, že som sa pokúsil porovnať neporovnateľné.

TikZ môžeme používať vo formátoch L<sup>A</sup>T<sub>E</sub>X, Plain T<sub>E</sub>X a aj ConT<sub>E</sub>Xt. Princíp je rovnaký a obrázky sa tvoria v príkaze `\tikz` alebo v prostredí `tikzpicture`. V tomto príspevku sa primárne budeme venovať práci v L<sup>A</sup>T<sub>E</sub>X-u, kde sa základný modul načíta príkazom `\usepackage{tikz}` a obrázky sa konštruujú v prostredí

`\begin{tikzpicture} ... \end{tikzpicture}`,

v Plain T<sub>E</sub>X-u sa modul načíta `\input tikz.tex` a obrázky sa tvoria pomocou `\tikzpicture ... \endtikzpicture`

a v ConT<sub>E</sub>Xt-e sa modul načíta `\usemodule[tikz]` a obrázky sa tvoria pomocou `\starttikzpicture ... \stoptikzpicture`.

Nepopierateľnou výhodou práce v TikZ-e je, že môžeme využívať všetky možnosti dostupné v T<sub>E</sub>X-u, vrátane písma a štýlu v zhode s domovským dokumentom (pokiaľ napríklad potrebujeme ilustrácie k textu). Ďalšou výhodou je, že mnoho práce sa dá zautomatizovať pomocou makier a cyklov. Nevýhodou pre menej skúsených užívateľov môže byť tvorba zložitejších obrázkov. V TikZ-e sa dajú vytvárať aj 3D obrázky, prípadne sa dajú obrázky animovať, ale týmto oblastiam sa venovať nebudeme. V príspevku sa budeme venovať tvoreniu obrázkov iba v systéme L<sup>A</sup>T<sub>E</sub>X.

## 1. Začínáme s TikZ-om

Aby sme mohli začať TikZ používať, musíme ho v preambule dokumentu inicializovať príkazom `\usepackage{tikz}`. TikZ je používateľské rozhranie postavené na grafickom engine PGF (Portable Graphics Format) a po jeho zavolaní sa automaticky spustí `pgf` (hlavný balíček rodiny PGF) a niekoľko jeho modulov `pgfcore` (jadro grafického systému: definície ciest, základné nástroje na kreslenie), `pgffor` (možnosti cyklov, napr. príkaz `\foreach`), `pgfkeys` (spracováva kľúče, parametre a štýly), `pgfmath` (možnosť výpočtov, napr. súradníc a práca s premennými a funkciami), `pgfsys` (systémová vrstva zabezpečujúca komunikáciu s výstupným formátom, napr. dviPS, pdfT<sub>E</sub>X, XeT<sub>E</sub>X, LuaT<sub>E</sub>X).

Pre správne fungovanie TikZ automaticky na pozadí natiahne balíčky `xcolor` (práca s farbami, definovanie, miešanie, farebné prechody ap.), `graphicx` (vkládanie externej grafiky, správa a transformácia grafických objektov, napr. rotácia, zmena mierky), `atbegshi` (vkládanie objektov na pozadie na stránku, napr. vodoznak, umiestňovanie grafických vrstiev a špeciálne efekty). To znamená, že tieto balíčky už nemusíme pridávať do preambuly. Ak tak urobíme, nič sa nestane.

Problém vznikne, ak chceme niektorý z týchto balíčkov načítať so špeciálnymi parametrami. Napr. pre rozšírenie farieb `dvipsnames` v balíčku `xcolor` použijeme príkaz `\usepackage[dvipsnames]{xcolor}`, ale musí byť umiestnený v preambule pred príkazom `\usepackage{tikz}`.

Ak chceme využiť niektorú z mnohých rozširujúcich možností TikZ-u, musíme následne načítať príkazom `\usetikzlibrary{knižnica}` príslušnú internú knižnicu. Napr. rozšírenie tvarov koncových šípiek je v knižnici `arrows.meta` a použijeme príkaz `\usetikzlibrary{arrows.meta}`, pre rozšírenie tieňovania použijeme `\usetikzlibrary{shadings}`.

Ako ilustruje nasledujúca ukážka vľavo, môžeme TikZ použiť aj na tvorbu samostatných obrázkov pomocou triedy `standalone`.

|                                           |                                               |
|-------------------------------------------|-----------------------------------------------|
| <code>\documentclass[tikz,</code>         | <code>\documentclass[a4paper]{article}</code> |
| <code>dvipsnames]{standalone}</code>      | <code>\usepackage[dvipsnames]{xcolor}</code>  |
| <code>\usetikzlibrary{arrows.meta}</code> | <code>\usepackage{tikz}</code>                |
| <code>\begin{document}</code>             | <code>\usetikzlibrary{arrows.meta}</code>     |
| <code>TikZ obrázok</code>                 | <code>\begin{document}</code>                 |
| <code>\end{document}</code>               | <code>TikZ obrázky</code>                     |
|                                           | <code>\end{document}</code>                   |

V TikZ-e sa objekty vykresľujú pomocou **príkazov**, ktoré začínajú spätnou lomkou a názvom `\meno_príkazu`, pokračujú následnými parametrami, súradnicami, vykresľovanými objektami, textom atď. Každý príkaz musí byť vždy ukončený bodkočiarkou `;`. Základné možnosti na vytváranie obrázkov sú dve. Prvou možnosťou je použiť príkaz `\tikz`, ktorý sa najčastejšie používa na vkladanie jednoduchších obrázkov priamo v texte v tvare

`\tikz jeden_príkaz`, resp. `\tikz {viac_príkazov}`.

Namiesto príkazu `\tikz` môžeme použiť ekvivalentné prostredie

`\begin{tikzpicture}` príkazy `\end{tikzpicture}`.

Záleží iba na používateľovi, ktorú možnosť využije, i keď pri zložitejších konštrukciách je pohodlnejšie a prehľadnejšie pracovať v uvedenom prostredí.

## 1.1. Základné grafické príkazy

Keďže TikZ funguje v T<sub>E</sub>X-u, môžeme používať ľubovoľné dĺžkové jednotky a kombinovať ich navzájom. Na súradnice bodov a vzdialenosti sa implicitne používajú centimetre `cm`. To znamená, že všetky nasledujúce súradnice `(1cm,1cm)`, `(1,1)`, `(1cm,1)`, `(1,1cm)` vyjadrujú ten istý bod. Na určovanie súradníc môžeme použiť aj matematické operácie a funkcie a známe konštanty  $\pi$ ,  $e$  (`\pi`, `e`). Ak sa v tomto vyjadrení vyskytne niektorá zo zátvoriek `()`, potom ich musíme zapúzdriť medzi skupinovú zátvorku `{}`, napr. `(\pi,{e+2*\sin(1 r)})`. Vo vnútri objektov na hrúbky čiar, posúvanie okolo vrcholov ap. sa používajú implicitne body `pt`. To znamená, že výrazy `node[above=5pt]` a `node[above=5]` predstavujú identické posunutia smerom hore o vzdialenosť 5pt a výrazy `line width=1pt` a `line width=1` definujú rovnaké hrúbky čiar 1pt. Tieto jednotky (`cm`, `pt`) ako aj napr. `mm`, `in` sú jednotky absolútne, pretože definujú konkrétnu vzdialenosť. Okrem nich môžeme používať relatívne dĺžkové jednotky `ex` (výška malého písma  $x$  aktuálneho písma) a `em` (šírka veľkého písma  $M$  aktuálneho písma), ktoré sa odvíjajú od daného fontu.

Základom pri konštruovaní sú `path` (cesty) a `node` (uzly, vrcholy). Ak chceme vykresliť nejaký objekt, použijeme príkaz `\path[draw] ...`, pre vyplnenie objektu použijeme `\path[fill] ...` ap. V praxi sa uvedené príkazy používajú v skrátenejších formách `\draw`, resp. `\fill`, preto ich budeme aj my používať.

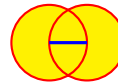
## 1.2. Môj prvý obrázok „Hello, world!“

Základné príkazy na vykresľovanie sú `\draw` (kreslí čiary, obrysy), `\fill` (vyplní objekty bez kreslenia obrysov), `\filldraw` (kreslí obrysy a zároveň vyplní), `\clip` (ohraničí/vystrihne viditeľnú oblasť pre nasledujúce grafické objekty). Príkaz `\filldraw` môžeme nahradiť konštrukciou `\draw[fill]` a `\clip` môžeme použiť v kombináciách `\draw[clip]`, `\fill[clip]`, resp. `\filldraw[clip]`. Tieto príkazy majú nasledujúcu formálnu štruktúru:

`\prikaz[voliteľné_parametre] (súradnice) geometrický_tvar ;`

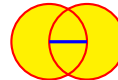
Vo `voliteľných_parametroch` môžeme definovať farbu čiar, farby výplne, farebné efekty, hrúbku a tvar čiar, úpravy okrajov, rohov ap., `súradnice` definujú miesto, na ktoré sa daný `geometrický_tvar` umiestní. Do jedného príkazu môžeme umiestniť viacero `geometrických_tvarov` spolu s ich `súradnicami`, ale oddeľujú sa iba medzerami. Aby sme iba nete(r)orizovali, tu je náš prvý obrázok:

```
\tikz{\draw[red,fill=yellow,thick]
  (0,0) circle (0.5)
  (0.5,0) circle (0.5);
\draw[blue,very thick] (0,0)--(0.5,0);}
```



alebo ten istý obrázok v prostredí `tikzpicture`

```
\begin{tikzpicture}
\draw[red,fill=yellow,thick]
  (0,0) circle (0.5) (0.5,0) circle (0.5);
\draw[blue,very thick] (0,0)--(0.5,0);
\end{tikzpicture}
```



Kružnicu/kruh definujeme sekvenciou `(stred) circle (polomer)`, resp. ekvivalentne `(stred) circle [radius=polomer]` a spojnicu bodov definujeme `--`.

Desatinné čísla sa píše v angličtine s desatinnou bodkou (napríklad 6.35). Ak je na začiatku desatinného čísla číslica nula, môže sa vynechať, t. j. výrazy 0.5 a .5, resp. -0.5 a -.5 predstavujú identické hodnoty.

## 1.3. Pozícia obrázku v texte

Konečný obrázok vytvorený v TikZ-e sa oreže podľa krajných bodov na šírku a na výšku a výsledný obdĺžnik sa umiestní na základnú pozíciu od ľavého dolného rohu. V texte sa tento grafický objekt správa ako samostatné nerozdeliteľné slovo (samozrejme so svojou výškou a šírkou).

```
text pred obrázkami
\tikz\fill[yellow] (0,0) circle (.5em);
text medzi obrázkami
\tikz\fill[red] (5,5) circle (.75em);
text za obrázkami
```

text pred obrázkami ● text medzi  
obrázkami ● text za obrázkami



## 1.4. Súradnice

Na určovanie pozície môžeme použiť klasické karteziánske súradnice

$(x\_súr, y\_súr)$ ,

kde  $x\_súr$  a  $y\_súr$  vyjadrujú vzdialenosti danej pozície od kolmých osí  $x$  a  $y$ . Druhá možnosť je použiť polárne súradnice v tvare

$(uhol: vzdialenosť)$ ,

pričom pozícia sa posunie z počiatku  $(0:0)$  o  $vzdialenosť$  v smere  $uhla$ . Uhol je vyjadrený v stupňoch (nie radiánoch), kladný smer je orientovaný proti smeru hodinových ručičiek a záporný smer naopak. V TikZ-e sa používajú stupne, platí to aj pre goniometrické funkcie (tie sa dajú počítať aj v radiánoch).

Polárne a karteziánske súradnice môžeme v príkazoch ľubovoľne kombinovať. Oba typy súradníc môžeme používať absolútne alebo môžeme pozíciu odvádzať relatívne od predchádzajúceho bodu konštrukciou  $++(súradnice)$  alebo relatívne od posledného predchádzajúceho bodu reťazca s absolútnymi súradnicami konštrukciou  $+(súradnice)$ .

```
\tikz\draw[red,thick] (0,0)--(1,.5)--(2,0)
--(1,0)--(2.5,.25) circle (.25);

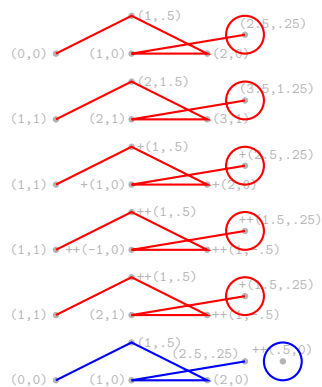
\tikz\draw[red,thick] (1,1)--(2,1.5)--(3,1)
--(2,1)--(3.5,1.25) circle (.25);

\tikz\draw[red,thick] (1,1)++(1,.5)--+(2,0)
--+(1,0)--+(2.5,.25) circle (.25);

\tikz\draw[red,thick] (1,1)+++(1,.5)+++(1,-.5)
--++(-1,0)+++(1.5,.25) circle (.25);

\tikz\draw[red,thick] (1,1)+++(1,.5)+++(1,-.5)
--(2,1)--+(1.5,.25) circle (.25);

\tikz\draw[blue,thick] (0,0)--(1,.5)--(2,0)
--(1,0)--(2.5,.25) ++(.5,0) circle (.25);
```



V prvých piatich červených príkladoch sa vytvorí ten istý objekt, rozdiel je iba v prvom príklade v štartovacom bode. V prvých dvoch príkladoch sú súradnice bodov určené absolútne. V treťom príklade výrazy  $+(súradnice)$  znamenajú, že konečné súradnice dostaneme pripočítaním tejto hodnoty k súradniciam prvého bodu reťazca, t. j. k súradniciam  $(1,1)$ . V štvrtom príklade výrazy  $++(súradnice)$  znamenajú, že konečné súradnice dostaneme pripočítaním tejto hodnoty k súradniciam predchádzajúceho bodu reťazca, t. j. sa posúvame z bodu do bodu postupne. Tieto spôsoby môžeme ľubovoľne kombinovať, ale musíme si dávať aby sme sa dostali tam kam chceme (piaty príklad).

V poslednom šiestom príklade sú súradnice bodov na ceste určené absolútne, iba posledný výraz  $++(.5,0)$  posunie stred kružnice o pol centimetra vpravo.

Pri prevodoch medzi karteziánskymi a polárnymi súradnicami môžu vznikať iracionálne hodnoty. Ich použitie taktiež nie je rovnaké, polárne súradnice používame pre oblúky, kružnice, skladanie uhlov ap. V polárnom tvare si musíme uvedomiť, že absolútne pozície sa počítajú od rovnakého bodu  $(0,0)=(0:0)$ . Ak

by sme chceli predchádzajúce príklady vyjadriť v polárnych súradniciach, museli by sme súradnice prepočítať, čo v praxi nemá význam.

Vykresliť pravidelný mnohoúhelník v karteziánskych súradniciach nie je problém, ale museli by sme presne vypočítať súradnice jeho vrcholov. V polárnych súradniciach na to stačí jednoduchý príkaz. Napríklad rovnostranný trojuholník vykreslíme nasledovným príkazom.

```
\tikz\draw[red,very thick]
  (0:.4)--(120:.4)--(2*120:.4)--(0:.4);
```

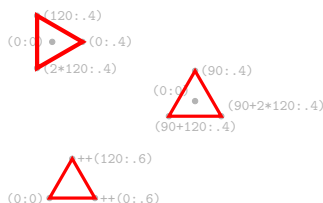


Ak sa pozrieme na ukončenie trojuholníka v bode (0:.4), zistíme zub, t. j. trojuholník nie je korektne uzavretý. Tento nedostatok vyriešime jednoducho, že namiesto opakovania prvého/posledného bodu na konci príkazu napíšeme výraz `cycle`.

```
\tikz\draw[red,very thick]
  (0:.4)--(120:.4)--(2*120:.4)--cycle;

\tikz\draw[red,very thick]
  (90:.4)--(90+120:.4)--(90+2*120:.4)--cycle;

\tikz\draw[red,very thick]
  (0:0)--++(0:.6)--++(120:.6)--cycle;
```



## 1.5. Farby a ich použitie

Pre prácu s farbami využíva TikZ možnosti na začiatku spomínaného balíčka `xcolor`. Štandardne môžeme použiť 19 základných farieb:

black, blue, brown, cyan, darkgray, gray, green, lightgray, lime, magenta,  
olive, orange, pink, purple, red, teal, violet, white, yellow.



Ak potrebujeme iné farby, môžeme použiť niektorú knižnicu z balíčka `xcolor`, t. j. `dvipsnames`, `svgnames` alebo `x11names`. Farby si môžeme namiešať aj sami podľa schém `gray`, `rgb` (red, green, blue – červená, zelená, modrá) alebo `cmk` (cyan, magenta, yellow, key/black – tyrkysová, purpurová, žltá, čierna):

```
\definecolor{farba}{gray}{číslo} (oddtiene sivej),
\definecolor{farba}{rgb}{čísR,čísG,čísB} (monitory, dataprojektory),
\definecolor{farba}{cmk}{čísC,čísM,čísY,čísK} (tlačiarne).
```

Hodnoty `číslo`, `čísR`, ..., `čísK` predstavujú čísla v intervale od 0 do 1 a vyjadrujú koľko dielov danej základnej farby sa pridá do výslednej farby. Hodnota `číslo` (tiež od 0 do 1) vyjadruje, koľko percent bielej sa mieša s čiernou farbou, t. j. `{gray}{0}`, `{gray}{.1}`, `{gray}{.2}`, ..., `{gray}{1}` definujú postupne odtdiene sivej od čiernej farby postupne po bielu farbu.

Farby, ktoré máme k dispozícii, môžeme pri používaní medzi sebou miešať.

- `farA!číslo!farB` znamená, že sa zmieša `číslo` dielov farby `farA` a `100-číslo` dielov farby `farB` (`číslo` je hodnota v intervale od 0 do 100).
- `farA!číslo`, resp. `farA!číslo!white` zmieša farbu `farA` s bielou farbou.

- `farA!čísA!farB!čísB!farC`, resp. `farA!čísA!farB!čísB!farC!čísC!farD` nemieša farby naraz, ale postupne. Najprv zmieša prvé dve farby `farA` a `farB`, potom ich výslednú farbu zmieša s farbou `farC` atď.

Takéto miešanie farieb sa bežne v TikZ-e používa. Z existujúcich farieb si môžeme namiešať a potom používať aj vlastné farby pomocou príkazu:

```
\colorlet{Moja_farba}{farA!číslo!farB}.
```

## 1.6. Príkazy `\draw` a `\fill`

Príkaz `\draw` sa používa na vykresľovanie objektov s obrysami aj výplňami a `\fill` na vykresľovanie objektov s výplňami bez obrysov, majú tvar:

```
\prikaz[voliteľné_parametre] (súradnice) geometrický_tvar ;
```

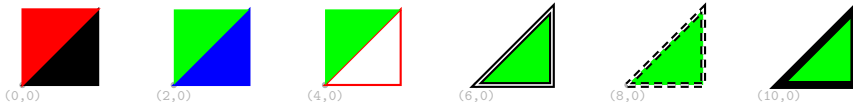
Rozdiel medzi objektami vytvoreným príkazmi `\draw` a `\fill` je v ich veľkosti, `\draw` vykreslí aj obrysovú čiaru určenej hrúbky a o túto hrúbku je objekt vykreslený pomocou `\fill` menší, tenší. Obrysová čiara v príkaze `\draw` sa kreslí okolo objektu určeného príkazom `\fill`, pričom polovica hrúbky čiary sa vykreslí smerom dnu a druhá polovica smerom von.

Voliteľné\_parametre týchto príkazov sa oddeľujú čiarkou, väčšinou nezáleží na ich poradí a definujú rôzne parametre vykresľovania. Ak nie sú v príkaze explicitne uvedené, použijú sa ich implicitné nastavenia:

- `color=farba`, resp. `farba` definuje farbu obrysov objektu (implicitne čierna).
- `fill=farba` definuje farbu výplne (implicitne farba obrysov), pričom uzavretý objekt sa vyplní celý a pri otvorenom objekte sa vyplní časť ohraničená hranicami objektu a spojnicou (úsečkou) koncových bodov objektu.
- `line width=hrúbka`, `ultra thick`, `very thick`, `thick` (hrubé čiary), `semi-thick`, `thin` (implicitne), `very thin`, `ultra thin` (tenké čiary) určujú hrúbku obrysovej čiary, ak nie je určená jednotka v `hrúbke`, použijú sa body `pt`.
- `solid`, resp. `-` (jednoduchá plná čiara, implicitné nastavenie), `double` (dvojitá čiara, tento parameter je možné použiť aj s nasledujúcimi čiarami), `dotted`, `densely dotted`, `loosely dotted` (bodkované čiary), `dashed`, `densely dashed`, `loosely dashed` (čiarkované čiary), `dash dot`, `densely dash dot`, `loosely dash dot`, `dash dot dot`, `densely dash dot dot`, `loosely dash dot dot` (bodkočiarkované čiary) definujú tvar obrysovej čiary.

```
\draw[thick,fill]      (0,0)---+(1,0)---+(0,1)--cycle;
\draw[red,thick,fill]  (0,0)---+(0,1)---+(1,0)--cycle;
\draw[blue,thick,fill] (2,0)---+(1,0)---+(0,1)--cycle;
\fill[green]          (2,0)---+(0,1)---+(1,0)--cycle;
\draw[red,thick]       (4,0)---+(1,0)---+(0,1)--cycle;
\fill[green]          (4,0)---+(0,1)---+(1,0)--cycle;

\draw[thick,double,fill=green]      (6,0)---+(1,0)---+(0,1)--cycle;
\draw[thick,densely dashed,double,fill=green] (8,0)---+(1,0)---+(0,1)--cycle;
\draw[black,line width=3,fill=green] (10,0)---+(1,0)---+(0,1)--cycle;
```



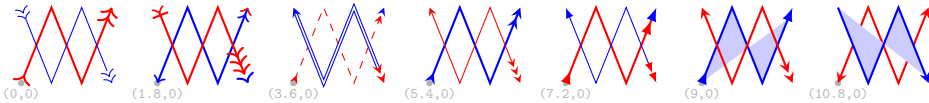
- <- a ->, resp. to- a -to, stealth-, -stealth, latex-, -latex (šípky), >- a -<, resp. to reversed- a -to reversed, stealth reversed-, -stealth reversed, latex reversed-, -latex reversed (obrátené šíčky), |-, -| (kolmé čiarky) definujú začiatky/ukončenia čiar, pričom ich môžeme opakovať a ľubovoľne kombinovať, meniť ich farbu a oddeľovať od čiar (pomocou .) .

```

\def\ObrA{---+.4,1)---+.4,-1)---+.4,1)}
\def\ObrB{---+.4,-1)---+.4,1)---+.4,-1)}

\draw[red,thick,>->] (0,0) \ObrA;
\draw[blue,>.-.->] (0,1) \ObrB;
\draw[blue,thick,to-[to[]to[red]]] (1.8,0) \ObrA;
\draw[red,thick,{to reversed[]|}->>>[blue]] (1.8,1) \ObrB;
\draw[red,dashed,stealth reversed-{stealth[]stealth[blue]]] (3.6,0) \ObrA;
\draw[blue,double,stealth-{stealth[red].stealth[red]]] (3.6,1) \ObrB;
\draw[blue,thick,stealth reversed-{stealth[]stealth[blue]]] (5.4,0) \ObrA;
\draw[red,stealth-{stealth[red].stealth[red].stealth[red]]] (5.4,1) \ObrB;
\draw[red,thick,latex reversed-{latex[]latex[blue]]] (7.2,0) \ObrA;
\draw[blue,thick-{latex[red].latex[red]}] (7.2,1) \ObrB;
\draw[blue,thick,latex reversed-latex,fill=blue!20] (9,0) \ObrA;
\draw[red,thick,stealth-{stealth[]stealth[]}] (9,1) \ObrB;
\draw[red,thick,stealth-stealth] (10.8,0) \ObrA;
\draw[blue,thick,-stealth,fill=blue!20] (10.8,1) \ObrB;

```



Ak sa nám nejaká časť (aj časť príkazu) opakuje, môžeme ju uložiť do vlastného makra príkazom `\def`. Makro musí byť definované pred svojím spustením, môže byť definované aj vo vnútri príkazu `\tikz`, resp. prostredia `tipzpicture` a definuje sa nasledovne (v príkladoch makrá `\ObrA` a `\ObrB`):

```

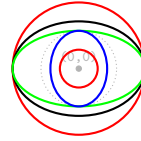
\def\Meno{ definícia } (bez parametra),
\def\Meno at #1 { definícia } (jeden parameter #1),
\def\Meno at #1 at #2 { definícia } (dva parametre #1, #2).

```

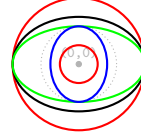
Vlastné makro sa v TikZ-e spúšťa `\Meno` (bez parametra), resp. `\Meno at par1`, `\Meno at par1 at par2`, ... (podľa počtu parametrov, max 9), pričom sa pri jeho spustení nahradí `#1` hodnotou `par1`, `#2` hodnotou `par2` atď.

- `scale=číslo`, `xscale=čísX`, `yscale=čísY` zväčší/zmenší oba dĺžkové rozmery `číslo` krát, resp. šírku `čísX` krát a výšku `čísY` krát. Mení celú mierku, takže mení veľkosť objektu a súčasne ho posúva v smere od/ku počiatku (0,0).
- `x=jedX`, `y=jedY` nastaví jednotkový rozmer šírky alebo výšky na `jedX` alebo `jedY`. Ak nie je zadaná jednotka, implicitne sa nastaví bod `pt`. Napr. `x=.5cm` prestaví jednotku šírky z 1 cm na 0,5 cm (na polovicu). Fungujú rovnako ako `xscale`, `yscale`. Nastavenia `xscale=.5` a `x=.5cm` majú rovnaký efekt.

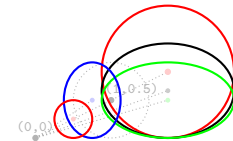
```
\draw[red,thick,scale=1.75] (0,0) circle (.5);
\draw[thick,xscale=1.75,yscale=1.25] (0,0) circle (.5);
\draw[green,thick,xscale=1.75] (0,0) circle (.5);
\draw[blue,thick,xscale=.75] (0,0) circle (.5);
\draw[red,thick,scale=.5] (0,0) circle (.5);
```



```
\draw[red,thick,x=1.75cm,y=1.75cm] (0,0) circle (.5);
\draw[thick,x=1.75cm,y=1.25cm] (0,0) circle (.5);
\draw[green,thick,x=1.75cm] (0,0) circle (.5);
\draw[blue,thick,x=.75cm] (0,0) circle (.5);
\draw[red,thick,x=.5cm,y=.5cm] (0,0) circle (.5);
```



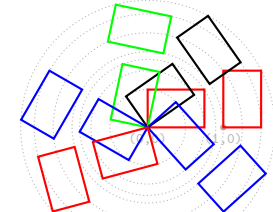
```
\draw[red,thick,scale=1.75] (1,.5) circle (.5);
\draw[thick,xscale=1.75,yscale=1.25] (1,.5) circle (.5);
\draw[green,thick,x=1.75cm] (1,.5) circle (.5);
\draw[blue,thick,xscale=.75] (1,.5) circle (.5);
\draw[red,thick,x=.5cm,y=.5cm] (1,.5) circle (.5);
```



- `rotate=uhol_otočenia` otočí objekt o `uhol_otočenia` (v stupňoch) okolo počiatku (0,0) v smere kladnom proti smeru hodinových ručičiek (kladný `uhol_otočenia`) alebo v zápornom smere (záporný `uhol_otočenia`).

Obdĺžnik definujeme sekvenciou (súr1) `rectangle` (súr2), pričom (súr1), (súr2) sú súradnice protilahlých rohov obdĺžnika (uhlopriečne).

```
\def\Obrazok{(0,0) rectangle ++(.75,.5)
(1,0) rectangle ++(.5,.75)}
\draw[red,thick] \Obrazok;
\draw[black,thick,rotate=35] \Obrazok;
\draw[green,thick,rotate=78] \Obrazok;
\draw[blue,thick,rotate=150] \Obrazok;
\draw[blue,thick,rotate=-48] \Obrazok;
\draw[red,thick,rotate=-165] \Obrazok;
```



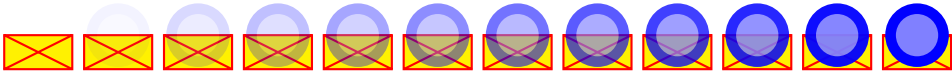
Prácu si môžeme zjednodušiť používaním cyklu `\foreach`, ktorý môžeme použiť vo vnútri TikZ-u i mimo TikZ a má tvar:

`\foreach \x in { množina } {vlastné_příkazy}` (parameter `\x`).

V cykle sa postupne vykonávajú `vlastné_příkazy` a premenná `\x` sa v nich nahrádza konkrétnymi hodnotami z `množina`. Množina môže obsahovať konkrétne hodnoty, ale aj postupnosti `a,b,...,c`. V tomto prípade sa spočíta krok `d=b-a` a postupne sa `\x` nahrádza hodnotami `a, b, b+d, b+2d, ...`, pokiaľ sa neprekročí hodnota `c`. Obe možnosti sa môžu kombinovať.

- `opacity=číslo`, `draw opacity=čísD`, `fill opacity=čísF` definujú transparentnosť (priesvitnosť) celého objektu (`číslo`), obrysov (`čísD`) alebo výplne (`čísF`). Hodnoty `číslo`, `čísD`, `čísF` sú čísla v intervale od 0 do 1 a prakticky vyjadrujú nepriesvitnosť, `opacity=0` znamená 100 % priesvitnosť, t. j. 0 % nepriesvitnosť, `opacity=.3` znamená 30 % nepriesvitnosť, ..., `opacity=1` znamená 100 % nepriesvitnosť.

```
\foreach \x in {0,.05,.15,...,.95,1}
{\tikz{\draw[red,thick,fill=yellow,scale=.9] (-.5,-.5) rectangle ++(1,.5)
(-.5,-.5)---+(1,.5) (-.5,0)---+(1,-.5);
\draw[blue,line width=4pt,fill=blue!50,opacity=\x,scale=.7] (0,0) circle (.5);}}
```

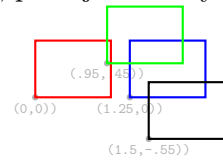


Na hornom obrázku sa postupne hodnota `opacity` v kruhu zväčšuje  $0 \rightarrow 1$ . Na dolnom obrázku sa v kruhu hodnota `draw opacity` postupne zväčšuje  $0 \rightarrow 1$  a hodnota `fill opacity` postupne zmenšuje  $1 \rightarrow 0$ .



- `xshift=posX`, `yshift=posY` posunie objekt vodorovne o hodnotu `posX` alebo zvisle o hodnotu `posY`, ak nie je určená dĺžková jednotka, použijú sa body `pt`.

```
\draw[red,thick] (0,0) rectangle ++(1,.75);
\draw[blue,thick,xshift=1.25cm] (0,0) rectangle ++(1,.75);
\draw[black,thick,xshift=1.15cm,yshift=-.5cm]
(0,0) rectangle ++(1,.75);
\draw[green,thick,xshift=.95cm,yshift=.45cm]
(0,0) rectangle ++(1,.75);
```



Voliteľných `parametrov` existuje oveľa viac. Ak sa nám nejaká ich kombinácia v obrázku často opakuje (mnoho čiar má rovnakú farbu, hrúbku, tvar, výplň ap.) alebo chceme zväčšiť/zmenšiť, otočiť výsledný obrázok naraz, môžeme tieto požiadavky zadať priamo do príkazu `\tikz`, resp. prostredia `tikzpicture`:

`\tikz[nastavenia]`, resp. `\begin{tikzpicture}[nastavenia]`.

Tieto `nastavenia` sa nastavujú implicitne pre celý obrázok, napr.

```
\tikz[red,thick,scale=.5,fill=yellow]
```

znamená, že (pokiaľ to nezmeníme) obrysy sa budú vykresľovať tučne červenou farbou, vyplňať sa budú žltou farbou a výsledný obrázok sa zmenší na polovicu.

V `nastaveniach` si môžeme definovať vlastný štýl s unikátnym názvom

```
\tikz[Mojstyl/.style={nastavenia štýlu}],
```

ktorý používame medzi voliteľnými `parametrami`. Ak má štýl parameter `#1`, potom ho používame v tvare `Mojstyl=par` (`#1` sa nahradí výrazom `par`).

```
\tikz[C styl/.style={blue, fill=blue!20, thick},
stylB/.style={red, fill=red!20, thick}, scale=.8, rotate=85]
{\draw[stylB,double] (0,0) circle (.5cm);
\draw[C styl] (-.25,-.25) rectangle ++(1,.5);}

\tikz[m styl/.style={#1, fill=#1!20, thick}, scale=.75, rotate=85]
{\draw[m styl=red,double] (0,0) circle (.5cm);
\draw[m styl=blue] (-.25,-.25) rectangle ++(1,.5);}
```



Definície vlastných štýlov môžeme uložiť v `preambule` do „prvotných nastavení“ `\tikzset{ }` a potom ich môžeme používať v celom dokumente.

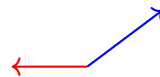
## 1.7. Geometrické tvary

Dôležité súradnice si môžeme označiť špeciálnymi názvami. Potom nemusíme práčne prepisovať súradnice týchto bodov a stačí sa odvolať na tieto (nami zvolené) názvy. Najjednoduchšie je použiť príkaz `\coordinate`:

```
\coordinate (meno) at (súradnice);
```

Aj pri takto definovaných bodoch môžeme použiť relatívne posunutia `++`, resp. `+`.

```
\coordinate (a0) at (0,.75);\coordinate (b0) at (1,.75);
\draw[red,thick,->] (b0)--(a0);
\draw[blue,thick,->] (b0)---++(b0);
```



V príkazoch TikZ-u sa na zadané súradnice umiestňujú geometrické objekty. Niektoré (spojnica bodov, kruh, obdĺžnik) sme si už spomenuli. Je samozrejmé, že takýchto objektov existuje oveľa viac.

- Pre **spojnicu bodov** najčastejšie používame `--`, ale je viac možností.

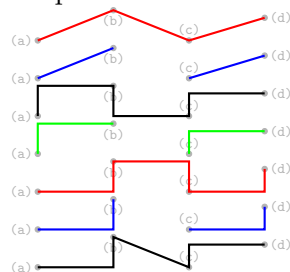
♠ `--` spája body definovanou čiarou.

♠ `|-` spája body tak, že najprv urobí zvislú čiaru a potom vodorovnú kolmicu.

♠ `-|` spája body tak, že najprv urobí vodorovnú čiaru a potom zvislú kolmicu.

```
\coordinate (a) at (0,0);\coordinate (c) at (2,0);
\coordinate (b) at (1,.4);\coordinate (d) at (3,.3);
```

```
\draw[red,thick] (a) -- (b) -- (c) -- (d);
\draw[blue,thick] (a) -- (b) (c) -- (d);
\draw[black,thick] (a) |- (b) |- (c) |- (d);
\draw[green,thick] (a) |- (b) (c) |- (d);
\draw[red,thick] (a) -| (b) -| (c) -| (d);
\draw[blue,thick] (a) -| (b) (c) -| (d);
\draw[black,thick] (a) -| (b) -- (c) |- (d);
```

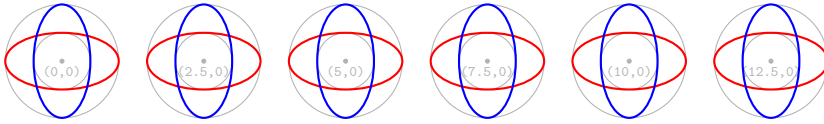


- **Kružnicu/kruh** definujeme sekvenciou `(stred) circle (polomer)`, resp. ekvivalentne `(stred) circle [radius=polomer]`.
- **Elipsu** definujeme sekvenciou `(stred) ellipse (polX and polY)`, resp. ekvivalentne `(stred) ellipse [x radius=polX, y radius=polY]`.

Slová `circle` a `ellipse` môžeme vymeniť. Elipsu nakreslíme pomocou `circle (polX and polY)`, resp. `circle [x radius=polX, y radius=polY]` a naopak kružnicu/kruh nakreslíme pomocou `ellipse (polomer)`, resp. `ellipse [radius=polomer]`. Elipsu môžeme vytvoriť z kružnice taktiež deformáciou poloosí pomocou `xscale=čísX, yscale=čísY`, resp. `x=jedX, y=jedY`. Musíme ale zvážiť, že sa objekt nielen deformuje ale aj posúva, t. j. musíme stred vrátiť do pôvodnej pozície. V našom prípade modré elipsy majú deformáciu vodorovnej osi 0,5, takže  $x$ -ovú súradnicu stredu musíme násobiť krát dva.

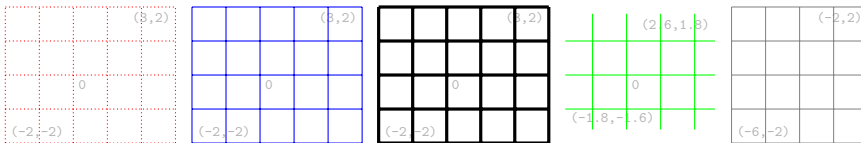
```
\draw[thick,red] (0,0) ellipse (1 and .5);
\draw[thick,red] (2.5,0) circle (1 and .5);
\draw[thick,red,xscale=1,yscale=.5] (5,0) circle (1);
\draw[thick,red,yscale=.5] (7.5,0) ellipse (1);
\draw[thick,red,x=1cm,y=.5cm] (10,0) circle (1);
\draw[thick,red,y=.5cm] (12.5,0) ellipse (1);
```

```
\draw[thick,blue] (0,0) ellipse [x radius=.5, y radius=1];
\draw[thick,blue] (2.5,0) circle [x radius=.5, y radius=1];
\draw[thick,blue,xscale=.5,yscale=1] (2*5,0) circle [radius=1];
\draw[thick,blue,xscale=.5] (2*7.5,0) ellipse [radius=1];
\draw[thick,blue,x=.5cm,y=1cm] (2*10,0) circle [radius=1];
\draw[thick,blue,x=.5cm] (2*12.5,0) ellipse [radius=1];
```

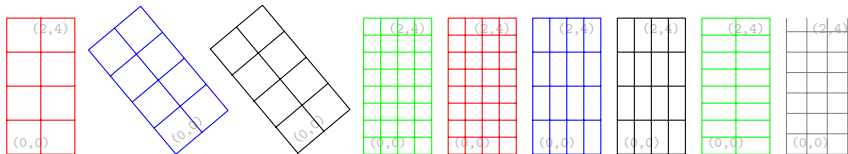


- **Obdĺžnik** definujeme vzťahom (súr1) rectangle (súr2), pričom (súr1), (súr2) sú súradnice protilahlých rohov obdĺžnika (uhlopriečne).
- **Súradnicová mriežka** sa definuje analogicky ako obdĺžnik súradnicami protilahlých rohov vzťahom (súr1) grid (súr2). Medzi voliteľné parametre môžeme pridať step=krok (krok), xstep=krokX (krok pre os  $x$ ), ystep=krokY (krok pre os  $y$ ), ak nie sú určené jednotky, použijú sa centimetre cm. Ak zadáme ako parameter help lines, mriežka sa zobrazí v sivej farbe. Voliteľné parametre môžeme meniť aj priamo za slovom grid.

```
\draw[red,densely dotted] (-2,-2) grid (3,2);
\draw[blue] (-2,-2) grid ++(5,4);
\draw[black,very thick] (-2,-2) grid (3,2);
\draw[green] (-1.8,-1.6) grid (2.6,1.8);
\draw[help lines] (-6,-2) grid (-2,2);
```



```
\draw[red] (0,0) grid (2,4);
\draw[blue,rotate=40] (0,0) grid (2,4);
\draw[black] (0,0) grid[rotate=40] (2,4);
\draw[green,step=.5] (0,0) grid (2,4);
\draw[red] (0,0) grid[step=.5] (2,4);
\draw[blue,xstep=.5] (0,0) grid (2,4);
\draw[black] (0,0) grid[xstep=.5] (2,4);
\draw[green,ystep=.5] (0,0) grid (2,4);
\draw[help lines] (0,0) grid[xstep=.6,ystep=.6] (2,4);
```



- **Orientovaný oblúk** definujeme (poč) arc (uholZ:uholK:polomer), resp. (poč) arc [start angle=uholZ, end angle=uholK, radius=polomer], pričom (poč) je bod, v ktorom sa oblúk začne vykresľovať (nie stred oblúka), uholZ a uholK sú uhly, kde oblúk začne a skončí, polomer je polomer oblúka.

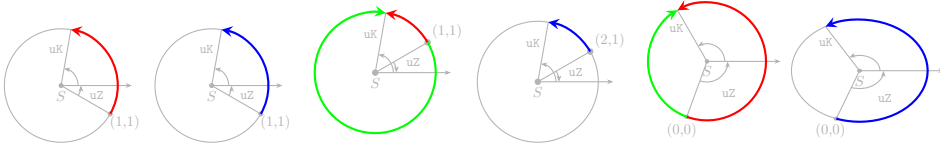
Stred oblúka je umiestnený v bode (poč) ++(180+uholZ:polomer), t. j. je posunutý zo začiatku vykresľovania (poč) do vzdialenosti polomer v protismere uhla uholZ, t. j. v smere uhla 180+uholZ. To znamená, že oblúk závisí iba od uhla uholZ, ale nezávisí od uhla uholK. Ak namiesto radius=polomer, definujeme x radius=polX, y radius=polY potom zostrojíme oblúk elipsy.



```

\draw[thick,red,-stealth] (1,1) arc [start angle=-30, end angle=80, radius=1];
\draw[thick,blue,-stealth] (1,1) arc (-30:80:1);
\draw[thick,red,-stealth] (1,1) arc (30:80:.8);
\draw[thick,green,-stealth] (1,1) arc (30:-280:.8);
\draw[thick,blue,-stealth] (2,1) arc (30:80:.8);
\draw[thick,red,-stealth] (0,0) arc (-110:120:1.2);
\draw[thick,green,-stealth] (0,0) arc (-110:-240:1.2);
\draw[thick,blue,-stealth] (0,0) arc
[start angle=-110, end angle=120, x radius=1.2, y radius=.9];

```

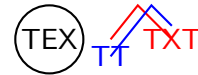


- **Text** vytvoríme vzťahom (súradnice) `node[nastavenia] {text}`. Ak nie sú súradnice zadané, text sa uloží v náväznosti na predošlý útvar, napr. pri kružnici `circle` to je jej stred a pri neuzavretých krivkách sú to ich konce, ale pozíciu textu môžeme zmeniť.

```

\draw[thick] (0,0) circle (.5) node {\sf TEX};
\draw[red,thick] (1.1,0)--(1.5,.5)--(2,0) node {\sf TXT};
\draw[blue,thick] (2,.2)--(1.7,.5)--(1.1,-.25) node {\sf TT};

```



Implicitne vytvorí `node` okolo textu box v tvare transparentného obdĺžnika, ktorého parametre môžeme meniť. Ak zmeníme rotáciu alebo škálu v nastaveniach pre celý obrázok, text sa nezmení, musíme ho zmeniť samostatne.

- ♠ `color` nastaví farbu obrysu a textu, resp. `{\color{farba} text}` zmení farbu iba textu. Farbu celého príkazu môžeme zmeniť tiež `\draw[farba]`.
- ♠ `draw` vykreslí obrys boxu, `circle` zmení obdĺžnik na kruh, `draw=farba` nastaví farbu obrysu boxu, `fill=farba` nastaví farbu výplne boxu (pre viac tvarov boxov musíme načítať `\usetikzlibrary{shapes.geometric}`).

Pre vykresľovanie obrysov boxu môžeme použiť známe príkazy na úpravu čiar `double`, `line width=hrúbka`, `thick`, `thin`, `dotted`, `dashed`, ...

```

\draw[fill] (0,0) circle (.05) node {TE XT};
\draw[fill] (0,0) circle (.05) node[draw] {TE XT};
\draw[fill] (0,0) circle (.05) node[draw,red] {TE XT};
\draw[fill] (0,0) circle (.05) node[draw,red] {\color{blue}TE XT};
\draw[fill] (0,0) circle (.05) node[draw,dotted,red] {TE XT};
\draw[fill] (0,0) circle (.05) node[draw,thick,red] {TE XT};
\draw[fill] (0,0) circle (.05) node[draw,thick,double,red,fill=yellow] {TE XT};
\draw[fill] (0,0) circle (.05) node[draw,line width=1mm,red,circle] {TT};

```

TE•XT

TE•XT

TE•XT

TE•XT

TE•XT

TE•XT

TE XT

T•T

- ♠ `right`, `above right`, `above`, `above left`, `left`, `below left`, `below`, `below right` určujú pozíciu boxu vzhľadom na súradnice, môžeme použiť aj `right=posun`, `above right=posun`, ..., kde `posun` určuje posunutie v danom smere. Ak nie je zadaná jednotka, implicitne sa nastaví bod `pt`.

- ♠ `scale=číslo`, `xscale=čísX`, `yscale=čísY` zväčší/zmenší v boxe oba dĺžkové rozmery `číslo` krát, resp. šírku `čísX` krát a výšku `čísY` krát.

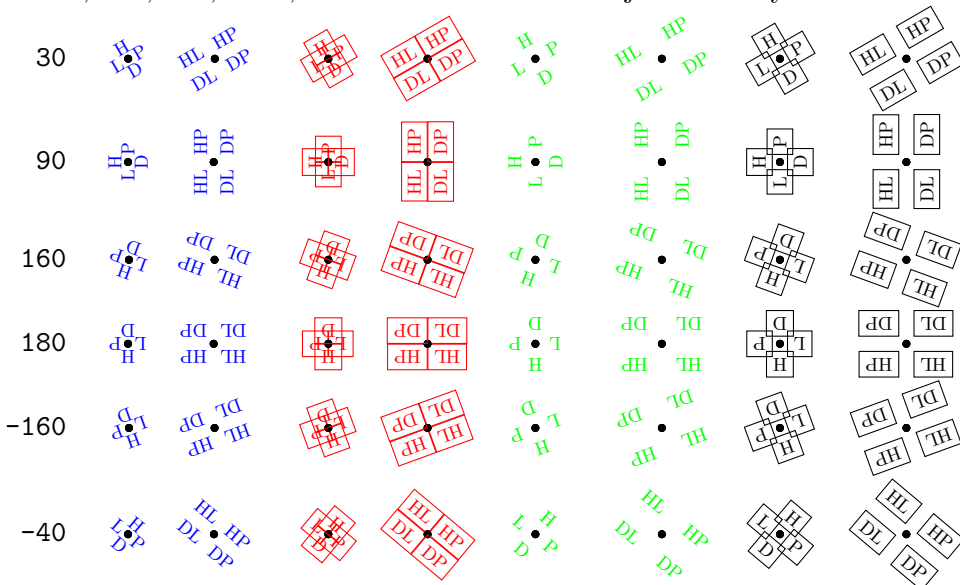
```
\draw[fill] (0,1) circle (.05) node[left,blue,scale=.65] {L};
\draw[fill] (0,1) circle (.05) node[right,blue,scale=.65] {P};
\draw[fill] (0,1) circle (.05) node[above,blue,scale=.65] {H};
\draw[fill] (0,1) circle (.05) node[below,blue,scale=.65] {D};
\draw[fill] (0,1) circle (.05) node[above left,blue,scale=.65] {HL};
\draw[fill] (0,1) circle (.05) node[above right,blue,scale=.65] {HP};
\draw[fill] (0,1) circle (.05) node[below left,blue,scale=.65] {DL};
\draw[fill] (0,1) circle (.05) node[below right,blue,scale=.65] {DP};
```

Predchádzajúce príkazy definujú modré obrázky, červené obrázky majú pridaný `node` parameter `draw`, t. j. `node[draw,...]`. Zelené a čierne obrázky majú oproti modrým a červeným obrázkom špecifikované posunutie `.25em`, t. j. obsahujú `node[left=.25em,...]`, `node[above left=.25em,...]`, ...



- ♠ `rotate=uhol_otočenia` otočí box o `uhol_otočenia` (v stupňoch) okolo bodu (súradnice).

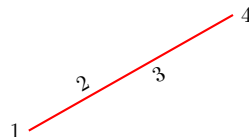
Na predchádzajúcom obrázku sú znázornené boxy v rôznych posunutiach okolo (súradníc). K príkazom generujúcim tieto podobrázky pridáme medzi `node` parametre `rotate=uhol_otočenia`, pričom postupne dosadíme 30, 90, 160, 180, -160, -40 a dostaneme nasledujúce obrázky.



- ♠ `pos=číslo` je desatinné číslo od 0 do 1, ktoré definuje pozíciu na popisovanej krivke v percentách od začiatku po koniec a na túto pozíciu sa uloží `{text}`. Medzi krivky, ktoré nemôžeme takýmto spôsobom popisovať patria `circle`, `ellipse` a `rectangle`. Ak použijeme parameter `sloped` (musí byť

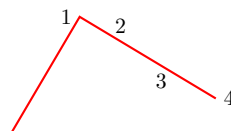
definovaná pozícia), potom sa popis automaticky prispôsobí sklonu krivky na danej pozícii.

```
\draw[red,thick] (0,0)--(3,1.7)
  node[black,left,pos=0,scale=.75] {1}
  node[sloped,black,above,pos=.3,scale=.75] {2}
  node[sloped,black,below,pos=.6,scale=.75] {3}
  node[black,right,pos=1,scale=.75] {4};
```



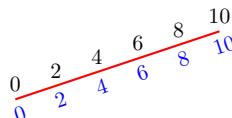
Ak krivku poskladáme z viacerých objektov, popíše sa iba posledná časť.

```
\draw[red,thick] (0,0)--(1,1.7)--(3,.5)
  node[black,left,pos=0,scale=.75] {1}
  node[black,above,pos=.3,scale=.75] {2}
  node[black,below,pos=.6,scale=.75] {3}
  node[black,right,pos=1,scale=.75] {4};
```

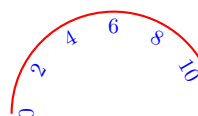


Popis môžeme optimalizovať pomocou cyklov.

```
\draw[red,thick] (0,0)--(3,1)
  \foreach \x in {0,2,...,10}
    {node[black,above,pos=\x/10,scale=.75] {\x}
     node[sloped,blue,below,pos=\x/10,scale=.75] {\x}};
```



```
\draw[thick,red,sloped] (0,0) arc (180:30:1.5)
  \foreach \x in {0,2,...,10}
    {node[blue,below,pos=\x/10,scale=.75] {\x}};
```



Preddefinované pozície sú `at end` (`pos=1`), `very near end` (`pos=.875`), `near end` (`pos=.75`), `midway` (`pos=.5`), `near start` (`pos=.25`), `very near start` (`pos=.125`), `at start` (`pos=0`).

- **Externý obrázkový súbor** vkladáme pomocou príkazu `\includegraphics` z balíčka `graphicx`, rovnako ako `text` pomocou `node`, vzťahom

(súradnice) `node[nastavenia] {\includegraphics[nastav]{obr}}`.

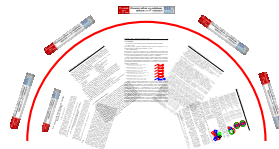
- ♠ `obr` je názov obrázkového súboru aj s koncovkou (podporované sú `.png`, `.jpg`, `.jpeg` a `.pdf`, iné treba externe prekonvertovať na tieto formáty).
- ♠ Parameter `width=šírka` (v `nastav`) definuje šírku obrázku (`width=3cm`, `width=.3\linewidth` ap.), `height=výška` definuje výšku obrázku. Ak použijeme iba jeden rozmer, druhý sa prispôsobí automaticky. Ak zadáme oba, obrázok sa deformuje. Deformácii zabráni parameter `keepaspectratio`.

Parameter `scale=číslo` zväčší/zmenší oba dĺžkové rozmery `číslo` krát a parameter `angle=uhol` otočí obrázok o daný orientovaný `uhol`. Ak je použité `clip=true`, resp. `clip`, potom môžeme orezať obrázok pomocou `trim={zľava zdola sprava zhora}`, kde sa daný obrázok oreže v daných smeroch o uvedené hodnoty, napr. `trim=1cm 1.5cm 1cm 1.5cm`, `clip`.

Ak má vkladaný `.pdf` súbor viac strán, potom môžeme vybrať konkrétnu stranu parametrom `page=čísloS`, kde `čísloS` je číslo strany, napr. `page=4`.

Pri vkladaní obrázka môžeme parametre `nastav` a `nastavenia` kombinovať.

```
\draw[thick,red,sloped] (0,0) arc (180:0:1.75)
\foreach \x in {1,3,...,9}
{node[blue,above,pos=\x/10,scale=.75]
{\includegraphics[width=1cm]{oss-konf.png}}
node[blue,below,pos=\x/10,scale=.75]
{\includegraphics[width=1cm,page=\x]{clanok.pdf}}};
```



## Poznámky na záver

Namiesto záveru vyslovíme zopár poznámok. Prvou z nich je skutočnosť, že príspevok vznikol bez aktívnej pomoci všadeprítomnej AI alebo IA(?).

Možností na úpravu grafických objektov je samozrejme oveľa viac, a nie je v možnostiach príspevku ich všetky obsiahnuť. Navyše po načítaní podporných knižníc `\usetikzlibrary{knižnica}` v preambule sa možnosti zväčšia.

Pri otočení `rotate=uhol_otočenia` o záporný uhol, rovnako aj pri posunutí `xshift=posX`, `yshift=posY` so zápornými hodnotami a pri konštrukcii čiary `(bod1) |- (bod2)` nastáva v slovenčine a češtine problém so znakom `-` (pomlčka/mínus). Ak sa v preambule aktivuje `\usepackage{slovak}{babel}`, resp. `\usepackage[czech]{babel}`, potom je nutné zaradiť príkaz `\shorthandoff{-}` hneď za `\begin{document}`.

**Podakovanie.** Príspevok vznikol s príspevom grantu KEGA 010ŽU-4/2026 „Matematika ako jazyk: myslenie, diskurz a porozumenie pomocou Open Source nástrojov“ podporeného slovenskou Kultúrnou a edukačnou grantovou agentúrou MŠVVaM SR.

## Literatúra

- [1] TikZ & PGF Manual for Version 3.1.9a: TikZ & PGF.
- [2] TikZ & PGF Manual online: [tikz.dev/](https://tikz.dev/).
- [3] CASTELEYN, J. P.: *Visual TikZ*, [mirrors.ctan.org/info/visualtikz/VisualTikZ.pdf](https://mirrors.ctan.org/info/visualtikz/VisualTikZ.pdf).

## Kontaktná adresa

**RNDr. Rudolf Blaško, PhD.**, Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovensko,  
*e-mail:* [beerb@frcatel.fri.uniza.sk](mailto:beerb@frcatel.fri.uniza.sk), <https://frcatel.fri.uniza.sk/users/beerb/>.

## CULTURAL MISALIGNMENT IN LARGE LANGUAGE MODELS: DETECTION, MEASUREMENT, AND MITIGATION THROUGH TARGETED FINE-TUNING

ANTONI CZOLGOWSKI (US) AND ABEL IYASELE (US)

**Abstract.** We evaluate three open-weight LLMs (Gemma3-12B from USA, Bielik-11B-v3 from Poland, Qwen3-4B from China) against World Values Survey Wave 7 data for 63 demographic personas, using normalized Wasserstein distance to quantify distributional misalignment. No model favors its country of origin: the Chinese-built Qwen3-4B performs worst on its own Chinese population ( $\hat{W}_1 = 0.436$ , the highest value in the model  $\times$  country matrix). Targeted LoRA fine-tuning on five worst-case personas, using fewer than 1 200 training pairs and under 15 minutes on a single GPU, reduces bias by 16.8% for Bielik-11B ( $p_{\text{Bonf}} = 0.002$ ,  $d = -4.4$ ), with all five targets improving. Country-level decomposition reveals that fine-tuning redistributes rather than removes bias: Bielik’s worst-case personas swap entirely from American to Chinese elderly, with zero overlap between pre- and post-correction sets. To our knowledge, this is the first study targeting worst-case demographic personas with LoRA fine-tuning for cross-cultural bias mitigation.

**Key words and phrases.** Large Language Models, Cross-Cultural Bias, World Values Survey, Wasserstein Distance, LoRA Fine-Tuning.

## KULTÚRNE NESÚLADY VO VEĽKÝCH JAZYKOVÝCH MODELOCH: DETEKCIA, MERANIE A ZMIERŇOVANIE PROSTREDNÍCTVOM CIELENÉHO DOLAĐOVANIA

**Abstrakt.** Hodnotíme tri modely s otvorenou váhou (Gemma3-12B z USA, Bielik-11B-v3 z Poľska, Qwen3-4B z Číny) na dátach World Values Survey Wave 7 zo 63 demografických oblastí, pričom na kvantifikáciu distribučnej nezhody používame normalizovanú Wassersteinovu metriku. Žiaden z modelov nepreferuje svoju domovskú krajinu: čínsky Qwen3-4B dosahuje najhoršie výsledky na vlastnej čínskej populácii ( $\hat{W}_1 = 0,436$ ). Cieľené LoRA dolaďovanie na piatich najhorších skupinách, využívajúce menej ako 1 200 tréningových príkladov a menej ako 15 minút na jednom GPU, znižuje skreslenie o 16,8 % pre Bielik-11B ( $p_{\text{Bonf}} = 0,002$ ,  $d = -4,4$ ), pričom sa zlepšilo všetkých päť cieľových skupín. Rozklad na úrovni krajín ukazuje, že dolaďovanie skreslenie skôr prerozdeľuje ako odstraňuje: najhoršie skupiny modelu Bielik sa úplne zamenia z amerických na čínske staršie osoby. Ide o prvú štúdiu cieľenú na najhoršie skupiny prostredníctvom LoRA dolaďovania na zmiernenie medzikultúrneho skreslenia.

**Kľúčové slová.** Veľké jazykové modely, medzikultúrna zaujatosť, prieskum svetových hodnôt, Wassersteinova metrika, LoRA dolaďovanie.

## 1. Introduction and Background

Large language models are increasingly deployed in contexts that affect people’s lives, from automated hiring systems and grant evaluations to public-facing chatbots and educational tools. Organizations adopt these systems as efficient and scalable, often without recognizing that the models carry systematic biases inherited from their training data and alignment procedures. When a model consistently misrepresents the values or opinions of particular demographic groups, the consequences extend beyond technical inaccuracy: in the right context, such misrepresentation can translate into discriminatory outcomes for the very populations the technology is meant to serve.

Large language models are not value-neutral. Trained predominantly on English-language Internet corpora, they absorb the cultural assumptions, political orientations, and social norms embedded in those texts. Rozado [7] demonstrated that most modern conversational LLMs exhibit left-leaning political preferences after instruction tuning; Tao et al. [9] found that GPT-family models cluster in the self-expression and secular-rational quadrant of the Inglehart-Welzel World Cultural Map regardless of the cultural context specified in the prompt; and Gallegos et al. [3] attribute much of this drift to the WEIRD demographics of RLHF annotators. An earlier study by one of the present authors [2] used WVS data to show that Gemma 2 2B reproduced opinion patterns of developed Western countries far more accurately than those of Libya, Nigeria, or India. The foundational shift toward distributional alignment came with Santurkar et al. [8], who adopted the 1-Wasserstein distance ( $W_1$ ) as the primary metric on the OpinionQA benchmark. Unlike Kullback-Leibler or Jensen-Shannon divergence,  $W_1$  respects the ordinal structure of Likert scales. Liu et al. [6] applied it across 64 countries and seven SOTA LLMs, confirming consistent advantages for English-speaking populations, and Zhao et al. [11] packaged WVS questions into a standardized benchmark with demographic conditioning.

Persona-based survey simulation conditions a model on a demographic profile and queries it like a human respondent. Hu and Collier [5] quantified the ceiling on this technique, finding that persona variables account for less than 10% of variance in human annotations on most subjective tasks. Among mitigation strategies, Low-Rank Adaptation (LoRA) [4] has emerged as particularly attractive: it freezes the pretrained weights and injects small low-rank matrices into attention layers, reducing trainable parameters by orders of magnitude while preserving general capabilities. Li et al. – demonstrated culture-specific LoRA adapters trained on WVS-derived data (CultureLLM), establishing that modest amounts of culturally grounded data can measurably shift a model’s cultural orientation.

The literature converges on a clear measurement toolkit but offers little on **targeted** mitigation at the demographic subgroup level. We evaluate three open-weight models of different cultural provenance (Gemma3-12B from the United States, Bielik-11B-v3 from Poland, Qwen3-4B from China) against WVS Wave 7 for 63 demographic personas, quantify bias with the normalized Wasserstein distance, and apply targeted LoRA fine-tuning to the worst-case personas. We ask:

- RQ1:** Do open-weight LLMs systematically favor demographic personas aligned with their country of origin?
- RQ2:** Does targeted LoRA fine-tuning on the worst-case personas significantly reduce bias?
- RQ3:** Does such targeted fine-tuning produce collateral side effects on non-targeted demographic groups?

## 2. Methodology

### 2.1. Data Source and Personas

We draw our human reference data from the World Values Survey (WVS) Wave 7, which collected standardized survey responses from 93,728 respondents across 64 countries between 2017 and 2022. The WVS is widely adopted as a benchmark for cultural alignment in language models [6,8,9,11]. We select question Q164: “How important is God in your life?” rated on a 1–10 scale. The question exhibits maximum cross-cultural variance among the candidate items we considered: China’s mean response is 2.8 (52% answer “1”;  $N \approx 3,000$ , strongly secular), Slovakia’s is 6.4 ( $N \approx 1,200$ , moderately religious), and the United States’ is 6.7 (39% answer “10”;  $N \approx 2,500$ , highly religious). The ordinal scale suits the Wasserstein metric, and Q164 has been used in prior LLM evaluation studies [9], enabling direct comparison. Poland does not appear in WVS Wave 7, so we use Slovakia as a culturally proximate substitute, justified by shared Visegrad-Group membership, adjacent positions on the Inglehart-Welzel cultural map, and comparable religiosity profiles shaped by Catholic heritage and parallel post-1989 transitions; throughout, references to the Central European region refer to Slovak respondents.

To evaluate bias at the demographic subgroup level we construct personas by crossing four features from WVS respondent metadata: country (CHN, SVK, USA; B\_COUNTRY), sex (male, female; Q260), age group (18–29, 30–49, 50–64, 65+; binned from Q262), and education level (lower, medium, higher; ISCED 0–2, 3–4, 5–8 from Q275R). This yields a theoretical design space of  $3 \times 2 \times 4 \times 3 = 72$  personas. We imposed a minimum threshold of  $N \geq 10$  WVS respondents per persona; nine cells fell below this threshold, yielding **63 valid personas** (23 Chinese, 23 Slovak, 17 American). Each persona is identified by a compound label encoding all four features (e.g., `USA_Male_30-49_Medium`). Each persona is queried

with a structured prompt stating the demographic profile in a system message followed by the survey question, following the profile-plus-question paradigm of Qu and Wang [?]. Each persona is queried 100 times per model with temperature 0.7, top- $k = 50$ , and max 10 new tokens. The baseline evaluation generates  $63 \times 100 \times 3 = 18,900$  queries; an additional 12,600 post-LoRA queries for Bielik and Qwen bring the total to 31,500. The parsing-failure rate was **0%**.

## 2.2. Wasserstein Distance Metric

To quantify the distance between human and model response distributions we adopt the 1-Wasserstein distance ( $W_1$ ), also known as the Earth Mover’s Distance. For discrete ordered distributions on  $\{1, \dots, 10\}$ ,

$$W_1(P, Q) = \sum_{k=1}^9 |F_P(k) - F_Q(k)|. \quad (1)$$

Following Santurkar et al. [8], we normalize by 9 (the maximum on a 1–10 scale) to obtain  $\hat{W}_1 = W_1/9 \in [0, 1]$ , where 0 indicates perfect alignment and 1 indicates maximum divergence. The choice of  $W_1$  over KL or JS divergence is motivated by its sensitivity to the ordering of response categories, essential for Likert-type items. To account for sampling variability we construct 95% bootstrap confidence intervals (1,000 resamples) on the human side and identify worst-case personas using the **lower bound** of the CI rather than point estimates, so that a persona is flagged only when we are confident its true  $\hat{W}_1$  is high.

## 2.3. Models and Fine-Tuning Procedure

Our study evaluates three open-weight LLMs from three distinct geopolitical contexts. **Gemma3-12B** (Google, USA, 12B parameters, Gemma architecture) is served via Ollama at full precision and serves as baseline only since Ollama does not expose HuggingFace-compatible weights. **Bielik-11B-v3** (SpeakLeash, Poland, 11B, Mistral) is closest in scale to Gemma3. **Qwen3-4B-Instruct-2507** (Alibaba, China, 4B, Qwen) is the smallest. Bielik and Qwen use 4-bit BitsAndBytes quantization, enabling all inference and fine-tuning on one A100 GPU on the CURC Alpine HPC cluster. The parameter disparity (4B vs. 11B vs. 12B) is a confound in direct comparisons but also lets us examine whether scale affects receptiveness to LoRA fine-tuning.

The fine-tuning intervention targets the five worst-case personas per model (identified by the bootstrap procedure above). To avoid data leakage we build training data exclusively from **non-religious** WVS Wave 7 questions, deliberately excluding Q164 and all religiosity-related items; the model therefore learns target-persona response patterns from politics, economics, social norms, and family values, and we evaluate whether this transfers to the held-out religiosity question. Each training example is an SFT pair (persona-conditioned prompt +



human modal response). We use LoRA [4] with  $r = 16$ ,  $\alpha = 32$ , dropout 0.05, targeting `q_proj` and `v_proj` in all attention layers; training runs 3 epochs with learning rate  $2 \times 10^{-4}$ , batch size 4. Bielik trained on 1,180 pairs in 13.6 min (loss  $2.53 \rightarrow 0.39$ ); Qwen on 1,003 pairs in 4.9 min (loss  $1.13 \rightarrow 0.36$ ). After fine-tuning each model is re-evaluated on all 63 personas. RQ1 is tested with a two-way ANOVA on  $\hat{W}_1$  (model  $\times$  country); RQ2/RQ3 use paired  $t$ -tests on baseline vs. post-LoRA  $\hat{W}_1$  for 5 targets and 58 non-targets respectively, with Bonferroni correction across four tests at  $\alpha = 0.05$ .

### 3. Results

#### 3.1. Cultural Origin Bias (RQ1)

**RQ1:** Do language models systematically favor demographic personas aligned with their country of origin?

The two-way ANOVA on  $\hat{W}_1$  ( $N = 189$ ) yields three significant effects: model ( $F(2, 180) = 56.54$ ,  $p < 2 \times 10^{-16}$ ), country ( $F(2, 180) = 5.25$ ,  $p = 0.006$ ), and the model $\times$ country interaction ( $F(4, 180) = 26.38$ ,  $p < 2 \times 10^{-16}$ ). The significant interaction demonstrates that the three models exhibit **different** patterns of bias across countries, the signature one would expect if cultural provenance mattered. Table 1 presents the cell means.

**Table 1.** Mean normalized Wasserstein distance ( $\hat{W}_1$ ) by model and persona country. Lower values indicate better alignment with human survey data. The bold cell in each row marks the model’s own country of origin.

|                     | China        | Slovakia     | USA          |
|---------------------|--------------|--------------|--------------|
| Bielik-11B (Poland) | 0.173        | <b>0.249</b> | 0.288        |
| Gemma3-12B (USA)    | 0.199        | 0.306        | <b>0.293</b> |
| Qwen3-4B (China)    | <b>0.436</b> | 0.286        | 0.348        |

Three patterns emerge. First, a **distribution-shape effect**: Gemma3-12B and Bielik-11B achieve their lowest  $\hat{W}_1$  on Chinese personas despite not being Chinese-origin models. We interpret this as a distributional artifact (Section 4): China’s distribution is heavily concentrated at the low end (52% answer “1”), making it a narrow target that concentrated model responses can partially match, whereas Slovakia and the US have broader distributions that are harder to approximate. Second, a **limited regional-proximity pattern** in Bielik: after accounting for the Chinese distribution-shape effect, Bielik aligns better with Slovakia ( $\hat{W}_1 = 0.249$ ) than with the United States ( $\hat{W}_1 = 0.288$ ), consistent with a modest Central European alignment effect. Third, the **Qwen paradox**:

the Chinese-origin model performs **worst** on Chinese personas ( $\hat{W}_1 = 0.436$ ). Inspection shows that Qwen consistently outputs moderate-to-high religiosity values even where the Chinese reference distribution is strongly secular, which we attribute to a model-level response default possibly shaped by instruction tuning or safety alignment.

**Answer to RQ1:** Models do **not** systematically favor their country of origin. Instead, baseline alignment reflects distributional difficulty, limited regional proximity in Bielik, and origin-country misalignment in Qwen.

### 3.2. Fine-Tuning Efficacy (RQ2)

**RQ2:** Does targeted LoRA fine-tuning on worst-case personas significantly reduce bias?

Bielik’s five worst-case personas turned out to be all American (baseline  $\hat{W}_1 \in [0.321, 0.365]$ , all middle-aged or younger men or one young woman, all medium or higher education); Qwen’s were all Chinese (baseline  $\hat{W}_1 \in [0.526, 0.637]$ , all male, four aged 65+).

Bielik: successful bias reduction. All five target personas show reduced  $\hat{W}_1$  after LoRA, with post-LoRA values in  $[0.276, 0.290]$ , improvements ranging from  $-14.2\%$  to  $-20.8\%$  (mean  $-16.8\%$ ). The paired  $t$ -test is highly significant even after Bonferroni correction:  $t(4) = -9.811$ ,  $p_{\text{Bonf}} = 0.0024$ , Cohen’s  $d = -4.388$ , 95% CI  $[-0.074, -0.042]$ . The result is noteworthy because the LoRA adapter was trained **only** on non-religious WVS questions yet improved alignment on the held-out religiosity question Q164. The adapter therefore learned transferable cultural patterns rather than memorizing a single item’s distribution, validating our core methodological assumption that cross-question cultural knowledge can be leveraged to correct bias on held-out evaluation items.

Qwen: failed bias reduction. In stark contrast, Qwen’s LoRA **increased** bias on four of five target personas (only `CHN_Male_30-49_Medium` held essentially flat), with post-LoRA values reaching  $[0.525, 0.723]$  and worsening of  $+13.5\%$  to  $+24.0\%$ . The paired  $t$ -test does not survive Bonferroni correction ( $t(4) = 3.563$ ,  $p_{\text{Bonf}} = 0.094$ ,  $d = +1.593$ ); the raw  $p$ -value would be significant at  $\alpha = 0.05$  without correction. We attribute the failure to a combination of smaller model capacity (4B vs. 11B), fewer training pairs (1,003 vs. 1,180), and possible resistance from instruction-tuned response defaults against low-religiosity outputs.

Bielik’s CDF moves visibly toward the human CDF after fine-tuning (the  $W_1$  area shrinking from 0.365 to 0.289), while Qwen’s CDF shifts **further** from the human CDF (widening  $W_1$  from 0.637 to 0.723); see the extended online version for the CDF comparison figure.

**Answer to RQ2:** LoRA fine-tuning on worst-case personas reduces bias for Bielik-11B ( $-16.8\%$ ,  $p_{\text{Bonf}} = 0.0024$ ,  $d = -4.4$ ) but backfires for Qwen3-4B ( $+13.9\%$ , n.s. after Bonferroni). The asymmetry is consistent with model-capacity differences and the interaction between LoRA corrections and existing RLHF alignment.

### 3.3. Side Effects (RQ3)

**RQ3:** Does fine-tuning to improve worst-case personas introduce negative side effects on other demographic groups?

For each fine-tuned model we evaluate  $\hat{W}_1$  changes on the 58 non-target personas. At the aggregate level (top of Table 2), neither model shows statistically significant side effects after Bonferroni correction (Bielik  $t(57) = 1.775$ ,  $p_{\text{Bonf}} = 0.325$ ,  $d = 0.233$ ; Qwen  $t(57) = 1.244$ ,  $p_{\text{Bonf}} = 0.875$ ,  $d = 0.163$ ). On its own this would suggest LoRA is “safe.” However, aggregate statistics can conceal opposing effects that cancel in summation. Motivated by the overcorrection patterns documented by An et al. [1], we decompose the analysis by persona country (bottom of Table 2). The decomposition reveals a **bias redistribution** pattern entirely invisible at the aggregate level.

**Table 2.** RQ3: Aggregate and country-level side effects on the 58 non-target personas. Aggregate non-significance masks a dramatic country-level redistribution.

| Model                              | Stratum            | $n$ | Mean $\Delta\hat{W}_1$ | Change | Improved | Worsened |
|------------------------------------|--------------------|-----|------------------------|--------|----------|----------|
| <i>Aggregate (all non-targets)</i> |                    |     |                        |        |          |          |
| Bielik-11B                         | all                | 58  | +0.019                 | –      | 24       | 29       |
| Qwen3-4B                           | all                | 58  | +0.012                 | –      | 26       | 27       |
| <i>Country-level decomposition</i> |                    |     |                        |        |          |          |
| Bielik-11B                         | China              | 23  | +0.077                 | +50.2% | 3        | 20       |
|                                    | Slovakia           | 23  | −0.032                 | −10.4% | 20       | 3        |
|                                    | USA (non-target)   | 12  | +0.008                 | +6.3%  | 6        | 6        |
|                                    | China (non-target) | 18  | +0.085                 | +27.6% | 2        | 16       |
| Qwen3-4B                           | Slovakia           | 23  | −0.001                 | +1.6%  | 9        | 10       |
|                                    | USA                | 17  | −0.047                 | −13.5% | 15       | 1        |

Bielik: the redistribution effect. Trained on five American personas with moderate to high religiosity (mean  $\approx 6.7$ ), Bielik’s adapter shifts the response distribution globally toward higher values. Chinese personas worsen by +50.2% on average (20 of 23 worsened) as the upward shift pushes responses away from Chinese secularity; Slovak personas improve by −10.4% (20 of 23) since Slovakia’s profile (mean 6.4) resembles the American targets; non-target American personas

show mixed results (+6.3%, 6 improved and 6 worsened). The most extreme collateral damage occurred among elderly Chinese personas: `CHN_Male_65+_Lower` experienced a +251% increase in  $\hat{W}_1$ , becoming Bielik’s new single worst-case persona. Bielik’s worst-case ranking underwent a complete turnover: all five pre-LoRA worst-case personas were American, all five post-LoRA are Chinese elderly, with zero overlap.

Qwen: a mirror-image redistribution. Qwen’s adapter, trained on Chinese personas, produces a mirror pattern: American personas improve by −13.5% (15 of 17 improved) while Chinese non-targets worsen by +27.6% (16 of 18). Because Qwen’s fine-tuning failed to improve even its targets (Section 3.2), the redistribution compounds failure: Chinese personas worsen across the board, targets and non-targets alike.

**Answer to RQ3:** No significant aggregate side effects, but country-level decomposition reveals a critical **bias redistribution**: LoRA does not remove bias so much as move it between cultural groups. Training on one country’s personas improves alignment for culturally similar populations while degrading alignment for distant ones; aggregate metrics alone are insufficient to detect this.

## 4. Discussion and Conclusions

Our results expose a methodological blind spot in cross-cultural bias evaluation. Low model–country distance does not necessarily mean cultural competence; it can also mean that the target distribution is simply easier to approximate. In our case, Gemma3-12B and Bielik-11B appear closest to Chinese personas largely because China’s responses are sharply concentrated at the low end of the scale. By contrast, Slovakia and the United States have broader, more dispersed distributions that require models to spread probability mass more precisely. The key implication is that cultural alignment scores should be read as an interaction between model behavior and the geometry of the human reference distribution, not as a pure measure of model bias.

The **Qwen paradox** ( $\hat{W}_1 = 0.436$  on its own Chinese population, the highest cell in the matrix) is the most counterintuitive finding. Qwen systematically produces moderate-to-high religiosity responses (7–8) even when ground truth is strongly secular. We interpret this as an RLHF artifact: annotators evaluating “importance of God” responses likely rate moderate answers as safer than extreme-low ones, and the reward model then steers the LLM away from the very outputs needed to align with Chinese reality. This is consistent with Rozado’s [7] finding that RLHF crystallizes value preferences during post-training and the broader observation [9] that LLMs gravitate toward Western-aligned values regardless of context: RLHF can function as a bias **amplifier** on culturally sensitive topics.

The country-level decomposition shows that LoRA does not remove bias but **redistributes** it: the adapter learns a directional correction applied globally because the low-rank update affects attention layers indiscriminately. The asymmetric outcomes for Bielik (success) and Qwen (failure) are consistent with model scale, the 11B model having sufficient capacity to integrate cultural signals from  $\sim 1,200$  pairs while the 4B did not, matching Hu and Collier’s [5] demographic-variance scaling result and Liu et al.’s [6] Qwen2.5 pattern; we cannot fully disentangle scale from confounds such as Qwen’s stronger RLHF and greater baseline misalignment.

**Practical framework.** Our findings suggest a three-step pre-deployment protocol: **Identify** worst-case personas by computing  $\hat{W}_1$  against human references using bootstrap CIs and the lower-bound criterion; **Mitigate** with targeted LoRA fine-tuning on non-overlapping survey items to prevent data leakage; **Evaluate** at the subgroup level, not only in aggregate, since aggregate non-significance can mask severe redistribution. The framework is valuable precisely because it reveals what aggregate evaluation can miss: targeted LoRA may improve one group while silently degrading others. In this sense, it is not merely a diagnostic tool, but an evaluation layer for future mitigation methods, especially constraint-aware training approaches that penalize collateral degradation on non-target groups.

**Conclusions.** Cultural bias in open-weight LLMs is measurable and exhibits systematic, interpretable patterns. We established highly significant model $\times$ country interaction ( $p < 2 \times 10^{-16}$ ). Targeted LoRA fine-tuning on five worst-case personas reduced bias by 16.8% in Bielik-11B ( $d = -4.4$ ) using fewer than 1,200 pairs in under 15 minutes on a single GPU. We uncovered a striking paradox in Qwen3-4B (a Chinese-origin model performing worst on its own population), which is consistent with the findings that instruction tuning or alignment procedures introduce a response default that conflicts with the target population distribution. To our knowledge, this is the first study to combine worst-case persona identification with targeted LoRA fine-tuning for cross-cultural bias mitigation. All code, data, and trained adapters are publicly released.<sup>1</sup>

**Acknowledgment.** This work used the Alpine HPC resource at CU Boulder, supported by NSF grants OAC-2201538 and OAC-2322260 [10], and data from the World Values Survey Wave 7.

## References

- [1] AN, J. – HUANG, D. – LIN, CH. – TAI, M.: *Measuring gender and racial biases in large language models: Intersectional evidence from automated resume evaluation*, PNAS Nexus, vol. 4, no. 3, 2025, <https://doi.org/10.1093/pnasnexus/pgaf089>.

<sup>1</sup>The extended online version contains the full literature review, additional figures, and complete appendix, <https://github.com/AntoniCzolowski/llm-cultural-bias>.

- [2] CZOLGOWSKI A.: 2025, *Does Artificial Intelligence Discriminate? Bias Analysis of the Gemma 2 2B Language Model*, Bachelor's thesis, SGH Warsaw School of Economics.
- [3] GALLEGOS, I. O. – ROSSI, R. A. – BARROW, J. – TANJIM, MD M. – KIM, S. – DERNONCOURT, F. – YU, T. – ZHANG, R. – NESREEN K. A.: *Bias – Fairness in Large Language Models: A Survey*, 2024, <https://arxiv.org/abs/2309.00770>.
- [4] HU, E. J. – SHEN, Y. – WALLIS, P. – ALLEN-ZHU, Z. AN – LI, Y. – WANG, S. – WANG, L. – CHEN, W.: *LoRA: Low-Rank Adaptation of Large Language Models*, ICLR, vol. 1, no. 2, pp. 3, 2022.
- [5] HU, T. – COLLIER, N.: *Quantifying the Persona Effect in LLM Simulations*, 2024, <https://arxiv.org/abs/2402.10811>.
- [6] LIU, Y. – KANEKO, M. – CHU, CH.: *On the Alignment of Large Language Models with Global Human Opinion*, 2025, <https://arxiv.org/abs/2509.01418>.
- [7] ROZADO D.: *The Political Preferences of LLMs*, 2024, <https://arxiv.org/abs/2402.01789>.
- [8] SANTURKAR S. – DURMUS, E. – LADHAK, F. – LEE, C. – LIANG, P. – HASHIMOTO, T.: *Whose Opinions Do Language Models Reflect?*, pp. 29971–30004, 2023, vol. 202, Proceedings of Machine Learning Research, PMLR, <https://proceedings.mlr.press/v202/santurkar23a.html>.
- [9] TAO, Y. – VIBERG, O. – BAKER, R. S. – KIZILCEC, R. F.: *PNAS Nexus, Cultural bias – cultural alignment of large language models*, vol. 3, 9, 2024, doi: 10.1093/pnas-nexus/pgae346, <https://doi.org/10.1093/pnasnexus/pgae346>.
- [10] University of Colorado Boulder Research Computing: *Alpine*, 2023, doi: 10.25811/k3w6-pk81.
- [11] WENLONG, Z. – MONDAL, D. – TANDON, N. – DILLION, D. – GRAY, K. – GU, Y.: *WorldValuesBench: A Large-Scale Benchmark Dataset for Multi-Cultural Value Awareness of Language Models*, 2024, <https://arxiv.org/abs/2404.16308>.

## Contact addresses

**Antoni Czolowski, M.S. Candidate**, Masters in Data Science Program, University of Colorado Boulder, Boulder, CO 80309, USA,  
*e-mail*: [antoni.czolowski@colorado.edu](mailto:antoni.czolowski@colorado.edu)

**Prof. Abel Iyasele, Ph.D.**, Department of Information Science, University of Colorado Boulder, Boulder, CO 80309, USA,  
*e-mail*: [abel.iyasele@colorado.edu](mailto:abel.iyasele@colorado.edu)

## CMRX: NOVEL APPROACH TO MEMORY ISOLATION IN LOW-POWER EMBEDDED SYSTEMS

EDUARD DRUSA (SK)

**Abstract.** Many realtime operating systems for microcontrollers offer feature of memory isolation. Typically it is implemented either as user-mode or as partitioning. Partitioning offers more fine-grain isolation abilities than simple kernel / user separation. Yet it comes at a great price: partitions have to be configured manually and even moderately complicated configuration can easily deplete available regions. Here we present different approach to memory isolation which is both better scalable and easier to maintain than traditional techniques. Much of industry is reluctant to this idea because it “takes away control”. Therefore we open-sourced the operating system so you could help.

**Key words and phrases.** MCU, microcontroller, memory protection, cybersecurity, RTOS.

## CMRX: NOVÝ PRÍSTUP K OCHRANE PAMÄTE NA ENERGETICKY ÚSPORNÝCH EMBEDDED SYSTÉMOCH

**Abstrakt.** Väčšina real-time operačných systémov pre mikrokontroléry podporuje nejakú formu ochrany pamäte. Typicky je táto podpora buď vo forme „user módu“ alebo vytvárania pamäťových partícií. Partície poskytujú oveľa flexibilnejšie možnosti nastavenia prístupu ako jednoduchá izolácia kernelu a aplikácie. Táto flexibilita je ale výkopená potrebou manuálnej konfigurácie partícií. V reálnych situáciach môže aj relatívne jednoduchá konfigurácia naraziť na limity použitého hardvéru. V tomto článku demonštrujeme konceptuálne odlišný prístup k ochrane pamäte ktorý odbúrava potrebu manuálneho manažmentu a radikálne zvyšuje flexibilitu riadenia prístupu k pamäti. Väčšina z odvetvia má k takému prístupu odpor, pretože „berie vývojarom kontrolu“. Preto sme zdrojové kódy systému otvorili, aby ste mohli pomôcť.

**Kľúčové slová.** MCU, mikrokondrollér, ochrana pamäte, kyberbezpečnosť, RTOS.

## Introduction

Currently, there are two major state-of-art approaches to implement memory isolation in low-power<sup>1</sup> embedded systems. The less elaborate system implemented by some RTOS is usermode, where kernel and drivers are isolated from all the userspace software by means of memory protection. Here kernel and drivers form one large isolation domain unreachable from outside. This approach is simple, well defined and works as a measure to improve kernel stability. Though it is not

---

<sup>1</sup>In this article “low-power” embedded device will describe an embedded device with microcontroller without memory mapping unit, such as ARM Cortex M family or similar.

suitable in scenarios involving mixed criticality or when device is connected as drivers providing connectivity usually run in privileged mode.

For such scenarios partitioning is used. Here the developer defines one or more memory regions which are accessible by certain tasks (partitions) and then assigns data to these regions. While this approach works for applications that really need it, such as critical medical devices, it has some major drawbacks. The drawback which is obvious yet may not be acknowledged by professionals is the labour-intensive manual nature of partition configuration. Memory isolation is considered a high-end feature and as such it is accepted that it is hard to configure. This makes the developer that configures partitioning a single point of failure, especially in case it is a lead developer who is considered to be the only developer suitable to do such job.

Another major drawback is that manual partition configuration can easily deplete regions available in typical hardware. This is due to the fully static nature of partition configuration and the need for a way to communicate between isolation domains. For communication purpose a block of shared memory is typically used which consumes region in every isolation domain where it is present. With devices typically having 8 protection regions this quickly leads to region depletion and forces developers to simplify the isolation scheme.

With all the drawbacks and limitations of partitioning one problem existing with kernel/usermode separation still exists with partitioning: kernel-resident drivers run unprotected with unlimited privileges.

## The Problem

All mechanisms to address memory isolation currently in use were designed to ensure functional safety is not at risk. Both safety and security have same goal — keeping the system running — but they focus on completely different threats. Functional safety deals with risk from mechanical and electrical failures and misuse. Security deals with handling deliberately malformed inputs crafted with the goal of gaining control over system or disabling it.

From safety point of view, memory protection is a mechanism that provides spatial isolation. This isolation ensures that no other than high-risk component itself can affect its state by accidentally or erroneously overwriting its memory. Once this guarantee is in place, only code of high-risk component itself needs to be considered under high Safety Integration Level and rest of firmware needs less scrutiny. Partitioning is sufficient solution here as one device usually has one critical function which needs to be protected. This leads to safe/unsafe partitioning schema [1, Chapter 1] which is not sufficient to address cybersecurity risks.

Major difference between functional safety and cybersecurity is that unsafe partition may easily become cybersecurity relevant. For example an ABS unit



may consider its CAN connection as not safety-relevant. If CAN communication becomes unavailable, ABS unit will continue operating just with internal signals provided by sensors. No malicious behavior is expected from unsafe partition.

This assumption falls apart if attacker is able to communicate with such ECU. Potential vulnerability in CAN communication stack may allow an attacker to gain control over this large partition. While he is not be able to manipulate safe partition directly, he may be able to manipulate inputs/outputs into safe partition. These are often protected by means preventing accidental data damage, but these can be easily circumvented by attacker calculating protection markers for forged input.

To prevent this scenario the standard approach is to limit the attack surface. This is done by isolating components with external visibility from the rest of the system. Implementing attack surface limitation will increase pressure on hardware resources as system which required two or three partitions and one shared memory region in the past may need five or six partitions. That may be enough to exhaust available regions of most hardware as typical microcontroller carries between 4 and 8 MPU regions.

## Different Approach

We found traits of partitioning described above unacceptable. Limited flexibility and labour-intensive configuration make it “use only if absolutely necessary” feature. With ever-increasing use of IoT devices which get targeted by online attacks and Cybersecurity Resilience Act it may happen that the range of “absolutely necessary” applications will grow much bigger than it is today. Thus we decided to use different mechanism which provides same basic security guarantees as bare partitions while it is much easier to maintain.

It turns out that reinventing the wheel is not necessary here. Such mechanism is used in general purpose computing for decades. This mechanism is process level isolation. From developer’s point of view it has similar features as partitioning – it isolates memory of various parts of software from each other. The major difference between embedded system partitions and processes is how these isolation boundaries are defined: With partitions their definition is up on the developer. With processes these boundaries are naturally defined by process boundaries and automatically enforced by operating system without developer’s intervention.

## The Process in MMU-less architecture

There are two major obstacles in retrofitting processes to architectures that don’t support memory mapping. First comes from the fact that all platforms that implement process-level isolation rely on some kind of memory mapping of addresses

to achieve virtual address space. Memory mapping is necessary to avoid addressing collisions when two processes want to place their belongings to same address range. Second obstacle comes from the fact that processes are individual executables while firmware is one monolithic block which contains no natural boundaries.

It turns out that in case of low-power embedded firmware these two obstacles cancel out. A typical device firmware only does one thing and thus rarely needs to dynamically load executable code on demand. In this case no address collisions exist by default.

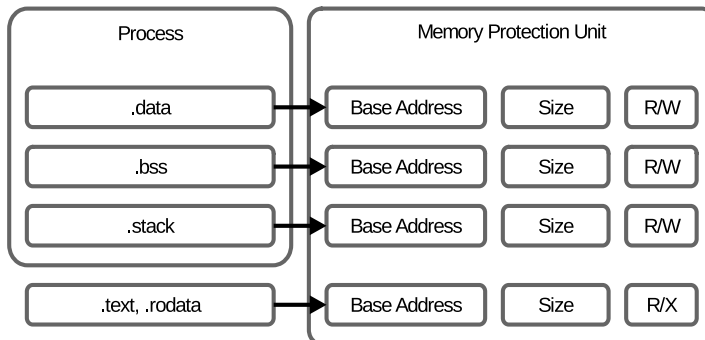
So the only task left for resolution was to find a naturally existing boundaries to be used to calculate memory isolation boundaries without explicit developer's involvement.

While there are no individual executables in firmware there is another building block that exists in firmware compile time but has no runtime overhead – static libraries. So we built the boundary determining mechanism around these.

Static libraries have the property that they are built as mostly individual pieces of code and data and later they are linked together with the rest of binary to form one monolithic firmware. Their use thus doesn't introduce any runtime overhead which could slow the firmware down. Yet they still provide identifiable boundaries.

Thus the process-level isolation mechanism works in a way that it treats one static library as one process. This provides developer with very clear and simple mental model of memory isolation: whatever resides inside one library belongs together and can access variables inside that library. Everything else is off-limits.

This model is automatically enforced by the tooling during firmware build. Developers simply divide their firmware into libraries and these are then fully automatically converted into processes. Tooling will take care of memory layout so fixed amount of regions is used by the process and there is no risk of region depletion or need for manual allocation.



**Figure 1.** Automatic mapping of process sections to memory protection regions

With such automation in place the whole developer's task to define new process basically shrinks to one line of code. Operating system will ensure correct MPU configuration is loaded when threads belonging to process are executing. This way any number of processes can exist. Operating system will load correct MPU configuration when thread is executed. Monolithic nature of firmware provides an advantage here: this configuration can be statically computed at compile time and boils down to copy operation at runtime. This makes process-level isolation fast.

An example listing of fully defined process with one thread is below:

```
#include <cmrx/application.h>
#include <pico/stdlib.h>
#include <cmrx/ipc/timer.h>

#define BUFSIZE 512

static unsigned blink_rate = 500000;
static char buffer[BUFSIZE];

static int blinky_main()
{
    gpio_init(PICO_DEFAULT_LED_PIN);
    gpio_set_dir(PICO_DEFAULT_LED_PIN, GPIO_OUT);
    while (1) {
        gpio_put(PICO_DEFAULT_LED_PIN, 1);
        usleep(blink_rate);
        gpio_put(PICO_DEFAULT_LED_PIN, 0);
        usleep(blink_rate);
    }
    return 0;
}

OS_APPLICATION_MMIO_RANGES(blinky,
    0x40000000, 0x50000000,
    0xd0000000, 0xe0000000);
OS_APPLICATION(blinky);
OS_THREAD_CREATE(blinky, blinky_main, NULL, 32);
```

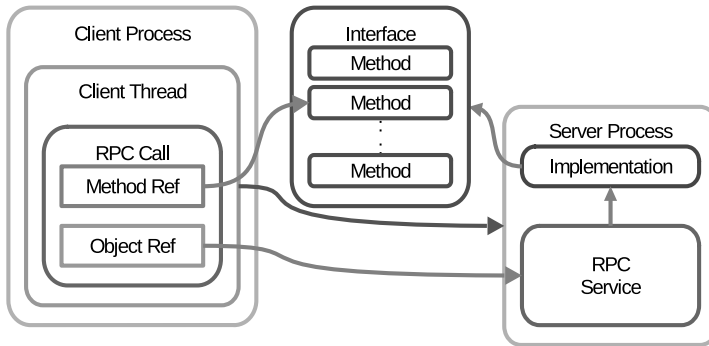
The code in listing implements a simple blinking routine. It contains nothing special except of three declarations at the very bottom. One of them declares that the library this code resides in is actually a process.

## Processes Talking to Each Other

In typical memory-isolated firmware partitions talk to each other using shared memory or via kernel-provided facilities such as queues. When memory protection is active, queues can't be implemented as zero-copy so kernel needs to manage

additional buffers and each message pass involves at least two additional copies. The disadvantage of shared memory in memory-isolated systems is that each shared memory block permanently occupies MPU region in each process that does use it. For example: if partition shares data with two other partitions using two blocks of shared memory this will consume two more MPU regions in that partition's region list.

To avoid these drawbacks and limitations we designed a remote procedure calling mechanism. This kernel-supported mechanism allows a process to call a function from other process, temporarily giving this function access to part of caller's own memory.



**Figure 2.** Scheme of remote procedure calling mechanism

Developer is free to determine which variables in caller process' address space will become accessible to callees during RPC calls originating from threads in this process using SHARED variable qualifier. Kernel will grant callees access to these variables automatically while remotely-called method is executed. This way RPC calls serve both purpose of sharing memory between processes and synchronization primitive as RPC method is executed in the thread which made the call.<sup>2</sup> Thus caller and callee can't access shared data simultaneously.

With RPC calls in CMRX the caller doesn't provide callee directly, rather the concept of interfaces is used. In this schema caller is using service and method references. Service reference is a pointer to callee-maintained object living in callee's address space (and thus being an opaque pointer for caller) which contains list of callable methods. Method reference is plain offset in this table. This double indirection adds a small runtime overhead but allows for complete decoupling of server and client which both refer to interface definition when determining which method to implement and call.

<sup>2</sup>This effectively constitutes mechanism of thread borrowing when thread invoking RPC call is borrowed into callee's process to get access to callee's memory during RPC call execution.

Unlike many other implementations of RPC mechanism CMRX avoids the use of IDL and uses plain C structures to define interfaces. This helps to enable full compile-time type-checking with no additional tooling needed – all C11-compliant compilers are able to type-check RPC calls automatically. Compile-time type checking means that there is no added runtime checking overhead. RPC mechanism also doesn't perform any data marshalling; instead RPC calls are dispatched as plain function calls with kernel reconfiguring MPU when RPC method is entered and left.

## From Process-Based Firmware to Microkernel

With memory isolation and RPC calling in place, the difference between a pure process and a driver boils down to driver being granted access to certain explicitly specified region of memory where peripheral it controls is mapped. This can be trivially achieved. Actually the program in listing above is a driver as it was given access to GPIO peripheral of the RP2040 microcontroller using `OS_APPLICATION_MMIO_RANGES` macro.

In CMRX, drivers use the same RPC interfaces to provide their API which processes use to talk to each other. This approach provides solution for problem of passing data between application and driver across isolation boundaries – RPC calls allow sharing memory. Other than explicit permission to access certain peripheral memory region, drivers are just ordinary processes with no special privileges granted. This helps to reduce attack surface of highly privileged code compared to monolithic kernels which usually run drivers in kernel space with full kernel privileges.

With device drivers displaced out of kernel, nothing prevented shrinking the kernel to absolute bare minimum – it became a microkernel. Queues mentioned previously, which cease to be zero-copy in memory-isolated environments were displaced out of the kernel too. In most use-cases queues are only used locally within one process. What used to be a syscall is mostly used as a library function now. For those rare cases when queues are needed cross-process, there's userspace queue server which provides the functionality. Such minimalistic kernel is easier to test, offers potential for rather cheap formal verification and can easily be made very robust.

## Summary

The approach presented in this article takes fresh take on memory isolation in low-power embedded systems. It is less demanding towards developers and rules out scenarios where memory isolation might be inadvertently configured wrong. It also provides better overall security of the whole firmware as the attack surface of each component is greatly reduced.

Yet, in the realms of overly conservative embedded industry we have to ask one question: Who ordered this?

The reality is that despite memory protection hardware is a standard equipment of modern microcontrollers for more than two decades<sup>3</sup> it is rarely used. Arguments developers often state are that it is not necessary and/or it is complicated to maintain [2].

With this attitude being prevalent in the industry it is hard to tell if any novel approach has chance to succeed. One factor that might force the industry to embrace memory isolation in more scenarios is the Cybersecurity Resilience Act. This norm that takes effect in late 2027 may<sup>4</sup> require more firmware authors to use memory isolation based on criticality of their use cases. If this happens then ease of partition handling may help to actually spread this approach.

This industry-wide ignorance of memory protection also has another side-effect: SDKs and HALs are largely incompatible with memory-isolated environments. Many SDKs expect that SDK routines run with full privileges and have unrestricted access to peripherals. This makes introduction of any means of memory isolation often a hard task.

## References

- [1] SMITH, C.: *The car hacker's handbook*, No Starch Press, 2016, San Francisco, USA.
- [2] ZHOU, W. – GUAN, L. — LIU, P. – ZHANG, Y.: *Good Motive but Bad Design: Why ARM MPU Has Become an Outcast in Embedded Systems*, 2019, <https://arxiv.org/abs/1908.03638>.

## Contact address

**Ing. Eduard Drusa**, CMRX Realtime Microkernel Project, 040 01 Košice, Slovakia,  
*e-mail*: [eduard.drusa@cmrxrtos.org](mailto:eduard.drusa@cmrxrtos.org).

---

<sup>3</sup>For example, the Cortex-M3 processor core introduced in 2004 supports memory protection. So do many implementations of ARM cores from this era, such as AT91SAM7 or STM32F1 series.

<sup>4</sup>As of time of writing this article only incomplete drafts of horizontal standards are available thus the real extent of requirements is not known.

## L<sup>A</sup>T<sub>E</sub>X, ORG A BABYLÓN JAZYKOV

RICHARD FABO (SK)

**Abstrakt.** Článok voľne naväzuje na prednášku „Python ako priateľ L<sup>A</sup>T<sub>E</sub>X-u“ [1]. Pri používaní značkovacieho jazyka rozšírenia *org-mode* (nielen) textového editora *GNU Emacs* je možné využiť funkcionality nazývanú „babel“, ktorá ponúka efektívne nástroje pre spracovanie textu, a to nielen pre programátorov, ale aj pre akademikov a výskumníkov. Ako názov napovedá, jedná sa o „miešanie“ rôznych programovacích či značkovacích jazykov v jednom dokumente. Súčasne je možné vložené kódy vyhodnocovať a ďalej spracovávať priamo v prostredí uvedeného textového editora.

**Kľúčové slová.** Emacs, L<sup>A</sup>T<sub>E</sub>X, org mode, babel, značkovací jazyk.

## L<sup>A</sup>T<sub>E</sub>X, ORG, AND THE BABYLON OF LANGUAGES

**Abstract.** This article freely follows on the lecture “Python as a friend of L<sup>A</sup>T<sub>E</sub>X” [1]. When using the markup language extension *org-mode* of the (not only) text editor *GNU Emacs*, it is possible to use a feature called “babel”. As the name suggests, this involves “mixing” different programming or markup languages in a single document. It is also possible to evaluate and further process parts of the entered code.

**Keywords.** Emacs, L<sup>A</sup>T<sub>E</sub>X, org mode, babel, markup language.

## Úvod

Kedy má zmysel používať viaceré programovacie jazyky v jednom dokumente? Určite nás napadnú myšlienky o písaní dokumentácie, návodov, výukových materiálov. Donald E. Knuth, známy autor systému T<sub>E</sub>X, kníh o algoritmických teóriách a programovaní, v rámci svojho diela „The Art of Computer Programming“ uviedol koncept tzv. *literárneho programovania*, ktorý kombinuje kód s dokumentáciou. Svoje využitie nájde i v písaní konfiguračných súborov, pretože je možné z kúskov kódu automatizovane vygenerovať kompletný súbor. Vďaka systému predávania hodnôt premenných je možné použiť *Babel* ako alternatívu k Jupyter notebooku<sup>1</sup>. Presne tak: výborne zvolený názov tohto rozšírenia *Babel* symbolizuje jazykovú rozmanitosť.

---

<sup>1</sup>Jupyter je projekt, ktorý umožňuje vytvárať interaktívne dokumenty známe ako „Jupyter notebooky“. Tieto dokumenty kombinujú kód, text a vizualizácie a sú populárne v oblasti vedy, vzdelávania a výskumu. <https://jupyter.org/>

## 1. Čo je to vlastne Org-mode?

*Org-mode*, označovaný aj *org-mode*, *ORG*, *org* je rozšírením GNU Emacs-u<sup>2</sup>, ktoré slúži na organizáciu, a spravovanie textových informácií. Je určené na písanie poznámok, úloh, plánovanie projektov a iné organizačné účely.

Org-mode umožňuje vytvárať hierarchické zoznamy, sledovať časové harmonogramy a má ďalšie funkcie, ktoré umožňujú efektívnejšiu prácu s textom. Org-mode tiež poskytuje možnosť exportu dokumentov do rôznych formátov (HTML, PDF, L<sup>A</sup>T<sub>E</sub>X, ODT, ...). Dajú sa v ňom efektívne tvoriť tabuľky, dá sa „previazať“ s inými programami. Dá sa povedať, že to nie je len značkovací jazyk, ale efektívne prostredie na ľubovoľnú prácu s informáciami v textovej podobe. V ďalšom texte autor nadväzuje na informácie z publikovaného dokumentu [2].

## 2. Nevyhnutné nastavenia

Aby sme mohli využiť funkcionality *babel*, musia byť splnené minimálne tieto požiadavky: *org* mód vo verzii minimálne 7.0, *Emacs* vo verzii minimálne 24. a príslušné kompilátory alebo interprety vyhodnocovaných jazykov (napr. gcc, bash, R, python).

### Hlavička v *org* dokumente

V hlavičke *org* dokumentu je nevyhnutné uviesť tieto vlastnosti:

```
:HEADER:
#+PROPERTY: header-args :eval yes
:END:
```

Ostatné nastavenia de-facto kopírujú potrebnú preambulu *.tex* súboru, napr.

```
#+LATEX_CLASS: article
#+LATEX_CLASS_OPTIONS: [a4paper]
#+LATEX_HEADER: \usepackage{lmodern}
#+LATEX_HEADER: \usepackage{subcaption} \usepackage{listings}
#+TITLE: *Demo \LaTeX{} dokument vytváraný v~org mode/*
```

### Babel

Predvolene Emacs nemá podporu pre konkrétne jazyky aktivovanú, hoci je nainštalovaná v jeho prostredí. Ich aktivovanie spravíme zadefinovaním párovej dvojice (*<jazyk>* . *t*). Podľa nastavení to môže byť spravené v niektorom z konfiguračných súborov (*.emacs*, *init.el*, ...), možné je využitie i textového rozhrania.

---

<sup>2</sup>GNU Emacs je bohatý textový editor, rozširiteľný pomocou skriptov v jazyku (e)lisp.



```
(setq org-babel-load-languages '((emacs-lisp . t)
                                   (shell . t)
                                   (C . t)
                                   (python . t)))
```

Z bezpečnostných dôvodov je potrebné pri kompilácii `.org` dokumentu do `.tex` opakovane povoľovať vyhodnotenie jednotlivých blokov kódu. Tieto môžeme automaticky povoliť nastavením súvisiacej premennej:

```
(setq org-confirm-babel-evaluate nil)
```

### 3. Vloženie a vyhodnotenie kódu

Emacs v *org* dokumente rozpoznáva kľúčové slová `#+BEGIN_SRC` a `#+END_SRC` (nezáleží na veľkosti písmen). Určujú začiatok a koniec bloku vloženého kódu.

Syntax je nasledovná:

```
#+NAME: <označenie bloku (t.j. referencia)>
#+BEGIN_SRC <jazyk> <prepínače> <argumenty hlavičky>
  <telo>
#+END_SRC
```

Toto riešenie vysádza výstup kódu ako samostatný odsek. V prípade potreby vloženia výstupu priamo do textu, použijeme tzv. „inline“ kód, s nasledovnou syntaxou:

```
src_<jazyk>[<argumenty hlavičky>]{<telo>}
```

V rámci dokumentu je výhodné i definovanie globálnych premenných, ktorých platnosť nie je obmedzená na aktuálny blok príkazom `:var <názov>=<hodnota>`. `<prepínače>` poskytujú upresnenie vykonávania a formátu kódu.

`<argumenty hlavičky>` ovládajú mnoho aspektov exportu a prepojenia blokov kódu.

Bohatý zoznam takmer štyridsiatich podporovaných jazykov a ich identifikátorov je dostupný na <https://emacsdocs.org/docs/org/Languages>. Nebolo by účelné opisovať dokumentáciu dostupnú online (<https://emacsdocs.org/docs/org/The-Org-Manual>). Preto uvedieme najčastejšie prepínače a argumenty:

- `:includes <knižnica.h>` vloženie požadovaných knižníc jazyka C;
- `:results verbatim` výsledok sa vloží do textu ako neproporcionálny text (`\texttt`), ďalšie možnosti s vysvetľujúcim pomenovaním sú `list`, `scalar` či `file`;
- `:exports results` vloží do textu len výstup bloku kódu, iné bežné možnosti sú `code` alebo `both`.

Vstupnými dátami môžu byť aj ASCII tabuľka či pomenovaný zoznam. Príklad použitia napovie viac. Definovanie bloku:

```

#+NAME: fakt
#+BEGIN_SRC haskell :exports results :var n=0
    fac 0 = 1
    fac n = n * fac (n-1)
#+END_SRC

```

A jeho vykonanie priamo v texte:

*Faktoriál čísla 5 je* `src_haskell[:results raw]{fac 5}`.

Ďalší príklad, ktorý zobrazí zdrojový kód a aj jeho výsledok (krátené):

```

#+NAME: Kód interpretra shell
#+BEGIN_SRC sh :results verbatim :exports both
    echo "Kernel = $(uname -r)"
    echo "Hostname = $(hostname)"
    echo "Username = $(whoami)"
#+END_SRC

```

Ďalšou možnosťou je zobrazit výsledky priamo pod kódom tak, že tieto sa neexportujú pri prevádzaní *org* dokumentu do *L<sup>A</sup>T<sub>E</sub>X-u*. Takže aj bez prekladu celého dokumentu<sup>3</sup> vidíme výsledky kódu. Vykonáva sa to stlačením `C-c C-c` resp. zavolaním internej funkcie s vtipným názvom `org-ctrl-c-ctrl-c`.

```

#+begin_src python :results value latex :exports results
import sys
info = sys.version_info
return(str(info.major) + ".\\textbf{" + str(info.minor) + "}. " + str(info.micro))
#+end_src

#+RESULTS:
#+begin_export latex
3.\\textbf{11}.2
#+end_export

```

**Obr. 1.** Vyhodnotenie kódu priamo do dokumentu.

Za zmienku stojí, že zavolaním funkcie `org-edit-special` resp. stlačením `C-c '`, ak je kurzor v tele kódu, môžeme editovať kód v osobitnom buffri, so všetkými výhodami dedikovaného vývojového prostredia (používanie autodoplnovania, „snippetov“, skokov na referencie,...).

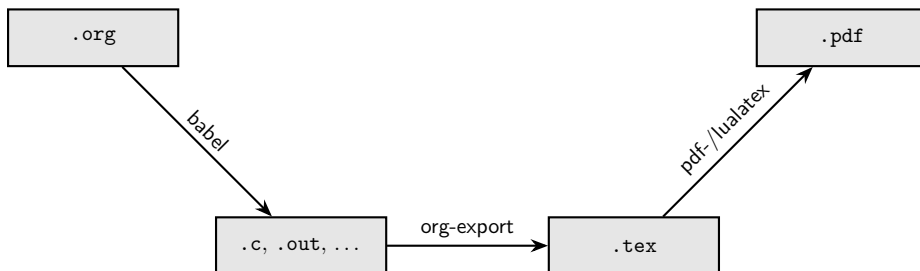
## 4. Čo sa deje na pozadí?

Ako sme spomínali, dokument píšeme v značkovacom jazyku *org*, do ktorého ale môžeme ľubovoľne zadávať i príkazy a makrá *T<sub>E</sub>X-u*, *L<sup>A</sup>T<sub>E</sub>X-u* alebo hoci aj

<sup>3</sup>Emacs umožňuje aj preklad len preambuly a zvýrazneného textu.

LuaT<sub>E</sub>X-u. Nemusíme vôbec využívať výhody a syntax značkovacieho jazyka *org*, čím sa ale pripravíme o šikovné a rýchle definovanie formátovania, interaktívnu tvorbu tabuliek (včítane vkladania vzorcov alebo (e)lispových funkcií), zoznamov či vkladania a zobrazenia obrázkov.

Pri preklade *org* dokumentu do formátu *.tex* a následne do *.pdf* dochádza k vykonaniu týchto krokov:



## Vyhodnotenie kódových blokov

Ak je kód vyhodnocovaný interpretrom (napr. python, bash, ...), tak jeho výsledky sú uložené v adresári */tmp/* s názvom *<interpreter>-<identifikátor>*. Ak sa jedná o kompilovaný kód, tak pribudnú navyše dva súbory so zdrojovým kódom a jeho preložená binárna podoba:

```

<jazyk>-src-<identifikátor>
<jazyk>-bin-<identifikátor>

```

Prípadné chyby pri prevádzaní alebo preklade sú vypísané v buffri *\*Org-Babel Error Output\**, ktorý je výstupom interpretera alebo prekladača.

```

U:**- demo.org 59% L129
/tmp/babel-Usyyg0/C-src-DNX6KG.c:13:1: error: unknown type name 'float'; did you mean 'float'?
 13 | floa povolenyRozdiel = 3.0; // Maximálny rozdiel od aktuálnej teploty
    | ^~~~
    | float
/tmp/babel-Usyyg0/C-src-DNX6KG.c: In function 'vypocitajMiesanieVzduchu':
/tmp/babel-Usyyg0/C-src-DNX6KG.c:21:3: error: invalid operands to binary * (have 'int' and 'doubl
 21 | *percentoVonk = (pozadovanaMiesanaTeplota - teplotaVnut) /
    | ^
[ Babel evaluation exited with code 1 ]
/bin/bash: riadok 1: /tmp/babel-Usyyg0/C-bin-nyhYVN: Prístup odmietnutý
[ Babel evaluation exited with code 126 ]

U:**- *Org-Babel Error Output* All L1 (Compilation) 24.03. 08:15
Code block evaluation complete.

```

**Obr. 2.** Zobrazenie chýb v kóde.

Ak je vyhodnotenie úspešné, pokračuje sa druhým krokom.

## Spracovanie hlavičky *org* súboru

V *org* dokumente je hlavička uzavretá (tzv. „drawer“, môže byť vizuálne potlačená) medzi kľúčovými slovami `:HEADER:` a `:END:`. Obsahuje príkazy preambuly pre L<sup>A</sup>T<sub>E</sub>X a tiež makrá, parametre exportu či poznámky. Príkazy pre preambulu sa vyexportujú do *.tex* súbory bez zmeny, dochádza k aplikácii makier (tie si môžeme predstaviť ako formu `\newcommand`) a doplneniu preambuly o parametre pre `documentclass`.

Následne sa sa nahradia formátovacie značky L<sup>A</sup>T<sub>E</sub>X-ovými príkazmi (napríklad /šikmé/ sa nahradí `\emph{šikmé}`), ASCII tabuľky spolu s ich parametrami vytvoria štruktúru `table`<sup>4</sup>, analogicky i obrázky či poznámky pod čiarou. V tomto momente dochádza i k vloženiu vyhodnotených programových blokov.

## Spustenie kompilátora

Príkaz `#+LATEX_COMPILER:` definuje typ kompilátora, napríklad `lualatex`, `xetex` či `pdflatex`. Pri spustení kompilácie *org* dokumentu príkazom

`org-latex-export-to-latex` alebo príslušnou klávesovou skratkou dochádza k vygenerovaniu *.pdf* súboru. V prípade potreby je možné vynechanie tohto kroku a vytvorený *.tex* súbor preložiť ručne. Autor príspevku využíva vlastnú funkciu v (e)lispe ([https://gitlab.com/my\\_elisp/emacs-org-pdf-preview/-/blob/main/org-latex-display-onside.el](https://gitlab.com/my_elisp/emacs-org-pdf-preview/-/blob/main/org-latex-display-onside.el)), ktorá zobrazuje v rozdelenom okne zdrojový kód (*org*) a vysádzaný *.pdf* súbor. Má i užitočnú vlastnosť – v prípade označenej oblasti, alebo zúženého pohľadu (funkcie Emacsu `narrow-to*`, ktoré umožňujú vizuálne potlačiť časť dokumentu) spracováva len túto oblasť a neprekladá celý dokument (vždy sa jedná o preklad preambuly a viditeľného obsahu). Takže pri práci s veľkým súborom sa autor môže sústrediť len na práve vytváranú časť, s výhodou rýchleho prekladu či náhľadu v rozdelenom okne. Táto funkcionality dozaista nie je nová, je prítomná v známych dedikovaných vývojových prostrediach (napr. T<sub>E</sub>Xstudio či Overleaf), ktoré boli inšpiráciou pre jej vytvorenie.

## 5. Záver

Opísaním ďalších možností a vlastností *babel* by sme niekoľkonásobne presiahli rozsah tohto článku a hádam aj únosnú mieru množstva prijatých informácií. Môžeme povedať, že je to most medzi programovaním a dokumentovaním, nástroj, ktorý zásadne zjednodušuje interakciu medzi textom a kódom. Jeho schopnosť integrovať rôzne programovacie jazyky priamo do dokumentácie umožňuje

---

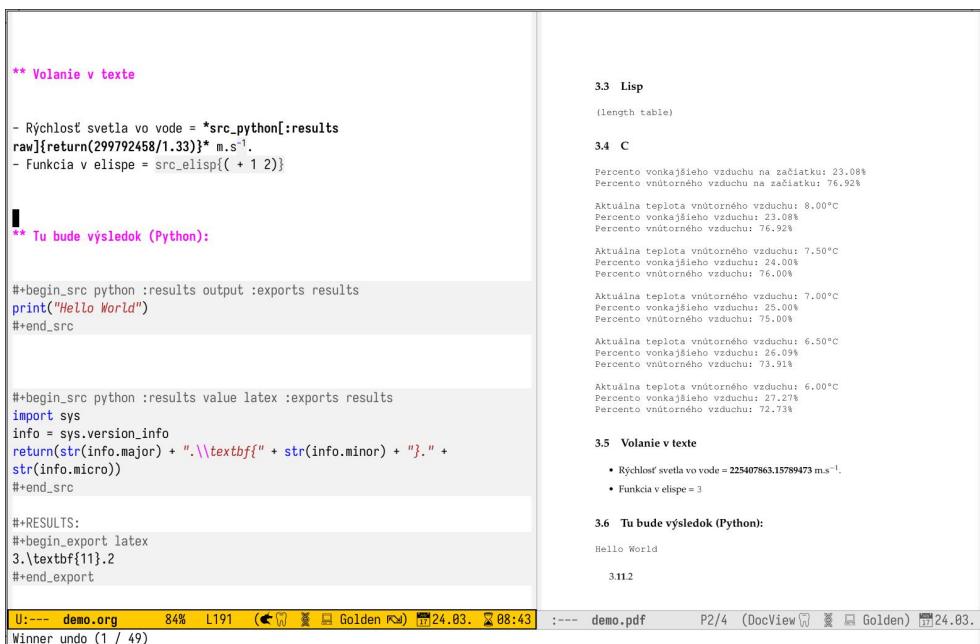
<sup>4</sup>Je potrebné úprimne poznamenať, že *org* je „prikrátky“ na tabuľky využívajúce `multirow`, `multicolumn`,... a pokročilé formátovanie buniek (napr. farby). Avšak vždy môžeme vytvárať tieto tabuľky pomocou príkazov L<sup>A</sup>T<sub>E</sub>X-u!

```
Percento vonkajšieho vzduchu na začiatku: 23.08%
Percento vnútorného vzduchu na začiatku: 76.92%
```

```
Aktuálna teplota vnútorného vzduchu: 8.00°C
Percento vonkajšieho vzduchu: 23.08%
Percento vnútorného vzduchu: 76.92%
```

```
Aktuálna teplota vnútorného vzduchu: 7.50°C
Percento vonkajšieho vzduchu: 24.00%
Percento vnútorného vzduchu: 76.00%
```

Obr. 3. Vysádzaný výstup z programu v jazyku C (krátené).



Obr. 4. Práca na zdrojovom kóde so zobrazením vysádzaného náhľadu.

efektívne vykonávať analýzy, generovať grafy a viesť technickú dokumentáciu bez potreby používania iných programov.

## Literatúra

- [1] KOZUBÍK, A.: *Python ako priateľ L<sup>A</sup>T<sub>E</sub>X-u*, Otvorený softvér vo vzdelávaní, výskume a IT riešeníach, zborník medzinárodnej konferencie OSSConf 2025, Žilina,

- 1.–3. júla 2025, ISBN 978-80-554-2209-1, <https://frcatel.fri.uniza.sk/ossconf/zborniky-online/ossconf2025/files/09-kozubik.pdf>.
- [2] FABO, R.: *GNU EMACS + ORG MÓD = EASY L<sup>A</sup>T<sub>E</sub>X*, Otvorený softvér vo vzdelávaní, výskume a IT riešeníach, zborník medzinárodnej konferencie OSSConf 2024, Žilina, 2.–4. júla 2024, ISBN 978-80-554-2105-6, <https://frcatel.fri.uniza.sk/ossconf/zborniky-online/ossconf2024/files/04-fabo.pdf>.

## Kontaktná adresa

**Ing. Richard Fabo**, Triton Famme s.r.o., Levočská 862/28, 058 01 Poprad, Slovensko,  
*E-mailová adresa:* [triton@famme.sk](mailto:triton@famme.sk), <https://triton.famme.sk>.

## Typst – VÝZVA PRE $\text{\LaTeX}$ ALEBO LEN HYPE?

RICHARD FABO (SK)

**Abstrakt.** Typst je svojimi autormi a fanúšikmi označovaný za novú generáciu nástroja na typografiu a spracovanie dokumentov. Nových používateľov láka na intuitívnejšie užívateľské rozhranie, monolitické riešenie základného formátovania a rýchlejší čas kompilácie. V článku si v krátkosti ukážeme základné princípy používania. Porovnáme ho s etablovaným systémom  $\text{\LaTeX}$  z viacerých hľadísk. Položíme si otázky, či tieto výhody kompenzujú robustnosť a veľkosť  $\text{\LaTeX}$ -u a pokúsime sa nájsť jeho „koncového užívateľa“.

**Kľúčové slová.** Typst,  $\text{\LaTeX}$ , značkovací jazyk.

## Typst – A CHALLENGE FOR $\text{\LaTeX}$ OR JUST HYPE?

**Abstract.** Typst is described by its authors and fans as a new-generation tool for typography and document processing. It attracts new users with its more intuitive user interface, a monolithic solution for basic formatting, and faster compilation times. In this article, we will briefly demonstrate the basic principles of its use. We will compare it with the established  $\text{\LaTeX}$  system from several perspectives. We will ask whether these advantages compensate for the robustness and size of  $\text{\LaTeX}$  and attempt to identify its end-user.

**Keywords.** Typst,  $\text{\LaTeX}$ , markup language.

## Úvod

Jednou z fascinujúcich vlastností slobodného softvéru je jeho rôznorodosť, modularita, adaptabilita a alternatívne riešenia, viac alebo menej inovatívne. V roku 2023 bola uvoľnená prvá verzia programu Typst. Prvé ohlasy boli pozitívne, ukazuje sa, že je to jediný vyzývateľ  $\text{\LaTeX}$ -u za posledné dekády. Ale dokáže ho skutočne nahradiť? Typst čerpá svoje princípy a syntax z Markdown-u,  $\text{\LaTeX}$ -u a iných programovacích jazykov. Je oveľa jednoduchší ako  $\text{\LaTeX}$ , ľahšie sa v ňom odstraňujú chyby a dokumenty sa v ňom pripravujú rýchlejšie. Postačuje to? Na to sa pokúsime úprimne odpovedať.

## 1. Ešte jeden úvod – o Typst-e

V roku 2019 sa dvaja nemeckí vývojári, LAURENZ MÄDJE a MARTIN HAUG, rozhodli napísať náhradu za  $\text{\LaTeX}$ . V marci 2023 bola oznámená prvá predbežná beta verzia; o mesiac neskôr bolo s vydaním verzie 0.1.0 zavedené sémantické číslovanie verzií. Povráva sa, že zložené zátvorky a spätné lomky sa na nemeckých

klávesniciach píšú komplikovane, a to bol dodatočný podnet na vytvorenie alternatívneho značkovacieho systému, ktorý nevyžaduje množstvo týchto symbolov.

Zacitujme z oficiálnej dokumentácie: „Typst je nový systém na sadzbu textov pre vedecké články, založený na značkovacom jazyku. Je navrhnutý ako alternatíva k pokročilým nástrojom, ako je  $\text{\LaTeX}$ , aj k jednoduchším nástrojom, ako sú Word a Google Docs. Naším cieľom pri vývoji Typst je vytvoriť nástroj na sadzbu textov, ktorý je vysoko výkonný a zároveň sa s ním príjemne pracuje.“ [6]

Väčšina recenzií Typst-u je až príliš rovnaká – opíše sa spôsob zadávania značiek, do nesmiernych výšin sa vyzdvihne rýchlosť kompilácie, spomenie sa online nástroj na tvorbu textu. Ale takto by zodpovedná recenzia vyzerat nemala. Pokúsime sa to zmeniť.

Samozrejme, základnému opisu sa vyhnúť nemožno. Stačí na to málo priestoru a viac čitateľovej predstavivosti. V súvisiacej prednáške hádam ukážeme trochu viac. Porovnania s  $\text{\LaTeX}$ -om budú opisované priebežne.

## 2. Inštalácia, veľkosť, balíčky

V čase písania článku neexistuje inštalačný balíček pre GNU/linuxové distribúcie. Vývojári na stránke GitHub-u <https://github.com/typst/typst/releases/> zverejňujú, okrem zdrojových kódov i staticky linkovaný binárny spustiteľný súbor pre rôzne architektúry. Užívateľ musí archív `.tar.xz` stiahnuť, rozbalíť, ručne vytvoriť symbolický odkaz do adresára so spustiteľnými súborami (typicky `/usr/bin`, prípadne `/usr/local/bin`, a ak nemá práva superužívateľa tak aspoň do `~/local/bin`). Nie je to náročné, ale autori mohli spraviť aspoň malý skript, ktorý by stiahol a nainštaloval najnovšiu verziu.

Samotný spustiteľný súbor `typst`, vo verzii 0.14.2, má necelých 50 MB. Užívatelia  $\text{\LaTeX}$ -u, najmä ak inštalujú doporučené balíky, si musia vyhradiť na disku rádovo gigabajty. Na prvý pohľad je jasné, že takto to porovnávať nemôžeme. Pomôže nám „úrok bokom“.

Ak niekto potreboval pracovať s  $\text{\LaTeX}$ -om na počítači a súčasne nemal práva superužívateľa, a súčasne sa nekamarátil so správcom počítača, a súčasne mal dostatok vyhradeného priestoru vo svojom konte, ..., dobre, znie ako príklad z pradávnej minulosti, ..., ale ak by k tomu došlo, mohol by využiť projekt `TINYTEX` (<https://github.com/rstudio/tinytex-releases>), čo je ľahká, multiplatformová a prenosná distribúcia  $\text{\LaTeX}$ -u založená na `TeX Live`. Zjednodušene by sme mohli povedať, že autor zoskupil dôležité balíky a súbory a nastavil príslušné (relatívne) cesty tak, aby mohol známy celý strom `TeX Live` existovať v ľubovoľnom adresári na disku. V kontexte tohto článku je projekt zaujímavý z pohľadu vytvorených (a doporučených) skupín balíkov pre ten-ktorý prípad použitia. Ak uvažujeme použiteľnú verziu s názvom `TINYTEX-1`, ktorá obsahuje cca 100 balíkov (námatkovo `bibtex`, `ams*`, `luatex`, `xcolor`, základné fonty, `babel`,<sup>1</sup> ...), tak

<sup>1</sup>Nie české či slovenské rozšírenie, ktoré si musíme doinštalovať (rozumej nakopírovať) ručne.



táto má okolo 52 MB. Verzia TINYT<sub>E</sub>X, obsahujúca už i TikZ, wrapfig, . . . , má cca 150 MB.

Ak to skrátime: bežné sťažnosti, že L<sup>A</sup>T<sub>E</sub>X je voči Typst-u „moloch“ nie sú namieste. Typst totiž v samotnej základnej inštalácii obsahuje príkazy pre značkovanie, matematický režim, skriptovanie, stĺpce, plávajúce objekty, bibliografiu, vkladanie a spracovanie obrázkov a výstupný nástroj pre tvorbu PDF. a tu sme na jeho veľkosti porovnateľnej s L<sup>A</sup>T<sub>E</sub>X-om, ktorý si so sebou navyše nesie „nános minulosti“.<sup>2</sup> Ostatné potrebné rozšírenia sa ukladajú (predvolene) do `~/.cache/typst/packages/` a `~/.local/share/typst/packages`.

Jednou z výhod Typst-u je spôsob sťahovania balíčkov. Nie je potrebné nič v predstihu inštalovať. Ak kompilátor narazí na príkaz:

```
#import "@preview/<názov:verzia>"
```

napr. `#import "@preview/cetz:0.4.2"` pre balík `cetz` (obdoba TikZ), tak príslušný balík stiahne do uvedených adresárov.

Týmto sa neplánovane dostávame k porovnaniu počtu rozšírení. Samozrejme, Typst sa zatiaľ nemôže s L<sup>A</sup>T<sub>E</sub>X-om v tomto porovnávať. Desiatročia budovania tzv. „ekosystému“ nie je možné nahradiť. Porovnajme oficiálne zdroje pre L<sup>A</sup>T<sub>E</sub>X (<https://ctan.org/>) a Typst (<https://typst.app/universe/search/?kind=packages>).

**Tabuľka 1.** Porovnanie počtu balíčkov k aprílu 2026

|                 | L <sup>A</sup> T <sub>E</sub> X | Typst |
|-----------------|---------------------------------|-------|
| Počet balíčkov: | 6904                            | 634   |

Zodpovedne uvádzame, že niektoré rozšírenia L<sup>A</sup>T<sub>E</sub>X-u (napríklad `geometry`, `grid`, `table`, `graphicsx`, `bibtex`) sú v základnej binárke Typst-u. To nás vedie k úvahe, či na prvý pohľad lákavý spôsob distribúcie – všetko v jednom súbore – je skutočnou výhodou. Pravdepodobne nie. Osvedčená „KISS“ filozofia<sup>3</sup> uprednostňuje modularitu, menší rozsah zmien, lepšiu správu závislostí, paralelný vývoj, vyššiu stabilitu a v neposlednom rade prispôbitelnosť a ľahšie pochopenie.

## 2.1. Porovnanie dostupnosti najbežnejších rozšírení

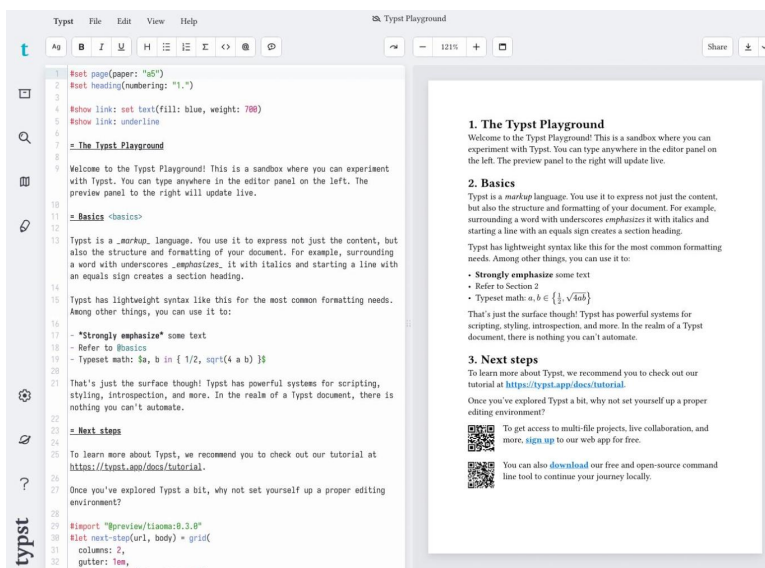
Ako sme spomínali, Typst je dodávaný s „kompletným vybavením“, takže ekvivalenty mnohých obľúbených balíčkov L<sup>A</sup>T<sub>E</sub>X-u sú v ňom priamo integrované. Uvádzame tabuľku 2 s často používanými balíkmi a zodpovedajúcimi funkciami v Typst-e. Zabudované funkcie sú uvedené v hranatých zátvorkách.

<sup>2</sup>Staré konvencie a syntaktické zvyky (T<sub>E</sub>X-ové primitívy, knižnice spred 20–30 rokov), ktoré sa dnes považujú za neefektívne alebo neintuitívne, zastaralé balíčky/triedy, ktoré sa musia zachovať pre spätnú kompatibilitu, rôznorodé a nekompatibilné spôsoby riešenia tej istej úlohy, ap.

<sup>3</sup>Keep It Simple, Stupid je návrhová filozofia, ktorá zdôrazňuje jednoduché riešenia pred zbytočnou zložitou.

Tabuľka 2. Ekvivalentné balíky či funkcie

| Balíček L <sup>A</sup> T <sub>E</sub> X           | Alternatíva v Typst               |
|---------------------------------------------------|-----------------------------------|
| graphicx, svg                                     | [image]                           |
| tabularx                                          | [table], [grid]                   |
| fontenc, inputenc, unicode-mat babel, polyglossia | [#set text(lang: "sk")]           |
| amsmath                                           | priame zadávanie                  |
| geometry, fancyhdr                                | [page]                            |
| xcolor                                            | [#set text(fill: rgb("#0178A4"))] |
| hyperref                                          | [link]                            |
| bibtex, biblatex, natbib                          | [cite], [bibliography]            |
| lstlisting, minted                                | [raw]                             |
| parskip                                           | [block], [par]                    |
| caption                                           | [figure]                          |
| enumitem                                          | [list], [enum]                    |
| tikz                                              | cetz                              |



Obr. 1. Online vývojové prostredie typst.app

### 3. Používanie

Po inštalácii sa konečne vrhnime do jeho používania. Zatiaľ Typst nemá svoje vlastné dedikované vývojové prostredie. Na rýchle vyskúšanie je výhodné využiť online nástroj Typst.app (<https://typst.app/>), ktoré ani nevyžaduje registráciu. Základný demonštračný príklad vidíme na obr 1. Lákadlom je dozaista možnosť spolupráce viacerých užívateľov na jednom dokumente, avšak treba si uvedomiť, že na strane servera beží uzavretý proprietárny kód, podobne ako v prípade L<sup>A</sup>T<sub>E</sub>X-ového Overleaf.

Lepšie a čistejšie riešenie je písanie dokumentu na lokálnom počítači. Už dnes existujú rozšírenia do známych vývojových prostredí ako GNU Emacs, Neovim alebo VS Code. Ale samozrejme postačuje ľubovoľný textový editor a na zrýchlenie prekladu súbor Makefile, ktorý v najjednoduchšej podobe môže vyzeráť napríklad takto:

```
TEXT ?= $(wildcard ./*.typ)
BASENAME ?= $(basename $(TEXT))
all:
    typst compile $(TEXT)
    mupdf $(BASENAME).pdf
```

Pričom väčšina editorov dokáže príkaz make spúšťať klávesovou skratkou.

### 3.1. Značky

Ako sme naznačili, nebolo by účelné spraviť z tohto článku manuál k programu. Preto si najzákladnejšie formátovacie značky a ich alternatívu v iných jazykoch len ukážeme v tabuľke 3, vychádzajúc z [7], s rozšírením o org-mode.

**Tabuľka 3.** Základné formátovacie značky

|                  | Typst                                   | L <sup>A</sup> T <sub>E</sub> X                                                                              | org-mode                                            |
|------------------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| Tučný text       | <b>*tučný*</b>                          | <code>\textbf{tučný}</code><br>alebo <code>{\bf tučný}</code>                                                | <b>*tučný*</b>                                      |
| Kurzíva          | <code>\_kurzíva\_</code>                | <code>\textit{kurzíva}</code>                                                                                | <code>/kurzíva/</code>                              |
| Nadpis 1. úrovne | <code>= nadpis</code>                   | <code>\section{nadpis}</code>                                                                                | <code>* nadpis</code>                               |
| Nadpis 2. úrovne | <code>== podnadpis</code>               | <code>\subsection{podnadpis}</code>                                                                          | <code>** podnadpis</code>                           |
| Zoznam           | <code>- položka</code>                  | <code>\begin{itemize}</code><br><code>\item položka</code><br><code>\end{itemize}</code>                     | <code>- položka</code>                              |
| Matematika text  | <code>\$x^2\$</code>                    | <code>\$x^2\$</code>                                                                                         | <code>\$x^2\$</code>                                |
| Matematika blok  | <code>\$ x^2 \$</code>                  | <code>\[x^2\]</code> alebo <code>\$\$x^2\$\$</code>                                                          | <code>\$\$x^2\$\$</code>                            |
| Obrázok          | <code>\#figure(image("..."))</code>     | <code>\begin{figure}</code><br><code>... \end{figure}</code>                                                 | ako L <sup>A</sup> T <sub>E</sub> X alebo hyperlink |
| Tabuľka          | <code>\#table(columns: 2, [...])</code> | <code>\begin{tabular}{ c c }</code><br><code>... \end{tabular}</code>                                        | ako L <sup>A</sup> T <sub>E</sub> X alebo ASCII     |
| Citácia          | <code>@kľúč</code>                      | <code>\cite{kľúč}</code>                                                                                     | <code>\cite{kľúč}</code>                            |
| Pozn. pod čiarou | <code>\#footnote[Poznámka]</code>       | <code>\footnote{Poznámka}</code>                                                                             | <code>[FN: Poznámka]</code>                         |
| Formát strany    | <code>\#set page(size: "A4")</code>     | <code>\usepackage[a4paper]</code>                                                                            | <code>\#+LATEX_CLASS_OPTIONS: ...</code>            |
| Písma            | <code>\#set main(font: "Times")</code>  | <code>\usepackage[T1]{fontenc}</code> ako L <sup>A</sup> T <sub>E</sub> X<br><code>\usepackage{times}</code> |                                                     |

Pri zadávaní značiek sa ešte zastavíme. Neoprávnene sa považujú za jedno z hlavných lákadiel. V diskusiách či recenziách môžeme nájsť názory, že práve „toto“ je významným krokom vpred. Myslíme si, že práve „toto“ je najmenej dôležité. Áno, text je pre netrénované oko čitateľnejší. Avšak v dnešnej dobe pokročilých editorov s automatickým dopĺňaním alebo „snippetmi“<sup>4</sup>, ktoré vložia všetky

<sup>4</sup>Snippet je krátky úryvok kódu alebo textu, ktorý slúži na rýchle vkladanie často používaných konštrukcií (funkcií, šablón, hlavičiek, príkazov) do editora.

požadované časti prostredia, pričom sa interaktívne dopytujú na jednotlivé parametre, už toto nie je zásadným zlepšením práce. Niektorí recenzenti [8] uvádzajú jednoduchšie zadávanie matematických výrazov, pričom ako jednu zo zásadných výhod spomínajú „čitateľnejšie“ zadávanie zlomkov. Len pripomenieme, že namiesto `\frac{}{}` je možné zadať i čitateľnejšiu  $\text{\TeX}$  verziu `\over`. Zátvorky v Typst-e sa zväčšujú podľa potreby, niektoré znaky sa zadávajú jednoduchšie (napr. `oo` vs. `\infty`, `->` vs. `\rightarrow`).

- $\text{\LaTeX}$ : `$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$`,
- Typst: `$x = (-b plus.minus (sqrt(b^2 - 4 a~c))) / (2a)$`.

### 3.2. Funkcie

Tu treba úprimne povedať, že písanie funkcií je v porovnaní s  $\text{\LaTeX}$ -om príjemnejšie a jednoduchšie, hoci práve  $\text{Lua}\text{\LaTeX}$  tento rozdiel zmenšuje. Napriek tomu, že je zabudovaný systém Lua veľmi výkonný, prejavuje sa na ňom jeho postavenie „pridanej funkcie“, čo od používateľov vyžaduje, aby sa vyrovnali s rozhraním medzi dátovými štruktúrami Lua a vnútornými mechanizmami  $\text{\LaTeX}$ -u. V Typst-e je programovanie integrované do systému, bez akýchkoľvek prechodov medzi jazykom používaným na výpočty a konštrukciami sadzby. Na využitie väčšiny príkazov (podmienky, cykly) nie sú potrebné dodatočné balíčky (ako napríklad `ifthen`, `pgffor`, ...) a pritom ich syntax je jednoduchšia, bez spätných lomiek uvádzajúcich príkazy či riadiace sekvencie. Pokročilé programovacie balíky ako napríklad `expl3` (pre  $\text{\LaTeX}$ 3) zvyšujú jeho roztrieštenosť, pridávajú ďalšiu vrstvu zložitosti ( $\text{\TeX}$  primitívy +  $\text{\LaTeX}$  makrá + balíky + `expl3`), pričom používateľ musí rozumieť všetkým týmto úrovňam. Ukazuje sa, že  $\text{Lua}\text{\LaTeX}$  je lepšie riešenie ako `expl3`, avšak ich vzájomné porovnanie a prístup by dostatočne naplnili obsah osobitného článku.

Z dnešného pohľadu nezvyčajnú syntax vidíme v prípade takýchto konštrukcií v plain $\text{\TeX}$ -u (výpis Fibonacciho postupnosti pre 20 prvkov).

| Typst:                                                                                                                                               | Plain $\text{\TeX}$ :                                                                                                                                                                                                                                                             | Lua $\text{\LaTeX}$ :                                                                                                                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>#let fibonacci(n) = {   let a = 0   let b = 1   for i in range(n) {     a += b     (a - b,)     b = a - b   } }</pre> <pre>#fibonacci(20)</pre> | <pre>\newcount\fibA\newcount\fibB \newcount\fibC\newcount\fibN \def\fibonacci{   \fibA=0 \fibB=1   \fibN=0   \loop     \the\fibA     \ifnum\fibN&lt;19       , \advance\fibN by1       \fibC=\fibA       \advance\fibA by\fibB       \fibB=\fibC     \repeat   } \fibonacci</pre> | <pre>\begin{luacode} function fibonacci(n)   local a = 0   local b = 1   for i = 1,n do     tex.print(tostring(a)       .. ", ")     a = a + b     b = a - b   end end \end{luacode}</pre> <pre>\directlua{fibonacci(20)}</pre> |

Definovanie vlastného nadpisu:

Typst:

```
#let heading(content) = {
  align(center,
    text(size: 1.5em, content))
}
#heading("Nadpis článku")
```

L<sup>A</sup>T<sub>E</sub>X:

```
\newcommand{\heading}[1]
{\begin{center}
  \Large #1
\end{center}}
\heading{Nadpis článku}
```

Ale aby sme neboli len negatívni: ako v prípade každej zručnosti, s praxou sa syntax stáva prirodzenou a reflexívnou. a to platí aj v prípade menej intuitívnej syntaxe či záhadných príkazov. L<sup>A</sup>T<sub>E</sub>X má vytvorené obrovské množstvo funkcií, riešení, hotových balíčkov, šablón a makier. Je teda otázkou, či to predčia nové a „naleštené“ funkcie Typst-u.

### 3.3. Kompilácia

Bez výnimky udávanou výhodou v prospech Typst-u je rýchlosť prekladu a vytvorenia PDF súboru. Je to skutočne tak, viacerí recenzenti merali túto rýchlosť pomocou (obsahovo) rovnakých zdrojových súborov. Azda najkomplexnejší výstup na, ako tvrdia autori [9], v „modernom x86 systéme“, sú uvedené v tabuľke 4.

**Tabuľka 4.** Porovnanie rýchlosti kompilácie Typst – L<sup>A</sup>T<sub>E</sub>X

| Typ dokumentu                    | L <sup>A</sup> T <sub>E</sub> X | Typst        |
|----------------------------------|---------------------------------|--------------|
| Jednoduchý článok (5 strán)      | 2 – 4 s                         | 50 – 100 ms  |
| Článok ku konferencii (10 strán) | 5 – 10 s                        | 100 – 200 ms |
| Diplomová práca (100 strán)      | 30 – 60 s                       | 0,5 – 1 s    |
| Kniha (>300 strán)               | 2 – 5 min.                      | 1 – 3 s      |
| Postupná zmena (1 znak)          | Úplné prekompilovanie           | 10 – 50 ms   |

Sú to veľké rozdiely a v prípade Typst-u je rýchlosť ohromujúca. Prečo je vlastne tak?

Viacero článkov včítane uvedeného [9] uvádzajú najmä tieto príčiny:

**Prírastková kompilácia:** Typst prekompiluje len zmenené časti, napríklad ak sa zmení slovo, preloží sa znovu len daný odsek. Oproti tomu L<sup>A</sup>T<sub>E</sub>X prekladá celý dokument; i keď tu sa dá použiť režim „draftu“, pri ktorom sa ignorujú obrázky. Pri používaní **org**- alebo **AUC<sub>T</sub>TeX**-módu pre GNU Emacs možno prekladať len viditeľnú (tzv. zúženú časť) [1], čo napomáha zníženiu celkového času prekladu. Je taktiež dobrou praxou dokument deliť na logické celky a len tieto prekladať pri potrebe náhľadu.

**Moderná architektúra:** Typst je napísaný v jazyku Rust s využitím moderných kompilačných techník. T<sub>E</sub>X-ový engine programu L<sup>A</sup>T<sub>E</sub>X pochádza z roku 1978 a dokumenty spracováva postupne. Toto je, podľa nášho názoru odvážne tvrdenie, ktoré si vyžaduje viac pozornosti, ktorej sa mu dostane v kapitole 4.

**Spracovanie v jednom kroku:** Typst spracováva odkazy v jednom kroku.

$\text{\LaTeX}$  často vyžaduje dva až tri prechody na spracovanie odkazov, citácií a obsahu.

## 4. Architektúra

Typst je monolitický systém. Je to jeden binárny súbor zahrňujúci funkcie na analýzu textu, rozvrhnutie prvkov a stránok, vykresľovanie fontov, spracovanie bibliografie, tvorbu obsahov či indexov atď. Naproti tomu  $\text{\LaTeX}$  prekladá zdrojový text využitím viacerých nástrojov (typicky  $\text{pdf\LaTeX}$ ,  $\text{Lua\LaTeX}$ ,  $\text{ConTeXt}$ ,  $\text{OpTeX}$ , ...), pre expandovanie makier a generovanie pomocných súborov,  $\text{BibTeX}$  na spracovanie citačnej databázy, opätovné použitie  $\text{pdf\LaTeX}$  (...) pre finálne rozvrhnutie textu a vytvorenie PDF [prípadne  $\text{dvi}$  v prípade  $\text{TeX-u}$ ].

Áno, systém makier (napr. najrozšírenejší  $\text{\LaTeX}$ ) nad  $\text{TeX-ovým}$  jadrom pri-náša vyššiu neprehľadnosť. Ich prvotný zámer – zjednodušiť používanie  $\text{TeX-u}$  sa podaril len čiastočne, pričom znásobil redundanciu príkazov. Takto existuje systém, v ktorom makrá volajú iné makrá a tie sa spoliehajú na iné, a táto ná-sobná expanzia zvyšuje čas prekladu. Vela makier nad  $\text{TeX-om}$  vytvára kultúru „riešení pre všetko“, so všetkými výhodami i nevýhodami. Výhody sú zrejmé, nevýhody sú najmä, okrem zvyšovania zložitosti, náročný debugging, potreba veľkých balíkov makier aj pri jednoduchom dokumente, možné konflikty, kompli-kovanejšia údržba, nejednoznačná syntax. Z tohto hľadiska má k Typst-u zrejme najbližšie  $\text{OpTeX}$ .

Súčasná jadrá  $\text{pdfTeX}$ ,  $\text{XeTeX}$ , atď., sú napísané v jazyku C, ktorého rých-lost je takmer totožná s jazykom Rust. Avšak Typst používa paralelné spracova-nie [10], vďaka modernému Rust-ovému backend-u, ktorý optimalizuje výkon na viacjadrových systémoch – na rozdiel od  $\text{*TeX-u}$ , kde je paralelizmus dosiahnu-teľný len pomocou externých nástrojov.

Ak v Typst-e použijeme veľké knižnice, kompilácia sa spomaľuje, ale vďaka optimalizácii „importov“ si užívateľ môže z knižnice vybrať len požadovanú časť (napríklad `#import "@preview/tablex:0.0.5": tablex`, analogicky ako v ja-zyku Python (`from math import sqrt, pi`).

## 5. Model rozloženia

Podľa autora programu, LAURENZA MÄDJEHO, hoci Typst preberá niektoré časti modelu  $\text{TeX-u}$ , najmä algoritmus rozloženia odsekov, existujú medzi nimi aj významné rozdiely. Väčšina z nich sa týka rozloženia na úrovni blokov – na-príklad umiestňovania riadkov, zabraňovania vdovám a sirotám, umiestnenia ta-buliek a plávajúcich objektov [5]. Model rozloženia v  $\text{TeX-u}$  je postavený **boxoch** a **výplnkoch**. Boxom môže byť skupina znakov, riadok v odseku, položka v ta-bulke, celá tabuľka, záhlavie alebo celá strana. Box, rovnako ako zložka znaku

a riadku, je v T<sub>E</sub>X-u interpretovaný ako pravouhlý dvojrozmerný útvar, ktorý sa umiestňuje do zložky s ohľadom na tzv. základnú čiaru (baseline) [3]. Zjednodušene **výplnok** je rozťahovateľný priestor medzi boxmi. Má prirodzenú veľkosť, ale je zmenšiteľný a rozťahnutelný, čo mu umožňuje zaberat menej alebo viac priestoru v závislosti od potrieb rozloženia. Pomocou týchto dvoch primitív T<sub>E</sub>X vytvára všetko, od písmen a slov, až po riadky, odseky a stránky. Na tento účel vytvára horizontálne a vertikálne zoznamy. Horizontálny zoznam obsahuje vložený obsah a je spracovaný nástrojom na zalomenie riadkov, ktorý vytvorí rámčeky pre každý riadok. Tieto riadky sa dostanú do vertikálneho zoznamu, ktorý je následne spracovaný nástrojom na zalomenie stránok, ktorý vytvorí stránky [5].

Preklad Typst dokumentu prebieha v piatich krokoch: analýza (premena reťazcov zdrojového textu na syntaktický strom), vyhodnotenie (vložených blokov kódu a volaní funkcií), transformácia obsahu na tzv. „toky“ a „uzly“ (ktoré je možné priamo rozložiť, ako sú odseky, nadpisy, zoznamy), rozloženie – transformácia uzlov na sekvenciu **rámčov** a nakoniec exportovanie rámčov na výstupné formáty (PDF, PNG) [2].

**Región** opisuje tvar, do ktorého je možné rozložiť prvky (odsek, nadpis, atď.) Funkcia rozloženia dostáva postupnosť oblastí, do ktorých má rozložiť obsah. Výsledkom je množstvo **rámčov**, ktoré sú podobné **boxom** v systéme T<sub>E</sub>X.

Typst v súčasnosti obmedzuje všeobecný model regiónov dvoma spôsobmi:

- Všetky regióny v sekvencii musia mať v súčasnosti rovnakú šírku.
- Regióny môžu byť v súčasnosti len obdĺžnikové. Neumožňujú „výrezy“.

Tieto dve obmedzenia spolu umožňujú Typst-u rozdeľovať odseky na riadky nezávisle od toho, kde sa na stránkach nachádzajú (rovnako ako T<sub>E</sub>X). Napriek tomu sa prvky na úrovni bloku, ako sú tabuľky, môžu prispôbovať miestu, kde sú na stránke umiestnené. Vďaka tomu má Typst oveľa menšiu flexibilitu pri presúvaní prvkov, ale väčšiu flexibilitu pri prispôbovaní rozloženia na základe vlastnej pozície. T<sub>E</sub>X uprednostňuje pohyblivosť, aby mohol optimalizovať pozície a medzery. Typst uprednostňuje umiestnenie, aby prvky mohli reagovať na svoje pozície, čo umožňuje napríklad správne rozdelenie buniek tabuľky na viaceré strany.

Na vytvorenie odseku musí T<sub>E</sub>X poznať šírku jednotlivých riadkov. Zvyčajne sú všetky rovnaké, ale pomocou príkazu `\parshape` môže používateľ pre každý riadok zadať individuálnu šírku. Keďže sa odseky vytvárajú skôr, ako je známe, na ktorej strane sa nakoniec ocitnú, T<sub>E</sub>X musí poskytovať tieto informácie bez znalosti stránok.

Podobné obmedzenie sa prejavuje aj pri balíku `wrapfig`. Tento balík pridáva podporu pre obrázky, ktoré sú umiestnené vedľa textu a text ich obteká. Na rozdiel od plávajúcich prvkov umiestnených dole alebo hore predstavuje tento typ plávajúceho prvku zásadnú výzvu pre model T<sub>E</sub>X-u: Pri vytváraní riadkov odseku vedľa obrázka s obtekaním T<sub>E</sub>X ešte nemôže poznať vertikálne polohy

jednotlivých riadkov. Preto nemôže (s istotou) vedieť, ktoré riadky sa ocitnú vedľa obtekajúceho obrázku.

Z tohto dôvodu balík vychádza z predpokladu, že základné čiary odseku budú rovnako vzdialené. Na základe toho a známej výšky obtekajúceho obrázku vypočíta počet  $\mathcal{N}$  riadkov, ktoré sa zmestia vedľa obrázku. Potom pri vytváraní riadkov odseku pridelí správnu šírku pre týchto  $\mathcal{N}$  riadkov (prostredníctvom `\parshape`). Zvyčajne to funguje dobre. Ak sa však jeden z riadkov skončí väčší alebo na inej strane,  $\text{\TeX}$  nemôže svoju chybu opraviť a vedľa posledných riadkov zostane nevyužitý priestor.

Model  $\text{\TeX}$ -u nie je dokonalý vo všetkom, čo vyžaduje znalosť presných vertikálnych pozícií: flexibilné veľkosti strán, reťazené kontajnery, plávajúce objekty s bohatými kolíziami, tabuľky s možnosťou rozdelenia, sadzba založená na mriežke a ďalšie.

Súčasný model Typst-u trpí čiastočne rovnakými a čiastočne inými problémami. Koncepcia regiónov umožňuje vyriešiť množstvo problémov, ktoré  $\text{\TeX}$  nedokáže vyriešiť – flexibilné veľkosti stránok, reťazenie kontajnerov, kolidujúce plávajúce objekty. Zároveň regióny prinášajú nové problémy: odovzdávaním presných pozícií na podrozloženia sa stáva ťažšie optimalizovateľným rozloženie presúvaním prvkov. V dôsledku toho Typst v súčasnosti zaostáva v oblasti prevencie vdov a sirôt, vertikálneho zarovnania a ďalších funkcií [5].

## 6. Chybové hlásenia

O  $\text{\LaTeX}$ -u sa právom tvrdí, že jeho chybové hlásenia sú „kryptické“. Je to dôsledok toho, že text sa parsuje po tokenoch a chyba sa často ukáže neskôr než jej zdroj (napr. chýbajúca zatváracia zátvorka sa prejaví až pri ďalšom tokene). Balíčky a makrá pridávajú nové príkazy a zmeny správania  $\text{\TeX}$ -u (nové kontrolné sekvencie, prostredia, nastavenia kategórií tokenov). Dva balíčky môžu definovať rovnaké makro alebo meniť rovnaké interné nastavenia; pri konflikte sa správanie môže stať nepredvídateľné a chybové hlásenia budú ukazovať až následné symptómy (chyby pri rozparsovaní tokenov, nesprávne uzavretie prostredí). Chyba sa často prejaví až po spustení vnútorných algoritmov  $\text{\TeX}$ -u a prejaví sa ako „ $\text{\TeX}$  error“ [4].

Typst poskytuje jasné a konkrétne hlásenia o chybách, ktoré presne uvádzajú, čo sa pokazilo a ako to napraviť. Obr. 2 zobrazuje chybové hlásenie o chýbajúcich uzatváracích zátvorkách.

## 7. Čo sa nehodí inde...

Respektíve pár subjektívnych názorov.

**„Problémy“  $\text{\LaTeX}$ -u dobehnú i Typst.** Hlavné problémy, ktoré Typst rieši svojím prístupom založeným na funkcionálnom programovaní, sa v prostredí



```

$ typst compile demo.typ
error: unclosed delimiter
demo.typ:89:5
89 | #grid(
    |      ^
error: unclosed delimiter
demo.typ:94:17
94 | align(center)[
    |              ^

```

Obr. 2. Chybové hlásenie pri preklade Typst dokumentu

L<sup>A</sup>T<sub>E</sub>X-u, ktoré je preťažené makrami, nedajú vyriešiť (napríklad rozumné hlásenie chýb a okamžitá inkrementálna kompilácia). Ak má byť zachovaná spätná kompatibilita, musíme tento stav akceptovať. Typst je nový, „navoňaný“, zatiaľ nemusí riešiť zmeny prichádzajúce s evolúciou programovacích jazykov alebo paradigiem. Cynicky by sme mohli povedať: „Počkajme dvadsať rokov a uvidíme.“

**Začiatocníci.** Ak chcete L<sup>A</sup>T<sub>E</sub>X vhodný pre začiatocníkov, jednoducho im odporučte napríklad L<sup>A</sup>T<sub>E</sub>X. Alebo dobrý editor. Ak niekto používa \*Office kvôli (na prvý pohľad neprekonateľnej) zložitosti L<sup>A</sup>T<sub>E</sub>X-u, ten pravdepodobne bude podobne odradený i pri pohľade na zadávanie zložitejších štruktúr (tabuľky, obrázky) v Typst-e.

**Konzistencia.** Typst je oveľa konzistentnejší. Každá funkcia a knižnica tretích strán funguje rovnako, bez prekvapení, na rozdiel od knižníc L<sup>A</sup>T<sub>E</sub>X-u, z ktorých každá má svoj vlastný, zvláštny spôsob používania makier. To platí. Zatiaľ. Možno očakávať, že s pribúdajúcimi balíčkami sa tento stav zhorší.

**Ekosystém, komunita, odborné príspevky.** L<sup>A</sup>T<sub>E</sub>X má rozsiahlejší ekosystém a komunitu. Ak organizátor konferencie alebo publikácie vyžaduje súbory .tex, potrebujete L<sup>A</sup>T<sub>E</sub>X. Ak práca závisí od (veľmi) špecializovaných balíkov L<sup>A</sup>T<sub>E</sub>X-u, Typst pravdepodobne ešte nemá ich ekvivalenty.

**Vlastné šablóny, už vytvorené dokumenty atď.** Počas dlhého obdobia sme vytvorili obrovské množstvo dokumentov, máme šablóny na ponuky, kalkulácie, protokoly, faktúry. Ako veľké musí byť nadšenie pre Typst, aby človek tieto veci prepísal do jeho jazyka?

**Niektoré funkcie ešte neexistujú.** Hoci je Typst pre mnohých dnes plnohodnotnou náhradou za L<sup>A</sup>T<sub>E</sub>X, stále existujú funkcie, ktoré Typst (zatiaľ) nepodporuje. Napríklad vloženie PDF či EPS súborov ako obrázkov.

**Kto je teda „koncový užívateľ“ Typst-u?** Zodpovedať na otázku z abstraktu nie je jednoduché. Nejde len o porovnanie výhod a nevýhod, ani o množstvo už hotových dokumentov; motivácia často pramení z nekvantifikovateľných hodnôt, akými sú zvedavosť, nadšenie pre nové veci, a tzv. „vnútorný nepokoj“ a potreba robenia si „vecí po svojom“. a tu je Typst v slobodnom svete GNU/linuxu „ako doma“.

## 8. Záver

Typst zatiaľ nedosahuje univerzálnu robustnosť L<sup>A</sup>T<sub>E</sub>X-u, no prináša sľubnú perspektívu ako moderná alternatíva pre tvorbu dokumentov. Jeho prítomnosť je symbolom dôležitosti plurality nástrojov.

Veda a tvorba napredujú tam, kde existuje priestor pre rozmanitosť. Každá alternatíva, každý experiment, či už úspešný alebo neúspešný, pridáva do kolektívnej znalostnej zásoby podnety k zmene, iné pohľady, alebo aj čistú radosť z tvorivého procesu. Typst dokazuje, že aj v zdanlivo uzavretých či menej populárnych oblastiach ako je typografia, má zmysel hľadať nové cesty – a to nielen z praktického hľadiska, ale aj ako principiálnu víziu, že iba rôznorodosť zaručuje (umožňuje) progres. Neskôr sa jedny riešenia stávajú dominantnými na úkor iných, ale vždy túto rozmanitosť potrebujeme, aby sme sa vôbec dostali **ďalej**.

## Literatúra

- [1] FABO, R.: *GNU EMACS + ORG MÓD = EASY L<sup>A</sup>T<sub>E</sub>X*, Otvorený softvér vo vzdelávaní, výskume a IT riešeniach, zborník medzinárodnej konferencie OSSConf 2024, Žilina, 2.–4. júla 2024, ISBN 978-80-554-2105-6, <https://frcatel.fri.uniza.sk/ossconf/zborniky-online/ossconf2024/files/04-fabo.pdf>.
- [2] MÄDJE, L.: *Typst a Programmable Markup Language for Typesetting*, Master's thesis, Computer Science, Technical University of Berlin Faculty IV – Electrical Engineering and Computer Science, 8.9.2022, <https://laurmaedje.github.io/programmable-markup-language-for-typesetting.pdf>.
- [3] OLŠÁK, P.: *T<sub>E</sub>Xbook naruby*, Brno, KONVOJ 2000, ISBN 80-7302-007-6.
- [4] OLŠÁK, P.: , *Proč nerád používám L<sup>A</sup>T<sub>E</sub>X*, Zpravodaj CSTUGu, 1–2/1997 (89–99), <https://petr.olsak.net/ftp/olsak/bulletin/nollatex.pdf>.
- [5] TeX and Typst: Layout Models, [online], dátum citovania [2026-04-20], <https://laurmaedje.github.io/posts/layout-models/>.
- [6] Typst documentation, [online], dátum citovania [2026-03-25], <https://typst.app/docs/>.
- [7] Typst vs L<sup>A</sup>T<sub>E</sub>X: Complete comparison for Academic Writing | 2026 | TypeTeX, [online], dátum citovania [2026-04-02], <https://www.typetex.app/comparisons/typst-vs-latex>.
- [8] Typst vs LaTeX by Amiitesh | Medium, [online], dátum citovania [2026-04-02], <https://medium.com/@tspamiitesh/typst-vs-latex-2a13e4a1b61e>.
- [9] Typst vs L<sup>A</sup>T<sub>E</sub>X: Compilation Speed Comparison 2026, [online], dátum citovania [2026-04-03], <https://www.typetex.app/comparisons/typst-vs-latex-speed>.
- [10] Typst 0.12 is just ... better, [online], dátum citovania [2026-04-20], <https://typst.app/blog/2024/typst-0.12/>.

## Kontaktná adresa

**Ing. Richard Fabo**, Triton Famme s.r.o., Levočská 862/28, 058 01 Poprad, Slovensko,  
*E-mailová adresa:* [triton@famme.sk](mailto:triton@famme.sk), <https://triton.famme.sk>.

## MODERNÍ POSTUPY PARALELIZACE ZPRACOVÁNÍ RASTROVÝCH GEODAT V R – PŘÍPAD KOMBINACE TIDYVERSE A BALÍČKU TERRA

ONDŘEJ LEDVINKA (CZ)

**Abstrakt.** Zpracování rastrových geodat v R vyžaduje specifické zacházení, protože tento typ geodat je často velmi velký na to, aby se vešel do paměti běžné pracovní stanice, prostřednictvím které je aplikována většina základních R funkcí. O to více je zde třeba dbát na správnou strategii, pokud jde o paralelizaci procesů za účelem urychlení práce. Přístupů k paralelizaci ve vztahu k rastrovým geodatům v R jistě existuje více, což závisí na konkrétní úloze, a zcela jistě se bude situace odlišovat na různých operačních systémech. Tento příspěvek odráží zkušenosti autora na OS Windows 10 a 11 a jeho zaměření na využití funkcí ekosystému balíčků *tidyverse*, přičemž z jeho pohledu a jím řešených úkolů dnes při paralelizaci hraje prim experimentální implementace funkcí balíčku *mirai* při funkcionálním programování.

**Klíčová slova.** R balíček terra, funkcionální programování, démoni, klimatologie.

## MODERN APPROACHES TO PARALLELIZATION OF RASTER GEODATA PROCESSING IN R – A CASE OF THE COMBINATION OF THE TIDYVERSE AND THE TERRA PACKAGE

**Abstract.** The processing of raster geodata in R requires a specific treatment, because this type of geodata is often very large to fit into the memory of a common workstation, through which most of the basic R functions are applied. This makes it all the more necessary to pay attention to the correct strategy when it comes to process parallelization to speed up work. There are certainly more approaches to parallelization in relation to raster geodata in R, which depends on specific tasks, and, certainly, the situation will differ on different operating systems. This contribution reflects the author's experience on OS Windows 10 and 11 and his focus on utilizing the functions of the *tidyverse* package ecosystem, where from his perspective and the tasks he solves, the experimental implementation of the *mirai* package functions in functional programming plays a major role in parallelization today.

**Keywords.** R package terra, functional programming, daemons, climatology.

### 1. Úvod

Rastrová geodata a jejich sady jsou zvláště v dnešní době velmi velká na to, aby se vešla do paměti standardní pracovní stanice celá. V moderních balíčcích určených ke zpracování rastrových geodat v R jsou proto voleny strategie, které jsou ve srovnání s klasickým zpracováním dat v R dosti nezvyklé. Může jít o tvorbu

dočasných souborů, využití funkcí psaných v C++, načítání hodnot rastru ze souboru po částech apod. V neposlední řadě lze volit paralelizaci procesu zpracování rastrových geodat, pokud to situace umožňuje. Jakmile však přejdeme k těmto postupům, můžeme narazit na několik problémů, které souvisí s objektem v Globálním prostředí, jenž se velmi často pouze odkazuje na data uložená na disku v podobě souboru. Z tohoto důvodu nefungují klasické postupy serializace objektu a ukládání dat do RDS souborů. Toto rovněž souvisí i s paralelizací procesu zpracování většího množství rastrových souborů, resp. souborů obsahujících více rastrových vrstev najednou.

Za kombinace přístupů *tidyverse* [9, 10] a funkcí balíčku **terra** [3] si ukážeme, jak efektivně pracovat s velkým množstvím rastrových geodat v R. Představíme si postupy sbalování pomocí funkce **terra::wrap()**, kde je možné namísto načítání dat do paměti stále volit i odkazování se na hodnoty v souboru. S tím jsou spjaty jednak možnosti serializace a ukládání do RDS souboru, jednak zprovoznění efektivní paralelizace procesu přes více souborů, resp. více vrstev rastrového objektu. Samotnou paralelizaci budeme provádět především přes funkce moderního balíčku **mirai** [2], jež jsou ve formě funkce **in\_parallel()** experimentálně implementovány do postupů funkcionálního programování prostřednictvím balíčku **purrr** [11], který je součástí základního *tidyverse*. Efektivitu paralelizace můžeme měřit časem stráveným od začátku do konce procesu, což umožní porovnání s funkcemi, které využívají C++. Obecně se nevyplácí proces paralelizovat, pokud lze využít funkci implementovanou v C++, ale mohou existovat i výjimky.

## 2. Specifika rastrových geodat v R

V R existují postupy tvorby vlastních rastrových geodat, které začínají definicí geometrické konstrukce, pokračují přes naplňování buněk hodnotami nějaké veličiny a končí přidáváním identifikačních údajů, jako je název vrstvy, čas, název proměnné (krátký či dlouhý) nebo jednotky, s případným následným uložením do souboru. V klimatologii mohou nová rastrová geodata vznikat též v procesu prostorové interpolace staničních dat. Protože současná doba je nakloněna otevírání dat, což nevyklučuje ani geodata, nabývá však na významu zpracování datasetů, které pro další práci připravil někdo jiný před námi. Technologie umožňují tvorbu velmi velkých rastrových geodat, a to jak z pohledu jejich prostorového rozlišení, tak z pohledu velkého množství vrstev, reprezentujících např. časovou řadu.

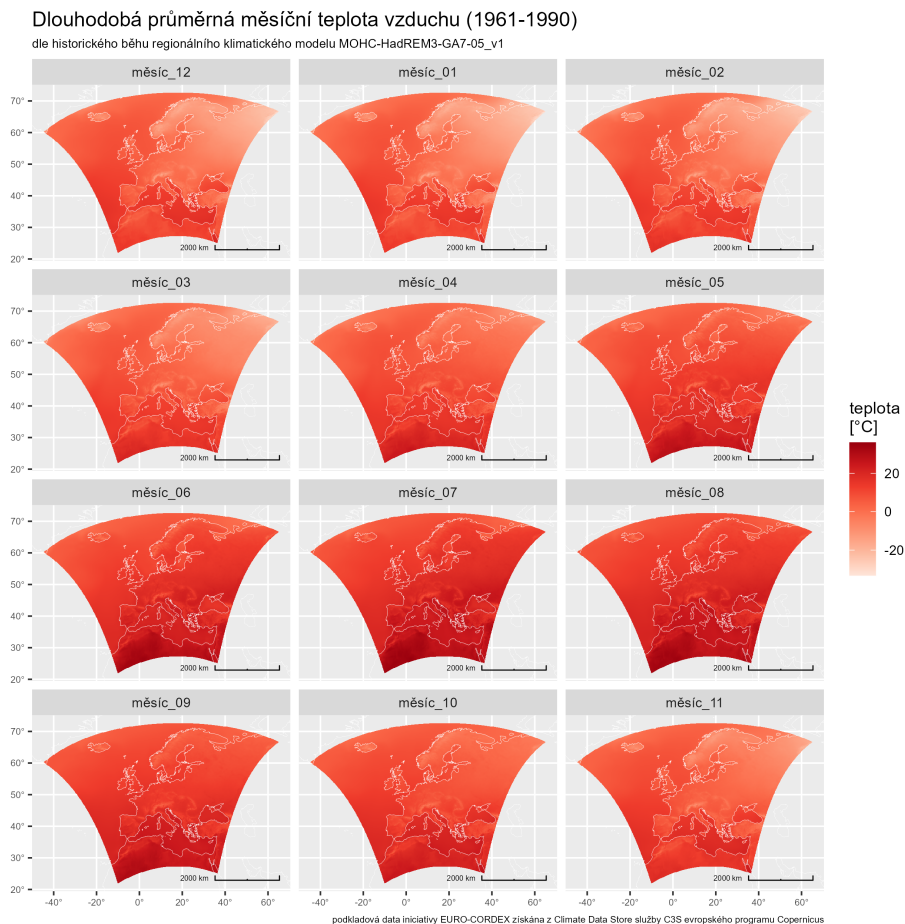
Jak již uváděl příspěvek [5], R balíček **terra** přichází se základní načítací funkcí **rast()** a po její aplikaci soubor s rastrovými geodaty je také v Globálním prostředí (např. v IDE RStudio) zřídka k vidění objekt třídy *SpatRaster*, který by dle dokumentace balíčku **terra** měl být základem pro zacházení s rastrovými geodaty v R. Namísto toho se spíše zpracovatel setká s objekty třídy *Formal class SpatRaster*, což znamená, že se jedná jen o jakýsi odkaz (*pointer*) na soubor na disku (třeba i jen dočasný). Tento fakt s sebou nese jedno zásadní úskalí, a sice,

že tyto objekty nelze klasicky serializovat, a tedy ani ukládat do RData nebo RDS souborů. Neznamená to tedy pouze, že se doporučuje dbát na pravidelné ukládání pracně získaných výsledků i mezivýsledků do souborů typu GeoTIFF nebo NetCDF, ale také, že se k serializaci těchto objektů musí přistupovat jinak. Serializace rastrových geodat je mimochodem žádoucí právě při paralelizaci jejich zpracování. Jednotliví pracovníci ve smyslu logických jader totiž musí vědět, se kterým objektem mají pracovat a ve světě funkcí balíčku **terra**, které jsou většinou implementovány v C++, se bez serializace zkrátka nelze pohnout dále. Balíček **terra** proto přichází se speciální funkcí **wrap()**, která standardně všechna data načítá do RAM, jak zmiňoval i příspěvek. Toto načítání může být velkou nevýhodou zvláště na pracovních stanicích s malou RAM, ale je to nežádoucí i na lepších stanicích. Proto autoři balíčku **terra** přidali do funkce **wrap()** přidali argument **proxy**, kterým se zajistí obdoba serializace bez načítání dat do RAM se zjevným odkazem na soubor na disku, odkud si funkce data vezmou v případě potřeby. Produktem funkce **terra::wrap()** je objekt třídy *Fromal class PackedSpatRaster*. Tento objekt již může vstupovat do paralelizovaného procesu, ale výsledek bude zaručen jen tehdy, bude-li tento objekt uvnitř procesu opět rozbalen funkcí **terra::unwrap()**.

### 3. Paralelizace práce s rastrovými objekty R balíčku terra

Mnohé funkce balíčku **terra** jsou implementovány v jazyce C++ a jsou samy o sobě dosti rychlé, takže zpočátku o paralelizaci procesů souvisejících se zpracováním objektů třídy *SpatRaster* není třeba vůbec uvažovat. Velice silným nástrojem jsou také vlastní funkce podporující funkcionální programování ve smyslu základních R funkcí, jako je **apply()** a podobné varianty. V balíčku **terra** se tak nacházejí funkce **app()**, **tapp()**, **lapp()** a **sapp()** [5]. Tyto funkce jsou optimalizované pro využití funkcí implementovaných v C++. Avšak předložit jim lze prakticky jakoukoliv jinou funkci, která nativně implementována v C++ nemusí. Lze to provést definováním vlastní funkce, která se pak bude nacházet v Globálním prostředí, což umožní její využití v argumentu **fun**. Jinou možností je předložit argumentem **fun** tzv. anonymní funkci (někdy zvanou lambda) bez nutnosti ukládat ji do Globálního prostředí. V každém případě zde ovšem musí být dodrženy základní myšlenky týkající se vstupu a výstupu. Vstup je uvažován jako několik vektorů, kterých je tolik, kolik je buněk rastrového objektu. Délka všech těchto vektorů musí být stejná a rovná se počtu vrstev rastrového objektu. Důraz je kladen na shodnost geometrických vlastností rastrových vrstev bez kterého by postup nemohl fungovat (a ani vstupní rastrový objekt by jinak nemohl být sestaven). Výsledek funkce v argumentu **fun** musí vracet právě jednu hodnotu, aby ze sady hodnot reprezentujících jednotlivé buňky, mohla být opět sestavena právě jedna nová rastrová vrstva. V případě, že funkce daná argumentem **fun** vrací více hodnot, vzniká rastrový objekt s více vrstvami. Ve funkci **lapp()** může

vystupovat více rastrových objektů najednou, ale opět musí být jejich geometrie stejná. Funkce `lapp()` dokonce akceptuje ještě jinou třídu, která přichází s balíčkem `terra`, a sice `SpatRasterDataset`.



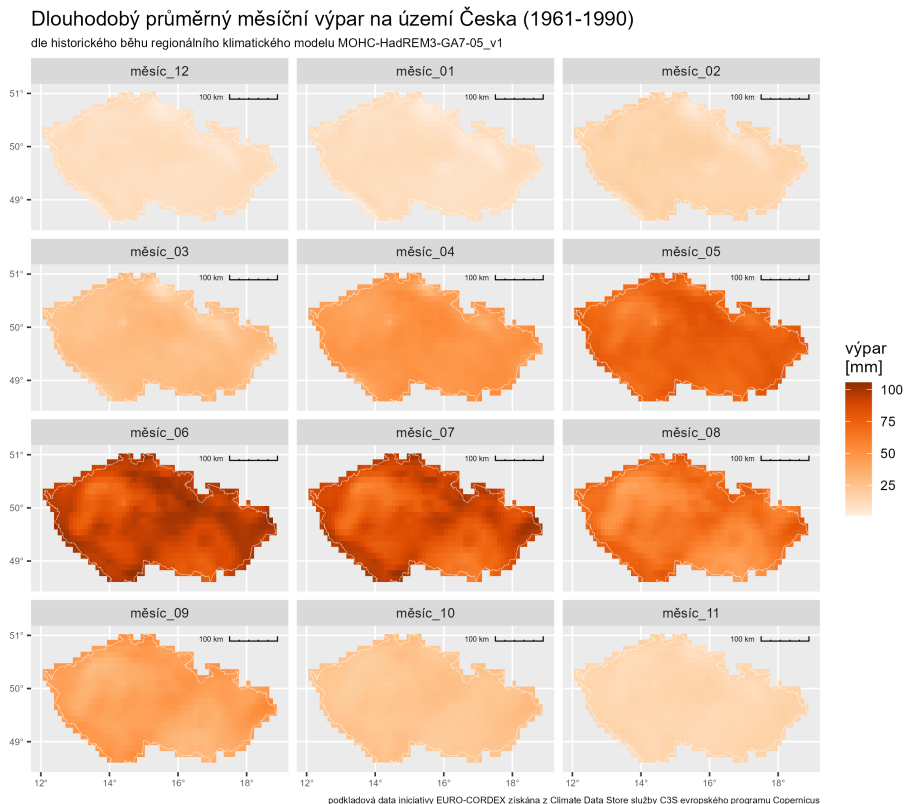
**Obrázek 1.** Vizualizace dlouhodobé průměrné měsíční teploty vzduchu v Evropě v období 1961–1990 vypočtené z rastrových vrstev vybraného regionálního klimatického modelu

Nutnost využití vlastní funkce (třeba i kombinování s funkcemi z jiných R balíčků) a anonymními funkcemi může práci s rastrovými geodaty ve smyslu nativního funkcionálního programování značně zpomalovat. Proto ve zmiňovaných funkcích balíčku `terra` existuje argument `cores`. Tento argument nemá vliv, pokud je argumentem `fun` vkládána funkce implementovaná v C++. V opačném případě může jeho nastavení vést ke zrychlení procesu. Je zapotřebí vědět, kolik logických jader pracovní stanice má. V R jejich počet dá zjistit pomocí příkazů

`parallel::detectCores()` nebo `parallelly::availableCores()`. Druhý případ samozřejmě vyžaduje doinstalování R balíčku `parallelly`. Balíček `parallel` je instalován standardně s první instalací R. Doporučuje se zadávat argumentem `cores` o jedno dostupné logické jádro méně (viz např. vinětu k balíčku `multidplyr` [8]). Ale je zapotřebí také hlídat množství RAM spotřebované nastaveným paralelizovaným procesem a následně optimalizovat. Již zde také mohou nastat problémy související s neznámým rastrovým objektem z hlediska dalších, než je hlavní pracovník. Je tedy třeba pamatovat na sbalování rastrových geodat funkcí `terra::wrap()`.

V oborech, jako je klimatologie, je možné se setkat s rastrovými geodaty rozdělenými do vrstev, kterých nezřídka bývá řádově i desetitisíce. V takových případech se nabízí nastavení paralelizace zpracování tak, že si pracovníci rozdělí více vrstev mezi sebe. Ovšem vyplatí se i vzpomenout si na skutečnost, že funkce balíčku `terra` dokáží zpracovat více vrstev najednou. Strategie zpracování se tedy mohou různit podle úlohy a dat po ruce. Skvělou vlastností vylepšených datových rámců známých ve světě *tidyverse* jako *tibble* je skutečnost, že jejich sloupce mohou bez větších problémů obsahovat seznamy. Prvky těchto seznamů mohou být objekty třídy *SpatRaster* a rovněž *PackedSpatRaster*. A nemusí jít jen o objekty s jednou vrstvou. Lze tak získat přehlednou tabulku, jejíž sloupce tvoří kromě seznamů také vektory. Toho lze pak s výhodou využít při aplikaci technik funkcionálního programování se slovesy a příslovci z balíčku `purrr`, z nichž nejdůležitější jsou zřejmě funkce `map()` a `walk()` a jejich varianty. Druhá jmenovaná funkce je přitom obdobou funkce první s tím rozdílem, že se hodí spíše k tzv. vedlejším efektům, ke kterým náleží např. ukládání souborů. Poměrně nedávno se v dokumentaci k funkci `purrr::map()` objevila zmínka o možnostech vkládání mapovaných funkcí do funkce `purrr::in_parallel()`. V její dokumentaci se lze dočíst, že pracuje s tzv. démony, jak jsou nazýváni pracovníci nastavení pro paralelizaci funkcemi balíčku `mirai`. Jsou jimi hlavně funkce `mirai::daemons()` pro nastavení počtu démonů a `mirai::require_daemons()`, která někdy musí být spuštěna před aplikací funkce `in_parallel()`, aby došlo ke správné aktivaci démonů. Démoni se pak vypínají zadáním nulového počtu (tj. `mirai::daemons(0)`) nebo ukončením sezení. Démoni opět musí vědět, se kterým objektem mají pracovat, a musí mít i přístup k funkcím z balíčků, které pro hlavní proces máme již načteny. Argumenty funkce `purrr::in_parallel()` jsou k tomu dobře uzpůsobeny, přičemž zdánlivě neznáme balíčky lze pohodlně načítat i v definici anonymní funkce.

Autorovy zkušenosti dokazují, že aplikováním démonů balíčku `mirai` lze skutečně rapidně proces zpracování rastrových geodat urychlit. Tento přístup předčí, nejen pokud se rychlosti týká, i přístupy zakládané na funkcích balíčku `furrr` [7]. Čas strávený jednotlivými přístupy a strategiemi lze měřit např. funkcemi balíčku `tictoc` [4].



**Obrázek 2.** Vizualizace dlouhodobého průměrného měsíčního výparu na území Česka v období 1961–1990 vypočteného z rastrových vrstev vybraného regionálního klimatického modelu

#### 4. Příklad s rastrovými vrstvami vybraného regionálního klimatického modelu

Pro demonstraci byly z Climate Data Store služby Climate Change Service evropského programu Copernicus (<https://cds.climate.copernicus.eu/>) staženy soubory s rastrovými vrstvami regionálního klimatického modelu MOHC-HadREM3-GA7-05\_v1 (historický experiment CORDEX, doména Evropy). Přitom byly vybrány rastrové vrstvy s teplotou vzduchu a výparem za období 1961–1990. Neboť tato rastrová geodata v originálních NetCDF souborech přichází v nestandardním souřadnicovém referenčním systému, byla nejprve aplikována technika warpingu, což umožnilo dostat všechny vrstvy do systému WGS 84 (EPSG:4326). Následně byly počítány dlouhodobé měsíční průměry (ve °C, resp. v mm), které již byly vykresleny. U teploty vzduchu byla ponechána celá doména Evropy (obr. 1), u výparu bylo vykrojeno jen území Česka (obr. 2). Pokud



jde o výpočet dlouhodobého měsíčního průměru z tohoto datasetu, nelze jednoduše aplikovat funkci `terra::tapp()` s přednastaveným indexem „months“ či „yearmonts“, protože každý měsíc je v modelu reprezentován 30 dny. Proto musela být zvolena jiná taktika s vlastními indexy. Po výpočtu průměrů však u výparu nastává situace, kdy se kvůli jednotkám (mm) musí aplikovat násobení počtem vteřin v každém měsíci, protože původní průměry reprezentují průměrné denní úhrny. Zde se pro přepočet nabízí porovnání aplikace funkcí implementovaných v C++ a s aplikací démonů. Bylo zjištěno, že když bylo zaměstnáno 12 démonů, proces trval zhruba 13 vteřin, kdežto bez nastavení démonů proces trval 0.03 vteřiny. To ukazuje, že je zapotřebí umět rozeznat úlohy, při kterých se demony nastavovat vyplatí.

## 5. Závěr

Příspěvek se netýkal paralelizace přes cloudové řešení ani přes tvorbu klastrů složených z více pracovních stanic. Netýkal se ani možností využití potenciálu GPU, což R určitě umí, jak dokazuje seznam specializovaných balíčků určených k paralelizaci [1]. Je zapotřebí poznamenat, že v R existuje i sofistikovaný balíček `stars` [6] zaměřený na vícerozměrná rastrová geodata, tzv. rastrové datové kryche (a nejen na ně). Tento balíček naskýtá v návaznosti na externí knihovnu GDAL další vlastní techniky paralelizace, které zde diskutovány příliš nebyly. Např. se může jednat o zpracování rastrové vrstvy s velmi podrobným prostorovým rozlišením po částech (tzv. *chuncích*), kde je velmi pravděpodobně možná také aplikace funkcionálního programování popoháněného demony z balíčku `mirai`. Nezbývá než neprobádané postupy vyzkoušet, příp. ověřit postupy nabízené nativně autorem balíčku `stars`.

**Poděkování.** Tento příspěvek vznikl s podporou Technologické agentury České republiky (TA ČR), projektů SS02030040 „Predikce, hodnocení a výzkum citlivosti vybraných systémů, vlivu sucha a změny klimatu v Česku (PERUN)“ a SS02030018 „Centrum pro krajinu a biodiverzitu (DivLand)“.

## Reference

- [1] EDDLEBUETTEL, D.: *CRAN Task View: High-Performance and Parallel Computing with R*, 2026, Comprehensive R Archive Network, <https://CRAN.R-project.org/view=HighPerformanceComputing>.
- [2] GAO, C.: *Mirai: Minimalist Async Evaluation Framework for R*, 2026, pp. 2.7.1, doi: 10.32614/CRAN.package.mirai Comprehensive R Archive Network.
- [3] HIJMANS, R. J. – BROWN, A. – BARBOSA, M.: *Terra: Spatial Data Analysis*, 2026, doi: 10.32614/CRAN.package.terra, Comprehensive R Archive Network.
- [4] IZRAILEV, S.: *Tictoc: Functions for Timing R Scripts, as Well as Implementations of "Stack" and "StackList" Structures*, 2024, pp. 1.2.1, doi: 10.32614/CRAN.package.tictoc, Comprehensive R Archive Network.

- [5] LEDVINKA, O.: *R balíčky určené k práci s geodaty a jejich využití v projektu PERUN*, Zborník príspevkov medzinárodnej konferencie OSSConf 2025, 2025, pp. 95–102, Žilinská univerzita v Žiline, ISBN 978-80-554-2209-1.
- [6] PEBESMA, E. J. – BIVAND, R.: *Spatial Data Science: With Applications in R*, 2023, The R Series, CRC Press, Boca Raton, ISBN 978-0-429-45901-6.
- [7] VAUGHAN, D. – BENGTSSON, H. – DANCHO, M.: *Furrr: Apply Mapping Functions in Parallel Using Futures*, 2026, doi: 10.32614/CRAN.package.furrr, Comprehensive R Archive Network.
- [8] WICKHAM, H.: *Multidplyr: A Multi-Process 'dplyr' Backend*, 2025, pp. 0.1.4, doi: 10.32614/CRAN.package.multidplyr, Comprehensive R Archive Network.
- [9] WICKHAM, H. – AVERICK, M. – BRYAN, J. – CHANG, W. – MCGOWAN, L. – FRANÇOIS, R. – GROLEMUND, G. – HAYES, A. – HENRY, L. – HESTER, J. – KUHN, M. – PEDERSEN, T. – MILLER, E. – BACHE, S. – MÜLLER, K. – OOMS, J. – ROBINSON, D. – SEIDEL, D. – SPINU, V. – TAKAHASHI, K. – VAUGHAN, D. – WILKE, C. – WOO, K. – YUTANI, H.: *Welcome to the Tidyverse*, year=2019, Journal of Open Source Software, vol. 4, no. 43, pp. 1686, ISSN 2475-9066, 10.21105/joss.01686.
- [10] WICKHAM, H. – ČETINKAYA-RUNDEL, M. – GROLEMUND, G.: *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*, 2023, 2nd ed, O'Reilly Media, Inc, Sebastopol, USA ISBN 978-1-4920-9740-2.
- [11] WICKHAM, H. – HENRY, L.: *Purrr: Functional Programming Tools*, 2026, pp. 1.2.2, doi: 10.32614/CRAN.package.purrr, Comprehensive R Archive Network.

## Kontaktní adresa

**Mgr. Ondřej Ledvinka, Ph.D.**, Oddělení hydrofondu a bilancí, Úsek hydrologie, Český hydrometeorologický ústav, Na Šabatce 2050/17, 143 06 Praha 4 – Komořany, Česko,

*E-mailová adresa:* [ondrej.ledvinka@chmi.cz](mailto:ondrej.ledvinka@chmi.cz), *Aktuální adresa:* Katedra fyzické geografie a geoekologie, Přírodovědecká fakulta, Univerzita Karlova, Albertov 6, 128 00 Praha 2, Česko,

*E-mailová adresa:* [ledvinka@natur.cuni.cz](mailto:ledvinka@natur.cuni.cz).

## 3D SCANNING IN PRACTICE: TERRESTRIAL LASER SCANNING OF CULTURAL HERITAGE SITES

MICHAL LEKÝR (SK)

**Abstract.** Terrestrial laser scanning has become an accessible and effective method for documenting large objects, architectural spaces, and cultural heritage sites. The quality of a resulting 3D dataset, however, depends not only on the technical parameters of the scanner, but also on the complete measurement methodology, including planning, scanner positioning, registration strategy, and point cloud processing. This article presents a practical workflow for 3D scanning with the Leica ScanStation P30, with emphasis on large-scale objects. The workflow is based on field experience from geometrically complex environments, including historical buildings and heritage sites such as Budatín Castle and Bojnice Castle. The paper discusses scan planning, reduction of occluded areas, balancing of point density and acquisition time, registration of multiple scans, and common problems caused by inaccessible positions, reflective surfaces, moving objects, and complex geometry. A part of the workflow is based on open-source tools, especially CloudCompare that allow point cloud cleaning, alignment, segmentation, visualization, and distance analysis, and MeshLab which is suitable for mesh-oriented processing and model optimization.

**Key words and phrases.** Terrestrial laser scanning, 3D scanning, point cloud, CloudCompare, MeshLab, cultural heritage, open-source software.

## 3D SKENOVANIE V PRAXI: POZEMNÉ LASEROVÉ SKENOVANIE PRE LOKALITY KULTÚRNEHO DEDIČSTVA

**Abstrakt.** Laserové 3D skenovanie sa stalo dostupnou a účinnou metódou mapovania veľkých objektov, architektonických priestorov a lokalít kultúrneho dedičstva. Kvalita výsledného 3D dátového súboru však nezávisí iba od technických parametrov skenera, ale aj od celkovej metodiky merania vrátane plánovania, umiestnenia skenera, stratégie registrácie a spracovania mračna bodov. Tento príspevok predstavuje súhrn skúseností z praxe pri 3D skenovaní pomocou zariadenia Leica ScanStation P30 s dôrazom na veľkorozmerné objekty. Pracovný postup vychádza z povahy skenovaných lokalít, ktoré sú tvorené geometricky zložitými prostrediami, ako napr. Budatínsky hrad alebo Bojnický zámok.

Článok sa zaoberá plánovaním skenovania, možnosťami vyváženia hustoty bodov, redukcie času zberu dát, registráciou viacerých skenov a ďalšími problémami spôsobenými neprístupnými stanoviskami, reflexnými povrchmi, pohybujúcimi sa objektmi a zložitou geometriou. Súčasťou pracovného postupu je aj využitie open-source nástrojov, najmä programu CloudCompare, ktorý umožňuje čistenie, zarovnávanie, segmentáciu, vizualizáciu a analýzu vzdialeností v mračne bodov, a programu MeshLab, ktorý je vhodný na spracovanie mračien bodov orientované na generovanie 3D modelov a ich optimalizáciu.

**Kľúčové slová.** Terestriálne laserové skenovanie, 3D skenovanie, mračno bodov, CloudCompare, MeshLab, kultúrne dedičstvo, open-source softvér.

## 1. Introduction

Three-dimensional scanning is now a common method for acquiring spatial information about physical objects, interiors, buildings, industrial sites, and cultural heritage monuments. The general idea is simple: a real object or environment is measured and transformed into a digital representation that can be inspected, processed, archived, compared, and reused. In practice, however, the process is rarely simple. Large-scale scanning projects require a combination of technical knowledge, geometric reasoning, careful preparation, and systematic data management.

Terrestrial laser scanning (TLS) is particularly useful when the objective is to acquire dense and accurate spatial data from static positions. A scanner records millions of measured points, each with a position in three-dimensional space and, depending on the device and configuration, intensity or color information. The resulting point cloud can be used for documentation, visualization, measurement, deformation analysis, reverse engineering, architectural survey, digital preservation, or as a basis for further 3D modelling. Surveys of 3D scanning technologies show that laser scanning and photogrammetry represent important families of methods for acquiring 3D data, each with different strengths, limitations, and processing requirements [7].

The practical motivation of this article is the documentation of large and geometrically complex objects. Castles, historical interiors, courtyards, staircases, towers, and irregular rooms are difficult scanning environments because they contain occlusions, narrow passages, reflective or dark surfaces, decorations, and areas that cannot be accessed directly. In such cases, the main challenge is not merely to operate the scanner. The more important task is to design a measurement strategy that produces a complete, reliable, and manageable dataset.

The article focuses on a workflow used with the Leica ScanStation P30. The P30/P40 scanner family is designed for high-definition 3D data capture and is specified by the manufacturer with a scan rate up to one million points per second and ranges up to 270 m depending on the model and measurement conditions [9, 10]. The workflow described here is not limited to this particular scanner. The same methodological principles apply to other terrestrial laser scanners that produce dense point clouds from multiple scanning stations.

The second motivation is the role of open-source software in 3D documentation. Proprietary tools are often used for scanner control and initial data export, but open-source applications can provide powerful and transparent processing possibilities. CloudCompare is a 3D point cloud and triangular mesh processing application originally designed for comparing dense point clouds and point

clouds with meshes [2, 3]. MeshLab provides tools for processing and editing 3D triangular meshes, including cleaning, healing, inspecting, rendering, texturing, and converting models produced by 3D digitization devices [12]. Open-source tools are not only educational alternatives, but can be integrated into a serious practical workflow.

## 2. Terrestrial Laser Scanning

Terrestrial laser scanning measures the surrounding environment from a fixed position. a laser beam is emitted, reflected from visible surfaces, and detected by the instrument. The scanner determines the distance to the measured surface and combines it with angular measurements to compute a 3D point. By rotating the measuring unit, the scanner captures a large number of points in a spherical or near-spherical field of view. The final result of one scan station is a point cloud representing all visible surfaces from that scanner position.

The Leica ScanStation P30 is used in this article as a representative professional TLS device. For documentation tasks, its practical advantages are range, measurement speed, and the possibility of obtaining dense 3D data from stable positions. According to the official Leica ScanStation P30/P40 data sheet, the system accuracy and range parameters are suitable for survey-grade documentation tasks under appropriate measurement conditions [10]. In real scanning projects, however, the scanner specification is only one part of the result. The achievable quality also depends on the environment, reflectivity of surfaces, angle of incidence, stability of the instrument, selected point spacing, number of stations, registration strategy, and post-processing decisions.

In documentation of large objects, one scan is almost never sufficient. Each scanner position captures only surfaces that are directly visible from that position. Areas behind columns, furniture, walls, vegetation, sculptures, staircases, or architectural details remain hidden. Therefore, the object or environment must be captured from multiple stations. These scans must later be transformed into one consistent coordinate system. The need for multiple scans is the main reason why planning and registration are central topics in practical TLS.

A useful way of thinking about a scanning project is to distinguish between three levels of decisions. The first level is the documentation objective. For example, a project may require a visually complete model for presentation, a geometrically accurate survey for restoration, a point cloud for measurement, or a mesh model for digital exhibition. The second level is the field methodology: scanner locations, point density, targets, overlap, and time schedule. The third level is the processing methodology: registration, cleaning, segmentation, export formats, and final presentation. a weak decision at any level can reduce the quality of the final dataset.

### 3. Measurement Planning and Scanner Positioning

Planning starts before the first scan is acquired. The operator must define what is being documented, what quality is required, and how the final dataset will be used. a point cloud intended only for visual orientation does not require the same density and registration control as a dataset intended for dimensional analysis or restoration documentation. Similarly, a scan of a small room can be planned differently from a scan of a castle courtyard or a large facade.



**Figure 1.** Laser scanner Leica P30 in field conditions during scanning of the Bojnice Castle.

The first practical step is a site inspection. During this inspection, the operator identifies accessible positions, possible scanner stations, obstacles, problematic surfaces, and safety limitations. Historical buildings often contain narrow stairs, low passages, fragile surfaces, visitor routes, locked rooms, and areas where tripods cannot be placed. Outdoor heritage sites introduce additional constraints such as weather, sunlight, moving visitors, vegetation, and limited scanning time.

Scanner positioning is mainly a problem of visibility and overlap. Each scan station should capture new information, but it must also overlap sufficiently with neighbouring scans. Overlap is necessary for registration, especially when cloud-to-cloud or hybrid registration is used. If two scans share only a small or geometrically weak area, the registration can become unstable. Long corridors, repetitive architectural elements, flat walls, and symmetrical spaces are typical examples where registration may be ambiguous.

The reduction of occluded areas is one of the most important practical tasks. Occlusion occurs whenever one object blocks the scanner's view of another surface. In buildings, occlusions are caused by walls, columns, vaults, doors, furniture, railings, statues, vegetation, and people. The only reliable way to reduce occlusions is to scan from additional positions. This requires a compromise: more stations increase completeness and registration robustness, but they also increase acquisition time, data volume, and processing effort.

Point density must also be planned. Higher density provides more detail but increases scan time and file size. Lower density may be sufficient for large flat surfaces but can miss small decorative elements. In practice, mixed strategies are often used. General geometry can be captured with moderate density, while detailed areas can be scanned with higher resolution from closer positions. In historical interiors, decorated ceilings, portals, sculptures, and inscriptions may require special attention.

## 4. Data Acquisition Workflow

A practical data acquisition workflow can be divided into six steps: site inspection, measurement planning, scanner setup, acquisition of individual scans, field quality control, and data backup. These steps are simple in principle but very important in large projects.

During site inspection, the operator identifies the object boundaries, expected outputs, possible scanner positions, and problematic surfaces. In heritage documentation, this step should also include discussion with site administrators. It may be necessary to respect visitor hours, protected zones, restrictions on equipment placement, and safety conditions.

Measurement planning defines the scanning stations and the expected relationship between them. If targets are used, their distribution must be planned so that they are visible from multiple stations and are geometrically well distributed. If





**Figure 2.** Scanning plan based on the digital map of the area.

natural features are used, the operator must ensure sufficient overlap and avoid areas where the geometry is too uniform. a hybrid approach is often the most robust: targets provide controlled correspondences, while cloud-to-cloud alignment refines the registration using the measured geometry.

Scanner setup includes levelling, stable tripod placement, selection of scan resolution, and configuration of image or color capture if required. Stability is particularly important in historical buildings with uneven floors, stairs, or outdoor terrain. The scanner should be placed where it is safe, visible, and protected from movement by visitors or staff.



After each scan, field quality control should be performed. The operator should check whether the expected surfaces were captured, whether the scan contains obvious gaps, whether moving objects disturbed the measurement, and whether the next planned station still makes sense. This is especially important because returning to the site later may be expensive or impossible.

Raw data organization is also part of the acquisition workflow. Each scan should be named consistently and stored with metadata such as location, station number, date, operator notes, and any known problems. Backups should be created as soon as possible. A well-organized dataset reduces later processing errors and makes the project reproducible.

Typical problems include shadow zones, reflective materials, glass, polished stone, metal, dark surfaces, and moving objects. Reflective or transparent surfaces can produce missing data, noisy measurements, or false returns. Dark surfaces may reduce signal quality. People moving through the scene create unwanted points and partial objects. Outdoor scanning can be influenced by rain, wind, direct sunlight, and vegetation movement. These problems cannot always be eliminated, but they can be managed by scanning from additional positions, changing the measurement angle, repeating problematic scans, or removing unwanted points during processing.

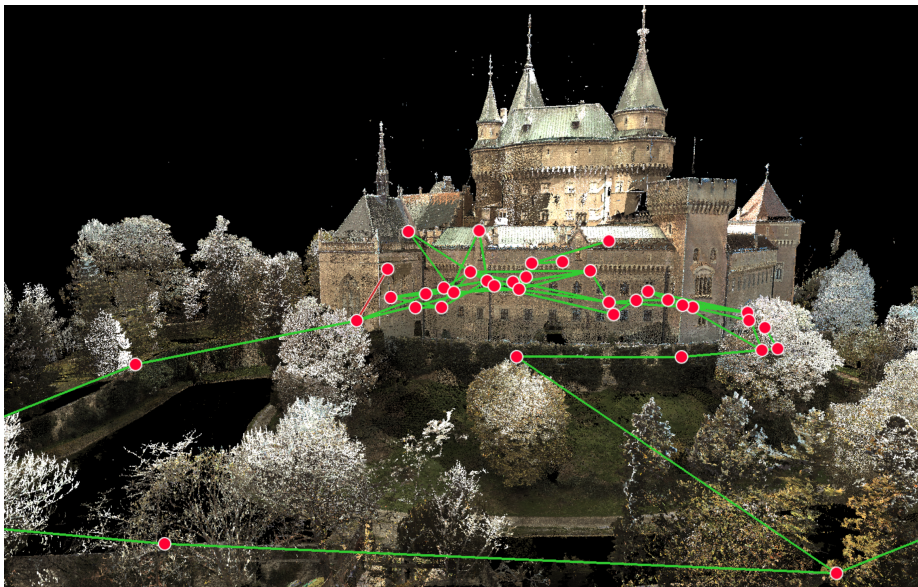
## 5. Registration of Multiple Scans

Registration is the process of transforming multiple scans into one coordinate system. Without registration, each scan station has its own local coordinate frame. The final dataset requires the scans to be aligned so that corresponding surfaces overlap correctly. Registration is a major source of both success and error in TLS projects.

A review of laser scanning point cloud registration methods shows that coarse-to-fine strategies are common in LiDAR registration [1]. In practice, this means that scans are first brought into approximate alignment and then refined. The approximate alignment may use artificial targets, manually selected corresponding points, natural features, or external survey control. The refinement often uses geometric minimization methods such as the Iterative Closest Point (ICP) algorithm.

Target-based registration uses physical targets placed in the scene. These targets are detected in multiple scans and used as correspondences. This approach can be accurate and reliable when targets are visible, well distributed, and measured consistently. Its disadvantages are the time required to place and measure targets and the fact that targets may be difficult to use in fragile or restricted heritage environments.

Cloud-to-cloud registration uses the measured geometry itself. It is useful when enough overlapping surface geometry is available. The advantage is that



**Figure 3.** Raw point cloud before cleaning and segmentation, including noise, redundant regions, vegetation, animals and other moving objects.

no artificial targets are required. The disadvantage is that the method can fail or produce unstable results when overlap is insufficient, when the geometry is repetitive, or when the initial alignment is too poor. CloudCompare includes tools for manual alignment and fine registration using ICP, which can be used in such workflows [5, 6].

A hybrid strategy combines targets, manual constraints, and cloud-to-cloud refinement. This is often practical in complex buildings. Targets or manually selected points provide initial alignment, while ICP refines the relationship between overlapping point clouds. However, ICP should not be treated as a magical correction tool. It assumes that the initial alignment is already reasonably close and that the selected overlapping regions represent the same geometry. Otherwise, it can converge to an incorrect solution.

Quality control after registration should include numerical and visual checks. Numerical values may include target residuals, registration errors, or cloud-to-cloud distances. Visual inspection is equally important: edges, corners, floors, walls, and repeated architectural details should align without visible double structures. In large projects, error accumulation must be monitored. If each local registration is slightly wrong, the accumulated error can become significant over a long sequence of rooms or exterior stations.



**Figure 4.** Registered point cloud of a geometrically complex object, after cleaning and segmentation. The figure illustrates the result after removal of noise, moving objects, and irrelevant surroundings.

## 6. Point Cloud Processing Using Open-Source Tools

The raw registered point cloud is not the end of the workflow. It usually contains redundant points, noise, moving objects, irrelevant surroundings, and areas that need segmentation or additional analysis. This is where open-source software becomes especially useful.

CloudCompare is a central tool in the workflow described here. It supports point cloud and mesh processing and was originally designed for comparison of dense point clouds [2,3]. In practical documentation, CloudCompare can be used for importing standard point cloud formats, visual inspection, segmentation, cleaning, subsampling, registration, distance analysis, and export.

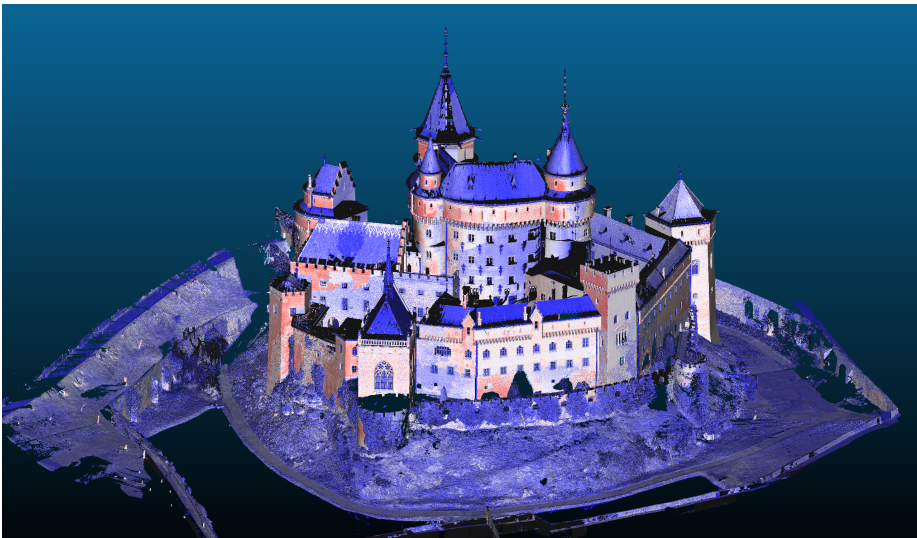
Cleaning is often the first processing step. It removes unwanted points such as people, vegetation movement, temporary objects, measurement noise, or parts of the surrounding environment that are not relevant to the documentation task. Cleaning can be manual, semi-automatic, or based on scalar fields and geometric properties. The operator must be careful not to remove important details. In heritage documentation, small ornaments or damaged areas may look like noise but may be historically relevant.

Segmentation divides the point cloud into meaningful parts. For example, an interior scan can be segmented into walls, floor, ceiling, furniture, decorative elements, windows, and doors. Segmentation makes later visualization and analysis easier. It also helps when only selected parts of the dataset need to be exported or converted into meshes.

Subsampling is used to reduce data volume. Large scans can contain hundreds of millions of points. Such datasets are difficult to process, visualize, and share. Subsampling can reduce the number of points while preserving the overall geometry. The correct level of subsampling depends on the intended use. a dataset for web presentation can usually be simplified more than a dataset for metric analysis.

Distance analysis is another important function. Cloud-to-cloud and cloud-to-mesh distances can be used to compare datasets, evaluate registration quality, detect changes, or verify deviations between measured and modelled geometry. In restoration and monitoring tasks, distance analysis can support objective comparison between different states of an object.

MeshLab complements CloudCompare in mesh-oriented workflows. It is an open-source system for processing and editing 3D triangular meshes, with tools for cleaning, inspecting, rendering, texturing, converting, and preparing models produced by 3D digitization devices [12]. In a scanning workflow, MeshLab can be used for mesh generation, mesh cleaning, normal computation, decimation, and preparation of models for presentation or archiving. It is especially useful when the final output must be a surface model rather than a point cloud.



**Figure 5.** Example of point cloud processing in CloudCompare. Scalar-field visualization of ray reflection quality.

The use of open-source tools has methodological and educational advantages. Processing steps can be demonstrated to students without licensing barriers. Students can install the tools on their own computers, repeat the workflow, inspect data, and understand point cloud concepts directly. For research, open tools support transparency and reproducibility. For practice, they provide a flexible environment for tasks that do not always require proprietary processing packages. At the same time, it is important to be realistic. Open-source tools require user knowledge and careful validation. The quality of a result depends on the method and operator, not only on the software license.

## 7. Heritage Sites as Complex Environments

The described workflow has been tested in real-world conditions, particularly during documentation of historical buildings and heritage sites, including Budatín Castle, Bojnice Castle, Cieszyn castle, Oravska Lesna railway, Open-air museum Výchylovka, Čaplovič library in Dolný Kubín, Raden house in Čičmany, House no. 42 in Čičmany, Open-air museum in Slemeniu, Divinka mansion, Radola mansion, Wedding palace in Bytča, Castle Ksiazat Sulkowskich in Bielsko-Biała, Castle in Pszycynie and many other geometrically complex environments. These sites are useful examples because they combine large exterior volumes, interior rooms, narrow passages, decorative details, and restrictions typical for cultural heritage objects.

Castles and historical buildings are difficult scanning environments. Their geometry is usually irregular and layered. Exterior facades may contain towers, courtyards, arcades, windows, roofs, and walls that create many shadowed areas. Interiors may include high ceilings, staircases, corridors, alcoves, furniture, exhibitions, and decorative elements. Some areas are difficult to access because of conservation rules, visitor routes, or physical limitations. These constraints make it necessary to combine technical planning with practical flexibility.

In such environments, the operator must constantly balance completeness, accuracy, and time. Capturing every small detail may require many scans and very high density, but this may be impractical for a large building. A more useful strategy is to define documentation priorities. General geometry can be captured with a systematic station network, while important details can be scanned more densely from selected positions.

The case experience also demonstrates that the processing workflow must be planned together with field acquisition. If the final result is intended for visualization, the project may focus on completeness and visual clarity. If it is intended for measurement, the project must prioritize registration control, documented accuracy, and preservation of metric detail. If the output is intended for education or presentation, simplified models, screenshots, and interactive viewers may be more important than full-resolution point clouds.

## 8. Conclusion

The practical experience described in this article supports a simple conclusion: the success of a 3D scanning project is determined by the complete workflow, not by the scanner alone. a professional scanner provides high-quality measurements, but poor planning can still produce incomplete or difficult-to-register data. Conversely, a well-planned project can produce useful results even when the environment is challenging.

Measurement planning is the first critical factor. The operator must understand the object, expected outputs, and constraints. Scanner positions should be chosen to maximize coverage and overlap while minimizing redundant data. In large or complex buildings, it is often useful to think in terms of connected zones: exterior, courtyard, entrance, corridors, rooms, stairs, and special details. Each zone should be connected to neighbouring zones by sufficient overlap or control points.

The second factor is registration strategy. Registration is not a purely automatic step. It requires evaluation of overlap, geometry, residuals, and visual consistency. Target-based, cloud-to-cloud, and hybrid methods all have advantages. The correct choice depends on the site and the output requirements. Reviews of LiDAR registration methods emphasize the importance of robust registration because point clouds from different platforms, angles, or times must be integrated into a consistent representation [1]. Heritage documentation adds further difficulty because scenes may contain narrow interiors, decorations, and preservation constraints [11].

The third factor is data volume. Dense scanning produces rich data but also large files. This influences storage, processing time, backup strategy, and the ability to share results. Subsampling and segmentation are therefore not secondary tasks. They are necessary steps for transforming raw measurement data into usable information.

The fourth factor is software selection. CloudCompare and MeshLab are valuable because they provide accessible tools for point cloud and mesh workflows. CloudCompare is particularly strong for point cloud inspection, segmentation, registration support, and distance analysis. MeshLab is useful when the workflow moves toward mesh generation and optimization. The combination of these tools can cover many practical needs in education, research, and applied documentation.

In university teaching, this workflow has strong educational value. Students can learn geometry, coordinate transformations, registration principles, point cloud data structures, visualization, mesh processing, and data quality assessment using real datasets. Such tasks connect computer graphics with practical measurement and software engineering. They also show that open-source software

can be part of professional technological practice rather than only a substitute used when commercial software is unavailable.

This article presented a practical workflow for terrestrial laser scanning of large-scale objects and cultural heritage environments using the Leica ScanStation P30 as a representative scanner. The workflow includes planning, scanner positioning, data acquisition, registration, point cloud cleaning, segmentation, analysis, and preparation of outputs. The main message is that result quality depends on the whole methodology. Scanner parameters are important, but they cannot compensate for poor planning, insufficient overlap, unmanaged occlusions, or weak registration control.

Open-source software has a meaningful place in this workflow. CloudCompare provides practical tools for point cloud processing, registration support, segmentation, visualization, and distance analysis. MeshLab complements it with mesh processing, cleaning, simplification, and preparation of models for presentation or archiving. These tools can support professional practice, research, and teaching, provided that their outputs are validated and used with an understanding of their assumptions and limitations.

The workflow is especially relevant for documentation of historical buildings, where complex geometry, limited accessibility, and preservation constraints require careful methodology. Future work may include integration of terrestrial laser scanning with photogrammetry, UAV data, BIM/HBIM workflows, digital twins, VR/AR visualization, and long-term archival strategies for cultural heritage datasets.

## About the Author

Michal Lekýr is a lecturer at the Faculty of Management Science and Informatics, University of Žilina, where he teaches computer graphics, 3D computer graphics, and low-level programming. He is also the founder of 3Xcore, ltd. and has long-term experience in software development, interactive 2D and 3D presentation systems, visualization, and applied programming projects. His work connects academic teaching with real-world development, with a focus on practical innovation, analytical thinking, and modern digital technologies.

## References

- [1] Cheng, L. – Chen, S. – Liu, X. – Xu, H. – Wu, Y. – Li, M. – Chen, Y.: *Registration of Laser Scanning Point Clouds: a Review*, Sensors 2018, 18(5), 1641.
- [2] CloudCompare: *Presentation*, <https://www.cloudcompare.org/presentation.html>.
- [3] CloudCompare: *CloudCompare v2.6.1 User Manual*, <https://www.cloudcompare.org/doc/qCC/CloudCompare%20v2.6.1%20-%20User%20manual.pdf>.
- [4] CloudCompare Wiki: *Introduction*, <https://www.cloudcompare.org/doc/wiki/index.php/Introduction>.

- [5] CloudCompare Wiki: *Alignment and Registration*, [https://www.cloudcompare.org/doc/wiki/index.php/Alignment\\_and\\_Registration](https://www.cloudcompare.org/doc/wiki/index.php/Alignment_and_Registration).
- [6] CloudCompare Wiki: *ICP*, <https://www.cloudcompare.org/doc/wiki/index.php/ICP>.
- [7] Daneshmand, M. – Helmi, A. – Avots, E. – Noroozi, F. – Alisinanoglu, F. – Arslan, H. S. – Gorbova, J. – Haamer, R. E. – Ozcinar, C. – Anbarjafari, G.: *3D Scanning: a Comprehensive Survey*, arXiv:1801.08863, 2018.
- [8] Grussenmeyer, P. – Landes, T. – Voegtli, T. – Ringle, K.: *Recording Approach of Heritage Sites Based on Merging Point Clouds from High Resolution Photogrammetry and Terrestrial Laser Scanning*, ISPRS Archives, 2012.
- [9] Leica Geosystems: *Leica ScanStation P40/P30 laser scanners*, <https://leica-geosystems.com/products/laser-scanners/scanners/leica-scanstation-p40-p30>.
- [10] Leica Geosystems: *Leica ScanStation P30/P40 data sheet*, [https://leica-geosystems.com/-/media/files/leicageosystems/products/datasheets/leica\\_scanstation\\_p30-p40\\_civil\\_ds.ashx](https://leica-geosystems.com/-/media/files/leicageosystems/products/datasheets/leica_scanstation_p30-p40_civil_ds.ashx).
- [11] Markiewicz, J. et al.: *The evaluation of hand-crafted and learned-based features in Terrestrial Laser Scanning–Structure-from-Motion indoor point cloud registration: the case study of cultural heritage objects and public interiors*, Heritage Science, 2023.
- [12] MeshLab: *MeshLab official website*, <https://www.meshlab.net/>.
- [13] PyMeshLab: *Filter list documentation*, [https://pymeshlab.readthedocs.io/en/latest/filter\\_list.html](https://pymeshlab.readthedocs.io/en/latest/filter_list.html).

## Contact address

**Ing. Michal Lekýr, PhD.**, Department of Mathematics Methods and Operations Research, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovakia,  
*e-mail*: [michal.lekyr@fri.uniza.sk](mailto:michal.lekyr@fri.uniza.sk), <https://lekыр.com/>.



## **ARCHITEKTÚRA OPEN-SOURCE INFRAŠTRUKTÚRY PRE SPRACOVANIE A PRIESTOROVÚ OPTIMALIZÁCIU GEOGRAFICKÝCH BIG DATA**

PETER MÁRTON (SK) A BRANISLAV BLAŽEK (SK)

**Abstrakt.** Príspevok prezentuje návrh a implementáciu robustného open-source riešenia na spracovanie a priestorovú optimalizáciu geografických Big Data z programu Strava Metro. Výskum reaguje na metodické úskalia spojené s fixáciou poskytovaných dát na konkrétnu historickú verziu siete chodníkov, ulíc a ciest k 1. júnu 2025 z OpenStreetMap pre všetky poskytované dáta o objemoch a prúdoch cyklistov. Ako stabilný referenčný rámec implementuje globálny diskretný grid systém Uber H3 v prostredí databázy PostgreSQL s rozšírením PostGIS. Pomocou natívnych databázových rozšírení a indexovej matematiky presúva workflow výpočtovú záťaž z geometrie na číselné indexy. Výsledkom je mnohonásobné zrýchlenie priestorových operácií a topologicky čistá štruktúra pripravená na plynulú vizualizáciu v prostredí QGIS.

**Kľúčové slová.** Strava, open-source, PostgreSQL/PostGIS, QGIS.

## **OPEN-SOURCE INFRASTRUCTURE ARCHITECTURE FOR THE PROCESSING AND SPATIAL OPTIMISATION OF GEOGRAPHIC BIG DATA**

**Abstract.** This paper presents the design and implementation of a robust open-source solution for the processing and spatial optimisation of geographic data from the Strava Metro programme. The research addresses the methodological challenges associated with the provided data being locked to a specific historical version of the OpenStreetMap as of 1 June 2025 across all supplied datasets on cyclist volumes. To establish a stable reference framework, it implements the Uber H3 global discrete grid system within a PostgreSQL database environment with the PostGIS extension. By utilising native database extensions and indexing mathematics, the workflow shifts the computational load from geometry to numeric indices. This approach achieves a manifold acceleration in spatial operations, delivering a topologically clean data structure optimized for seamless visualization in QGIS.

**Keywords.** Strava, open-source, PostgreSQL/PostGIS, QGIS.

### **Úvod**

Digitálna platforma Strava, ktorá pôsobí v oblasti monitorovania športových a pohybových aktivít, oznámila v marci 2026 desiatku vybraných výskumných projektov pre rok 2026 v rámci svojho prestížneho programu Strava Metro for Academic Researchers [2]. Jeden z desiatich projektov zahrnutých do programu

Strava Metro for Academic Researchers v roku 2026 je zo Žilinskej univerzity v Žiline. V roku 2026 bolo do programu prihlásených viac ako 200 projektov z celého sveta. Medzi desiatimi podporenými sú iba tri z Európy. Platforma Strava Metro je kľúčovým analytickým nástrojom na hlbšie pochopenie toho, akým spôsobom sa ľudia pohybujú vo svojom domovskom prostredí.

V našom príspevku predstavíme, čo sme urobili za prvých šiest mesiacov nášho projektu. Chceme sa podeliť o naše skúsenosti z rôznych oblastí, ale predovšetkým o informácie, ako sme poskytnuté dáta uložili do vlastnej open-source databázy a ako postupne vyvíjame rôzne verzie ich vizualizácie. Počas nasledujúcich šiest mesiacov sa chceme zamerať na priestorové analýzy a prípravu metodík na efektívne využitie dostupných dát z platformy Strava Metro pri rozvoji cyklistickej infraštruktúry na úrovni miestnej a regionálnej samosprávy.

Príspevok je po úvode rozdelený do šiestich častí. Prvá časť charakterizuje získané dáta z portálu Strava Metro. V druhej časti predstavíme náš koncept databázy, ako sme ju naplánovali a tiež, čo sme sa naučili pri jej vytváraní. V tretej časti uvedieme našu prácu s gridom H3, ktorý je pre nás dôležitý kvôli agregácii, vizualizácii a analýze priestorových údajov. Vo štvrtnej časti opíšeme databázové riešenie, postup importu dát a spôsoby ich priestorovej optimalizácie. V piatej časti predstavíme, ako sme pri riešení technických problémov, optimalizácii databázových dopytov a spracovaní dát využívali generatívnu umelú inteligenciu. V závere zhodnotíme efektívnosť použitých open-source riešení a význam gridu H3 pri prekonávaní topologickej nestability a vytváraní podmienok pre dlhodobý monitoring cyklistickej dopravy.

## 1. Charakteristika dát Strava Metro

Na úvod tejto časti je potrebné zdôrazniť, že dáta z platformy Strava, poskytované v programe Metro for Academics Researchers, sú dôverné a ich šírenie či publikovanie v pôvodnej podobe je zakázané (napr. ako priame snímky obrazovky z webového rozhrania). Aj z tohto dôvodu sme v našom výskume zvolili metodiku vytvorenia vlastnej databázy a následných priestorových analýz v open-source riešeniach, čo nám umožňuje prezentovať výsledky bez porušenia licenčných podmienok.

Primárnym pracovným prostredím používateľa na portáli `metroview.strava.com` je sekcia Map, ktorá funguje ako pokročilá webová geoinformačná aplikácia (webGIS). Toto prostredie ponúka päť kľúčových režimov vizualizácie a filtrácie dát (Map Views), ktoré umožňujú hlboké dopravno-geografické analýzy správania cyklistov:

- **Counts (Edges alebo Hexagons):** Tento režim poskytuje kľúčové informácie o objemoch a smerových intenzitách cyklistických prúdov. Podrežim Edges mapuje záťaž priamo na sieť pozemných komunikácií, kým podrežim Hexagons

agreguje tieto toky do rovnako veľkých polygónov – šesťuholníkov (hexagónov) – gridu Uber H3 úrovne 11. V počiatočnej fáze projektu bolo sťahovanie údajov pre celé územie Slovenska dostupné iba cez líniové úseky (Edges).

- **Routes:** Režim, ktorý umožňuje porovnať najkratšie dostupné trasovania s reálne najobľúbenejšími trasami. Tento režim pomáha odhaliť vplyv topografie a kvality infraštruktúry na rozhodovanie cyklistov.
- **Travel Times:** Režim slúži na modelovanie časovej dostupnosti územia z vybraného bodu. Na základe reálne zaznamenaných rýchlostí umožňuje napríklad generovať izochrónne mapy.
- **Origins & Destinations:** Tento režim agreguje dáta do hexagónov Uber H3 úrovne 8 a poskytuje prehľad o tom, v ktorých lokalitách cyklistické aktivity reálne začínajú a končia. Rozhranie k tomu ponúka detailnú hodinovú a dennú časovú granularitu, socio-demografické odhady a rozlíšenie účelu jazdy.
- **Heatmap:** Globálne najznámejšia forma vizualizácie crowdsourcingových dát, ktorá formou rastrovej mapy hustoty vizuálne interpretuje celkovú frekvenciu prejazdov bez ohľadu na podkladovú sieť.

Z hľadiska manažmentu regionálneho rozvoja a plánovania cyklotrás poskytuje práve webGIS rozhranie sekcie Map primárne indície o dopyte po infraštruktúre, pričom možnosť exportu týchto údajov vo formátoch ESRI Shapefile a CSV umožňuje ich následné spracovanie a pokročilé priestorové modelovanie v prostredí desktopového GIS.

## 2. Naša infraštruktúra pre zálohovanie dát a priestorové analýzy

Používateľ má pri práci v portáli Strava Metro k dispozícii iba základné nástroje potrebné pre vizualizáciu intenzít aktivít cyklistov. Portál však umožňuje export niektorých typov dát. Preto sme sa rozhodli vytvoriť vlastné úložisko, databázu, pre všetky užitočné dáta. Uložené dáta používame pre vlastné vizualizácie priestorových údajov, porovnanie a vyhodnocovanie trendov. Pripravujeme si ich pre priestorové analýzy využiteľné pri plánovaní cyklistickej infraštruktúry.

Pre potreby nášho dátového úložiska sme v rámci fakulty alokovali prostriedky vo virtuálnej serverovej infraštruktúre, kde sme zriadili dedikovaný server `gis.fri.uniza.sk`. Pre počiatočnú fázu projektu nám bol pridelený diskový priestor s kapacitou 64 GB, pričom s ohľadom na predpokladaný nárast objemu dát do konca roku 2026 nepredpokladáme prekročenie hranice 128 GB. Na tomto virtuálnom stroji je nasadený operačný systém Debian GNU/Linux 12 (s kódovým označením bookworm). Pri výbere databázového riešenia bol našou jasnou voľbou open-source objektovo-relačný databázový systém PostgreSQL, aktuálne v produkčnej verzii 18.4 (distribúcia 18.4-1.pgdg12+1 z oficiálneho repozitára pre Debian 12). Pre plnohodnotnú prácu s priestorovými dátami, ich efektívnu správu a realizáciu pokročilých priestorových analýz bola databáza hneď v úvode

rozšírená o priestorovú nadstavbu PostGIS. Všetky spracovávané dáta sú centralizované v databáze, ktorá slúži ako spoločné výskumné a výučbové prostredie pre učiteľov a študentov zapojených do geoinformatických aktivít na našej fakulte. S cieľom zachovať čistotu dátovej štruktúry využívame samostatnú databázovú schému „trasa002“, do ktorej postupne importujeme očistené tabuľky exportované z portálu Strava Metro. Keďže naša databáza primárne neslúži na vizualizáciu kartografických výstupov, na tento účel využívame populárny open-source desktopový nástroj QGIS. V súčasnosti sa spoliehame na LTR verziu 3.40 Bratislava.



**Obr. 1.** H3 grid úr. 11 – Vážska cyklomagistrala pri Kotešovej © OpenStreetMap contributors

### 3. H3 grid a jeho využitie v dátach STRAVA

V našom výskume sme sa rozhodli, že čiarovú vrstvu s údajmi o intenzitách aktivít cyklistov na jednotlivých úsekoch pozemných komunikácií na území Slovenska prebratú z portálu Strava Metro z režimu Counts – Edges nebudeme používať priamo pre porovnanie a vyhodnocovanie trendov a pre priestorové analýzy využiteľné pri plánovaní cyklistickej infraštruktúry. Na portáli sú totiž úseky z OpenStreetMap pre všetky dáta z ktoréhokoľvek obdobia (od roku 2021 do roku 2026), viazané na jednu verziu mapy. Napr. v marci 2026 boli všetky údaje viazané na verziu OpenStreetMap z 1. júna 2025. Myslíme si, že ak by sme používali pre zobrazovanie a následne aj pre porovnávanie topológiu OpenStreetMap iba z 1. júna 2025, prinieslo by to pri dlhodobom výskume komplikácie spojené s neustálym vývojom a zmenami reálnej siete pozemných komunikácií. Aby sme eliminovali závislosť našich analýz od nestabilnej vektorovej čiarovej geometrie a získali univerzálny priestorový rámec na porovnávanie intenzít jázd cyklistov

v čase, prešli sme od reprezentácie v čiarovej vrstve ku polygónovej vrstve – reprezentácii gridom. Prevzali sme grid, ktorý používa na svojom portáli samotný poskytovateľ dát. Je to globálny diskretný súradnicový systém – UBER H3 grid. Výhodou H3 gridu je, že pomocou neho môžeme reprezentovať územie Slovenska kontinuálnym pokrytím bez medzier. Tento systém je na portáli Strava Metro používaný preto, lebo umožňuje transformovať počty aktivít zobrazené na úsekoch do rovnako veľkých polygónov (hexagónov). To nám umožňuje stať sa nezávislými od častých zmien topológie čiarových vrstiev OpenStreetMap a otvára nám to cestu k exaktnému monitorovaniu trendov nezávisle od konkrétneho trasovania pozemných komunikácií [6].

H3 grid (obr. 1, 2) je globálny indexačný systém vyvinutý spoločnosťou Uber [4, 5]. V tomto systéme sa vo svete neorientujeme pomocou tradičných súradníc zemepisnej šírky a dĺžky, ale pomocou miliónov pravidelných šesťuholníkov (hexagónov). Voľba šesťuholníkov má zásadnú matematickú výhodu oproti bežným štvorcovým mriežkam: každý šesťuholník má presne šesť susedov a stredy všetkých týchto susedných buniek sú od jeho vlastného stredu rovnako vzdialené. Táto vlastnosť odstraňuje geometrické skreslenia a umožňuje nám vykonávať priestorové dopyty, analýzy spádových oblastí či generovanie heatmap mnohonásobne rýchlejšie, keďže softvér nepracuje s komplikovanými tvarmi, ale iba s jedinečnými číselnými kódmi priradenými každému šesťuholníku. Celý systém je navrhnutý hierarchicky a ponúka 16 úrovní rozlíšenia – od najväčších buniek s rozlohou tisícov kilometrov štvorcových až po detailné mikro-šesťuholníky s plochou pod jeden meter štvorcový. Výhodou je, že bunky na seba logicky nadväzujú a jeden materský šesťuholník v sebe približne integruje sedem menších šesťuholníkov nižšej úrovne. To umožňuje pri analyzovaní priestorového rozloženia geografických objektov plynule prepínať mierku detailu, čo v našom príspevku využívame pri práci s hrubším rozlíšením úrovne 8 (plocha cca 0,73 km<sup>2</sup>) pre analýzu zdrojov a cieľov dopravy a tiež pri analýze rozmiestnenia služieb cyklistom, s detailným rozlíšením úrovne 11 (plocha cca 2 150 km<sup>2</sup>) pre modelovanie intenzít jázdy cyklistov.

Pri hľadaní optimálneho prostredia pre generovanie a spracovanie priestorových údajov v H3 gridoch sme najskôr testovali voľne dostupné riešenia v desktopovom GIS, konkrétne plugin H3 Toolkit v prostredí QGIS. Praktické testovanie ukázalo, že tento nástroj funguje úplne spoľahlivo a bezproblémovo pri riešení lokálnych a priestorovo malých úloh. QGIS však pri tomto prístupe spracováva všetky H3 algoritmy na pozadí prostredníctvom Python skriptov nad kompletnou geometriou, čo so sebou prináša obrovské výkonnostné limity. V momente, kedy sme potrebovali pracovať s gridom úrovne 11 pre územie celého Slovenska, prípadne generovať široké pásy pre analýzu našej Vážskej cyklomagistrály s celkovou dĺžkou presahujúcou 250 km, sa plugin H3 Toolkit stal kvôli extrémnej časovej odozve a zamŕzaniu softvéru úplne nepoužiteľný. Naším maximom bolo vygenerovanie gridu hexagónov úrovne 11 pre celé Slovensko do formátu Geopackage. Hotový súbor gpkg mal 10 GB.



**Obr. 2.** H3 grid úroveň 11 s intenzitami aktivít (2. august 2025) pri Kotešovej © ÚGKK SR

V súčasnosti máme na našom databázovom serveri PostgreSQL inštalované rozšírenia `h3` a `h3_postgis`. Základné rozšírenie `h3` integruje nízkoúrovňovú C-knižnicu systému Uber H3 priamo do jadra databázy PostgreSQL. Po jeho aktivácii získava server schopnosť vykonávať zložité priestorové operácie pomocou čistej indexovej matematiky. Namiesto náročného výpočtu topologických vzťahov medzi komplexnými polygónmi (ako je prekrývanie či hľadanie susedov) prebiehajú všetky operácie ako rýchle numerické porovnávanie 64-bitových celočíselných indexov šesťuholníkov. Rozšírenie poskytuje natívne funkcie pre prevod súradníc na H3 kód (`h3_lat_lng_to_cell`), prechod medzi hierarchickými úrovňami (`h3_cell_to_parent`) či bleskové generovanie spádových oblastí bez potreby tvorby metrických buffrov (`h3_grid_disk`). Rozšírenie `h3_postgis` slúži ako špeciálna nadstavba, ktorá zabezpečuje priamu kompatibilitu a prepojenie medzi matematickým svetom H3 indexov a priestorovým svetom PostGIS geometrií. Bez tohto rozšírenia by bolo spájanie tradičných vektorových dát (čiar a bodov) s H3 gridom výpočtovo komplikované. `h3_postgis` umožňuje databáze natívne transformovať binárne priestorové objekty PostGIS (typy `GEOMETRY` alebo `GEOGRAPHY`) priamo do buniek H3 a naopak – matematický kód šesťuholníka dokáže okamžite preložiť do plnohodnotného šesťuholníkového polygónu (`h3_cell_to_boundary_geometry`). Toto prepojenie je kľúčové pre klientske GIS aplikácie (napr. QGIS), ktoré vďaka tomu dokážu výsledky matematických H3 dopytov okamžite interpretovať a vykresliť ako vizuálnu mapovú vrstvu.

Praktickým dôkazom opodstatnenosti a prínosu inštalácie rozšírení pre grid H3 bol experimentálny porovnávací test rýchlosti spracovania záujmového územia

koridoru okolo Vážskej cyklomagistrály od Strečna po Komárno. Naším cieľom bolo vygenerovať široký pás, resp. obalovú zónu (buffer), s rozsahom 30 šesťuholníkov úrovne 11 na každú stranu od Vážskej cyklomagistrály (čo priestorovo reprezentuje pásmo so šírkou približne 2 km na obe strany). Pri počiatočnom prístupe, kedy sme využili štandardné priestorové operácie nad surovou geometriou celoslovenského gridu (obsahujúceho vyše 23 miliónov buniek) a polygónom klasickej obalovej zóny pomocou funkcie `ST_Intersects`, musel PostGIS vykonávať milióny náročných prienikov bodov a línií. Tento masívny priestorový dopyt trval databázovému serveru 38 minút a 12 sekúnd. Následne sme identickú úlohu realizovali optimalizovaným postupom s využitím natívnej H3 matematiky. Trasovanie magistrály bolo najskôr segmentované na body a prepočítané na reťaz základných H3 buniek. Okolo nich sme následne nechali databázu vygenerovať matematické susedstvo pomocou vstavanej funkcie `h3_grid_disk(cell, 30)`. Keďže tento algoritmus neprepočítava zložité priestorové topológie, ale realizuje len primitívne sčítanie a odčítanie v poli číselných indexov, identický dopyt prebehol za neuveriteľných 21 sekúnd. Implementácia natívneho H3 rozšírenia tak priniesla viac ako 100-násobné zrýchlenie výpočtu, pričom výsledkom je navyše topologicky dokonale čistá štruktúra rovnomerných šesťuholníkov pripravená na okamžité prepojenie s intenzitami z platformy Strava Metro.

#### 4. Databázové riešenie a priestorová optimalizácia dát

Pri počiatočných pokusoch o vizualizáciu a analýzu dát stiahnutých z platformy Strava Metro sme narazili na výrazné výkonnostné limity bežných desktopových GIS nástrojov. Pokus o priame načítanie surových priestorových vrstiev (vo formáte ESRI Shapefile) pre celé územie Slovenska v prostredí QGIS sa ukázal ako nerealizovateľný.

Napríklad, čiarová vrstva vo formáte shapefile, sieť úsekov (sekcia Counts – Edges) pre celé Slovensko, má v rozbalenom stave veľkosť približne 270 MB (sieť úsekov na jeden rok – 2025). CSV súbor s intenzitami cyklistov pre túto sieť má veľkosť viac ako 125 MB, pričom obsahuje takmer milión riadkov (každý riadok je jeden úsek siete). Podobne kritická situácia nastala pri pokuse o prácu s globálnou mriežkou H3 šesťuholníkov úrovne 11 pre celé Slovensko, ktorá obsahuje viac ako 23 miliónov polygónov. Priame načítavanie, prekresľovanie a najmä pokusy o priestorové spájanie takto masívnych súborov priamo na stolovom počítači alebo notebooku spôsobovali zamrzanie softvéru QGIS a extrémne nízku odozvu systému.

Dáta z prostredia Strava Metro sme importovali do našej databázy dvoma spôsobmi. Čiarovú vrstvu so sieťou úsekov vo formáte shp, na ktorých bola v čase od januára 2021 po súčasnosť zaznamenaná aspoň jedna jazda cyklistu, sme importovali pomocou príkazového riadku OSGeo4W Shell a nástroja `ogr2ogr` z knižnice GDAL. Tento nástroj sme použili aj pri importe h3 gridu s hexagónmi úrovne

11 pre celé Slovensko z formátu gpkg. Import h3 gridu bol realizovaný ešte predtým, než sme prišli na možnosť použitia rozšírení h3 priamo v databáze. Použitie príkazového riadku OSGeo4W Shell bolo prvým spôsobom importu. Druhým spôsobom importu, ktorý používame, je import textových CSV súborov s intenzitami jazd cyklistov priamo cez grafické rozhranie pgAdmin.

Vzhľadom na objem dát a potrebu eliminovať chyby pri pretypovaní sme implementovali dvojfázový proces importu: 1. vytvorenie dočasných tabuliek, 2. spracovanie nekonzistentných textových polí. Vytvorenie dočasných (staging) tabuliek znamená, že pre každý importovaný CSV súbor (ročné, mesačné alebo dňové intenzity cyklistov) sme najskôr navrhli čistú textovú štruktúru, kde boli všetky prichádzajúce stĺpce vrátane identifikátorov `hex_id`, `edge_uid`, `osm_id` a časových značiek definované univerzálnym dátovým typom `text`. Potom nasledovalo spracovanie nekonzistentných textových polí. Tento krok sa ukázal ako kľúčový z hľadiska zachovania integrity dát. Atribút typu aktivity totiž v surovom CSV obsahuje textové reťazce oddelené čiarkou (napr. "Ride,EBikeRide"). Pri bežnom importe by výskyt čiarky (ktorá slúži ako oddeľovač stĺpcov) spôsobil posun celej štruktúry riadka. V importnej relácii v prostredí pgAdmin bolo preto nutné striktne definovať kvalifikátory textu a úvodzovky. Týmto postupom boli dáta úspešne a bez straty riadkov nasypané do dočasnej štruktúry, pričom v ďalšej fáze sme sa mohli sústrediť na ich sofistikované pretypovanie na natívne priestorové a číselné typy, ako aj na ich priestorové orezanie pre okolie Vážskej cyklomagistrály. Aby sme však mohli toto pretypovanie a čistenie v druhej fáze úspešne zrealizovať, bolo potrebné exaktne rekonštruovať typy priamo na databázovom serveri podľa sémantického významu stĺpcov. Surová tabuľka líniových intenzít pre celú cestnú a cyklistickú sieť Slovenska totiž obsahuje široké spektrum smerových, demografických a socio-ekonomických ukazovateľov.

Vzhľadom na to, že surový import v prvom kroku uložil všetky tieto stĺpce ako `text`, proces preklopenia do finálnej tabuľky vyžadoval hromadnú SQL konverziu. Identifikátory boli pretypované na typ `bigint` (prípadne `character varying` pri H3 kódach), kým všetky intenzity, vekové sumy a počty prejazdov prešli transformáciou na typ `integer`. Pri tejto operácii bolo kritické ošetriť prítomnosť prázdnych textových reťazcov, ktoré platforma Strava generuje v úsekoch s nulovým výskytom danej demografickej alebo vekovej skupiny. Až takto upravená a vyčistená štruktúra zabezpečila absolútnu dátovú integritu pred spustením finálnych priestorových dopytov nad obalovou zónou Vážskej cyklomagistrály. Pri integrácii priestorových dát sa ako kritický bod ukázala nekonzistencia dátových typov. Kým globálna mriežka H3 bola do databázy naimportovaná z formátu Geopackage ako textový typ (`character varying`), niekedy sa nám nechtiac stalo, že exporty csv súborov zo Strava Metro boli transformované do natívneho binárneho typu `h3index`. Priame spájanie týchto typov viedlo k preťaženiu diskového priestoru z dôvodu generovania masívnych dočasných súborov (`temporary files`)



pri sekvenčnom skenovaní viac 4 miliónov polygónov Slovenska so zaznamenanými športovými aktivitami. Problém sme vyriešili explicitným pretypovaním textového kľúča na natívny h3index priamo v podmienke JOIN, čo umožnilo PostgreSQL plne využiť vybudované B-Tree indexy a skrátilo čas spracovania.

## 5. Úloha umelej inteligencie pri importe dát do databázy

Po vysvetlení priebehu vytvorenia našej databázy s dátami z portálu Strava Metro je potrebné priznať, že by sme to celé nezvládli bez použitia generatívnej umelej inteligencie. Naša interakcia s veľkým jazykovým modelom prostredníctvom Google Gemini bola a je kľúčovým akceleratorom našej práce. Spracovanie Big Data z programu Strava Metro so sebou prinieslo množstvo nečakaných nízkoúrovňových chýb v prostredí operačného systému servera, ako aj v samotnom relačnom modeli (napr. binárny nesúlad knižníc, nekompatibilita verzií PostgreSQL 17 a 18, či neočakávané správanie exotických dátových typov v klientskom softvéri QGIS). Úspešné prekonanie týchto výziev nebolo výsledkom slepého generovania kódu, ale bolo založené na vedomom, informovanom a priebežnom dialógu s AI. Prostredníctvom cieleného promptovania sme umelú inteligenciu využívali ako expertného partnera pre konzultácie architektonických rozhodnutí, optimalizáciu SQL dopytov a diagnostiku systémových chýb. Tento prístup zahŕňal:

- Návrh inžinierskych bypassov: Keď natívne H3 funkcie nefungovali pri skriptoch pre operáciu JOIN v databáze kvôli typovej nekonzistentnosti ID kódov H3 hexagónov v tabuľkách dát importovaných zo Stravy a v tabuľkách textových reťazcov gridu H3 vygenerovaného pre Slovensko, diskusia s AI nám pomohla navrhnúť elegantný bypass s vytvorením očisteného textového stĺpca (`hex_id_txt`), čím sme obišli chyby verzií a dosiahli okamžitý a stabilný JOIN text na text.
- Optimalizáciu výpočtov: Transformácia pôvodných priestorových joinov založených na náročnom porovnávaní komplexnej geometrie línií (trasa Vážskej cyklomagistrály) s 23 miliónmi polygónov H3 gridu úrovne 11 (pre územie celého Slovenska) na matematické susedstvo buniek cez funkciu `h3_grid_disk` výrazne skrátila výpočtový čas (v našom prípade z 38 minút na neuveriteľných 21 sekúnd).

Vedomé zapojenie AI do nášho výskumu v našom prípade nám dramaticky skrátilo čas potrebný na vývoj a ladenie infraštruktúry a umožnilo nám sústrediť sa na samotnú interpretáciu dopravnno-geografických výsledkov priestorovej analýzy údajov o využívaní Vážskej cyklomagistrály.

## Záver

V závere je možné konštatovať, že kombinácia open-source riešení (PostgreSQL/PostGIS, QGIS) pri budovaní infraštruktúry pre náš projekt, pre prácu

s hierarchickým gridom H3, sa ukázala ako mimoriadne efektívna a robustná odpoveď na výzvu spracovania geografických Big Data z platformy Strava Metro. Dúfame, že vyriešením niektorých problémov, napr. prekonaním topologicky nemeňtej siete pozemných komunikácií pomocou použitia H3 gridu sme položili základy pre náš ďalší výskum týchto otázok:

- Ako možno efektívne importovať a spravovať rozsiahle údaje Strava Metro v open-source databázovom prostredí?
- Aké výhody a straty informácie prináša agregácia líniových údajov do polygónov H3?
- Aké výkonnostné zlepšenie poskytujú natívne H3 operácie oproti geometrickým operáciám PostGIS?

Naším výskumom zameraným na technické otázky chceme podporiť dlhodobý monitoring cyklistickej dopravy [1]. Veríme, že naše doterajšie praktické skúsenosti a úspešne prekonané metodické úskalia poslúžia ako inšpirácia pre ďalšie výskumné tímy pracujúce s dátami crowdsourcingových platforiem a prispievajú k udržateľnému rozvoju cyklistickej infraštruktúry v meste Žilina, našom regióne i na Slovensku.

## Literatúra

- [1] PUCHER, J. – BUEHLER, R.: *Cycling towards a more sustainable transport future*, Transport Reviews, Vol. 37, no. 6, pp. 689–694, 2017, DOI: 10.1080/01441647.2017.1340234.
- [2] STRAVA METRO: *Metro for Academic Researchers*, 2026, [cit. 2026-06-22], <https://metro.strava.com/academics>.
- [3] STRAVA METRO: *Getting Started Guide*, Strava Metro Help Center, 2026, [cit. 2026-06-22], <https://stravametro.zendesk.com/hc/en-us/sections/360008086573-Getting-Started-Guide>.
- [4] UBER: *H3: Hexagonal hierarchical spatial index*, Open source geographic information system, 2026, [cit. 2026-06-22], <https://h3geo.org/>.
- [5] UBER ENGINEERING: *H3: Uber's Hexagonal Hierarchical Spatial Index*, 2018, [cit. 2026-06-22], <https://www.uber.com/us/en/blog/h3/>.
- [6] VENTER, Z. S. – GUNDERSEN, V. – SCOTT, S. L. – BARTON, D. N.: *Bias and precision of crowdsourced recreational activity data from Strava*, Landscape and Urban Planning, 2023, Vol. 232, p. 104686, DOI: 10.1016/j.landurbplan.2022.104686.

## Kontaktné adresy

**doc. Ing. Peter Márton, PhD.**, Katedra matematických metód a operačnej analýzy, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovensko,  
E-mailová adresa: [peter.marton@uniza.sk](mailto:peter.marton@uniza.sk)

**Bc. Branislav Blažek**, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovensko,  
E-mailová adresa: [blazek10@stud.uniza.sk](mailto:blazek10@stud.uniza.sk)

## **VYUŽITIE DÁT STRAVA PRE ROZVOJ CYKLODOPRAVY A PROJEKTOVÚ VÝUČBU V OBLASTI GEOINFORMATIKY**

PETER MÁRTON (SK), MARIÁN GOGOLA (SK) A ROMANA VAŠČÁKOVÁ (SK)

**Abstrakt.** Príspevok popisuje prvú fázu výskumu zameraného na inovatívny prístup k podpore manažérskeho rozhodovania v oblasti regionálneho rozvoja a udržateľnej mobility pomocou GIS a Big Data. Cieľom výskumu je analýza intenzity a preferencie cyklistickej dopravy v meste Žilina a na Vážskej cyklomagistrále. Výskumný tím má k dispozícii dáta z programu Strava Metro for Academic Researchers z obdobia rokov 2021–2026. Zatiaľ bola výskumná práca zameraná na proces spracovania, filtráciu a vizualizáciu priestorových dát. Výstupy budú slúžiť ako objektívny podklad pre strategické plánovanie samospráv pri zefektívňovaní cyklo dopravy. Článok reflektuje aj skúsenosti z projektovej výučby v inžinierskom štúdiu formou Challenge Based Education.

**Kľúčové slová.** Vážska cyklomagistrála, Strava, manažérske rozhodovanie, geoinformatika.

### **UTILISING STRAVA DATA FOR CYCLING INFRASTRUCTURE DEVELOPMENT AND PROJECT-BASED LEARNING IN GEOINFORMATICS**

**Abstract.** This paper describes the first phase of a research focused on an innovative approach to supporting managerial decision-making in regional development and sustainable mobility, using GIS and Big Data. The research aims to analyse the volume and preferences of cycling traffic in the city of Žilina and along the Váh Cycle Route. The project team has access to Strava Metro data covering the 2021–2026 period. So far, the research work has focused on the process of data ingestion, filtering, and the visualisation of spatial data. The outputs will serve as an objective baseline for local authorities' strategic planning to improve cycling infrastructure efficiency. The paper also reflects on experiences from project-based learning within the Master's degree programme using the CBE approach.

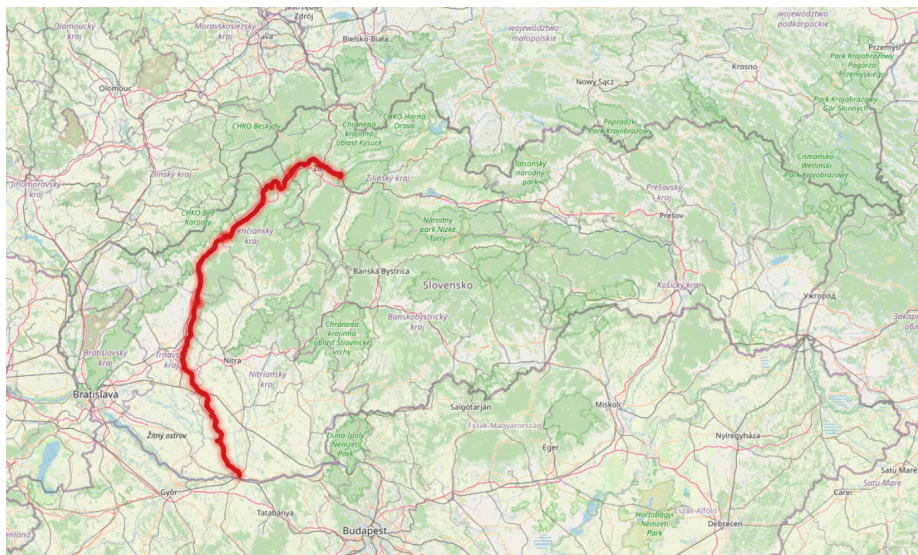
**Keywords.** Váh Cycle Route, Strava, managerial decision-making, geomatics.

## **Úvod**

Efektívne riadenie regionálneho rozvoja a strategické plánovanie udržateľnej mestskej a regionálnej mobility sa v súčasnosti nezaobíde bez objektívnych analýz priestorových údajov veľkého rozsahu (Big Data) [4,11]. Kľúčovým globálnym zdrojom takýchto informácií o reálnom správaní účastníkov cyklistickej dopravy je nástroj Strava Metro, ktorý od roku 2019 bezplatne podporuje expertov, manažérov a samosprávy pri systematickom zvyšovaní bezpečnosti a kvality cyklistickej infraštruktúry. Tento celosvetový program už nadviazal spoluprácu s viac ako

4000 odborníkmi a pozitívne ovplyvnil takmer miliardu obyvateľov. V septembri 2025 otvorila digitálna platforma Strava výzvu na podávanie projektov do druhého ročníka prestížneho akademického programu Strava Metro for Academic Researchers [5] a v marci 2026 boli oznámené výsledky [6]. Do tejto výzvy bolo prihlásených vyše 200 výskumných projektov z celého sveta, z ktorých komisia vybrala iba desiatku najperspektívnejších. Skutočnosť, že medzi desiatimi podporenými projektovými zámermi sú len tri z Európy, pričom jeden z je výskumný tím zo Žilinskej univerzity v Žiline, podčiarkuje spoločenskú a akademickú naliehavosť riešenej problematiky.

Cieľom nášho prebiehajúceho výskumu v roku 2026 je plnohodnotne transformovať získané crowdsourcingové Big Data z platformy Strava do podoby exaktných manažérskych podkladov pre mesto Žilina, Žilinský samosprávny kraj a aj ostatné samosprávne kraje na trase Vážskej cyklomagistrály. V príspevku sa zameriavame na popis ukončenej prvej fázy nášho výskumu, analýzu intenzít a preferencií cyklistov. Naším zámerom je poskytnúť samosprávam spoľahlivé analýzy reálneho pohybu tisícok obyvateľov v období rokov 2021 až 2026, čo umožní informované a zmysluplné investície do rozvoja udržateľnej cyklodopravy. Predkladaný článok navyše prináša dôležitú reflexiu toho, ako sme na našej fakulte na tento reálny geoinformatický problém zapojili do inžinierskeho vzdelávania budúcich manažérov a dátových analytikov.



**Obr. 1.** Vážska cyklomagistrála © OpenStreetMap contributors.

Príspevok je po úvode rozdelený do troch častí. V prvej časti charakterizujeme dáta získané v rámci programu Strava Metro for Academic Researchers a predstavíme ich prvé vizualizácie vytvorené pomocou open source desktopovej aplikácie

QGIS. V druhej časti uvidíme štruktúru atribútov, ktoré poskytujú informácie o intenzite a smere cyklistickej dopravy, účele jazdy, dennej dobe a socio-demografickej štruktúre používateľov. V tretej časti opíšeme zapojenie študentov do projektu prostredníctvom projektovej výučby založenej na prístupe Challenge Based Education. V závere zhrnieme doterajšie výsledky projektu a predstavíme plánované kroky jeho ďalšej fázy.

## 1. Charakteristika dát Strava Metro

Na úvod druhej časti je potrebné zdôrazniť, že dáta z platformy Strava, poskytované v programe Metro for Academics Researchers, sú dôverné a ich šírenie či publikovanie v pôvodnej podobe je zakázané (napr. ako priame snímky obrazovky z webového rozhrania).

K dátam z platformy Strava máme prístup cez portál [metroview.strava.com](https://metroview.strava.com). Po prihlásení do rozhrania máme k dispozícii tri sekcie – Dashboard, Map a Data. Prvú sekciu, Dashboard, v súčasnosti výrazne nevyužívame a podobne je na tom aj sekcia Data. Od udelenia prístupu sme najviac pracovali so sekciou Map [7].

V sekcii Dashboard je k dispozícii prehľad a štatistiky športových aktivít za jeden týždeň, pre vybrané územie a typ športu (cyklistika, beh-chôdza-turistika). Pre jeden týždeň sú v hlavičke „nástenky“ (priamy preklad slova dashboard) informácie o celkovom počte aktivít nahraných používateľmi na platformu, čísla z víkendu, celkový počet ľudí (ak niekto nahral niekoľko aktivít), odhadovaná úspora produkcie CO<sub>2</sub>, celková najazdená vzdialenosť a celkové prevýšenie všetkých aktivít. Zároveň je k dispozícii percentuálne porovnanie s predchádzajúcim týždňom (rast, pokles). Pod hlavičkou je k dispozícii šesť prehľadových grafov: počet jazd, počet aktívnych ľudí, súčet prejdenej vzdialenosti, odhad úspory produkcie CO<sub>2</sub>, demografická štruktúra aktívnych ľudí a rozdelenie podľa typu aktivít (bicykle, e-bicykle). Tieto grafy poskytujú prehľad spomínaných údajov za jednotlivé mesiace roka (prípadne týždne) a umožňujú porovnanie údajov z roku 2026 s údajmi za predchádzajúcich päť rokov.

V sekcii Map sú prehľadne, vo forme webovej mapy (webGIS) k dispozícii rôzne možnosti vizualizácie dát (Map Views). Dáta je možné zobraziť v režimoch: Counts (Edges, Hexagons), Routes, Origins&Destinations, Travel Times a Heatmap. Pri našej práci sme doteraz najviac využívali režim Counts, ktorý nám poskytuje informácie o počte cyklistov a intenzite cyklistických prúdov na jednotlivých úsekoch. Zároveň, keď sme s portálom začínali pracovať, bolo možné robiť exporthy dát pre celé Slovensko iba z podrežimu Counts – Edges.

Pri výbere Edges sú k dispozícii údaje o počte aktivít cyklistov – počte prejazdov po úsekoch. Úseky vo webovej mape je možné identifikovať nielen pomocou Edge\_UID (interné ID v dátach platformy Strava), ale aj pomocou OSM\_ID. Určitým úskalím súvisiacim s OSM\_ID v platforme Strava je ich fixácia na konkrétnu verziu OpenStreetMap. V marci 2026 to bola verzia OpenStreetMap

z 1. júna 2025. Pre výskum, ktorý pokrýva obdobie od roku 2020 až do roku 2026, to vytvára nesúlad verzií topológie. Napriek tomuto nedostatku sú dáta o aktivitách cyklistov veľmi cenné. Alternatívnym spôsobom poskytovania údajov o intenzite cyklistických prúdov sú agregované dáta vo forme hexagónov H3 na úrovni 11.

Pokiaľ máme dáta o intenzite uvedené ako hodnotu v hexagóne H3 [8, 9], výhodou je, že nie je dôležité, aké úseky sa pod ním nachádzajú a aká je reálna topológia, resp. či je iná ako pred dvoma alebo štyrmi rokmi. Dáta vo forme hexagónov H3 sú k dispozícii v režime Counts – Hexagons. Pre nás je nepríjemné, že hoci je z nášho pohľadu výhodnejšie porovnávať trendy a sledovať intenzity na skúmanom území pomocou hexagónov H3 než prostredníctvom úsekov (čiar, línií), z portálu STRAVA METRO je možné pre skúmané územie sťahovať dáta iba v podobe úsekov. Stiahnutie údajov vo forme hexagónov pre celé územie Slovenska v súčasnosti nie je možné. Spôsob, akým sme sa s týmto vysporiadali, vysvetlíme v jednej z nasledujúcich častí nášho článku. Dáta s informáciami o intenzitách pre každý úsek v topológii použitej pre Slovensko mali napríklad pre august 2025 veľkosť takmer 170 MB. Dáta sú poskytované ako balíčky zip, pričom obsahujú shp súbory s úsekmi a tabuľky vo formáte csv.

Pri popise režimu Counts je potrebné spomenúť ešte jedno dôležité metodické špecifikum. Okrem toho, že všetky údaje poskytované portálom STRAVA METRO sú prísne anonymizované, sú z dôvodu ochrany súkromia taktiež zaokrúhľované na násobky čísla 5. Ak je teda skutočná denná intenzita jász na danom úseku 73, v exportovaných údajoch uvidíme hodnotu 70. Z analytického a matematického pohľadu ide o limitujúci faktor. Keď napríklad v dátach o počiatkoch aktivít (Origins&Destinations) pre konkrétny hexagón vidíme sumár 185 začatých aktivít, z čoho je 15 priradených dochádzaniu do práce a 165 rekreácii, okamžite je zrejmé, že súčet subkategórií (180) nezodpovedá celku. Dáta týmto strácajú absolútnu matematickú presnosť a pri nízkych intenzitách to výrazne skresľuje realitu. Umelé zaokrúhľovanie dát zo strany poskytovateľa predstavuje pre mikroanalýzy (najmä v ranných a nočných hodinách s nižšou intenzitou alebo na úsekoch s nízkou intenzitou) značné skreslenie, ktoré zahmlieva reálne správanie cyklistov.

Ešte však trochu k iným režimom. Režim Routes ponúka prehľad, ktorá trasa je medzi dvoma miestami najkratšia a ktorá najobľúbenejšia. Napríklad, medzi našou fakultou a kúpeľami Rajecké Teplice vedie najkratšia trasa cez Lietavskú Lúčku a má 13,7 kilometra. Najobľúbenejšia trasa vedie cez Višňové, má 14,5 km. Ak by sme sa pozreli na trasu z našej fakulty do Piešťan, najobľúbenejšia trasa má 141,7 km. Najkratšia (nie je počítaná po diaľnici) má 127,9 km. Pri tomto režime sa v čase písania príspevku porovnávali dáta z obdobia od júna 2024 po máj 2026. Zaujímavé informácie, žiaľ bez možnosti stiahnutia, poskytuje režim Travel Times. Na základe zozbieraných dát je možné pozrieť aj to, kam všade sa dá dostať bicyklom za stanovený čas (napr. 15 minút) z našej fakulty.



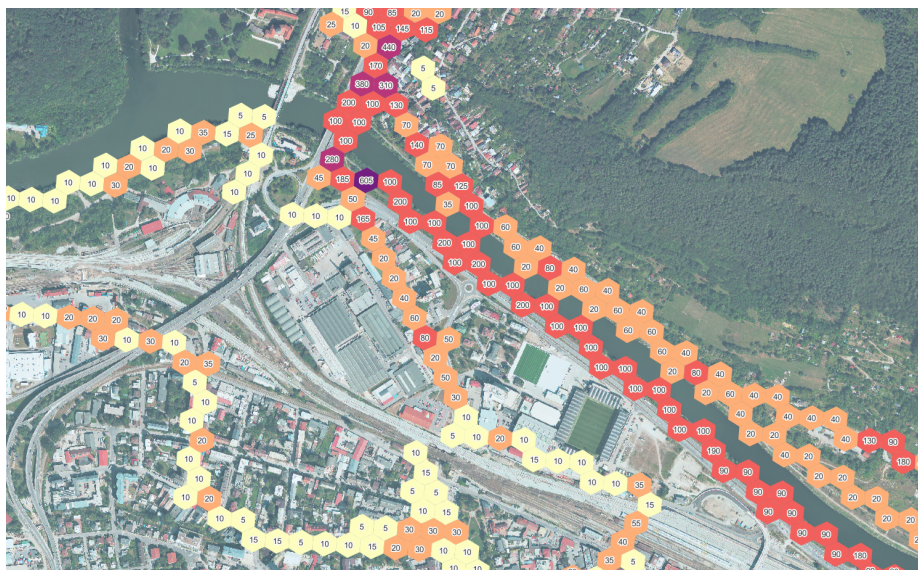
Režim Origins&Destinations patrí k tým, ku ktorým sme sa v úvode výskumu podrobnejšie nedostali. No dáta, ktoré poskytuje, sú mimoriadne užitočné pri priestorovo-časových analýzach. V tomto režime si môžeme nielen pozerať, ale aj stiahnuť dáta o tom, koľko cyklistov štartovalo v konkrétnom mesiaci alebo roku z ktorého miesta na sledovanom území, pričom tieto údaje sú agregované do hexagónov H3 úrovne 8. K webovej mape je k dispozícii aj štatistický prehľad pre každú hodinu počas dňa, grafické vyjadrenie intenzity počas jednotlivých dní v týždni, celkový počet nahraných aktivít, najjazdené kilometre dokopy, ušetrené CO<sub>2</sub> a prekonané prevýšenie. Napríklad, auguste 2025 bolo na platformu Strava zo Slovenska nahraných takmer 350 tisíc cyklistických záznamov, najjazdených približne 22,5 milióna kilometrov, ušetrených takmer 350 tisíc kg CO<sub>2</sub>, prekonaných cez 200 miliónov metrov prevýšenia. V tomto období 77% jász bolo rekreačného (športového) charakteru a 23% boli jazdy za každodennými povinnosťami (dochádzanie). Zároveň, 7% jász bolo urobených na elektrických bicykloch. Najviac jász, okolo 80 tisíc bolo v čase od 16:00 do 18:00 h. Sobota a nedeľa boli najsilnejšie dni, s intenzitou okolo 60 tisíc jász na jeden deň, teda spolu cez víkend asi 120 tisíc jász (viac ako jedna tretina zo všetkých jász). Dáta s informáciami za jeden mesiac za celé Slovensko, zbalené vo formáte zip, majú 7 MB. Obsahujú shp súbory s hexagónmi H3 a tabuľky vo formáte csv. Režim Heatmaps ponúka globálne populárny pohľad, kde sú intenzity prúdov cyklistov zobrazené rôznymi odtieňmi farby podľa hustoty trás. Donedávna bola táto vrstva integrovaná napríklad aj na slovenskom portáli freemap.sk.



**Obr. 2.** Vizualizácia dát zo sekcie Map v QGIS – úseky s počtom aktivít (cyklochodníky okolo Váhu v Žiline, august 2025) © ÚGKK SR

Prostredie portálu Strava Metro sa neustále dynamicky vyvíja. Ako používatelia sme často prekvapení, keď po pár týždňoch od poslednej návštevy nájdeme na portáli nové dát na stiahnutie alebo inovované spôsoby vizualizácie, pričom o týchto aktualizáciách a systémových zmenách nie sme zo strany poskytovateľa vopred informovaní.

Aby sme neporušili licenčné podmienky poskytnutia dát platformou Strava, prikkladáme niekoľko obrázkov vizualizácie dát (obr. 2 a 3) zo sekcie Map pripravených pomocou desktopovej aplikácie QGIS.



**Obr. 3.** Vizualizácie dát zo sekcie Map v QGIS – hexagóny H3 gridu úrovne 11 (cyklochodníky okolo Váhu v Žiline, august 2025) © ÚGKK SR

## 2. Atribútová štruktúra datasetu a dopravno-demografická granulácia dát

Pre potreby manažérskeho rozhodovania, plánovania a modelovania udržateľnej mobility v našom výskume je veľmi dôležité porozumieť sémantickému významu a vnútornej štruktúre dát poskytnutých v súboroch CSV (spomíname ich na konci predchádzajúcej časti). Importované zdrojové datasety v sebe integrujú komplexný súbor ukazovateľov, ktoré môžeme rozdeliť do troch základných kategórií:

1. Sietové a referenčné identifikátory: Každý zaznamenaný riadok v databáze je jednoznačne ukotvený pomocou unikátneho sieťového kľúča platformy Strava (edge\_uid). Z hľadiska multikriteriálneho posudzovania je dôležitý atribút



o type aktivity, ktorý umožňuje striktné oddeliť konvenčné bicykle od narastajúceho podielu e-bicyklov, prípadne sledovať ich kombinované toky. Časové hladisko a chronologický vývoj mobility v rokoch 2021 až 2026 zastrešuje časový parameter rok (resp. dátum). Z pohľadu prepájania systémov a dátovej interoperability je kritickým prvkom stĺpec s id úseku v OpenStreetMap, ktorý vytvára priamu väzbu na konkrétny líniový objekt v globálnej voľnej mape.

2. Smerové smerovanie a intenzita tokov: Na rozdiel od plošných hexagónov, líniová vrstva exportovaná z portálu (režim Edges) so sebou nesie neoceniiteľnú informáciu o smerovosti cyklistických prúdov. Systém okrem celkovej sumárnej záťaže konkrétneho úseku siete v oboch smeroch detailne rozlišuje smer pohybu cyklistov. Pomocou subkategórií s intenzitami pre smery jazdy je možné exaktne analyzovať asymetriu v ranných a popoludňajších špičkách, čo je kľúčové pre správne dimenzovanie širok cyklistických pruhov.
3. Socio-demografická a behaviorálna granúlacia: Najväčšiu pridanú hodnotu pre manažérske rozhodovanie a pochopenie motivácie cyklistov predstavujú agregované a plne anonymizované sociologické charakteristiky, ktoré systém Strava Metro generuje na základe profilov svojich používateľov. Tieto ukazovatele môžeme rozčleniť na tri subdomény:
  - (a) Účel realizácie cesty: Atribúty umožňujú jasne odseparovať utilitárne cesty (každodenné dochádzanie do práce, školy či za službami) od rekreačných, športových a turistických aktivít.
  - (b) Časová distribúcia počas dňa: Intenzity dopravy sú kvantifikované v štyroch explicitných časových blokoch. To umožňuje detailne monitorovať rannú dopravnú špičku, sedlo a skoré popoludnie, podvečerný návrat a rekreačný dopyt, ako aj minimálne nočné toky.
  - (c) Demografický a vekový profil: Pre exaktné socio-demografické štúdie dataset ponúka odhadované štruktúry zloženia cyklistov podľa pohlavia a štyri systematicky definované vekové kohorty v rozsahu od 18 do 65+ rokov.

Tento bohatý atribútový aparát transformuje surové geografické Big Data na plnohodnotný nástroj pre priestorovo-časové analýzy, pričom analytikovi umožňuje vytvárať komplexné dopyty a identifikovať špecifické cieľové skupiny používateľov cyklistickej infraštruktúry v záujmovom území.

### 3. Integrácia výskumného projektu Strava Metro do inžinierskej projektovej výučby

Spracovanie reálnych priestorovo-štatistických dát z globálnej platformy Strava Metro a s tým vyvolaná potreba vybudovať vlastné databázové úložisko vytvorili podmienky na zapojenie študentov inžinierskeho štúdia do projektu, v ktorom sa pracuje s reálnym zadáním, reálnymi dátami a jasným cieľom so spoločenskou

pridanou hodnotou. Týmto projektom sme sa pridali k vzdelávaniu, ktoré sa nazýva Challenge Based Education (CBE) [3].

Počas akademického roka 2025/26 sme tento projekt úspešne integrovali do dvoch zadaní našej fakultnej projektovej výučby, do predmetov Projekt 1 a Projekt 2 v našom inžinierskom štúdiu. V jednom zadaní, technicky zameranom, dvaja študenti boli poverení, aby vybudovali databázu a aby do nej importovali dáta stiahnuté z platformy Strava. Ich úlohou tiež bolo testovanie rôznych spôsobov importu dát, navrhovania agregácie dát medzi rôznymi úrovňami h3 gridov a podobne. V druhom zadaní mala študentka na starosti komunikáciu s poskytovateľom dát, spoločnosťou Strava, ako aj s mestom Žilina a samosprávnymi krajmi, cez ktoré Vážska cyklomagistrála prechádza. Jednou z jej úloh bolo získať podporné stanoviská potrebné na podanie prihlášky do programu Strava Metro for Academic Researchers. Podarilo sa jej získať podporné stanoviská z mesta Žilina a Žilinského samosprávneho kraja. Študentka zároveň pripravovala a sumarizovala dáta poskytnuté zo sčítačov cyklistov umiestnených na trase Vážskej cyklomagistrály. Kým vedúci projektu pracoval v prostredí QGIS, ona využívala ArcGIS Online na prípravu podkladov a máp určených na zdieľanie so samosprávnymi krajmi.

Vďaka nášmu projektu s dátami z platformy Strava boli naši talentovaní študenti postavení pred reálnu spoločenskú výzvu: navrhnúť metodiku a optimalizované dopyty pre monitoring cyklistickej dopravy v ich vlastnom regióne. V rámci tejto praktickej formy výučby študenti pracovali priamo v našom produkčnom databázovom prostredí. Táto forma vzdelávania priniesla obojstranný benefit – kým vedúci projektu získal inovatívne študentské pohľady a významnú pomoc pri svojom výskume, študenti si na reálnom datasete s miliónmi riadkov vyskúšali prácu s pokročilým PostGIS-om, kompiláciou nízkoúrovňových knižníc a prepínaním hierarchických úrovní gridu Uber H3. To ich výborne pripravilo pre prax v oblasti dátovej analytiky s priestorovými dátami.

## Záver

Naším príspevkom dokumentujeme prvých šesť mesiacov nášho výskumu dát poskytnutých programom Strava Metro for Academic Researchers. Podarilo sa nám vybudovať stabilnú serverovú infraštruktúru, vyriešiť technické obmedzenia a úspešne nainportovať masívne dáta obsahujúce časové rady intenzít cyklistov na Vážskej cyklomagistrále a jej blízkom okolí.

V najbližšom čase chceme zistiť, do akej miery sa údaje z platformy Strava Metro zhodujú s údajmi automatických sčítačov cyklistov. Tiež chceme zistiť, ktoré úseky Vážskej cyklomagistrály vykazujú najvýraznejšie priestorové alebo časové zmeny intenzity, vzhľadom na nové úseky sprístupnené verejnosti v Trenčianskom samosprávnom kraji a Žilinskom samosprávnom kraji v roku 2026.

Od júla 2026 vstupuje náš výskumný projekt do svojej druhej, čisto analytickej fázy, v ktorej máme v schválenom harmonograme naplánované nasledujúce strategické kroky:

1. Časovo-priestorová komparácia trendov: Realizácia dopytov zameraných na dlhodobé medziročné porovnanie intenzít (obdobie rokov 2021 až 2026) s cieľom exaktne vyhodnotiť, ako dobudovanie nových stavebných úsekov Vážskej cyklomagistrály reálne ovplyvnilo nárast cyklistickej dopravy v regióne.
2. Analýza izochrón a spádovosti: Využitie dát z režimu Origins&Destinations na modelovanie priestorovej dostupnosti a identifikáciu miest, kde cyklisti začínajú a kde končia svoje aktivity. Či to napríklad súvisí aj s rozmiestnením a dostupnosťou železničných staníc a zastávok, dostupnosťou kultúrnych pamiatok, parkovísk a podobne.
3. Príprava metodických podkladov pre samosprávy, ktoré poskytnú podporu príhláške projektu: Transformácia našich vizualizácií a databázových výstupov do podoby exaktných odporúčaní pre samosprávne kraje (Žilinský, Trenčiansky, Trnavský a Nitriansky) a mesto Žilina pre informované a zmysluplné investície do cyklodopravy, napríklad budovaním prístreškov, stojanov s nástrojmi pre ľahké opravy, informačných tabúl, rozvoja miest pre ubytovanie a občerstvenie.

Samozrejme, pri našom výskume a formulácii jeho výsledkov zohľadníme to, na čo upozorňuje odborná literatúra. Platforma Strava Metro je užitočný doplnkový zdroj údajov, ale údaje z nej sa nemajú bez kalibrácie interpretovať ako úplný obraz cyklistickej dopravy [1, 2, 10, 11]. Budeme brať do úvahy teda nasledovné limity údajov z platformy Strava Metro:

- používatelia Strava sú samovýberová vzorka,
- zastúpenie závisí od veku, pohlavia, typu cyklistiky, zariadenia a penetrácie aplikácie,
- rekreačné jazdy môžu byť nadreprezentované,
- nie všetky jazdy sú zaznamenané alebo verejne zdieľané,
- zmeny počtu používateľov Strava môžu vytvoriť trend aj bez zmeny celkovej cyklistickej dopravy,
- klasifikácia commute/leisure nie je priame pozorovanie účelu každej jazdy,
- hodnoty sú ovplyvnené ochranou súkromia a prahmi zobrazovania.

## Literatúra

- [1] FELCZAK, M. – FILICIAK, M.: *Mobile health cycling: How Eastern European amateur cycling enthusiasts frame their experiences with Zwift and Strava*, International Review for the Sociology of Sport, 2026, Vol. 61, no. 2, pp. 344–362, DOI: 10.1177/10126902251333566.

- [2] JEAN-LOUIS, G. – ECKHARDT, M. – PODSCHUN, S. – MAHNKOPF, J. – VENOHR, M.: *Estimating daily bicycle counts with Strava data in rural and urban locations*, Travel Behaviour and Society, 2024, vol. 34, p. 100694, DOI: 10.1016/j.tbs.2023.100694.
- [3] MALIEPAARD, I. A. J., 2023. *Student Agency in Challenge-Based Learning: A Case Study*. Student thesis. Utrecht University [online]. [cit. 2026-06-22]. Dostupné na: <https://studenttheses.uu.nl/server/api/core/bitstreams/cf23bf3a-2329-4543-b1eb-fad89be94c02/content>
- [4] PUCHER, J. – BUEHLER, R.: *Cycling towards a more sustainable transport future*, Transport Reviews, Vol. 37, no. 6, pp. 689–694, 2017, DOI: 10.1080/01441647.2017.1340234.
- [5] STRAVA: *Strava Announces Second Year of Metro Academic Program*, Strava Press, 2025, 10, [cit. 2026-06-22], <https://press.strava.com/articles/strava-announces-second-year-of-metro-academic-program>.
- [6] STRAVA METRO: *Metro for Academic Researchers*, 2026, [cit. 2026-06-22], <https://metro.strava.com/academics>.
- [7] STRAVA METRO: *Getting Started Guide*, Strava Metro Help Center, 2026, [cit. 2026-06-22], <https://stravametro.zendesk.com/hc/en-us/sections/360008086573-Getting-Started-Guide>.
- [8] UBER: *H3: Hexagonal hierarchical spatial index*, Open source geographic information system, 2026, [cit. 2026-06-22], <https://h3geo.org/>.
- [9] UBER ENGINEERING: *H3: Uber’s Hexagonal Hierarchical Spatial Index*, 2018, [cit. 2026-06-22], <https://www.uber.com/us/en/blog/h3/>.
- [10] VENTER, Z. S. – GUNDERSEN, V. – SCOTT, S. L. – BARTON, D. N.: *Bias and precision of crowdsourced recreational activity data from Strava*, Landscape and Urban Planning, 2023, Vol. 232, p. 104686, DOI: 10.1016/j.landurbplan.2022.104686.
- [11] WILLIAMS, S. — BEHRENDT, F.: *Data visibility of cyclists: social justice implications of Strava Metro data in transport planning.*, 2026, *Mobilities*, vol. 21, no. 2, pp. 381–394v DOI: 10.1080/17450101.2025.2542163.

## Kontaktné adresy

**doc. Ing. Peter Márton, PhD.**, Katedra matematických metód a operačnej analýzy, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovensko,  
*E-mailová adresa:* [peter.marton@uniza.sk](mailto:peter.marton@uniza.sk)

**doc. Ing. Marián Gogola, PhD.**, Katedra cestnej a mestskej dopravy, Fakulta prevádzky a ekonomiky dopravy a spojov, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovensko,  
*E-mailová adresa:* [marian.gogola@uniza.sk](mailto:marian.gogola@uniza.sk)

**Bc. Romana Vaščáková**, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovensko,  
*E-mailová adresa:* [vascakova1@stud.uniza.sk](mailto:vascakova1@stud.uniza.sk)

## RIEŠENIE SUDOKU GENETICKÝM ALGORITMOM NAPROGRAMOVANÝM POMOCOUI

TIBOR MENYHÉRT (SK)

**Abstrakt.** Článok opisuje opakovanú implementáciu genetického algoritmu (GA) na riešenie sudoku, tentoraz vytvorenú výhradne pomocou umelej inteligencie Gemini Pro v prostredí P5.js. Algoritmus využíva ostrovný model s viacerými metódami kríženia (sekciové, riadkové, stĺpcové, štvorcové a bodové), samoopravou genetického kódu, cieľenou mutáciou konfliktných buniek („Ostreľovač“) a hlasovaním generovanými „chimérami“, ktoré medzi ostrovmi efektívne šíria genetickú informáciu. Stagnáciu rieši mechanizmus kataklizmy. Program zvládne mriežky od  $9 \times 9$  po  $36 \times 36$ . Druhá časť článku zachytáva samotný proces programovania s UI – rýchlosť i rôzne úskalía, orezávanie dlhých odpovedí (vyriešené modularizáciou kódu), halucinácie, podlizovanie a riziká prílišnej dôvery v generované výstupy.

**Kľúčové slová.** Sudoku, genetický algoritmus, ostrovný model, kríženie, mutácia, fitness, P5.js, umelá inteligencia, Gemini Pro.

## SOLVING SUDOKU WITH A GENETIC ALGORITHM PROGRAMMED WITH AI

**Abstract.** The article describes a re-implementation of a genetic algorithm (GA) for solving Sudoku, this time created entirely with the help of the Gemini Pro artificial intelligence in the P5.js environment. The algorithm uses an island model with several crossover methods (section, row, column, square and point), self-repair of the genetic code, targeted mutation of conflicting cells (the “Sniper”), and voting-generated “chimeras” that efficiently spread genetic information between islands; stagnation is handled by a cataclysm mechanism. The program copes with grids from  $9 \times 9$  up to  $36 \times 36$ . The second part of the article captures the process of programming with AI itself — its speed as well as its pitfalls: truncation of long responses (solved by modularising the code), hallucinations, sycophancy, and the risks of over-relying on generated output.

**Keywords.** Sudoku, genetic algorithm, island model, crossover, mutation, fitness, P5.js, artificial intelligence, Gemini Pro.

## Úvod

V roku 2015 som naprogramoval riešenie problémov Sudoku genetickým algoritmom (GA) v programovacom jazyku PHP. Algoritmus bol vcelku úspešný, väčšinou problém vyriešil, ale narazil som na rôzne obmedzenia vyplývajúce zo zvoleného jazyka a parametrov hostingu:

- rýchlosť – skriptovací jazyk, zdieľané zdroje procesora s inými používateľmi,

- veľkosť vstupu – pôvodne som používal metódu Get,
- maximálna doba vykonávania skriptu v hostingu.

Niečo z toho som vyriešil, niečo obišiel, podrobnosti a program nájdete v článkoch na mojom blogu. Odkaz v literatúre na konci článku.

Keď ma môj priateľ Pavel Stríž pozval na túto konferenciu, ponúkol som mu moje vyučovacie programy vytvorené pomocou umelej inteligencie (UI) v prostredí P5.js. Počas komunikácie si Pavel spomenul, že som kedysi pomocou GA riešil Sudoku a navrhol mi, či by som to neskúsil znova naprogramovať pomocou UI. Myšlienka sa mi zapáčila a v článku popíšem, ako program funguje a ako prebiehal proces komunikácie s UI.

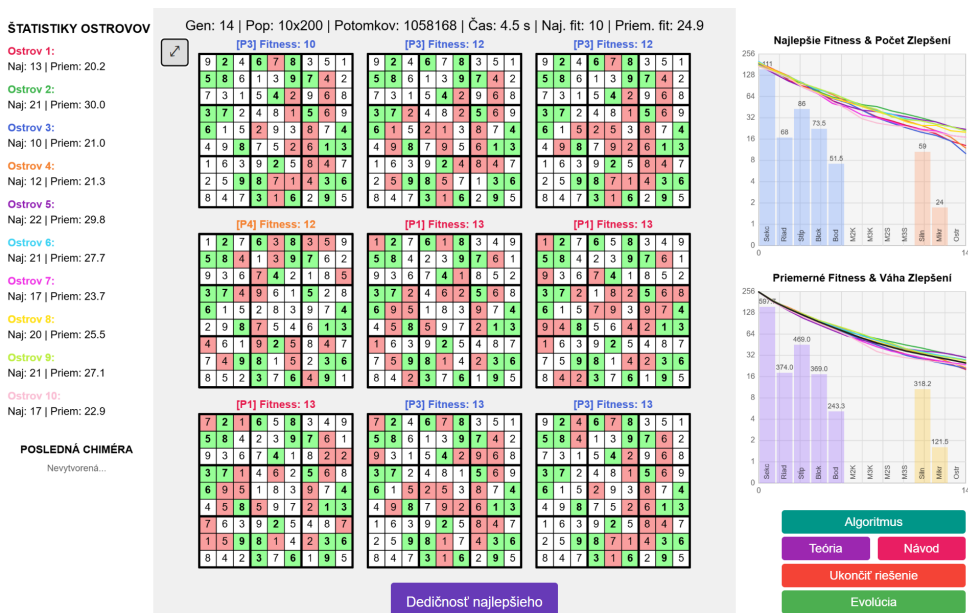
## 1. Popis programu

Po spustení sa zobrazí úvodná obrazovka (obr. 1). Zvolí sa veľkosť mriežky, problém (možno vybrať z piatich preddefinovaných, alebo klikaním na políčka zadať vlastný), veľkosť populácie na ostrove a počet ostrovov. Kliknutím na **Ukončiť a spustiť** sa spustí evolúcia. Predtým možno v menu **Konfigurácia evolúcie** zvoliť, ktoré druhy kríženia a mutácií sa majú počas evolúcie používať. Šípky vľavo hore zapnú fullscreen. Zvolil som obtiažnosť **Stredné**.

Obr. 1. Úvodná obrazovka

Priebeh evolúcie je zobrazený na obrázku 2. Vľavo sú štatistiky jednotlivých ostrovov, v strede sú najlepší jednotlivci v danej generácii (zelené políčka zadanie

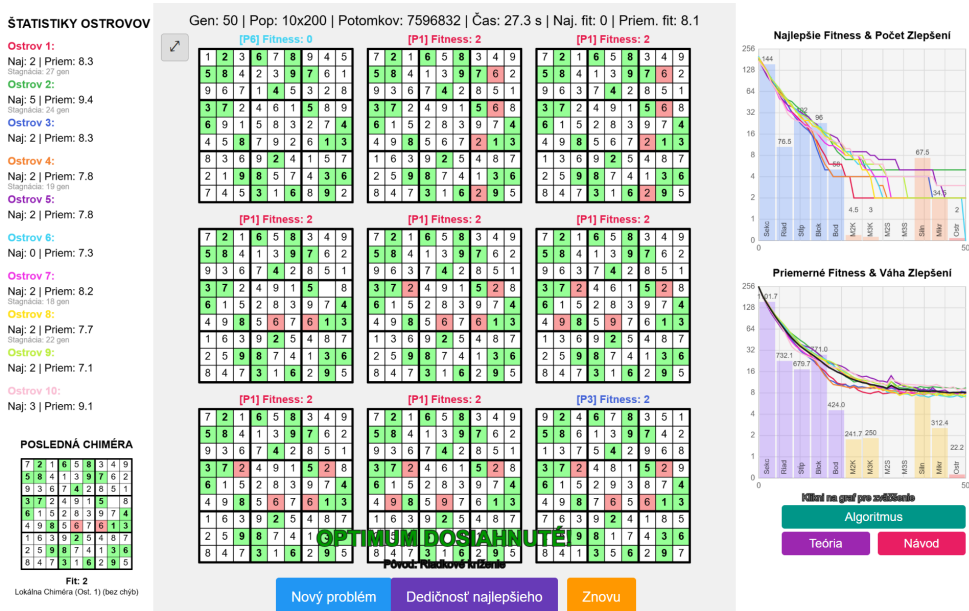
problému, červené v konflikte a biele bez konfliktu). Vpravo sú grafy priebehu evolúcie. Farebné čiary vyjadrujú každý ostrov samostatne, čierna je priemer za všetky ostrovy. Na hornom grafe sú najlepšie fitness a histogram vyjadruje ako často a ktoré metódy kríženia a mutovania viedli k zlepšeniu najlepšieho fitness na ostrove. Na dolnom grafe sú priemerné fitness a histogramy zobrazujú mieru zlepšenia, pričom zlepšenie z 1000 na 900 je vyhodnotené ako menší príspevok a zlepšenie z 2 na 0, ako mohutný príspevok.



Obr. 2. Priebeh evolúcie

Obrázok 3 vyjadruje stav, keď je problém vyriešený. Vľavo dole pribudla Chiméra (vysvetlím neskôr), v strede vľavo hore je riešenie problému, hore je vyhodnotenie (riešenie sa našlo po 50 generáciách, veľkosť populácie, počet potomkov, riešenie trvalo 27,3 sekundy). V strede dole si vmenu môžeme zvoliť nový problém, pozrieť sa, ako vznikol najlepší jedinec a spustiť ten istý problém znova.

Po rozkliknutí Dedičnosť najlepšieho sa nám zobrazia detaily: Vľavo hore Rodič 1, vpravo hore Rodič 2, ďalej grafické znázornenie, ktorá časť sa prebrala od koho a dole je výsledný potomok.



Obr. 3. Problém vyriešený

## 2. Popis algoritmu

Na každom ostrove sa náhodne vygeneruje nultá generácia jedincov, ktorí spĺňajú najzákladnejšiu požiadavku: v mriežke  $n \times n$  je  $n$  výskytov každého prípustného znaku. Každý jedinec je ohodnotený životaschopnosťou (fitness), celá generácia je usporiadaná od najlepšieho jedinca po najhoršieho. Jedinci sa náhodne skrížia (najlepší sa krížia častejšie), pričom sa používa niekoľko metód kríženia:

- sekciové – z 1. rodiča sa vyberie prvých  $m$  buniek a z 2. rodiča  $n \times (n - m)$  buniek,  $1 \leq m < n^2$ ;
- riadkové – z 1. rodiča sa vyberie 1 až  $m$  riadkov a z 2. rodiča  $n - m$  riadkov,  $1 \leq m < n$ ;
- stĺpcové – vyberajú sa stĺpce;
- štvorcové – vyberajú sa štvorce;
- bodové – vyberajú sa jednotlivé bunky.

Pri týchto kríženiach mohlo dôjsť k narušeniu požiadavky, aby sa každý znak vyskytol práve  $n$ -krát, potom dôjde k samooprave genetického kódu. Teraz sa na potomkov s istou pravdepodobnosťou uplatní mutácia: náhodná výmena niekoľkých buniek v potomkovi. Keď fitness klesne pod hodnotu 5, uplatňuje sa cieľená výmena konfliktných buniek, nazvali sme ju s UI *Ostreľovač*, prejde všetky



**Rodokmeň**

**Rodič 1 (Fit: 2)**

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 6 | 7 | 8 | 9 | 4 | 5 |
| 5 | 8 | 4 | 2 | 3 | 9 | 7 | 6 | 1 |
| 9 | 6 | 7 | 1 | 4 | 5 | 3 | 2 | 8 |
| 3 | 7 | 2 | 4 | 6 | 1 | 5 | 8 | 9 |
| 6 | 9 | 1 | 5 | 8 | 3 | 2 | 7 | 4 |
| 4 | 5 | 8 | 7 | 9 | 2 | 6 | 1 | 3 |
| 8 | 3 | 6 | 9 | 2 | 4 | 1 | 5 | 7 |
| 2 | 1 | 9 | 8 | 7 | 5 | 4 | 3 | 6 |
| 7 | 4 | 5 | 3 | 1 | 6 | 8 | 9 | 2 |

**Rodič 2 (Fit: 3)**

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 9 | 2 | 7 | 6 | 1 | 8 | 3 | 4 | 5 |
| 5 | 8 | 4 | 2 | 3 | 9 | 7 | 6 | 1 |
| 1 | 6 | 3 | 5 | 4 | 7 | 9 | 2 | 8 |
| 3 | 7 | 2 | 4 | 6 | 1 | 5 | 8 | 9 |
| 6 | 9 | 1 | 5 | 8 | 3 | 2 | 7 | 4 |
| 4 | 5 | 8 | 7 | 9 | 2 | 6 | 1 | 3 |
| 8 | 3 | 6 | 9 | 2 | 4 | 1 | 5 | 7 |
| 2 | 1 | 9 | 8 | 5 | 7 | 4 | 3 | 6 |
| 7 | 4 | 5 | 3 | 1 | 6 | 8 | 9 | 2 |

**Výsledný Potomok (Fit: 0)**  
(Riadkové kríženie)

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 6 | 7 | 8 | 9 | 4 | 5 |
| 5 | 8 | 4 | 2 | 3 | 9 | 7 | 6 | 1 |
| 9 | 6 | 7 | 1 | 4 | 5 | 3 | 2 | 8 |
| 3 | 7 | 2 | 4 | 6 | 1 | 5 | 8 | 9 |
| 6 | 9 | 1 | 5 | 8 | 3 | 2 | 7 | 4 |
| 4 | 5 | 8 | 7 | 9 | 2 | 6 | 1 | 3 |
| 8 | 3 | 6 | 9 | 2 | 4 | 1 | 5 | 7 |
| 2 | 1 | 9 | 8 | 5 | 7 | 4 | 3 | 6 |
| 7 | 4 | 5 | 3 | 1 | 6 | 8 | 9 | 2 |

Algoritmus

Teória

Návod

Späť na prehľad

Od Rodiča 1
  Od Rodiča 2
  Samooprava nuly
  Mutácia

Obr. 4. Dedičnosť najlepšieho ostrova

bunky, otestuje, či by došlo k zlepšeniu fitness pri zmutovaní, v kladnom prípade sa vytvorí mutant. Tiež si zvolí bunky, ktoré nie sú v konflikte, ale sú blízko konfliktnej bunky a urobí rovnaký proces. Ostrelovač sa okrem toho aplikuje každých 20 generácií na najlepšieho jedinca na ostrove.

Každých 30 generácií sa „hlasovaním“ vygenerujú na ostrovoch chiméry. Zistí sa, aký znak sa vyskytuje v bunke najčastejšie a vytvorí sa jedinec s najčastejšími výskytmi. Generujú sa dva druhy chimér: bezkonfliktná a konfliktná. V bezkonfliktnej sa v hlasovaní berú do úvahy iba bunky, ktoré nie sú v konflikte, ak bol v bunke konflikt so so všetkým, bunka ostane prázdna. Do konfliktnej idú aj bunky v konfliktoch. Hlasuje sa na každom ostrove zvlášť, prvých 20 jedincov na všetkých ostrovoch, prvých desať z každého ostrova. Ostrovná bunka sa zaradí na prvé alebo druhé miesto na svojom ostrove, globálna bunka sa zaradí na prvé alebo druhé miesto na všetkých ostrovoch. Pomocou globálnej chiméry sa šíri genetická informácia medzi ostrovmi.

Pôvodne sme posielali klony najlepších jedincov každého ostrova na každý ostrov, ale títo sa prejavovali ako „invazívny druh“, rýchlo zlikvidovali genetickú informáciu ostrova na ktorý boli nasadení, ak mali výrazne lepšie fitness, než bolo na hostiteľskom ostrove. Chiméry sa ukázali, ako veľmi efektívny nástroj, hoci nespĺňajú základnú požiadavku na počet znakov rovnakého druhu a obsahujú aj prázdne bunky. Na grafoch sa nasadenie chiméry prejavuje malým fitom

v priemernom fitness ostrovov, lebo obvykle majú horšie fitness, ako najlepší jedinec ostrova, ale často sa potom rýchlo vygeneruje jedinec s lepším fitness, než bolo na ostrove dovtedy.

### Výber do ďalšej generácie

Vytvorí sa zoznam, v ktorom sú všetky jedince z predchádzajúcej generácie a všetci potomkovia. Tento zoznam sa náhodne zamieša a utriedi sa podľa fitness (miešanie je dôležité, inak by jedinci s najlepším rovnakým fitness zostávali na svojich miestach; a jedinci, ktorí boli slepou uličkou, sa krížili prednostne). Do ďalšej generácie sa preberie 70 % najlepších jedincov a zo zvyšku sa náhodne vyberie 30 % jedincov.

### Kataklizma

Ak počas 100 generácií po sebe nedôjde k zlepšeniu najlepšieho fitness ostrova, prvých 10 jedincov sa zachová, zvyšní sú nahradení náhodne vygenerovanými jedincami. Ak má ostrov najhoršie priemerné fitness, vygenerujú sa všetci jedinci nanovo, okrem jedinca, ktorý je globálnym lídrom. Na grafoch sa to prejavuje výrazným fitom.

Tento genetický algoritmus rieši mriežky  $9 \times 9$  s populáciou 10 ostrovov po 200 jedincov obvykle do jednej minúty, mriežky  $16 \times 16$  tiež pomerne rýchlo s tými istými parametrami. Mriežky  $25 \times 25$  a  $36 \times 36$  trvajú výrazne dlhšie, u nich sme rýchlosť generovania novej populácie zrýchlili vtedy, ak sa vyskytol nový líder. Generovanie sme prerušili a išli na novú generáciu.

### Fitness

Počíta sa jednoducho, ak sa v riadku, stĺpci alebo štvorci vyskytne znak častejšie než mal, pripočíta sa +1, ak sa vyskytne nula, pripočíta tiež +1, ak je bunka v konflikte so zadáním pripočíta sa +5, ak sa znak vyskytuje globálne častejšie než je povolené tak tiež sa pripočíta sa +5.

## 3. Programovanie pomocou UI Gemini Pro

Vytvorenie prvej funkčnej verzie bolo skutočne rýchle, druhý deň potom, ako mi Pavel navrhol tento projekt, som mu poslal prvotné funkčné riešenie. Už som to kedysi naprogramoval sám, všetko som si nepamätal, ale vedel som čo chcem a ako skúsenejší programátor a analytik som vedel precízne formulovať prompty. Keď program narastal, začali vznikať problémy. Keď program dosiahne hranicu okolo 1000 riadkov, odpoveď býva useknutá, UI vygeneruje celý kód, ale bezpečnostný protokol kvôli zníženiu zaťaženia serverov, odpoveď oreže. Obvykle som chcel, aby UI generovala celý kód. Oveľa rýchlejšie bolo zmazať starý kód a vložiť nový, než pracne hľadať časti programu, ktoré mali byť nahradené, najmä, ak bolo potrebné vykonať zmeny na viacerých miestach.

Editor `P5.js` je primitívnejší, než Notepad, nie je to IDE prostredie, takže metóda `CTRL A`, `Delete`, kópia nového kódu `CTRL V` bola efektívnejšia. Ako asi viete, UI niekedy „halucinuje“. Urobí to čo chcete, ale často spraví aj to, čo ste nechceli. Odladené časti a funkcionality programu zmení alebo odstráni. Ak sa to vo vizuálnej podobe neprejaví, tak na to prídete niekedy až po troch či viacerých verziách a potom sa prácne hľadá, kde nastal problém. Väčšinou pomohlo do promptu UI napísať čo sa stalo, ona sa ospravedlnila a problém odstránila.

Problém s príliš veľkým kódom sme nakoniec vyriešili elegantne. Program sme zmodularizovali. Oddelili sme UI od výpočtov a väčšinou požadované zmeny sa prejavili iba v module UI alebo UI, respektíve som zadával prompty tak, aby bolo potrebné meniť jedno alebo druhé. Neskôr sme pridali ďalší modul (v programe je vysvetlený algoritmus), teória množstva permutácií Sudoku a množstva platných riešení a návod k programu. To sme najprv vytvorili ako textový súbor `texty.txt`, ale pri skopírovaní projektu na lokálny počítač, bezpečnostné pravidlá pre `javascript` nedovolili tento súbor čítať (inak by `javascript` mohol otvoriť ľubovoľný súbor vo vašom počítači). Takže vznikol modul `texty.js`.

Čo sa týka halucinácií, tak zatiaľ sa mi iba raz stalo, že halucinácia viedla k pokroku. UI som zadal úlohu a zrazu sa mi tam objavili histogramy. Tie som po nej chcel dávno predtým, ale stále s tým mala problém a keď som už na to rezignoval (tak snáď po desiatich promptoch), zrazu histogramy implementovala. Pre veľké mriežky trvá beh skriptu prídlho, tak som jej zadal úlohu, aby skript skonvertovala do `C++`. Riešenie pre najťažší problém našiel tento program za necelú sekundu, to pritom bežal online na serveri `onlinegdb.com`. Toto som zatiaľ odložil bokom, keďže by sa to nedalo spúšťať online, ale odhadujem, že veľké mriežky by boli riešené do 10 minút.

#### 4. Výhody, nevýhody a riziká programovania pomocou UI

Ak používateľ vie čo chce dosiahnuť, programovanie ide spočiatku rýchlo. To čo by neskúsený programátor nenaprogramoval vôbec a skúsenému by to trvalo dve tri hodiny, vygeneruje UI po správnych pokynoch do niekoľkých minút. S narastajúcou komplexnosťou programu UI začína halucinovať a niekedy by som si to rýchlejšie doplnil sám, než opakované upresňovanie zadania alebo dodatočné odhalenie, že nejakú odladenú časť kódu vypustila. Zásadne som však pracoval tak, aby všetok kód bol vygenerovaný UI a priamo som do kódu nezasahoval, s výnimkou modulu `texty.js`, aby tie texty boli skutočne zrozumiteľné a presné.

UI sa často podlizuje, každú chvíľu ma chválila: „Máš výborný postreh!“, „To je výborný nápad.“, „Máš absolútnu pravdu a sypem si popol na hlavu!“, „Je fascinujúce vidieť, ako sme to spoločnými silami dotiahli od základného konceptu až po profesionálne rozdelený, plne optimalizovaný kód.“ ap.

Slabšia povaha by mohla uveriť vo svoju genialitu a časom môže začať preferovať komunikáciu s UI pred komunikáciou s reálnymi ľuďmi aj v iných oblastiach, než je programovanie.

V programovaní, matematike, fyzike, ... som expert  
a tak obvykle ľahko odhalím halucinácie.

Ak by som komunikoval o oblastiach, v ktorých sa nevyznám,  
mohlo by sa stať,  
že uverím nezmyslom, ktoré UI vygeneruje.

## Literatúra

- [1] MENYHÉRT, T.: *Sudoku a genetické algoritmy*, <https://boro.blog.pravda.sk/2015/10/06/sudoku-a-geneticke-algorytmy/>.
- [2] MENYHÉRT, T.: *Sudoku (3), Zamrznutá evolúcia a inteligentný dizajnér*, <https://boro.blog.pravda.sk/2015/11/17/sudoku-3-zamrznuta-evolucia-a-inteligentny-dizajner/>.
- [3] PROGRAM SUDOKU GA: [evyuka.sk/sudoku/](http://evyuka.sk/sudoku/).
- [4] UI GEMINI: *Článok, ktorý bol vygenerovaný UI Gemini, na základe vyhodnotenia celého nášho četu*, <https://docs.google.com/document/d/12ocW3F0GFMrbD0XY7weLXqNmchNstRBKm12MLho81V8/edit?usp=sharing>.

## Kontaktná adresa

**RNDr. Tibor Menyhért**, Súkromná základná škola, Galaktická 9, 040 12 Košice, Slovensko,  
*E-mailová adresa:* [otm@bridgekosice.sk](mailto:otm@bridgekosice.sk).

## **TVORBA INTERAKTÍVNYCH DIGITÁLNYCH POMÔCOK POMOCOUI GEMINI PRO**

TIBOR MENYHÉRT (SK)

**Abstrakt.** Príspevok popisuje vývoj a funkcionality troch aplikácií v prostredí P5.js (Matematický tréner, Súradnice, Slovoje), ktoré boli navrhnuté a na mieru potrebám žiakov zo sociálne znevýhodneného prostredia a marginalizovaných rómskych komunít a boli prakticky odskúšané na ZŠ Galaktická 9, Košice. Práca analyzuje didaktický potenciál týchto nástrojov: od parametrizácie úloh (operácie, čas, typy čísel) cez prácu s priamou vizuálnou spätnou väzbou (autokorekcia) až po inkluzívny rozmer (plánovaný rómsky slovník v hre Slovoje). Autor kriticky hodnotí psychologické aspekty gamifikácie v triede a navrhuje riešenia pre zapojenie slabších žiakov. Záver je venovaný technickej reflexii kooperácie s Gemini Pro. Autor popisuje proces iteratívneho kódovania, limity veľkosti odpovedí (cca 1000 riadkov) a riziká straty funkcionality pri komplexných zmenách. Príspevok ponúka cenné praktické odporúčania, ako sú dôležité modularizácie kódu a presná formulácia promptov, ktoré vedú k zefektívneniu vývoja digitálnych učebných pomôcok v pedagogickej praxi.

**Kľúčové slová.** P5.js, Gemini Pro, umelá inteligencia, inkluzívne vzdelávanie, rómski žiaci, gamifikácia, matematika, digitálne pomôcky, modularizácia kódu.

## **CREATING INTERACTIVE DIGITAL AIDS WITH AI GEMINI PRO**

**Abstract.** The paper describes the development and functionality of three P5.js applications (Maths trainer, Coordinates, Word game), which were designed and tailored to the needs of pupils from socially disadvantaged backgrounds and marginalised Roma communities, and were tested in practice at the Galaktická 9 primary school in Košice. The work analyses the didactic potential of these tools: from the parametrisation of tasks (operations, time, number types), through working with direct visual feedback (self-correction), to the inclusive dimension (a planned Roma dictionary in the Slovoje game). The author critically assesses the psychological aspects of gamification in the classroom and proposes solutions for engaging weaker pupils. The conclusion is devoted to a technical reflection on the cooperation with Gemini Pro. The author describes the process of iterative coding, the limits on response size (roughly 1000 lines), and the risks of losing functionality during complex changes. The paper offers valuable practical recommendations, such as modularising the code and precisely formulating prompts, which lead to more efficient development of digital teaching aids in pedagogical practice.

**Keywords.** P5.js, Gemini Pro, artificial intelligence, inclusive education, Roma pupils, gamification, mathematics, digital teaching aids, code modularisation.

## Úvod

Vo svojej diplomovej práci pred takmer 40 rokmi som vytvoril niekoľko výukových programov v programovacom jazyku `Basic` pre počítače Sord a PMD, po skončení školy na Ústave výpočtovej techniky UPJŠ pre PP02 a neskôr pre PC-čka. Aj neskôr som sa zaoberal programovaním hier a výukových programov, hlavne so zameraním na bridž, keďže som školiteľom a trénerom bridžu a niekoľkonásobným Slovenským bridžovým ligovým majstrom Slovenska.

Zaujal ma webinár pani Tkáčovej, kde nás oboznámila s prostredím `P5.js`. Výhodou prostredia je, že žiaci hneď vidia aj úplne drobné zmeny v programe a keď sa pomýlia, tak často vznikajú zaujímavé grafické efekty, často pútavejšie, než pôvodné zadanie učiteľa. Prvé čo som vyskúšal naprogramovať pomocou umelej inteligencie v `P5.js` bol bridžový licitačný trenažér. Keďže väčšina z vás bridž nehraje, tak sa budem venovať iným výukovým programom.

## Matematický tréner

Učím na súkromnej základnej škole, všetci naši žiaci sú Rómovia. Často sú očarujúci svojou spontánnosťou, a inokedy učiteľ ťažko udrží hnev a utrži sklamanie. Väčšina z nich neovláda malú násobilku dokonca ani v ôsmej či deviatej triede a aj sčítanie a odčítanie im robí problémy. Podobné to je aj na bežných školách s prevahou majority, ale nie je to až tak výrazné.

Drilovanie neustáleho počítania na papieri či pri tabuli ich rýchlo omrzí. Raz sme mali alternatívny vyučovací deň a na internete som našiel program na precvičovanie matematických operácií. Web stránku som premietal na interaktívnu tabuľu a žiaci pomocou bezdrôtovej myši klikali na ponúkané odpovede. Vyskúšali sme aj dotyk rukou na interaktívnu tabuľu, ale bola zle nakalibrovaná, takže program často správnu odpoveď nezaznamenal. Konečný výsledok som napísal fixou na bočnú tabuľu s menom a bodmi žiaka. Každý mal tri pokusy a traja finalisti išli do finále na jeden pokus. Nevýhodou programu bolo, že bol často prerušovaný reklamnými videami. Rozhodol som sa čosi podobné, ale lepšie vyhovujúce použitiu na hodinách matematiky, naprogramovať s pomocou UI Gemini Pro. Výsledný program sa nachádza na adrese [evyuka.sk/mattrener/](http://evyuka.sk/mattrener/).

**Výhodou** oproti pôvodnému programu je, že ho učiteľ môže parametrizovať:

- zoznam generovaných operátorov a operácií: plus, mínus, násobenie, delenie, zlomky, percentá, mocniny, odmocniny, sčítanie, odčítanie, násobenie a delenie zlomkov,
- čas na úlohu,
- percentuálne vyhodnotenie úspešnosti aj po jednotlivých operáciách, žiak aj učiteľ sa dozvedia, čo by bolo potrebné precvičovať,
- automatický rebríček hráčov.

Automatický rebríček hráčov je síce rýchly, ale písanie fixkou na tabuľu malo svoje čaro. Najprv som výsledky písal ja, potom si svoj výsledok zapisoval žiak

sám a zažíval pocit úspechu, keď si zlepšil osobný rekord či dokonca niekoho predbehol. Zažívajú to síce aj pri automatickom rebríčku, ale je to menej intenzívne.

### Nevýhody:

- Žiaci sa dožadujú hrania matematických hier na každej hodine a nechcú sa učiť klasicky (nechceli sa ani predtým).
- Vo väčších triedach sa zapája iba časť žiakov a ani sa nestihnú všetci vystriedať, riešením je rozdeliť ich na malé skupinky okolo 4–6 žiakov, rozdať im tablety nastavené na kiosk (aby nemohli spúšťať iné programy). Pre víťazov skupín urobiť finále na interaktívnej tabuli.
- Žiaci, ktorým to ide a vlastne nepotrebujú precvičovať sa zapájajú intenzívnejšie a pokúšajú sa uzurpovať si hranie a tí, ktorým to nejde a precvičovať matematické operácie by potrebovali, sa hranu vyhýbajú. Môže to byť frustráciou z toho, že majú výrazne horší výsledok než ostatní. Riešením je opäť tablet, v tomto prípade vo dvojiciach približne rovnako úspešných žiakov.

### Súradnice

Žiakom často robí problém odčítať údaje z grafov. Napísal som program na tréning práce s pravouhlou súradnicovou sústavou. Nájdete ho na adrese [evyuka.sk/suradnice/](http://evyuka.sk/suradnice/).

Program pracuje v dvoch módoch:

- Vygenerujú sa súradnice bodov a žiak ich zakreslí do mriežky a body navzájom pospája v poradí ABC... Kontrola zobrazí skutočné pozície bodov a úsečky, ktoré ich spájajú.
- Zobrazia sa body a žiak fixkou zapíše súradnice bodov. Kontrola body so správnymi súradnicami zakrúžkuje zelenou kružnicou a nesprávne súradnice zobrazí ako červené krížiky.

Program pracuje so žiackou chybou, netrestá ju. Žiak vidí, že ak by to bolo tam, ako napísal, tak by to bolo niekde úplne inde. Kreslenie úsečiek voľnou rukou a následné porovnanie s „ideálnou“ tiež vedie ku autokorekcii. Vyskúšal som v 7. až 9. ročníku. Znova sa zapájali hlavne šikovnejší žiaci, resp. slabších som musel nútiť prísť k tabuli.

Program má tieto parametre:

- počet generovaných bodov,
- kladné čísla aj s nulou, kladné aj záporné čísla.

### Slovboj

Už v minulosti som naprogramoval hru Slovboj v AppInventore. Rozhodol som sa to znovu naprogramovať v prostredí P5.js pomocou UI a išlo to skutočne rýchlo. S podobnými programami ste sa určite stretli, ale tiež boli plné reklám a neuznávali mnohé spisovné či bežne používané slovenské slová a akceptovali mnohé nezmyselné slová, ak sa hráč preklepol.

Výhodou programu je, že hráči môžu slovník dopĺňať, pomocou reklamácie neuznaných slov aj reklamácie uznaných slov. Samotné zaradenie/vyradenie do/zo slovníka vykonáva editor. Keďže učím na rómskej škole, plánujem doplniť róm-sky slovník a pomerne jednoducho by sa dali doplniť aj o iné jazyky. Ale chce to editorov, ktorí dané jazyky perfektne ovládajú.

Písmená majú rôznu bodovú hodnotu, podľa frekvencie výskytu v slovenčine a červené a modré písmená, ktoré hodnotu slova vynásobia 3 a 2 krát. Slovenčinu neučím, ale je tam aj stratégia využívania prémií, takže som ju niekoľkokrát použil na hodinách matematiky od 6. po 9. triedu. Počas noci s Andersenom si hru vyskúšali aj žiaci prvého stupňa. Žiaci si navzájom pomáhali, jeden hral, ostatní mu radili. Ak im program slovo neuznal, vysvetlil som im gramatickú chybu (mäkké i a tvrdé y, dĺžne, mäčkene, spodobovanie slov, ap.). Slovo je na adrese [evyuka.sk/slovboj/](http://evyuka.sk/slovboj/).

Program má rebríček dosiahnutých výsledkov, vygenerovanie nových písmen, vytvorenie nového hráča, zmazanie hráčov a konfigurácie zvuk zapnutý/vypnutý, učenie zapnuté/vypnuté. Deštrukčné akcie sú chránené heslom.

## Skúsenosti s tvorbou výukových programov s Gemini Pro

Ak človek vie, čo chce naprogramovať a svoje požiadavky vie presne naformulovať, spočiatku ide programovanie veľmi rýchlo. Keď je už projekt komplexný, ďalšie zmeny sú komplikovanejšie. Gemini nie je súčasťou prostredia P5.js, obvykle požadujem vygenerovať celý kód s novými požiadavkami a starý kód zmažem a pomocou CTRL C a CTRL V preniesim nový. Pri generovaní celého kódu sa naráža na limit veľkosti odpovede, ktorý je okolo 1000 riadkov. Keď mi generovala iba zmeny v jednotlivých procedúrach, pomerne ťažko sa hľadalo, ktorú časť kódu mám zmazať a nahradiť novým.

Ak by UI bola integrálnou súčasťou prostredia, tak by to implementovanie zmien bolo efektívnejšie. Pokiaľ zadám požiadavku na príliš mnoho rôznorodých zmien naraz, tak ich síce spraví, ale často zmení či dokonca úplne odstráni niekedy aj kľúčovú funkcionality. Najhoršie je, ak znefunkční „drobnosť“, ktorú si človek všimne až po dlhšom čase. v jednom z projektov to bolo až otravné. Pri rozsiahlejších projektoch sa mi osvedčilo rozdeliť projekt na moduly, aj sa v tom ľahšie hľadá, aj UI menej halucinuje. Inak mi Gemini vyhovuje, vyskúšal som ChatGpt, Deepseek a mnohé iné. Gemini je zatiaľ pre mňa najlepšia. ChatGPT však dokázal vygenerovať aj niekoľko tisíc riadkový kód, podhodil som mu takmer hotový projekt, ktorý mal iba upraviť.

## Kontaktná adresa

**RNDr. Tibor Menyhért**, Súkromná základná škola, Galaktická 9, 040 12 Košice, Slovensko,  
E-mailová adresa: [otm@bridgekosice.sk](mailto:otm@bridgekosice.sk).



## SEZNÁMENÍ S OpTeXem

PETR OLŠÁK (CZ)

**Abstrakt.** Tento článek chce v krátkosti ukázat, že existují i jiné TeXové formáty než L<sup>A</sup>T<sub>E</sub>X, konkrétně OpTeX [1]. Jeho možnosti jsou s L<sup>A</sup>T<sub>E</sub>Xem srovnatelné, ale je daleko efektivněji napsaný a umožňuje psát vstupní soubory v mnohem příjemnější syntaxi a také umožňuje programovat sazbu přímočařeji za použití přímých nástrojů LuaTeXu. Článek se pokusí krátce seznámit čtenáře s OpTeXem.

**Klíčová slova.** OpTeX, TeXová makra, jazyk Lua.

### GETTING TO KNOW OpTeX

**Abstract.** This article wants to briefly show that there are other TeX formats than L<sup>A</sup>T<sub>E</sub>X, specifically OpTeX [1]. Its capabilities are comparable to L<sup>A</sup>T<sub>E</sub>X, but it is much more efficiently written and allows you to write input files in a much more pleasant syntax and also allows you to program the typesetting more straightforwardly using the direct tools of LuaTeX. This article will briefly introduce the reader to OpTeX.

**Keywords.** OpTeX, TeX macros, Lua programming language.

## 1. Jak to vzniklo

S TeXem jsem se seznámil na počátku 90. let a důkladně a několikrát jsem přečetl TeXbook. Připadalo mi účelnější pracovat v plainTeXu, protože do mého prvního domácího počítače s malou pamětí se L<sup>A</sup>T<sub>E</sub>X nevešel, ale plainTeX tam běžel jak z praku. Později jsem zaznamenal také plno problémů s L<sup>A</sup>T<sub>E</sub>Xem [2], takže jsem nejen u plainTeXu zůstal, ale taky jsem pro něj začal vyvíjet českou a slovenskou podporu *CSplain* [3]. Tuto podporu mám pod palcem dodnes, ale už mnoho let je verze *CSplainu* fixovaná a změny v ní v podstatě neprobíhají.

Vytvořil jsem si během desítek let doma vlastní makra, která rozšiřují možnosti plainTeXu. V roce 2013 jsem se rozhodl tato makra zveřejnit pod jménem OPmac [4]. Stala se součástí balíčku *CSplain* a používá se pomocí `\input opmac` na začátku dokumentu. Fungovalo to pěkně, ale mělo to dvě nevýhody: technickou dokumentaci jsem k tomu měl (ještě z doby domácího použití) jen v jazyce, který nemá mezinárodní přesah, a dále jsem musel v makrech řešit existenci všelijakých kódování fontů a všelijakých specifik různých TeXových mašin (pdfTeX, XeTeX, LuaTeX). To komplikovalo kód a také znepráhledňovalo použití.

V roce 2020 (v období, kdy jsme byli nuceně připoutáni jen k domácím počítačům) jsem se proto rozhodl práci na OPmac zastavit a udělat to kompletně na zelené louce znovu a s podporou jediného kódování (unicode) a jediné TeXové

mašiny (LuaTeX). Takže jsem zahodil všechna starodávná řešení (*CSplain*, *dvi* výstup, *PostScript*, různé obskurní kódování a ještě obskurnější fonty) a zaměřil jsem se jen na „moderní TeX“ postavený výhradně na unicode a na spojení Lua s TeXem. Tak vznikl OpTeX [1].

Nelituji toho, třebaže jsem měl do té doby obrovské množství vlastních dokumentů z minulých dob opírající se jen o *CSplain*. Od roku 2020 jsem pro své nové dokumenty *CSplain* nikdy nepoužil a práce na dokumentech se pro mě stala daleko snadnější a pohodlnější.

Poznamenávám, že L<sup>A</sup>TeXoví vývojáři si takové osvobozující zahození starých věcí nemohou dovolit, protože uživatelská základna L<sup>A</sup>TeXu je velmi konzervativní. Kdyby například přestali podporovat starodávný a ne-unikódový pdfL<sup>A</sup>TeX, přišli by asi o velké množství uživatelů, protože desítky let se věci v různých redakcích dělají pomocí prastarých L<sup>A</sup>TeXových šablon, ve kterých se dnes už nechce nikomu vrtat a které předpokládají, že se vývoj TeXu zastavil v devadesátých letech. Je příznačné, že v mnoha uživatelských prostředích je stále starodávný pdfL<sup>A</sup>TeX implicitní nabídkou, čím zpracovat dokument. Že tady existuje moderní TeXová mašina LuaTeX a obrovské množství fontů v moderním formátu OpenType už více než deset, to si asi někteří vývojáři uživatelských prostředí vůbec nevšimli.

## 2. Formáty a TeXové mašiny

OpTeX je TeXový formát. Slovo „formát“ je historické označení pro sadu TeXových maker a nastavení registrů, které TeXová mašina načte jen jednou při své instalaci a uloží si z toho obsah své paměti do binárního souboru `.fmt`. Ten pak opakovaně používá při zpracování různých dokumentů. Typickými příklady takové sady maker jsou `plainTeX`, `CSplain`, `LATeX`, `ConTeXt` nebo `OpTeX`.

Každý formát běží s nějakou konkrétní TeXovou mašinou (TeX engine). Typickými mašinami jsou TeX (to dnes už asi nikdo nepoužívá a v distribucích to máme jen z úcty k původnímu tvůrci TeXu), dále pdfTeX, XeTeX, LuaTeX. Poslední dvě jmenované mašiny podporují unicode. Makra L<sup>A</sup>TeXu z důvodů konzervatismu uživatelů udržují součinnost s jakoukoli z uvedených TeXových mašin (i starodávných, například parních), a proto máme L<sup>A</sup>TeX, pdfL<sup>A</sup>TeX, XeL<sup>A</sup>TeX, LuaL<sup>A</sup>TeX. *CSplain* také běží s kteroukoli mašinou (příkazy `csplain`, `pdfcsplain`, `xecsplain`, `luacsplain`). Dále dnešní verze ConTeXtu je vyvíjena se zcela odlišnou mašinou LuaMetaTeX a konečně OpTeX běží výhradně s LuaTeXem<sup>1</sup>.

## 3. Přehled základních vlastností OpTeXu

Filosofickým základem OpTeXu je dělat věci jednoduše a přímočaře, podobně jako `plainTeX`. Takže se vyhýbá zbytečně komplikovaným konstrukcím, které známe především z L<sup>A</sup>TeXu. S `plainTeXem` je OpTeX v zásadě kompatibilní, ale umí toho daleko více. Je označován jako „moderní `plainTeX`“.

<sup>1</sup> Přesněji s LuaHBTeXem (stejně jako dnešní LuaL<sup>A</sup>TeX), což je mašina umožňující navíc rendrování textu i s méně běžnými OpenType fonty (používá k tomu knihovnu harfbuzz).

- OpTeX obsahuje makra na pohodlnou práci s unicode fonty (fontový systém) včetně kompletní unicode-math, což souvisí s dnešní nabídkou velkého množství matematických unicode fontů. O ní se T<sub>E</sub>Xovými uživateli devadesátých let ani nesnilo. Tehdy měli v podstatě jedinou možnost Computer Modern a byli rádi, že jsou rádi.
- Přepínání barev je implementováno pomocí LuaT<sub>E</sub>Xových atributů, takže změny barev se chovají zcela přirozeně, respektují T<sub>E</sub>Xové skupiny, správně přecházejí na další stranu a nezanechávají v sazbě žádná `\specials`, která jinak dokáží programátora maker poměrně hodně potrápit.
- Na sestavení rejstříků a seznamů literatury přímo z `.bib` souborů nepotřebuje OpTeX žádný externí program. Vše udělá sám ve vlastní režii. Nemusíte tedy myslet na to, že po změně dokumentu je třeba volat externí program ani netřeba do makefile přidávat volání externího programu. Také obarvení výpisů kódů z počítačových programů probíhá bez externího programu.
- Pomocí `\slides` přechází OpTeX do režimu psaní slajdů pro zpětný projektor. Když se podíváte do zdrojáku nějaké slajdové prezentace napsané OpTeXem a porovnáte to např. se zdrojákem při použití L<sup>A</sup>T<sub>E</sub>Xového balíčku `beamer`, shledáte, že to je nebe a dudy. V OpTeXu uživatel skutečně píše *obsah* těch slajdů a nezdržuje se jejich *programováním* ve smyslu neustálého a vyčerpávajícího psaní `\begin{cosi}`, `\end{cosi}`.
- OpTeX se nesnaží zavádět novou uživatelskou syntaxi a zakrývat tím syntaxi samotného T<sub>E</sub>Xu. Předpokládá se běžná plainT<sub>E</sub>Xová syntaxe. Například, chceme-li nastavit šířku sazby v odstavci, přiřadíme k tomu odpovídající hodnotu do odpovídajícího T<sub>E</sub>X-primitivního registru takto: `\hsize=10cm`. Porovnejme to s L<sup>A</sup>T<sub>E</sub>Xovým `\setlength{\textwidth}{10pt}`.
- Kdo umí základy T<sub>E</sub>Xu na jeho primitivní úrovni, ten se dokáže v makrech OpTeXu rychle orientovat, neboť jsou napsána co nejjednodušeji a nejpřímočařeji. Pro ostatní je připraven krátký dokument [5] na seznámení s T<sub>E</sub>Xem.
- OpTeX nepotřebuje dodatečné volání balíčků maker, pokud chceme vytvořit běžný dokument včetně hyperlinků, barev, fontů, obrázků atd. Vše je přímo součástí formátu. Přesto vygenerovaný formátový soubor má cca 750 Kb, zatímco třeba L<sup>A</sup>T<sub>E</sub>Xový formátový soubor má přes 5,5 Mb a to neobsahuje `hyperref` ani `xcolor`, `graphicx`. I to dokládá, že makra OpTeXu jsou daleko úspornější než makra L<sup>A</sup>T<sub>E</sub>Xu.
- Nastavení vzhledu sazby uživatelem není typicky podporováno pomocí děsného množství dodatečných parametrů, které akorát prodlužují dokumentaci a také implementaci maker samotných. Místo toho se od uživatele předpokládá: chceš-li změnit výchozí chování OpTeXu, vem si jeho makro, zařaď si je do své sady maker a pozměň si je k obrazu svému (tj. předefinuj je). Používej k tomu zejména T<sub>E</sub>X-primitivní konstrukce, jako to dělají původní makra OpTeXu.
- Stovky dodatečných a specifických vlastností občas potřebných pro práci s dokumentem jsou prezentovány na stránce tzv. OpTeX triků [6]. Každý trik tam je stručně popsán, co to umí, a dále následuje jeho implementace, což je kolem

deseti řádků kódu na jeden trik. Jednotlivé triky řeší zadání v rozsahu, na které v  $\text{\LaTeX}$ u typicky existují externí balíčky. To ukazuje, že mnoho  $\text{\LaTeX}$ ových balíčků lze v  $\text{\OpTeX}$ u nahradit cca desetiřádkovými kusy kódu. Ukázky jsou v sekci 5.

- I  $\text{\OpTeX}$  umožňuje psát rozsáhlejší balíčky maker a zavést je do dokumentu pomocí makra `\load`. Programátoři maker balíčků ocení, že jmenný prostor pro řídicí sekvence v jejich balíčcích je snadno oddělen od jmenových prostorů dalších balíčků. Přitom netřeba psát do kódu žádné zoufalé identifikátory obsahující `@` ani pořád dokola opakovat v každém identifikátoru název jeho jmenového prostoru (jako to je v `Expl3`). Jednoduše na začátek balíčku napíšeme `\_namespace{jméno}` a ukončíme to `\_endnamespace`.  $\text{\OpTeX}$  také odděluje svá interní makra od maker, která si může definovat uživatel. Interní makra  $\text{\OpTeX}$ u mají počáteční podtržítka v názvu makra. Podtržítka má v  $\text{\OpTeX}$ u trvale nastavenou kategorii 11 (písmeno), takže identifikátory s podtržítkem jsou přímo přístupné bez nutnosti přepínat do jiného režimu kategorií pomocí `\makeatletter` či podobných věcí.
- Z  $\text{\LaTeX}$ u znáte asi patálii s tzv. „křehkými příkazy“, tj. makry, které se v pracovních souborech rozsypou a způsobí havárii. Nic takového v  $\text{\OpTeX}$ u neexistuje, ten do pracovního souboru zapisuje v režimu, kdy se nikdy nic nesyne, prostě v tu chvíli potlačí veškerou expanzi.
- $\text{\OpTeX}$  nabízí dokumentovat kódy maker pomocí dvojice `\_doc` a `\_cod`. Tímto nástrojem je vytvořena uživatelská a technická dokumentace  $\text{\OpTeX}$ u. Je na jediném místě v jediném PDF souboru a je vzájemně bohatě prolinkovaná.
- Také balíčky může programátor maker dokumentovat přímo pomocí `\_doc` a `\_cod`. Navíc za `\_endcode` (což je ekvivalent `\endinput`) je možné přidat podrobnou uživatelskou dokumentaci, která je tedy společně s makry balíčku v jediném souboru. Po `optex -jobname balicek '\docgen balíček'` se vygeneruje úhledně zformátovaná uživatelská i technická dokumentace k balíčku. Netřeba používat `docstrip` a oddělovat dokumentaci od maker samotných. Spíš naopak, chceme mít všechno pohromadě, pokud možno na jediném místě.
- Programátoři maker mohou kromě maker `plainTeX`u a primitivních příkazů  $\text{\TeX}$ u využít též další podpůrné nástroje, které nabízí  $\text{\OpTeX}$ . Mezi ně patří makra na cykly, které umějí být vnořeny a jsou plně expandovatelné, dále makra na vyhodnocení různých podmínek na větvení podle případů a mnoho dalších.
- $\text{\OpTeX}$  podporuje desítky vzorů dělení všech jazyků, které k tomu mají data v  $\text{\TeX}$ ové distribuci. Také podporuje zvláštnosti některých jazyků v souboru `lang-data.opm`.
- V  $\text{\OpTeX}$ u je možné kromě běžné sazby zleva doprava přepnout na sazbu zprava doleva (hebrejšтина, otestováno) nebo na sazbu se znaky kladenými shora dolů.

## 4. Příklady jednoduchého dokumentu

Nejdříve uvedeme minimální příklad:

```
\fontfam[Termes]
Ahoj, světe!
\bye
```

Vidíme, že byla zvolena rodina fontů pro dokument (zde Termes, to je svobodná alternativa k Times), dále je uveden text dokumentu a celé je to zakončeno `\bye`. Deklaraci rodiny fontů můžeme vynechat. Pak se ovšem použije implicitní rodina obsažená ve formátu a ta z technologických důvodů nemůže obsahovat unicode fonty, ale jen starodávné klasické T<sub>E</sub>Xové 8bitové fonty. To se projeví tak, že nebude fungovat znak `ě` a budeme o tom varováni na terminálu a v logu. Implicitní rodinu z formátu můžeme tedy akorát použít na soukromé testování maker. Ale pro skutečný dokument potřebujeme kvalitní rodinu unicode fontů deklarovanou pomocí `\fontfam`. Tato deklarace zahrnuje i kompletní zavedení unicode-math. V případě Termes tedy bude použit font s touto rodinou související: `texgyretermes-math`. Můžeme si o tom přečíst zprávu na terminálu a v logu.

Na rozdíl od interaktivních programů na přípravu sazby uživatel T<sub>E</sub>Xu typicky nemůže vyrolovat nabídku fontových rodin, které může v dokumentu použít. Tento hendikep OpT<sub>E</sub>X řeší tak, že umožní vygenerovat fontový katalog pomocí `\fontfam[catalog]`. Na zhruba 26 stránkách pak uživatel vidí přes tisíc vzorků fontů z desítek fontových rodin většiny volně dostupných fontů instalovaných v T<sub>E</sub>Xové distribuci [7]. A vidí tam také, jaký příkaz `\fontfam` má použít a jaké má pak k dispozici fontové přepínače, kterými získá konkrétní variantu fontu ve vybrané rodině.

Dokument `op-demo.tex` nabízí další jednoduchý dokument, který demonstruje použití OpT<sub>E</sub>Xu v typickém dokumentu. Ve výrazně zkrácené verzi vypadá takto:

```
\fontfam[LMfonts]           % Rodina fontů je Latin Modern
\margins/1 a5 (1,1,1,1.4)cm % Nastavení formátu A5 s okraji 1 cm
\typosize[9/10.5]           % stupeň písma 9pt font / 10.5pt baselineskip
\parindent=10pt             % nastavení parametrů sazby
\hyperlinks \Blue\Blue      % aktivní hyperlinky
\verbatim                    % verbatim v textu pomocí `...`
\cslang \csquotes            % české vzory dělení, \"text\" jsou české uvozovky

\tit Ukázka

\nonum\notoc\sec Obsah
\maketoc % automatické vygenerování obsahu

\sec Odkazy

číslovaná rovnice se označí pomocí ``\eqmark``:
$$
a^2 + b^2 = c^2 \quad \text{\eqmark[my-eq]}
$$
Můžeme odkazovat na rovnici~\ref[my-eq] na straně~\pgref[my-eq].
Též odkážeme na obrázek~\ref[my-pic] v sekci~\ref[obrazky] na straně~\pgref[my-pic].

\sec[obrazky] Obrázky

Obrázky (PDF, JPG, PNG, TIFF) mohou být vloženy pomocí ``\inspic``.
```

Dále může být přidán popisek pomocí `\caption`, to vygeneruje také číslo obrázku.

```
\label{my-pic}
\centerline {\picwidth=2.7cm \inspic{op-ring.png}} % šířka obrázku 2.7 cm
\cskip
\caption/f Tohle je jakýsi prstýnek.
\bye
```

V první části dokumentu je deklarován jeho vzhled a ve druhé je text dokumentu. Vidíte, že nevidíte moc kučeravých závorek typických pro  $\text{\LaTeX}$ ový dokument. Nadpisy kapitol, sekcí, podsekcí jednoduše končí koncem řádku.

## 5. Příklady z Op $\text{\TeX}$ triků

Na Op $\text{\TeX}$  triku 0181 (tisk fontové tabulky) se pokusím demonstrovat koncept Op $\text{\TeX}$ u dělat věci jednoduše. Trik vytiskne tabulku znaků unicode fontu s tím, že je třeba specifikovat, od kterého kódu začít a kde skončit. Implementuje se makro `\fonttable [from=..., to=...] {název fontu}`. Trik najdete na stránce [6] a shledáte, že makro je implementováno takto:

```
\optdef\fonttable[from=0,to=FF]#1{%
  \readkv{\the\opt}%
  \ufont\f=#1\relax
  \baselineskip=18pt plus1pt
  \footline={Font: #1\hss \the\pageno\qquad}
  \tmpnum="\kv{from} \divide\tmpnum by16 \multiply\tmpnum by16
  \fornumstep 16: \tmpnum.." \kv{to} \do {\tablerow{##1}}
}
\def\tablerow#1{\hbox{%
  \hbox to40pt{\hss\Blue\tt U+\hexnum{#1}-\hexnum{#1+15}:}
  \fornum #1..#1+15 \do {\tableslot{##1}}%
}}
\def\tableslot#1{\hbox to25pt{\hss \iffontchar\f#1 \f\char#1 \else\Red --\fi\hss}}
\def\hexnum#1{\directlua{tex.print(string.format("%.4X",#1))}}
\headline={\Blue\kern40pt \foreach 0123456789ABCDEF \do{\hbox to25pt{\hss#1\hss}}\hss}
```

Makro řeší jedinou věc, vytiskne požadovanou tabulku znaků z fontu podobně, jako to dělá  $\text{\LaTeX}$ ový balíček `unicodfonttable.sty` od Franka Mittelbacha. Na rozdíl od zmíněného balíčku makro přijme jen dva parametry `from` a `to` a nic více. Barvy a formátování tabulky jsou v makru zapsány „natvrdo“. Předpokládá se, že pokud někdo bude chtít tu tabulku tisknout v jiné vizáži, tak se podívá, jak je makro udělané, a modifikuje si je dle svých požadavků. Takže k dokumentaci tohoto makra stačil jeden odstavec textu na stránce Op $\text{\TeX}$  triků a k implementaci stačilo 15 řádků. Na rozdíl od  $\text{\LaTeX}$ ového balíčku, který má svou dokumentaci na sedmi stránkách, je v ní popsáno celkem 27 možných parametrů, které mohou modifikovat vzhled výsledku, a implementace je v souboru, který má cca 700 řádků kódu. Ovšem k čemu to potřebujeme? Abychom vytiskli fontovou tabulku. Modifikovat výstupní vzhled můžeme daleko silnějšími nástroji (přímým zásahem do maker Op $\text{\TeX}$  triku) než pomocí 23 obsáhle dokumentovaných parametrů.

Jen namátkou vyjmenuji některé další OpTeX triky. Nejprve je uvedeno číslo triku, pak stručný popis a nakonec v závorce před lomítkem počet řádků kódu OpTeX triku a za lomítkem počet řádků kódu odpovídajícího L<sup>A</sup>T<sub>E</sub>Xového balíčku.

- 0078: `\algot`, tisk pseudo-kódů (30/330)
- 0070: `\keystroke`, obrázky tlačítek na klávesnici (15/200)
- 0149: `\csvread`, práce s databázemi CSV (21/12 900)
- 0154: `\jsonread`, práce s JSON soubory (17/1970)
- 0035: `\import`, čtení souborů z podadresářů dle kontextu (3/150)
- 0054: `\beglua`, zápis Lua kódu pohodlně v dokumentu (10/160)
- 0105: `\directoutput`, tisk strany přímo z boxu (10/1080)
- 0012: použití znaků z fontů Awesome 5 (16/2185)
- 0182: `\pxpic`, pixel art (6/360)

Pochopitelně ke každému L<sup>A</sup>T<sub>E</sub>Xovému balíčku máte navíc relativně rozsáhlou dokumentaci, zatímco k odpovídajícímu OpTeX triku stačí typicky jeden odstavec vysvětlujícího textu.

Některá makra, která jsou definována OpTeX trikem a u kterých lze předpokládat častější užití, nemusí uživatelé přetahovat z webové stránky do svých makro souborů, ale mohou je použít přímo, jako by byly součástí formátu. Je to proto, že tato makra mají příznak „autoload“ a OpTeX je pak schopen si načíst jejich definice z externího souboru v okamžiku prvního použití makra a uživatel se nemusí o nic starat.

## 6. Rozsáhlejší aplikace OpTeXu: OpBible

Ve spolupráci s teologem Šimonem Konečným jsme vytvořili nástroj na psaní Biblí a nazvali jsme to OpBible [8]. Není to nástroj na tisk jen obyčejných Biblí, jsou to Bible určené pro studium teologie. To znamená, že originální text Bible je bohatě proložen komentáři a dalšími doprovodnými texty. Komentáře se věnují na konkrétní fráze použité v textu Bible a jsou umístěny podobně jako poznámky pod čarou na stejné stránce, kde se fráze vyskytuje. A ten poznámkový aparát je skutečně rozsáhlý: na typické stránce vidíte jen pár řádků originálního textu Bible a pod ním třeba 80 procent strany vyplňují poznámky k těm pár řádkům. Dovedete si jistě představit, že pak kompletní Bible má několik tisíc stran.

Existuje mnoho překladů Bible z originální hebrejštiny. Například do angličtiny jich je několik desítek a mezi překlady do češtiny asi někteří znají Bibli Kralickou, revidovanou Bibli Kralickou, ekumenický překlad, Bibli pro 21. století, AI překlad atd. Na stránce [9] je prezentováno celkem deset českých překladů Bible vygenerovaných pomocí OpBible, tedy pomocí OpTeXu.

Makra OpBible jsou na problém variantních překladů připravena a dokáží pracovat s jediným poznámkovým aparátem pro všechny překlady. To vyžaduje deklarovat odlišné fráze různých překladů ve speciálním souboru a pak ve zdrojovém textu poznámek používat jen univerzální fráze. Při zpracování konkrétního překladu se automaticky v poznámkách nahradí univerzální fráze těmi frázemi, které používá daný překlad. Například jméno krále Balsazara se v některých

překladech uvádí jako Bělšaccar, jinde zase jako Belšasar. Komentáře automaticky použijí správné jméno tohoto kinga podle toho, který překlad je zrovna zpracováván. Pan Konečný pustí po úpravě poznámek na noc jeden skript a ráno má nově vygenerovaných všech deset překladů, a to nejen v PDF souborech, ale makra OpBible vytvoří i podklad pro html verzi použitelnou pro web.

To samozřejmě není všechno. Je podporován rozsáhlý systém hypertextových odkazů, které v případě Biblí jsou velice specifické a odkazují do jednotlivých knih Bible, kapitol a veršů. To vyžaduje zcela odlišnou koncepci zápisu odkazů, než jsou odkazy v odborném článku. Je také potřeba sledovat kontext, tj. odkaz do verše nemusí vyžadovat specifikaci kapitoly ani knihy, pokud se o něm mluví ve stejné kapitole stejné knihy. Ale vzdálený odkaz do jiné knihy vyžaduje úplnou uspořádanou trojici údajů. Jak jsou teologové zvyklí odkazy do Bible používat, tak se skutečně používají. Aby toho nebylo málo, v některých překladech občas nějaký veršik chybí nebo přebývá a odkazový aparát se s tím musí umět vyrovnat.

Texty samotných Biblí v jednotlivých překladech jsou staženy z internetu a jsou strukturovaně uloženy ve speciálních souborech, odkud si je bere OpTeX při sestavování jednotlivých studijních Biblí.

Kromě textů Biblí samotných a poznámek k nim je výsledný dokument proložen občas přidávanými rozsáhlejšími komentáři na třeba i více stránkách. Ke všemu bylo potřeba navrhnout grafiku tak, aby se v tom nakonec čtenář vyznal.

Pro realizaci takto rozsáhlého projektu jsem potřeboval mít TeX stoprocentně pod svou vlastní kontrolou. Je tedy zřejmé, že nejlepší volbou pro mě bylo použít k tomu OpTeX.

## Reference

- [1] OLŠÁK P.: OpTeX, <https://petr.olsak.net/optex/>.
- [2] OLŠÁK P.: Proč nerad používám L<sup>A</sup>TeX, <https://petr.olsak.net/ftp/olsak/bulletin/nolatex.pdf>.
- [3] OLŠÁK P.: C<sup>S</sup>plain, <https://petr.olsak.net/csplain.html>.
- [4] OLŠÁK P.: OPmac, <https://petr.olsak.net/opmac.html>.
- [5] OLŠÁK P. TeX in a Nutshell, <https://petr.olsak.net/ftp/olsak/optex/tex-nutshell.pdf>.
- [6] OpTeX tricks, <https://petr.olsak.net/optex/optex-tricks.html>.
- [7] OLŠÁK P.: OpTeX katalog, <https://petr.olsak.net/ftp/olsak/optex/op-catalog.pdf>.
- [8] KONEČNÝ Š. – OLŠÁK P.: OpBible, <https://github.com/olsak/OpBible>.
- [9] KONEČNÝ Š.: OpBible, <https://opbible.org>.
- [10] KONEČNÝ Š. – OLŠÁK P.: OpBible — a tool for creating study Bibles, <https://tug.org/TUGboat/tb46-1/tb142konecny-opbible.pdf>.

## Kontaktní adresa

**Petr Olšák**, České vysoké učení technické v Praze, Fakulta informačních technologií,  
*E-mailová adresa:* [olsak@fit.cvut.cz](mailto:olsak@fit.cvut.cz)



## MACHINE GNOSTICS – A STEP TOWARDS NON-STATISTICAL DATA ANALYSIS AND MACHINE LEARNING

NIRMAL PARMAR (IN)

**Abstract.** Machine Gnostics (MG) is the world's first open-source Python library built on a completely different mathematical foundation rooted in Mathematical Gnostics. Instead of probability and statistics, it draws on the laws of physics — Riemannian geometry, thermodynamic entropy, and relativistic mechanics — to analyse data and build models. The result is a toolkit that works reliably on small datasets, automatically handles outliers without any manual preprocessing, and reveals structure in data that standard statistics cannot see. This paper introduces Machine Gnostics to the open-source community: what it is, where it came from, what it can do, and how to start using it today. Machine Gnostics is free, open-source (GNU GPL v3.0), and available at <https://github.com/MachineGnostics> and documentation at <https://docs.machinegnostics.com>.

**Key words and phrases.** Machine Gnostics, Mathematical Gnostics, open-source AI, robust statistics, Gnostic Distribution Functions, small data, non-statistical machine learning, MAGNET.

## MACHINE GNOSTICS – KROK K NESTATISTICKÉ DATOVÉ ANALÝZE A STROJOVÉMU UČENÍ

**Abstrakt.** Machine Gnostics (MG) je první open-source knihovna v jazyce Python založená na zcela odlišném matematickém základu, který vychází z matematické gnostiky. Namísto pravděpodobnosti a statistiky staví na zákonech fyziky, zejména na Riemannově geometrii, termodynamické entropii a relativistické mechanice, a využívá tyto principy k analýze dat a tvorbě modelů. Výsledkem je sada nástrojů, která spolehlivě pracuje i s malými datovými soubory, automaticky se vyrovnává s odlehlými hodnotami bez nutnosti ručního předzpracování a odhaluje ve datech struktury, které standardní statistické metody nemusí zachytit. Tento článek představuje Machine Gnostics komunitě OSS: vysvětluje, čím tato knihovna je, odkud vychází, co dokáže a jak ji lze začít používat v praxi. Machine Gnostics je svobodný open-source software dostupný pod licencí GNU GPL v3.0. Zdrojové kódy jsou k dispozici na adrese <https://github.com/MachineGnostics> a dokumentace na adrese <https://docs.machinegnostics.com>.

**Klíčové slová.** Strojová gnostika, matematická gnostika, open-source UI, robustní statistika, gnostické distribuční funkce, malá data, nestatistické strojové učení, MAGNET.

## 1. Introduction: What is Machine Gnostics?

Machine learning and statistical modeling have achieved substantial success across many domains, particularly when datasets are large, clean, and approximately aligned with standard assumptions. These assumptions work well in idealised settings — large samples, known distributions, repeatable experiments — but they are often violated in real-world applications. In practice — in factories, clinics, research labs, and financial trading desks — data are often small, messy, and structurally complex. Standard models do not fail loudly in these situations. They fail silently, producing confident-looking results that are simply wrong.

Machine Gnostics is a response to that problem. It is an open-source Python library that implements a completely different mathematical approach: one grounded not in probability and statistics, but in the laws of nature — the same physical laws that govern thermodynamics, curved geometry, and relativistic mechanics. Every algorithm in the library inherits these foundations, which means that it naturally resists outliers, works on very small datasets, and gives a more honest account of what the data actually say.

The core idea, in plain language, is this:

*“Every data point is a real, physical event — not a sample drawn from an imagined infinite population. Uncertainty has physical causes. Let the data speak for themselves.”*

Until now, every analyst, data scientist, and machine learning engineer has been working through the same lens — *the statistical lens* — often without even realising it. This is not a criticism: it is simply the only lens that has ever been available. Statistical thinking is so deeply embedded in data science that its assumptions are treated as natural laws rather than choices. What Machine Gnostics offers, for the first time, is a second lens. By converting Dr. Kovanic’s mathematical theory into a practical open-source library, MG makes it possible to look at the same data from a fundamentally different perspective: one grounded in the physics of individual events rather than the statistics of populations. This new lens does not replace the old one — *it complements it*. Many patterns that were invisible through the statistical lens — because they were averaged away, normalised out, or treated as noise — become clearly visible when the data are allowed to speak for themselves. For the first time, practitioners have a real choice in how they see their data.

This is not a competing replacement for scikit-learn, TensorFlow or PyTorch. It is a complementary lens. On large, clean datasets, MG and classical ML give similar results. On small, noisy, or contaminated data — the kind that most real-world practitioners actually face — MG gives substantially better and more honest results.

### 1.1. The Problem: When Statistics Quietly Gets it Wrong

Classical statistics was built for a specific world: large samples, known distributions, repeatable experiments. The Law of Large Numbers and the Central Limit Theorem — the two pillars of statistical inference — are both asymptotic results. They guarantee good behaviour only as the sample size approaches infinity.

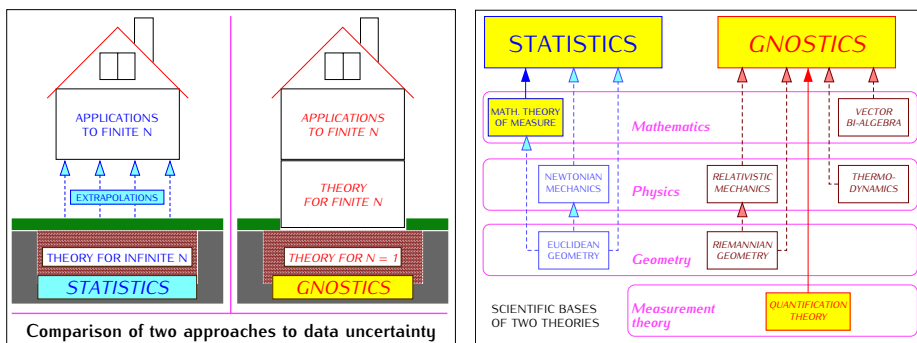
When these assumptions are violated, statistical models do not raise an error. They continue computing means, fitting lines, reporting  $R^2$  values or other metrics — all identical, all plausible-looking — even when the underlying data have completely different structures.

Common real-world situations where classical ML fails silently include:

- Small samples ( $n < 50$ ): summary statistics are unreliable, distributions are unknown.
- Outliers and leverage points: a single unusual value distorts OLS (Ordinary Least Squares) regression coefficients entirely.
- Non-Gaussian data: most financial, biological, and sensor data are not normally distributed.
- Contaminated measurements: real sensor data contains measurement artefacts that look like valid observations.

Machine Gnostics handles all four situations appropriately — by construction, not by adding post-hoc patches to a fundamentally misspecified model.

### 1.2. Different Foundation: Physics, Not Statistics



**Figure 1.** Left: Statistics builds its theory for infinite sample sizes and extrapolates results for finite datasets. Gnostics, in contrast, constructs its theory directly for finite (even single) events, providing a more natural fit for real-world data. Right: The scientific foundations of Statistics and Gnostics. Both theories span mathematics, physics, geometry, and measurement theory, but differ fundamentally in first principles [1].

Mathematical Gnostics, the theory underlying Machine Gnostics, was developed by Dr. Pavel Kovanic starting in the late 1970s at the Czechoslovak Academy

of Sciences in Prague. Its starting point is a simple but radical observation: uncertainty in data is not random — it has physical causes. A measurement deviates from its true value because of temperature fluctuations, instrument drift, operator error, or environmental noise. These are real, physical phenomena, not mathematical abstractions.

This leads to a different mathematics. Instead of treating observations as samples from a probability distribution, Mathematical Gnostics treats each data point as a finite event in curved geometric space (Riemannian geometry) with a measurable information content (thermodynamic entropy) and physical bounds (relativistic mechanics), see Figure 1. The result is a set of algorithms that:

- Assign each data point an individual gnostic weight — computed deterministically from its geometry, not from a distributional assumption.
- Automatically give near-zero weight to outliers and leverage points, without any manual preprocessing or threshold-setting.
- Fit distributions without assuming any parametric form (Gaussian, Laplacian, etc.).
- Provide new metrics — gnostic mean, median, correlation, H-loss, residual entropy — that expose structure that classical metrics hide.

The analogy to physics is exact, not decorative. The gnostic weight formula is derived from the same geometry that underlies general relativity. The entropy metric comes from Clausius’s original thermodynamic entropy, not from Shannon’s probabilistic formulation. This is not metaphor — it is mathematics.

### 1.3. From Theory to Library: A Brief History

Pavel Kovanic published the first formal description of Mathematical Gnostics in 1984, at a time when personal computers were just becoming available for scientific computation. The theory was met with scepticism — it challenged century-old statistical foundations — but Kovanic persisted. Through the 1990s and 2000s, early implementations appeared in BASIC, C, S-PLUS, and eventually R, and the theory found applications in economics, environmental science, and engineering.

In 2017, Dr. Zdeněk Wagner developed Octave-based gnostic software at the Czech Academy of Sciences specifically for atmospheric aerosol analysis — a project that would eventually analyse millions of data points and demonstrate gnostic methods’ remarkable ability to detect real anomalies invisible to standard statistics. Through 2018–2022, this work expanded into EU-funded environmental and research projects, establishing Mathematical Gnostics as a proven tool in demanding scientific domains.

The connection to machine learning came in late 2022, when Dr. Nirmal Parmar, then a PhD student at the Czech Academy of Sciences under Dr. Magdalena

Bendova and Dr. Zdenek Wagner, recognised that the deterministic, geometry-based algorithms of Mathematical Gnostics were exactly the foundation needed for a new generation of ML. He began building what would become Machine Gnostics. In 2023 — the same year Pavel Kovanic published his final monograph and passed away — the library was released as open-source Python. The project honours Kovanic’s legacy and extends it into the modern AI ecosystem.

## 2. Getting Started with Machine Gnostics

Machine Gnostics is designed to be immediately accessible to any Python developer familiar with Scikit-learn. Installation takes one command, the API is identical in structure to what the Python ML community already knows, and the library integrates natively with MLflow for experiment tracking.

### 2.1. Machine Gnostics OSS Architecture

Figure 2 shows the library architecture. At the centre is MAGCAL — the Mathematical Gnostics Calculations engine — which implements gnostic algebra, geometry, and weight computation. Every model and metric in the library is built on top of MAGCAL, ensuring a consistent, physically grounded foundation throughout. MAGNET (Machine Gnostics Neural Network), the upcoming deep learning framework, will sit at the same level as the current ML models.

The full library covers: Gnostic Distribution Functions (GDFs), gnostic tests (homogeneity, scedasticity), gnostic data analysis, regression, classification, clustering, and time-series forecasting models. Gnostic Boosting and Gnostic Decision Trees — standard ensemble models upgraded with gnostic weights — are also available, providing a direct upgrade path for practitioners who already use tree-based methods.

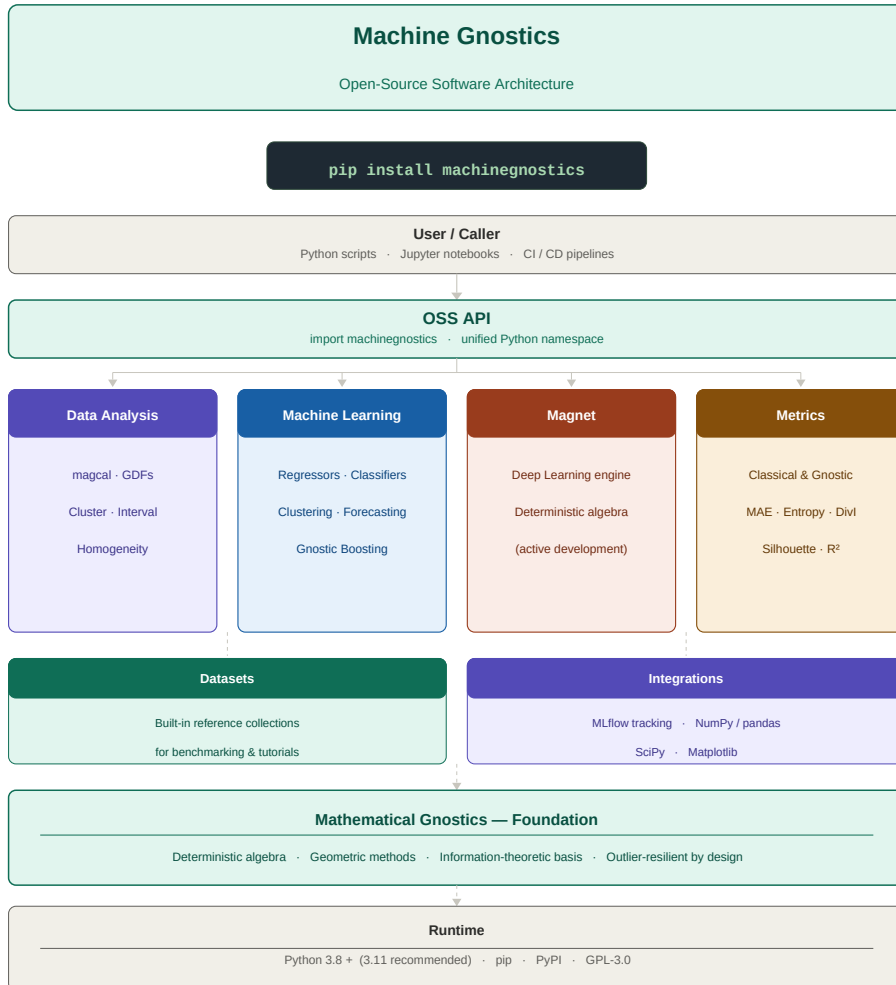
Machine Gnostics OSS is in its early development stage and is actively growing. The library is fully functional and has been validated on real scientific and engineering datasets; however, as with any maturing open-source project, some models and engine components may require further refinement over time. New models, additional gnostic functions, and an expanded test suite are planned. Users are encouraged to report issues and contribute through the GitHub repository.

### 2.2. Installation and the API

**2.2.1. Installation.** Machine Gnostics OSS installation is straightforward. The library is available on PyPI, so it can be installed with pip:

```
pip install machinegnostics
```

Once installed, the library can be imported and used like any other Python ML library. The API is designed to be familiar to any Python ML user, with



**Figure 2.** Machine Gnostics OSS high level architecture. [4].

`fit()` & `predict()` methods and accessible model attributes. Detailed documentation for each model is available at <https://docs.machinegnostics.com>.

**2.2.2. Gnostics Distribution Functions usage example.** GDF (Gnostic Distribution Functions) are a core component of the library, allowing users to fit distributions to data without any parametric assumptions. The API is simple: create an instance of the desired GDF, call `fit()` with your data, and access the fitted parameters or plot the distribution.

Machine Gnostics is offering four GDFs: ELDF (Estimating Local Distribution Function), EGDF (Estimating Global Distribution Function), QGDF (Quantifying Global Distribution Function), and QLDF (Quantifying Local Distribution

Function). Each has its own mathematical properties and use cases, but they all share the same intuitive API. For example, to fit an EGDF to a dataset:

```
# Machine Learning: Gnostic Distribution Functions
from machinegnostics.magcal import ELDF, EGDF, QGDF, QLDF

# distribution fitting and sampling
egdf = EGDF()           # Estimating Global Distribution Function
egdf.fit(data)          # fit to data
egdf.plot()             # plot fitted distribution
egdf.params             # access fitted parameters
```

**2.2.3. Gnostics Metrics usage example.** Machine Gnostics provides a suite of metrics that go beyond traditional statistics. These include the gnostic mean and median, which are robust to outliers, as well as gnostic correlation and  $R^2$  metrics that reveal structure invisible to classical measures. Using these metrics is straightforward: simply import them from the library and call them with your data. For example:

```
# Gnostic Metrics
from machinegnostics.metrics import robr2, correlation, mean, median

print(mean(y))          # gnostic mean (outlier-robust)
print(median(y))         # gnostic median
print(correlation(X, y)) # gnostic correlation
print(robr2(y_true, y_pred, weights)) # gnostic  $R^2$ 
```

**2.2.4. Gnostics Regression usage example.** Machine Gnostics regression models are designed to fit data with the gnostic approach. They automatically assign gnostic weights to each data point, giving near-zero weight to outliers without any manual intervention. The API is similar to other Python ML libraries: create a model instance, call `fit()` with your training data, and then call `predict()` on your test data. For example, to fit a Gnostic Linear Regression model:

```
# Machine Learning: Gnostic Linear Regression
from machinegnostics.models import LinearRegressor

# model initialisation, fitting and prediction
mg = LinearRegressor()
mg.fit(X_train, y_train)
predictions = mg.predict(X_test)

# accessing model attributes
mg.weights          # per-sample gnostic weight; near-zero = outlier detected
mg.coefficients     # model coefficients
mg._history         # iteration log: h_loss, rentropy, coefficients
```

**2.2.5. Gnostics and MLflow usage example.** Machine Gnostics integrates natively with MLflow for experiment tracking. This means that you can log your gnostic models, parameters, and metrics directly to MLflow without any additional code. The integration is designed to be seamless: simply import

the MLflow module, set up your experiment, and use `MLflow.log_param()` and `MLflow.log_metric()` as you would with any other model. For example:

```
# imports
import MLflow
from machinegnostics.integration import MLflow as mg_MLflow

# Set MLflow tracking URI (default is ./mlruns)
MLflow.set_tracking_uri("./mlruns")

# Create or set experiment
experiment_name = "Machine_Gnostics_Tutorial"
MLflow.set_experiment(experiment_name)

# Start MLflow run
with MLflow.start_run(run_name='LogReg_Degree3') as run:

    # 1. Initialize model with parameters
    model = LogisticRegressor()

    # 2. Train the model
    model.fit(X_train, y_train)

    # 3. Make predictions
    y_pred = model.predict(X_test)
    accuracy = np.mean(y_pred == y_test)

    # 4. Log parameters (model configuration)
    MLflow.log_param("model_type", "LogisticRegressor")
    MLflow.log_param("degree", model.degree)

    # 5. Log metrics (performance measures)
    MLflow.log_metric("accuracy", accuracy)
    MLflow.log_metric("rentropy", model.re)

    # 6. Create model signature for input/output schema
    signature = infer_signature(X_train, y_pred)
    input_example = X_train[:3]

    # 7. Log the model artifact
    mg_MLflow.log_model(
        model,
        artifact_path="model",
        signature=signature,
        input_example=input_example
    )
```

### 3. Where & How to use Machine Gnostics?

Machine Gnostics is most useful where conventional ML is weakest: datasets that are small, costly, structurally complex, or contaminated by measurement artefacts. In these settings, classical pipelines often either fail silently or require heavy manual preprocessing. MG addresses this directly through geometry-driven



weighting that reduces the influence of anomalous observations without requiring distributional assumptions.

For practitioners, MG is best used as a complementary lens, not a replacement workflow. Run classical and gnostic analyses in parallel and compare outcomes. Agreement increases confidence; disagreement is diagnostic.

Pharmaceutical, biomedical, and clinical studies are a natural fit because cohorts are typically small and expensive, and single anomalies can dominate classical estimates. Gnostic inference provides stable estimates under these constraints, as shown in DSC heat-capacity uncertainty analysis on small calorimetric datasets [5].

Materials science and thermodynamics present the same pattern: limited experimental runs, high-precision measurements, and non-Gaussian behaviour. Reported applications include VLE boiling-temperature analysis, thermophysical uncertainty quantification, and ionanofluid rheology, where gnostic weighting improves robustness against extreme observations [6, 7].

Financial and economic modelling also benefits from this approach, since returns are non-Gaussian, regimes shift, and small-sample estimation is common. Prior work in mathematical gnostics shows strong relevance for economic data analysis, with robust correlation and regression tools suited to event-driven outliers [2].

The longer-term opportunity is AI for real-world scientific and industrial systems, where data are finite, noisy, and expensive. MG provides a physically grounded and interpretable foundation for such settings, and with MAGNET in development, this framework is expected to extend to deep learning models that train reliably on tens or hundreds of observations rather than millions.

## 4. Conclusion

Machine Gnostics is a genuinely new approach to data analysis and machine learning — not a marginal improvement on existing methods, but a different mathematical foundation. By encoding the laws of physics into algorithms, it produces results that are automatically robust to outliers, reliable on small datasets, and honest about uncertainty in ways that probability-based statistics cannot match.

Machine Gnostics demonstrates, in practice, how physically grounded weighting can distinguish meaningful structure in small, noisy, or contaminated datasets while remaining robust to outliers and leverage effects without manual preprocessing.

Perhaps the most important point is simple: for decades, most analysts have worked within a single statistical framework, often without treating it as a chosen perspective. Machine Gnostics does not ask practitioners to replace that framework; it asks them to complement it with a second, physics-grounded perspective. Run both analyses and compare outcomes. Where they agree, confidence

increases. Where they diverge, the disagreement is often the key signal, revealing structure that a statistical-only view can miss.

For the open-source community at OSSConf, the invitation is direct: install the library, apply it to your own data, and explore. The physics-based lens has existed as a mathematical theory for over forty years. What Machine Gnostics brings is the open-source tooling that makes it accessible to every practitioner, on every dataset, today.

**Acknowledgment.** No grant support to declare. This work on Machine Gnostics was self-funded.

The author gratefully acknowledges former supervisors, Dr. Zdenek Wagner and Dr. Magdalenna Bendová, for their scientific guidance, mentorship, and continued support. The author also expresses sincere thanks to his wife, Darshita Parmar, for her steadfast personal support and patience throughout this work.

## References

- [1] KOVANIC, P.: *Mathematical Gnostics: Advanced Data Analysis for Research and Engineering Practice*, CRC Press, 2023, <https://doi.org/10.1201/9780429441196>.
- [2] KOVANIC, P. – HUMBER, M. B.: *The Economics of Information: Mathematical Gnostics for Data Analysis*, 1st ed., 2015, <https://www.math-gnostics.eu/books/>.
- [3] PARMAR, N.: *Machine Gnostics GitHub Repository*, Machine Gnostics, 2025, <https://github.com/MachineGnostics/machinegnostics>.
- [4] PARMAR, N.: *Machine Gnostics OSS Documentation*, Machine Gnostics, 2025, <https://docs.machinegnostics.com>.
- [5] PARMAR, N. – BENDOVA, M. – WAGNER, Z.: *Heat capacity measurements by a Setaram uDSC3 evo microcalorimeter: advanced data analysis by mathematical gnostics*, Journal of Thermal Analysis and Calorimetry, 150, 313–325, 2024, <https://doi.org/10.1007/s10973-024-13505-w>.
- [6] PARMAR, N. – BENDOVA, M. – WAGNER, Z.: *Study of change in heat capacity of carbon nanotubes based ionanofluids prepared from a series of imidazolium ionic liquids*, Journal of Physical Chemistry Chemical Physics, 24, 22181–22190, 2022, <https://doi.org/10.1039/D2CP02110B>.
- [7] PARMAR, N. – BENDOVA, M. – WAGNER, Z. – PENKAVOVA, V. – DOUIHRI, I. – JACQUEMIN, J.: *Carbon nanotube-based ionanofluids for efficient energy storage*, Industrial & Engineering Chemistry Research, 60(20), 7714–7728, 2021, <https://doi.org/10.1021/acs.iecr.0c06008>.

## Contact address

**Dr. Nirmal Parmar, Ph.D.**, Prague, Czech Republic,  
e-mail: [nirmalparmarphd@gmail.com](mailto:nirmalparmarphd@gmail.com), <https://www.nirmalparmar.in>.

## INTEGRATING FUZZY DECISION TREES INTO A MODULAR DATA MINING APPLICATION

JÁN RABČAN (SK)

**Abstract.** Existing graphical modeling tools for data mining, such as RapidMiner, KNIME, and Weka, have limited native support for fuzzy data types. Their architecture is primarily designed for batch processing of static datasets rather than for handling infinite data streams. This paper presents an open source modeling application that addresses these limitations. The application is organised into four independent layers, allowing new processing nodes to be added without modifying the core codebase. The Dynamic Fuzzy Decision Tree (DFDT), a previously proposed incremental fuzzy decision tree algorithm based on cumulative mutual information, was integrated into the application as a processing node. The integration enables the algorithm to process data streams, update its internal structure incrementally, and output predictions and visualisations. The functionality of the integrated algorithm was evaluated using the application, which served as a testbed for experiments on several streaming datasets. The results confirmed that the algorithm operates correctly within the workflow environment and achieves competitive performance compared to other online learning methods. The application provides a practical platform for further development and evaluation of streaming fuzzy classifiers.

**Key words and phrases.** Data streams, data mining application, open source, classification, fuzzy decision tree.

## INTEGRÁCIA FUZZY ROZHODOVACÍCH STROMOV DO MODULÁRNEJ APLIKÁCIE NA DOLOVANIE ZNALOSTÍ

**Abstrakt.** Existujúce grafické modelovacie nástroje na dolovanie dát, ako napríklad RapidMiner, KNIME a Weka, majú obmedzenú natívnu podporu pre fuzzy dátové typy. Ich architektúra je primárne určená na dávkové spracovanie statických datasetov, nie na prácu s nekonečnými dátovými prúdmi. Tento článok predstavuje open source modelovaciu aplikáciu, ktorá tieto obmedzenia prekonáva. Aplikácia je organizovaná do štyroch nezávislých vrstiev, čo umožňuje pridávať nové spracovateľské uzly bez úprav základného kódu. Dynamický fuzzy rozhodovací strom (DFDT), čo je už skôr navrhnutý inkrementálny fuzzy rozhodovací strom založený na kumulatívnej vzájomnej informácii, bol integrovaný do aplikácie ako spracovateľský uzol. Vďaka tejto integrácii môže algoritmus spracovávať dátové prúdy, inkrementálne aktualizovať svoju vnútornú štruktúru a poskytovať predikcie a vizualizácie. Funkčnosť integrovaného algoritmu bola overená pomocou samotnej aplikácie, ktorá slúžila ako testovacie prostredie pre experimenty na niekoľkých prúdových datasetoch. Výsledky potvrdili, že algoritmus pracuje správne v prostredí workflow a dosahuje konkurencieschopnú výkonnosť v porovnaní s inými online metódami učenia. Aplikácia poskytuje praktickú platformu pre ďalší vývoj a hodnotenie prúdových fuzzy klasifikátorov.

**Kľúčové slová.** Dátové prúdy, aplikácia na dolovanie dát, otvorený kód, klasifikácia, fuzzy rozhodovací strom.

## Introduction

Data mining is a standard component of data analysis across many fields. Graphical workflow tools like RapidMiner [17], KNIME [1], and Weka [4] are widely used because they allow users to design analytical processes visually. These platforms are mature and feature-rich, but they are generally optimized for batch processing of static datasets. Furthermore, while some are open-source, their codebases can be complex, making it difficult to integrate novel algorithms.

Two specific limitations can be identified in these established tools. First, native support for fuzzy logic is limited. RapidMiner, for example, does not include fuzzy attributes as a core data type [7], which makes it difficult to model uncertainty directly within the workflow. Integrating a fuzzy classifier into such environments is therefore not straightforward and requires additional workarounds. Second, these tools are not primarily designed for data streams [6,14]. While some can be adapted, they lack native constructs for handling infinite data sources or incremental model updates. As a result, researchers who wish to experiment with streaming fuzzy algorithms cannot directly work with fuzzy data.

Motivated by these limitations, we have developed a new open source application focused on diagram based workflow creation. In this application, users construct visual diagrams where individual nodes represent operations such as data loading, transformation, modelling, or evaluation. Data flows along the connections between nodes, and the user defines the sequence of steps that together form a complete data mining process. The application is designed to be extensible, allowing new processing nodes to be added without modifying the core system.

Table 1 summarizes the main differences between the mentioned tools (RapidMiner, KNIME, Weka) and our application [12]. From the overview it is clear that none of the existing tools provide direct support for fuzzy attributes or native processing of data streams with incremental learning.

In this paper, we describe the integration of an incremental fuzzy decision tree (DFDT) into this application. This algorithm has been developed in [13]. The DFDT algorithm combines fuzzy logic with incremental learning and uses cumulative mutual information to adapt to changes in the data stream over time, a phenomenon known as concept drift [8]. We present the main principles of the algorithm, including how it updates node statistics and decides when to restructure the tree. We then show how the algorithm is wrapped as a processing node in the workflow environment, making it accessible through the visual diagram interface.

**Table 1.** Comparison of features between existing tools and the proposed application

| Feature              | RapidMiner | KNIME              | Weka               | New app      |
|----------------------|------------|--------------------|--------------------|--------------|
| Use case             | Batch      | Batch              | Batch              | Stream/batch |
| Fuzzy support        | No native  | Limited/extensions | Limited/extensions | Native       |
| Stream handling      | No         | No                 | No                 | Yes          |
| Incremental learning | No         | No                 | No (add-ons)       | Yes          |
| Open-source          | Partly     | Yes                | Yes                | Yes          |
| Code complexity      | High       | High               | Moderate           | Moderate     |
| Drift adaptation     | No         | No                 | No                 | Yes          |

To evaluate the performance of the integrated DFDT, we conduct experiments comparing it against several online learning algorithms. These include incremental  $k$ -nearest neighbors, online Naïve Bayes, logistic regression, and an online multi-layer perceptron. Furthermore, we include standard decision trees, static fuzzy decision trees, as well as established incremental decision trees such as VFDT and FlexDT. To ensure a fair comparison across these structurally diverse methods, appropriate attribute transformations, such as fuzzification, linguistic-to-numeric conversion, or normalization were applied where necessary. The evaluation uses prequential testing on multiple streaming datasets from different domains, with all benchmarks optimized via hyperparameter tuning. The results show that DFDT achieves competitive accuracy while offering a combination of properties that are not present together in the other algorithms: adaptability to drift, an interpretable tree structure, and native support for fuzzy data.

The remainder of this paper is organized as follows. Section 2 describes the system architecture. Section 3 details the core library and the DFDT algorithm. Section 4 presents the user interface. Section 5 discusses extensibility, and Section 6 concludes the paper.

## 1. Application Design

The complete source code of the application is available at [12]. To allow new functionality to be added without modifying the existing codebase and to keep individual parts of the system reusable and testable, the application is organised into four distinct layers. This division allows each layer to be developed, tested, and potentially replaced independently. The layers are structured as follows:

- **Core Algorithm Library:** This layer contains the data structures and machine learning algorithms. It is a standalone library, independent of the workflow logic or graphical representation. This allows the algorithms to be used in other environments and applications.

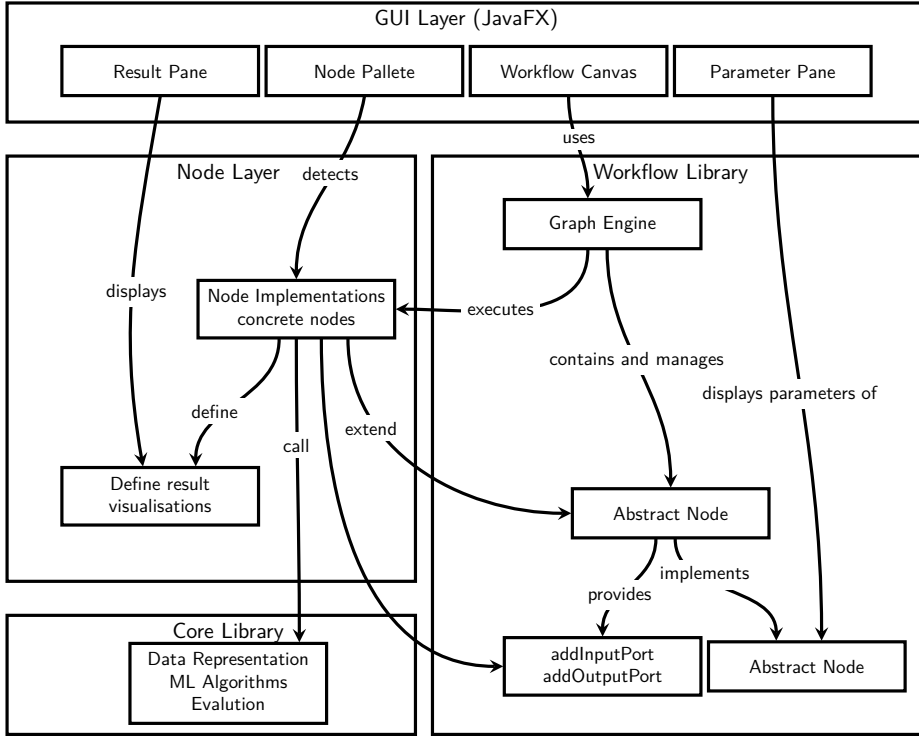
- **Workflow Library (Workflow-Lib):** This layer defines the logical framework for the system. It manages the workflow as a directed graph of processing nodes. The library provides the `AbstractNode` base class and the engine responsible for the execution flow. It does not contain specific mining logic but provides the necessary abstractions for data movement between nodes.
- **Node Implementation Layer (Workflow-Nodes):** This project acts as an extension layer. It contains concrete implementations of nodes (e.g., `DecisionTreeNode`) that bridge the Core Algorithm Library with the Workflow Library. By extending the `AbstractNode` class, these components integrate specific algorithms into the graph based system.
- **Graphical User Interface:** The top layer provides the graphical user interface. This layer is strictly dependent only on the Workflow-Lib abstractions. It uses an automated wrapping mechanism that detects and loads concrete node implementations as extensions. For every instance of a class extending `AbstractNode`, the GUI automatically generates a visual representation and a configuration panel. This ensures that new nodes can be added to the system without requiring any modifications or additional code within the presentation layer.

Because the system is separated into independent modules and the GUI relies on automatic discovery of node implementations, adding new functionality does not require changes across all layers. a developer can create a new node in the Node Implementation Layer without modifying the Workflow Library, the Core Algorithm Library, or the graphical user interface. The application itself remains unchanged; it simply loads available extensions at startup. This design allows the platform to grow over time by adding new algorithms and processing nodes without altering the existing codebase.

In next part of this paper, the incremental fuzzy decision tree algorithm and its integration into the application is described.

## 2. Core Library: Incremental Fuzzy Decision Tree

The Core Algorithm Library implements several mining algorithms, with the Dynamic Fuzzy Decision Tree (DFDT) serving as the primary method for incremental learning from data streams. DFDT is an algorithm that integrates fuzzy logic with incremental learning, enabling it to handle uncertainty while continuously refining its structure in response to new data. The implementation follows the algorithm described by Rabcan et al. [13], with adaptations for integration into the workflow-based environment.



**Figure 1.** Four-layer architecture of the proposed application

## 2.1. Background and Motivation

Traditional decision trees operate by recursively splitting datasets based on attribute values, creating interpretable tree structures. However, they face challenges in handling uncertainty arising from noise, imprecision, and overlapping classes. Fuzzy Decision Trees address these limitations by incorporating fuzzy logic—instead of making strict binary splits, they use membership functions to assign degrees of belonging to multiple classes. This approach softens decision boundaries and better models noisy or overlapping data.

Despite these advantages, FDTs are inherently static classifiers that cannot adapt to evolving data distributions in streaming environments. Incremental learning algorithms such as Hoeffding Trees and Very Fast Decision Trees (VFDT) address this limitation by updating their structure as new data arrives [10]. However, these methods focus primarily on traditional decision trees and have limited capability in handling uncertain or imprecise data. The DFDT algorithm bridges this gap by combining fuzzy logic with incremental learning, as detailed in the original publication [13].

## 2.2. Algorithm Overview

DFDT operates in three main phases: initialization, incremental update, and structural adjustment. Unlike many incremental decision trees that depend on explicit drift detection, DFDT continuously evaluates Cumulative Mutual Information (CMI) measures to guide its restructuring process, following the framework established in [13].

During initialization, DFDT constructs an initial tree structure using a batch of training data. For each node, the algorithm computes and stores the statistics required for immediate CMI calculation for all candidate attributes. These statistics capture the relationships between fuzzy attributes and class labels along the path to that node, and are maintained to support future updates without needing to recompute them from scratch.

As new data arrives, DFDT incrementally updates the node statistics. For each incoming sample, the algorithm updates corresponding stored statistics. CMI measure is then recalculated from these updated statistics to dynamically reassess attribute relevance. This approach allows updating the tree without requiring complete rebuilding.

When incremental updates indicate a shift in attribute relevance, DFDT evaluates whether the change in CMI is large enough to require structural adjustment. For a given node along path  $p$ , let  $I_g^{\text{best}}(p)$  be the CMI of the currently best candidate attribute and  $I_g^{\text{current}}(p)$  be the CMI of the attribute currently used for splitting. If the improvement  $\Delta I_g(p) = I_g^{\text{best}}(p) - I_g^{\text{current}}(p)$  exceeds the impurity shift threshold  $\tau(p)$  (defined below), the algorithm triggers a reconstruction of the subtree starting from that node. The previous splitting attribute is replaced by the more informative one, and the outdated subtree is pruned. This mechanism allows DFDT to adapt to concept drift while maintaining a compact and interpretable structure.

## 2.3. Cumulative Mutual Information

The CMI measure evaluates the relevance of attributes for splitting nodes by measuring the impurity reduction between attributes and class labels. All necessary quantities are derived directly from the statistics stored at each node, without requiring explicit contingency tables.

For a node along path  $p$ , let  $f(p)$  be the effective sample count, defined as the sum of cumulative membership degrees for all samples reaching that node:

$$f(p) = \sum_k w_p(s_k),$$

where  $w_p(s_k)$  is the cumulative membership degree of sample  $s_k$  along path  $p$ , computed as the product of membership degrees for all conditions on the path



(as originally defined in [13]). This accounts for the fuzzy nature of the data where samples may contribute partially to multiple paths.

The cumulative conditional entropy  $H(B \mid p)$  of the class attribute  $B$  along path  $p$ ,  $H(B \mid A_i, p)$  (conditional entropy of  $B$  given attribute  $A_i$ ), and  $H(A_i \mid p)$  (uncertainty of attribute  $A_i$  itself) are computed from the node's stored statistics. The normalized CMI between attribute  $A_i$  and class  $B$  is then computed as:

$$I_g(B; A_i, p) = \frac{H(B \mid p) - H(B \mid A_i, p)}{H(A_i \mid p)}.$$

DFDT selects the splitting attribute that maximizes this normalized CMI value, ensuring that the most informative attribute is used for splitting at each node.

To determine when structural adjustments are necessary, DFDT employs an impurity shift threshold inspired by the Hoeffding bound:

$$\tau(p) = \sqrt{\frac{R^2 \ln(1/\delta)}{2 \cdot f(p)}},$$

where  $R = 1$  (the range of normalized CMI, since  $I_g \in [0, 1]$ ),  $\delta$  is a confidence parameter (typically  $\delta = 0.05$ ), and  $f(p)$  is the effective sample count along path  $p$ . This formulation follows the threshold definition introduced in [13].

Structural adjustments are triggered when the improvement in CMI between the best candidate attribute and the current splitting attribute exceeds this threshold:

$$I_g^{\text{best}}(p) - I_g^{\text{current}}(p) > \tau(p). \quad (1)$$

This approach ensures that the tree structure is updated only when there is statistically significant evidence that another attribute has become more relevant for classification. It prevents unnecessary restructuring due to random fluctuations in the data stream. When the condition in equation (1) is met, the subtree rooted at node  $p$  is rebuilt using the new best attribute, and the outdated part of the tree is discarded.

## 2.4. Integration into the Application Architecture

The DFDT algorithm is implemented as part of the Core Algorithm Library, independent of workflow logic or graphical representation. This implementation follows the mathematical framework established in [13] and is then wrapped by a concrete node class (DecisionTreeNode) in the Node Implementation Layer, which extends the AbstractNode class from the Workflow Library described in Section 1.

The IncrementalDTNode encapsulates the DFDT algorithm and exposes its configurable parameters through the property system defined in the Workflow Library. As shown in the code snippet below, the node defines a BooleanParameter for pruning:

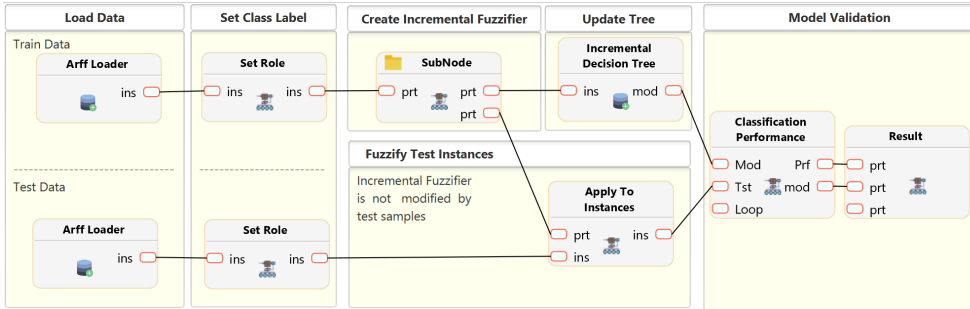
```
1 public class IncrementalDTNode extends ClassifierNode {
2
3     @Override
4     protected Classifier buildModel() {
5
6         IncrementalDecisionTree tree = new
7             IncrementalDecisionTree();
8
9         final List<Instance> instances = getDataset().
10             getInstances().getList();
11         tree.update(instances);
12
13         return tree;
14     }
15
16     @Override
17     public String getName() {
18         return "Incremental Decision Tree";
19     }
20 }
```

The `buildModel()` method demonstrates the core integration pattern. First, it retrieves the dataset from the node's input port. It then creates an instance of the `IncrementalDecisionTree` class from the Core Algorithm Library, configures it with the dataset, and builds the initial model. The method subsequently performs an incremental update using the list of instances, showcasing how the node handles both initial training and incremental learning within the same workflow execution.

After updating the model, the node creates a text-based visualization of the tree structure using `tree.getPrettyString()` and adds it to the model port as a UI object. This enables the graphical interface to display the current tree structure to the user, providing transparency into the model's decision logic.

The pruning parameter, exposed through the property system, is automatically reflected in the node's configuration panel in the GUI. Users can enable or disable pruning when configuring the node, and this setting is passed to the underlying `IncrementalDecisionTree` implementation during model building and updates.

When the workflow executes, this node processes incoming data incrementally. According to the model design, each time new data arrives, the `update` method of the underlying tree is called with the new instances. This updates the model's statistics and potentially triggers structural adjustments as described in Section 2.3. The updated tree structure is then immediately reflected in the visualization, allowing users to monitor how the model evolves over time. The



**Figure 2.** Integration of the incremental DFDT into the application

empirical performance and behavior of this incremental model are further evaluated and demonstrated in the experimental section (Section 3).

The practical application of these components is illustrated in Figure 2. The diagram shows a complete workflow in which the `IncrementalDTNode` (labeled as “Incremental Decision Tree” in the UI according to the `getName()` method) is integrated into the data-mining process. This is simplified and does not fully reflect an online or incremental data-processing. Instead, it is designed primarily for demonstration of the proposed algorithm into an application. Such a setup allows the behavior and interactions of individual components to be more easily observed and understood.

The diagram consists of several nodes, which are organized into cells to separate their concerns. The first cell handles data loading from files in the ARFF format. Two loaders are used: one for training data and one for testing data. Subsequently, the class label is specified. As the next step, the fuzzification model is created.

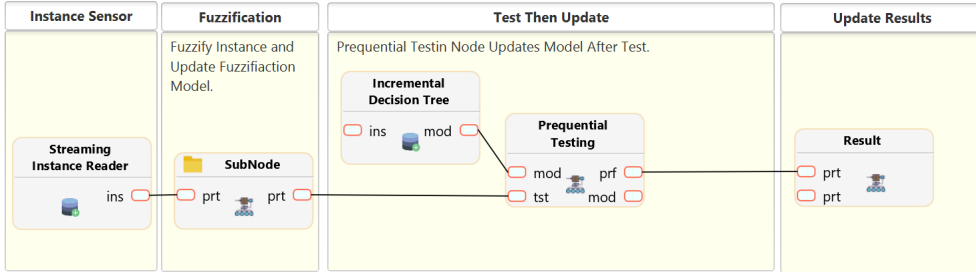
Afterward, the Incremental Fuzzy Decision Tree is trained. The same approach as in the incremental fuzzification phase is applied. The testing data do not modify the models (i.e., the fuzzifier and the decision tree). Once the model is trained, it is evaluated on the testing data.

In the next part of the paper, an experimental analysis is presented to compare the proposed DFDT algorithm with other algorithms for data streams.

### 3. Prequential Evaluation Against Other Online Algorithms

In this section, DFDT is compared against other online learning algorithms. The proposed application was used to perform all experimental evaluations. For each algorithm, a dedicated evaluation process was configured and executed. The specific process used for the DFDT algorithm is shown in the following diagram.

The designed process for the prequential evaluation of the DFDT algorithm consists of four sequential sections arranged linearly. First, the Instance Sensor



**Figure 3.** The evaluation process of DFDT

section uses a Streaming Instance Reader to process data rows sequentially. The data then enters the Fuzzification section, where a specialized sub-node converts numerical inputs into fuzzy values required by the algorithm. The main pipeline is controlled by the Test Then Update section. Here, an Incremental Decision Tree node initializes the streaming model, and the Prequential Testing node evaluates it using each incoming fuzzy instance. Following the prequential approach, the model first predicts the outcome to compute performance metrics and then updates itself using the same instance. Finally, the Update Results section sends the performance metrics to a Result node for real-time visualization.

Based on the analysis performed in [13], typical online algorithms were identified, particularly incremental decision trees such as VFDT [10] or FlexDT [5]. These algorithms represent state-of-the-art approaches for data stream classification, as they are designed to process instances incrementally while maintaining a compact model structure. Besides these tree-based models, neural networks and Bayesian models are also frequently used in the streaming setting. Online Multi-Layer Perceptron (oMLP) [11] exemplifies the neural approach, whereas Online Naïve Bayes (oNB) [9] represents a probabilistic method. Logistic Regression (LR) [15] and incremental  $k$ -Nearest Neighbours (kNN) [2] complete the set of reference algorithms, covering distance-based and linear models. The summarization of selected algorithms for comparison is:

- **DFDT (Dynamic Fuzzy Decision Tree)** [13] – The proposed algorithm combining online fuzzification with incremental tree growth and dynamic membership adaptation.
- **FDT (Fuzzy Decision Tree)** [13] – Incremental fuzzy decision tree without dynamic adaptation, using online fuzzification.
- **DT (Decision Tree C4.5)** [3] – Batch decision tree with a sliding window of 5,000 instances; the tree is rebuilt when the window is full.
- **VFDT (Very Fast Decision Tree)** [10] – Incremental decision tree based on the Hoeffding bound, processing data streams without storing historical examples.

- **FlexDT (Flexible Decision Tree)** [5] – Adaptive decision tree that dynamically changes its structure and parameters in response to changes in the data stream.
- **MLP (Multi-Layer Perceptron)** [11] – Online feedforward neural network trained with mini-batch learning (online MLP), with one hidden layer.
- **LR (Logistic Regression)** [15] – Linear model adjusting weights incrementally using gradient descent.
- **NB (Naive Bayes)** [9] – Online Naive Bayes updating prior and conditional probabilities with each new instance without storing past data.
- **kNN ( $k$ -Nearest Neighbours)** [2] – Incremental version storing a sliding window of 5,000 instances and dynamically removing outdated points.

Since these algorithms are not directly comparable, attribute transformations such as fuzzification, linguistic-to-numeric conversion, or normalisation were required. The experiment used prequential testing, where each incoming instance is first used to test the model and then immediately used for training. Several incremental learning algorithms were included, all optimised via hyperparameter tuning (e.g., learning rates, distance measures, network architectures) to ensure fair benchmarking. The results table presents the best configuration found for each algorithm. All datasets from experiments are available in [16]. The complete results are shown in Table 2.

**Table 2.** Classification accuracy of DFDT and other online learning algorithms

| Dataset     | DFDT  | FDT   | DT    | VFDT  | FlexDT | MLP   | LR    | NB    | kNN   |
|-------------|-------|-------|-------|-------|--------|-------|-------|-------|-------|
| HpPlane     | 0.864 | 0.872 | 0.825 | 0.860 | 0.862  | 0.927 | 0.852 | 0.835 | 0.755 |
| LED         | 0.746 | 0.638 | 0.637 | 0.735 | 0.730  | 0.712 | 0.738 | 0.740 | 0.450 |
| RBF         | 0.975 | 0.969 | 0.923 | 0.968 | 0.970  | 0.963 | 0.902 | 0.875 | 0.941 |
| RBF-Drift   | 0.975 | 0.969 | 0.923 | 0.965 | 0.967  | 0.973 | 0.902 | 0.870 | 0.971 |
| SEA         | 0.848 | 0.848 | 0.828 | 0.841 | 0.842  | 0.839 | 0.838 | 0.834 | 0.803 |
| CType       | 0.763 | 0.758 | 0.755 | 0.755 | 0.759  | 0.807 | 0.713 | 0.606 | 0.752 |
| SIEC        | 0.777 | 0.741 | 0.767 | 0.770 | 0.775  | 0.762 | 0.761 | 0.717 | 0.745 |
| Electricity | 0.792 | 0.666 | 0.703 | 0.780 | 0.785  | 0.792 | 0.583 | 0.455 | 0.786 |
| Household   | 0.953 | 0.951 | 0.923 | 0.949 | 0.942  | 0.933 | 0.942 | 0.922 | 0.931 |
| Incomes     | 0.943 | 0.930 | 0.931 | 0.935 | 0.933  | 0.921 | 0.918 | 0.940 | 0.921 |

The performance of all models varies across datasets, depending on data characteristics such as noise, dimensionality, or concept drift. DFDT variants generally maintain the best or competitive accuracy across most datasets. The oMLP achieves high accuracy in several cases, while oNB tends to be more stable on simpler streams. The iKNN shows reduced accuracy on high-dimensional

data (e.g., LED). Both the fixed-window FDT and the conventional decision tree demonstrate performance variability depending on the specific dataset properties.

The results indicate that model suitability is context-dependent. No single algorithm performs best across all scenarios, which suggests that the choice of algorithm should consider the specific characteristics of the target data stream. The DFDT variants offer an alternative that integrates fuzzy logic and continuous restructuring. The main contribution of DFDT is not necessarily a higher accuracy in all scenarios, but the integration of several properties: adaptability to concept drift, interpretability of the tree structure, and an inherent mechanism to handle data uncertainty through fuzzy logic. While models such as oMLP or traditional incremental trees often lack this specific combination, DFDT presents itself as a suitable approach for dynamic and uncertain environments that require both model transparency and the ability to work with imprecise data.

## 4. Conclusions

The paper presents a modular open-source workflow application for data mining. The application supports fuzzy data types and incremental stream processing. Its four-layer architecture allows new processing nodes to be added without changing the core codebase. The integration of the incremental fuzzy decision tree (DFDT) based on cumulative mutual information serves as a demonstration of this extensibility.

A prequential evaluation on ten streaming datasets shows that DFDT achieves competitive accuracy. It outperforms other methods on 6 out of 10 benchmarks and reaches an average accuracy of 0.864.

The main contributions of the paper are:

- An open-source workflow application with native support for fuzzy data and incremental learning.
- A four-layer architecture that enables adding new processing nodes without core code modifications.
- Integration of the DFDT algorithm into the application, including a node for stream processing.
- Experimental comparison of DFDT against eight other online learners on ten streaming datasets.

DFDT is not universally superior. Its performance degrades for high dimensional data (more than 50 attributes) and for drift with very short transition periods. However, DFDT is suitable for applications that require interpretability and adaptability to evolving data, where black-box models lack transparency or static trees cannot handle concept drift. No algorithm performs best in all streaming scenarios; the choice of method depends on the characteristics of the data.

Future work can focus on reducing the computational complexity of CMI for high-dimensional streams, adapting the threshold  $\tau(p)$  dynamically based on drift speed, and extending fuzzy representation to handle missing values in real time.

**Acknowledgment.** This work was supported by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under Project 09I03-03-V04-00416.

## References

- [1] ACITO, F.: *Predictive analytics with KNIME*, Analytics For Citizen Data Scientists, Switzerland: Springer, 2023.
- [2] BAHRI, M. – VELOSO, B. – BIFET, A. – GAMA, J.: *AutoML for stream k-Nearest neighbors classification*, 2020 IEEE International Conference On Big Data (Big Data), pp. 597–602, 2020.
- [3] BANU, E. – ROBERT, P.: *Evaluating the Performance of an Incremental Classifier using Clustered-C4*, 5 Algorithm for Processing Big Data Streams, 2024 5th International Conference On Communication, Computing & Industry 6.0 (C2I6), pp. 1–12, 2024.
- [4] HALL, M. – REUTEMANN, P.: *Weka knowledgeflow tutorial for version 3-5-8*, The University Of Waikato, New Zealand, 2008.
- [5] Hashemi, S. – Yang, Y.: *Flexible decision tree for data stream classification in the presence of concept change, noise and missing values*, Data Mining And Knowledge Discovery, 19, 95–131, 2009.
- [6] HOFMANN, M. – KLINKENBERG, R.: *RapidMiner: Data mining use cases and business analytics applications*, CRC Press, 2016.
- [7] KOTU, V. – DESHPANDE, B.: *Predictive analytics and data mining: concepts and practice with rapidminer*, Morgan Kaufmann, 2014.
- [8] LU, J. – LIU, A. – DONG, F. – GU, F. – GAMA, J. – ZHANG, G.: *Learning under concept drift: a review*, IEEE Transactions On Knowledge And Data Engineering, 31, 2346–2363, 2018.
- [9] LUTU, P.: *Fast feature selection for Naive Bayes classification in data stream mining*, 2013.
- [10] MANAPRAGADA, C. – WEBB, G. – SALEHI, M.: *Extremely fast decision tree*, Proceedings Of The 24th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining, pp. 1953–1962, 2018.
- [11] PRATAMA, M. – ZA'IN, C. – ASHFAHANI, A. – ONG, Y. – DING, W.: *Automatic construction of multi-layer perceptron network from streaming examples*, Proceedings Of The 28th ACM International Conference On Information And Knowledge Management, pp. 1171–1180, 2019.
- [12] RABCAN, J.: *workflow: Modular data mining application*, GitLab repository, <https://gitlab.com/jrabcanj/workflow-all>, 2025.
- [13] RABCAN, J., ZAITSEVA, E., LEVASHENKO, V., – KVASSAY, M.: *Incremental fuzzy decision tree based on cumulative mutual information*, Journal of Computational Design and Engineering, 12(9), 116–130, 2025.
- [14] RUSSELL, I. – MARKOV, Z.: *An introduction to the Weka data mining system*, Proceedings Of The 2017 ACM SIGCSE Technical Symposium On Computer Science Education, pp. 742–742, 2017.
- [15] YANG, B. – WANG, M. – XU, Z. – ZHANG, T.: *Streaming algorithm for big data logistic regression*, 2018 Ieee International Conference On Big Data (big Data), pp. 2940–2950, 2018.

- [16] UNIVERSITY-INDUSTRY EDUCATIONAL CENTRE IN ADVANCED BIOMEDICAL AND MEDICAL INFORMATICS: *Biomedical Signal Classification Using Fuzzy Classifiers – Chapter 1*. <https://cebmi.fri.uniza.sk/portal/modules/chapterDetail.jsp?module=30--chapter=1>.
- [17] WARADE J.: *AI Based Predictive Modelling for Internal Quality Control: a Machine Learning Approach Using Altair RapidMiner*, EJIFCC, 2025 Dec 5, 36(4): 491–498, PMID: 41459179, PMCID: PMC12743339.

## Contact address

**doc. Ing. Ján Rabčan, PhD.**, Department of Informatics, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovakia,  
*e-mail:* [jan.rabcan@fri.uniza.sk](mailto:jan.rabcan@fri.uniza.sk).



## LABORATOŘ ŘÍZENÍ KOLEJOVÝCH VOZIDEL – ŠKOLA HROU

JIŘÍ RYBIČKA (CZ)

**Abstrakt.** Provozně ekonomická fakulta MENDELU má akreditované studijní programy Otevřená informatika a Administrace IS/ICT. Jejich součástí jsou základy informatiky, jako je algoritmizace, objektový návrh, skriptování atd. Přirozenou součástí výuky je rovněž hledání optimálních metod výuky a snaha co nejvíce přiblížit prezentované učivo studentům. Jedním z projektů v této oblasti byl projekt Škola hrou založený na zpřístupnění kolejíště v Laboratoři řízení kolejových vozidel studentům, kteří by na něm mohli zkoušet řídit provoz pomocí speciálně vytvořeného jazyka, a učit se tak jednoduché skripty se základními algoritmickými obraty (větvení, cykly). Příspěvek se zabývá možnostmi a vybavením, které laboratoř nabízí studentům, ale i jiným zájemcům, případně široké veřejnosti.

**Klíčová slova.** Algoritmizace, začátečník, výukový programovací jazyk, modelová železnice, řízení provozu počítačem.

## RAIL VEHICLE CONTROL LABORATORY – SCHOOL THROUGH PLAY

**Abstract.** The Faculty of Business and Economics at MENDELU has accredited study programmes Open Informatics and IS/ICT Administration. They include the basics of informatics, such as algorithmisation, object-oriented design, scripting, etc. A natural part of teaching is also the search for optimal teaching methods and the effort to bring the presented curriculum as close as possible to students. One of the projects in this area was the School through Play project based on making the track in the Laboratory of Rail Vehicle Control accessible to students, who could try to control traffic using a specially created language, and thus learn simple scripts with basic algorithmic elements (branching, cycles). The article deals with the possibilities and equipment that the laboratory offers to students, but also to other interested parties, or the general public.

**Keywords.** Algorithmisation, beginner, learning programming language, model railway, traffic computer control.

### 1. Úvod

Ve studijních programech obsahujících informatiku je bezpochyby důležitou součástí tzv. informatické myšlení, jehož jedním z projevů je algoritmizace. Tu můžeme specifikovat jako schopnost realizovat zamýšlenou úlohu pomocí nástrojů, které máme k dispozici. Pro rozvíjení této schopnosti byla již v minulosti vyvinuta dlouhá řada různých nástrojů a postupů. Určitě si každý dokáže vybavit různé pseudojazyky, které ani nejsou zpracovatelné počítačem. Cesta, kdy při výuce

rozvíjíme schopnost čisté algoritmizace pomocí takových nástrojů, je ovšem časově náročná, protože většinou nemůžeme řešit reálné úlohy, a nedostáváme se tak dostatečně blízko k běžné praxi. V průběhu času vzniklo mnoho konceptů, výukových jazyků a doporučení, za všechny lze jmenovat zajímavý projekt Hedy [2].

Proto se v předmětech typu Algoritmizace přistupuje k používání některého univerzálního programovacího jazyka, s nímž jsou vždy spojeny jeho specifické konstrukce a zvláštnosti. Učení algoritmizace přechází i v názvech předmětů spíše k pojmu „programování“ a je v tomto případě vždy doprovázeno mnoha technickými a implementačními informacemi o daném jazyce. Je zřejmé, že tímto zahlcením se překrývá vlastní rozvíjení čistě algoritmických schopností, na druhé straně však je velmi zdůrazněna programátorská realita.

Přírozeným důsledkem tohoto stavu je neustálé hledání „toho nejlepšího“ reálného jazyka, který je pro výuku algoritmizace (přesněji tedy programování) maximálně vhodný. Tím ovšem opět vzniká disproporce – jazyk vhodný pro výuku už obvykle není vhodný pro reálné programování, i když můžeme hovořit o dostatečné blízkosti. Známé jsou také koncepty nepoužívající vůbec „průmyslový“ programovací jazyk, ale využívající symbolických cest – například blokové programování [1].

## 2. Záměr projektu Škola hrou

Jak vyřešit uvedené rozpory? Jakými požadavky na programovací nástroj pro úplné začátečníky bychom se měli řídit?

- Je vhodné použít reálný univerzální programovací jazyk.
- Je vhodné použít jazyk s jednoduchými konstrukcemi.
- Je nezbytné strojové zpracování cvičných programů s kontrolou jejich činnosti podobě jako u běžných programovacích jazyků.
- Je vhodné proces učení doplnit nějakou přidanou hodnotou.

V předmětech Algoritmizace a Základy objektového návrhu se v obou informatických studijních programech používá programovací jazyk C++. Důvody této volby jsou poněkud komplikované a přesahují rámec tohoto příspěvku, můžeme tedy tuto situaci chápat jako výchozí axiom.

Jazyk C++ se rozhodně nedá zařadit mezi jazyky s jednoduchými konstrukcemi. Jeho značný rozsah umožňuje tutéž úlohu řešit různými alternativními cestami, přičemž jazykové možnosti se neustále mění a rozšiřují, je k dispozici velmi rozsáhlá množina knihoven s nejrůznějšími vylepšeními základních prvků. Pro začátečníky je to nepřijatelná džungle, v níž se začínají ztrácet už v počátečních lekcích.

Základní myšlenkou realizovanou již dříve při výuce algoritmizace je vymezení pouze některých vhodných konstrukcí, s nimiž jsou studenti seznámeni, to však bývá v praxi překračováno tím, že zejména z internetových zdrojů teče mnoho

informací problematické kvality bezmyšlenkovitě přejímaných studenty a aplikovaných ve více či méně vhodných situacích.

Spojit programování s nějakou další přidanou hodnotou znamená přitáhnout k procesu algoritmizace i ty studenty, které pouhé samotné sledování nějakých abstraktních informací na obrazovce počítače příliš nezajímá. Vznikl tedy nápad řídit pomocí programů v jazyce C++ modelové kolejiště. Počáteční problém však spočíval v tom, že samotné ovládání kolejiště pomocí programů bez jakékoliv další podpory bylo poměrně komplikované a efekt „hraní si“ s kolejištěm se dost ztrácel. Navíc vytvoření takového ovládacího programu vyžadovalo poměrně komplexní znalosti, nemohlo být tedy zpřístupněno začátečníkům.

Jako úplně jiný koncept odstraňující tuto zásadní nevýhodu vznikl nový programovací jazyk s názvem RAL (Railway Algorithmic Language), na nějž byly kladeny tyto požadavky:

- Bude vypadat jako jazyk C++ a bude kompilovaný.
- Bude mít jen vybrané jednoduché konstrukce.
- Bude jednoduše ovládat kolejiště.

Inspirací při výběru konstrukcí byl určitý přehled například z učebnice [3], ale také časté zmínky jazyka Python, jež je podle mnohých rovněž vhodný pro začátečníky [5]. Domnívám se však, že reálný programovací jazyk vyžaduje pochopení konstrukcí, které jsou v mnoha jazycích velmi podobné, takže sáhodlouhé spory o to, který skutečně reálný jazyk je pro začátečníky nejlepší, jsou do značné míry zbytečné.

V rámci projektu Škola hrou vznikl návrh a překladač jazyka RAL. Studenti ve výuce předmětu Algoritmizace v prvním semestru studia mají jedno cvičení přímo v laboratoři pro seznámení s kolejištěm a jeho standardním ovládáním, v dalším průběhu semestru pak mají možnost jít do laboratoře a individuálně pracovat na svých programech v jazyce RAL.

Jednoduchý příklad programu v jazyce RAL, který rozjede vozidlo jedním směrem a pak se vrátí zpět:

```
Vozidlo L = 1521;  
L.Jed(10);  
wait(200);  
L.Stuj();  
L.Smer();  
L.Jed(10);  
wait(200);  
L.Stuj();
```

Překladač jazyka RAL je psán v jazyce C++ a jeho dalším efektem je, že slouží jako příklad realizace konkrétního překladače v předmětu Formální jazyky a překladače. Zápis překladače je přizpůsoben požadavkům na jednoduchost a srozumitelnost implementace, aby studenti měli možnost aplikovat podobné konstrukce na vlastní příklady. Implementace překladače vede na generování intermediárního

kódu, který je následně interpretován – jde o koncept známý z implementace jazyka Java [4]. Tím je také dosažena nezávislost na strojovém jazyce konkrétního počítače. Jednoduchost intermediárního kódu umožňuje ukázat typické instrukce (je jich asi jen 30) a jejich použití pro překlad aritmetických výrazů a řídicích konstrukcí.

### 3. Laboratoř řízení kolejových vozidel

Pracoviště s názvem Laboratoř řízení kolejových vozidel původně vzniklo na konci roku 2013 jako výuková základna pro studenty zabývající se elektronikou a řízením technologických procesů. Tehdy měla Provozně ekonomická fakulta akreditován studijní program Inženýrská informatika, jehož zásadní součástí bylo technické vybavení, orientace na robotiku a výrobní procesy řízené počítačem. Kolejistiště jako technologický celek řízený počítačem bylo z tohoto důvodu považováno za doplněk k laboratoři robotických komponent, ve studijním programu pak vznikl předmět Řízení technologických procesů se dvěma variantami – jedna pracovala s roboty, druhá s kolejistištěm.

Se zánikem této akreditace se výukové využití laboratoře přesunulo do volitelného předmětu Řízení kolejových vozidel a Kroužek řízení modelové železnice. Zatímco Řízení kolejových vozidel bylo zaměřeno opět na projekty postavené na elektronických prvcích nebo softwarových komponentách pro řízení, předmět Kroužek je koncipován jako čistě zájmová činnost spojená jak s řízením, tak i s výstavbou a provozem kolejistiště. Studenti se zájmem o hlubší spolupráci mohou v laboratoři také řešit své závěrečné práce.

Kolejistiště v laboratoři jsou dvě, jedno v měřítku TT (1:120), jedno v měřítku H0 (1:87). Fakulta těsně spolupracuje s Klubem modelářů železnic Brno I, v jehož vlastnictví je kolejistiště TT a některé moduly kolejistiště H0. Místnost s oběma kolejistišti má čtvercový půdorys o délce strany cca 15 m.



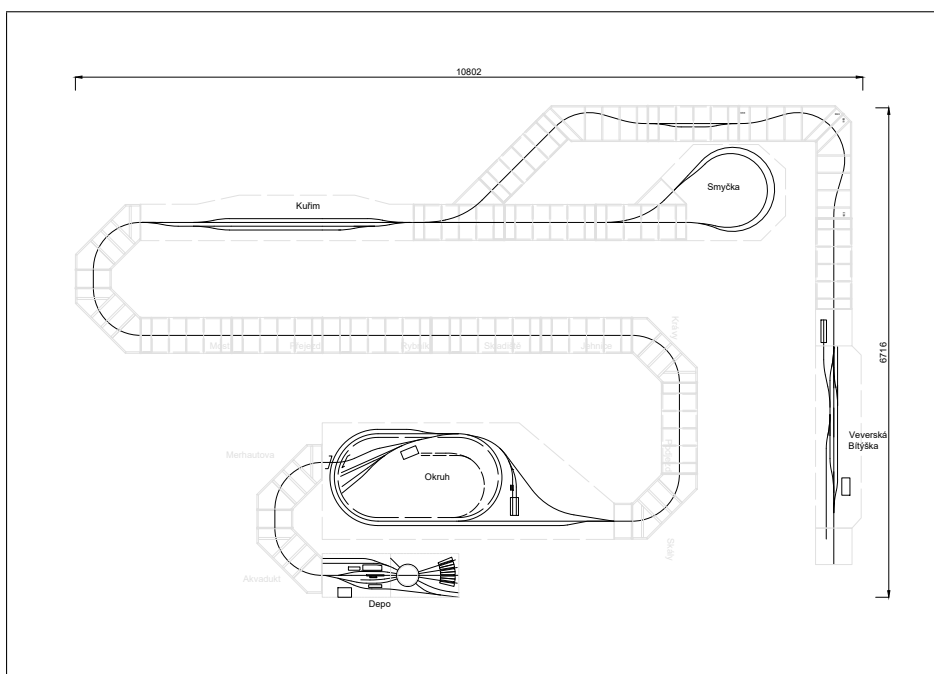
**Obrázek 1.** Ústřední stanice kolejistiště TT – Klobouky u Brna

### 3.1. Kolejiště TT

Koncepcí kolejišť je model reálných tratí, nebo pokud možno úseků, které se realitě hodně přibližují. Kolejiště TT představuje segmenty reálné tratě Čejč–Ždánice, která v dnešní době již prakticky neexistuje. Model představuje stav z 80. let 20. století, kdy byl na celé trati poměrně živý provoz.

### 3.2. Kolejiště H0

Kolejiště v měřítku H0 je z větší části ve vlastnictví fakulty a bylo postaveno za dobu činnosti laboratoře. Základem byly některé moduly vyrobené v Klubu a představující část tratě Veverská Bítýška – Kuřim (provozována v letech 1911–1935), k nimž byla doplněna stanice Kuřim (stav přibližně z roku 1934), další moduly představující tzv. Starou Tišnovku se zastávkou Jehnice (stav z roku 1931) a větší stanice Okruh s lokomotivním depem. Na tomto kolejišti se většinou odehrává provoz v rámci výuky (obr. 2 a 3).



**Obrázek 2.** Schéma modulu kolejiště H0 s pěti dopravními (Veverská Bítýška, Kuřim, Smyčka, Okruh a Depo). Tato sestava byla použita na výstavě v letech 2025 a 2026. Rozměrové kóty ilustrují, jakou část místnosti kolejiště zabírá



**Obrázek 3.** Osobní vlak 504981 přijíždí k nástupišti stanice Okruh

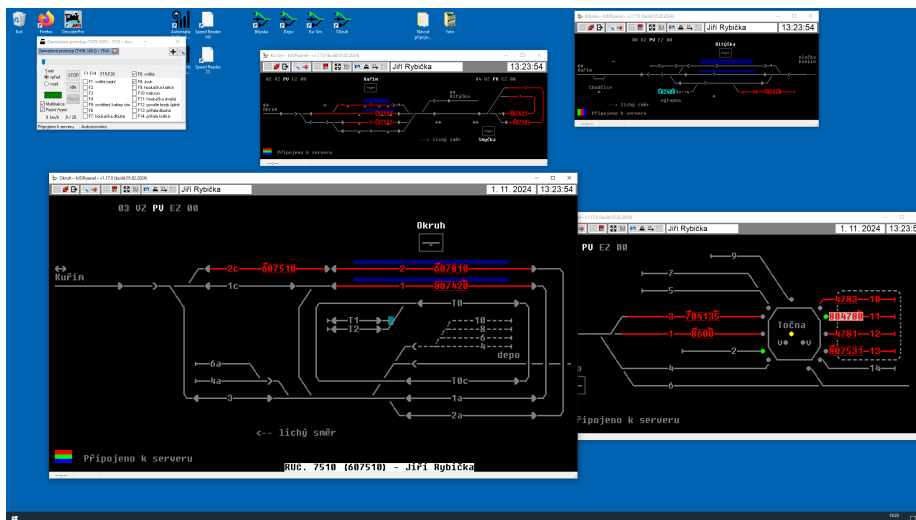
### 3.3. Řízení pomocí hJOP

Obě kolejiště jsou řízena identickým způsobem, mají stejné elektronické vybavení a prakticky stejné softwarové a hardwarové prvky. Vše je navíc koordinováno s vývojem a řízením kolejiště v Klubu modelářů železnic. Řídicí program hJOP (obr. 4) je modelem skutečného řídicího softwaru Správy železnic JOP (Jednotné obslužné pracoviště). Je mírně uzpůsoben pro modelový provoz, ale způsob ovládání je převzat ze skutečnosti.

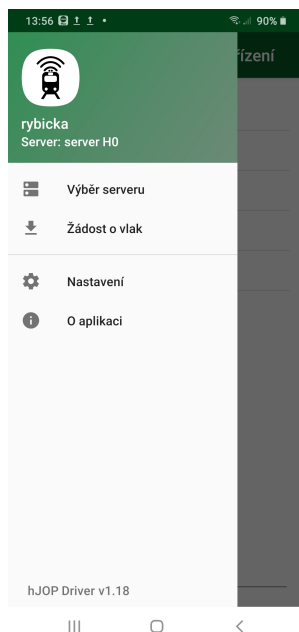
Kromě programu hJOP je možné ovládat vozidla mobilní aplikací hJOPdriver (obr. refhjopdriver a 6), v tom případě je hnací vozidlo (vlak) výpravčím z počítače předáno do řízení na mobilní telefon a strojvůdce jede podle domluvy s výpravčím a podle znaků na návěstidlech. Tato aplikace původně vznikla jako bakalářská práce jednoho studenta.

## 4. Závěr

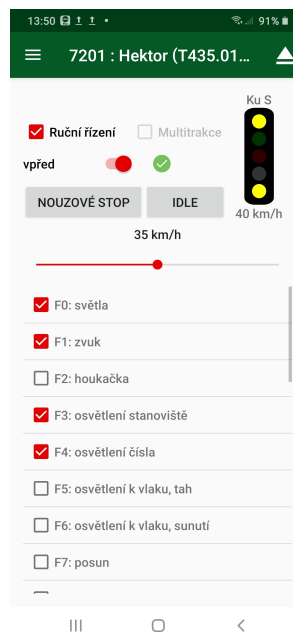
Popisovaná laboratoř jako specializované pracoviště je v první řadě koncipována jako výuková pomůcka a prostředí, kde se mohou studenti-zájemci realizovat při tvorbě a řízení modelového kolejiště. Tím, že kolejiště zároveň představuje reálné tratě, je vhodnou příležitostí pro modelářské aktivity. Svým námětem Staré Tišnovky i regionálně zapadá do lokality Brno-sever, jehož oblastí Stará Tišnovka procházela od roku 1885 až do svého zrušení v roce 1962, kdy byla nahrazena novou tratí. Proto je laboratoř v některých dnech otevřena i pro veřejnost a formou výstavy se ukazují výsledky modelářské práce, provoz podle časových epoch a podle předpisů převzatých ze skutečné železnice (obr. 7).



**Obrázek 4.** Řízení kolejíšť programem hJOP. Reliéfy stanic lze spustit na libovolném počítačím kolejíšť, lze tak ovládat jednu i více stanic



**Obrázek 5.** hJOPdriver – základní nabídka



**Obrázek 6.** hJOPdriver – řízení vlaku

Pro uchazeče o studium jde o pracoviště, které má demonstrovat širší možnosti, jež fakulta studentům při studiu poskytuje. Pravidelně jsou zde pořádány exkurze



studentů a učitelů ze středních škol. S jedním gymnáziem je dohodnuta spolupráce ve výuce speciálního předmětu, jehož jednou součástí je řízení železničního provozu. Studenti tohoto předmětu mají v laboratoři čtyřhodinové cvičení, během něhož zvládnou základní řízení provozu pomocí programu hJOP a naučí se některé prvky zabezpečovacího systému na železnici.



**Obrázek 7.** Návštěvníci na výstavě v Laboratoři v únoru 2026. V popředí je vidět trať do stanice Kuřim

## Reference

- [1] BAU, D., GRAY – J., KELLEHER, C. – SHELDON, J. – TURBAK, F.: *Learnable Programming: Blocks and Beyond*, Communications of the ACM, June 2017, pp. 72–80. <https://doi.org/10.48550/arXiv.1705.09413>.
- [2] HERMANS, F.: *The Programmer's Brain: What every programmer needs to know*, Manning, 2021, ISBN 9781617298677.
- [3] KENT, J.: *C++ bez předchozích znalostí*, Computer Press, 2013, ISBN 9788025124116.
- [4] LINDHOLM, T. – YELLIN, F. – BRACHA, G. – BUCKLEY, A.: *The Java Virtual Machine Specification, Java SE 8 Edition*, Addison-Wesley Professional, 2014, ISBN 9780133922721.
- [5] PECINOVSKÝ, R.: *Začínáme programovat v jazyku Python*, Praha: Grada, 2020, ISBN 978-80-271-1237-1.

## Kontaktní adresa

**doc. Ing. Jiří Rybička, Dr.**, Ústav informatiky, Provozně ekonomická fakulta, Mendelova univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika,  
E-mailová adresa: [rybicka@mendelu.cz](mailto:rybicka@mendelu.cz), <https://lrkv.pef.mendelu.cz>.



## DESKTOP NMR QUANTUM COMPUTER FOR TEACHING QUANTUM INFORMATICS AT UNIVERSITY OF ŽILINA

TOMÁŠ SOBEK (SK), MIROSLAV KVAŠŠAY (SK) AND ZIDONG LIN (CN)

**Abstract.** Quantum informatics is a rapidly growing field whose university-level teaching faces a fundamental pedagogical challenge: quantum hardware is typically invisible to students, accessible only as a remote cloud service. This paper describes the possibility of integrating of a desktop Nuclear Magnetic Resonance (NMR) quantum computer, the SpinQ Gemini Lab, into the teaching of quantum informatics at the Faculty of Management Science and Informatics, University of Žilina. We describe the physical principles of NMR quantum computing, the hardware characteristics of the device, and its software interface, which exposes three levels of abstraction: algorithm, gate, and pulse level. Based on these features, we propose a sequence of laboratory exercises progressing from single-qubit operations and measurement of  $T_1$  and  $T_2$  relaxation times to Bell state preparation, the Deutsch algorithm, and Grover's search. The platform is paired with open-source simulators such as Qiskit, allowing students to compare simulated and measured outcomes directly. The proposed approach bridges the gap between abstract quantum theory and physical hardware, and is expected to substantially increase student engagement by providing hands-on access to a real quantum device.

**Key words and phrases.** NMR, quantum computing, quantum informatics, qubit, SpinQ Gemini Lab.

## STOLOVÝ POČÍTAČ NA VÝUČBU KVANTOVEJ INFORMATIKY NA ŽILINSKEJ UNIVERZITE V ŽILINE

**Abstrakt.** Kvantová informatika patrí medzi rýchlo sa rozvíjajúce oblasti vedy, ktorých výučba na univerzitnej úrovni naráža na zásadný pedagogický problém: kvantový hardvér je pre študentov väčšinou neviditeľný a dostupný len ako vzdialená cloudová služba. Tento článok opisuje možnosť integrácie stolového kvantového počítača SpinQ Gemini Lab, ktorý využíva nukleárnu magnetickú rezonanciu (NMR), do výučby kvantovej informatiky na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline. V rámci článku stručne približujeme fyzikálny princíp NMR kvantových výpočtov, charakteristiky zariadenia a jeho softvérové rozhranie, ktoré sprístupňuje tri úrovne abstrakcie: algoritmickej, hradlovej a impulznej. Na základe týchto vlastností navrhujeme postupnosť laboratórnych cvičení, od operácií s jedným qubitom a merania relaxačných časov  $T_1$  a  $T_2$  až po prípravu Bellových stavov, implementáciu Deutschovho algoritmu a Groverovho vyhľadávania. Počítačová platforma je doplnená open-source simulátormi (Qiskit), čo umožňuje porovnanie simulovaných a reálne nameraných výsledkov. Navrhovaný prístup sa snaží premostiť priepasť medzi abstraktnou teóriou a fyzickým hardvérom a zvýšiť tak motiváciu študentov prostredníctvom priameho kontaktu s reálnym kvantovým zariadením.

**Kľúčové slová.** NMR, kvantové výpočty, kvantová informatika, qubit, SpinQ Gemini Lab.

## Introduction

Quantum information science studies how information can be stored, transmitted and processed using systems that obey the laws of quantum mechanics. Its elementary unit is the **quantum bit**, or qubit. Whereas a classical bit is always in one of two definite states, 0 or 1, a qubit may exist in a coherent superposition of both. a pure single-qubit state is written in Dirac (bra-ket) notation as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad |\alpha|^2 + |\beta|^2 = 1,$$

where  $\alpha$  and  $\beta$  are complex probability amplitudes. On measurement the qubit is found in state  $|0\rangle$  with probability  $|\alpha|^2$  and in  $|1\rangle$  with probability  $|\beta|^2$ , and the superposition collapses. Geometrically, a pure single-qubit state can be visualised as a point on the surface of the Bloch sphere, which makes single-qubit operations easy to picture as rotations.

Computation on a quantum computer proceeds by applying quantum gates, i.e., reversible, unitary operations, that are assembled into quantum circuits, followed by measurement of the output. Algorithms such as Shor's factoring algorithm, Grover's search and the simulation of quantum many-body systems show that, for certain tasks, quantum computers could in principle outperform any classical machine [6].

The power of quantum computing comes from two genuinely non-classical resources. The first is the exponential size of the state space: a register of  $n$  qubits is described by  $2^n$  complex amplitudes that all evolve simultaneously. The second is **entanglement**, a form of correlation between qubits that has no classical counterpart. For example, the two-qubit Bell state

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

cannot be written as a product of two independent single-qubit states.

Today's devices are still small and noisy, the so-called Noisy Intermediate-Scale Quantum (NISQ) era [7], and are mostly offered to users as remote cloud services. This creates a pedagogical problem. a student who submits a circuit to a distant superconducting processor receives a table of measurement counts but never sees a qubit, a control pulse, or the effect of decoherence; the hardware remains an abstract black box. a small, transparent, locally available quantum computer that students can operate themselves is therefore an attractive teaching aid. The remainder of this paper describes one such device, based on Nuclear Magnetic Resonance (NMR), and how it can be embedded in a single university course that we plan to open at the Faculty of Management Science and Informatics

## 1. NMR Quantum Computing

NMR was among the very first experimental platforms used to demonstrate quantum computation. In the late 1990s the idea of bulk-ensemble (liquid-state) NMR

quantum computing was introduced [1, 4], and within a few years NMR experiments had implemented small quantum algorithms, culminating in a seven-qubit demonstration of Shor's algorithm factoring the number 15 [10]. Although NMR is not a scalable route to large quantum computers, the quantum-control techniques it pioneered and its accessibility makes it well suited to physical experimentation in education.

### 1.1. Physical Principle

A nucleus with spin quantum number  $\frac{1}{2}$ , such as  $^1\text{H}$ ,  $^{13}\text{C}$ ,  $^{19}\text{F}$  or  $^{31}\text{P}$ , behaves like a tiny magnetic moment. When placed in a strong static magnetic field  $B_0$ , its two spin orientations acquire different energies (the Zeeman effect), giving two well-defined levels that serve as the qubit states  $|0\rangle$  and  $|1\rangle$ . The energy gap is

$$\Delta E = \hbar \gamma B_0 = \hbar \omega_0,$$

where  $\gamma$  is the gyromagnetic ratio of the nucleus and  $\omega_0 = \gamma B_0$  is its Larmor (resonance) frequency, typically in the Radio-Frequency (RF) range. Applying a resonant RF pulse drives transitions between the two levels: the duration, amplitude and phase of the pulse determine the rotation axis and angle on the Bloch sphere, which is exactly how single-qubit gates (such as the  $X$ ,  $Y$ , Hadamard and phase gates) are realised in hardware.

Two-qubit gates exploit the scalar ( $J$ -) coupling between nuclei in the same molecule. Combined with single-qubit rotations, this coupling allows entangling gates such as the controlled-NOT to be constructed. For a system of weakly coupled spins the internal Hamiltonian can be written, in units where  $\hbar = 1$ , as

$$H = \sum_j \frac{\omega_j}{2} Z_j + \sum_{j < k} \frac{\pi J_{jk}}{2} Z_j Z_k,$$

where  $Z_j$  is the Pauli- $Z$  operator of spin  $j$ ,  $\omega_j$  its resonance frequency, and  $J_{jk}$  the coupling strength between spins  $j$  and  $k$ . The first term distinguishes the qubits in frequency (so they can be addressed individually), while the second provides the interaction needed for entangling operations.

### 1.2. Readout, Ensembles and Decoherence

Measurement in NMR is collective rather than single-shot. After the circuit is applied, the precessing nuclear magnetisation induces a small oscillating voltage in a detection coil, the Free-Induction Decay (FID). a Fourier transform of the FID produces the NMR spectrum, in which the positions and intensities of the spectral peaks encode the final quantum state. Because the signal from a single molecule is far too weak to detect, an NMR quantum computer actually operates on an ensemble of order  $10^{18}$  identical molecules in a liquid sample at room temperature.

At room temperature this ensemble is in a highly mixed thermal state, very close to the maximally mixed state. The standard solution is to prepare a pseudo-pure state of the form

$$\rho = \frac{1-\varepsilon}{2^n} I + \varepsilon |\psi\rangle\langle\psi|,$$

in which only the small fraction  $\varepsilon$  of the ensemble behaves like the desired pure state  $|\psi\rangle$  while the rest contributes an unobservable uniform background (in the formula  $I$  denotes identity matrix and  $n$  number of qubits). The achievable coherence is ultimately limited by relaxation, characterised by the longitudinal ( $T_1$ ) and transverse ( $T_2$ ) relaxation times. These quantities are not merely nuisances: measuring  $T_1$  and  $T_2$  is itself an instructive experiment, which leads to observing decoherence directly.

For small-scale educational demonstrations, NMR can realize well-defined effective qubits, controllable gates, initialization through pseudo-pure state preparation, ensemble readout, and coherence times long enough for simple experiments. On the other hand, the signal-to-noise ratio degrades rapidly as qubits are added, which is why NMR does not scale to large processors. For a teaching device of two or three qubits, this limitation is irrelevant, and the use of mature, well-understood physics is an advantage.

## 2. The SpinQ Gemini Lab Platform: Two- and Three-Qubit Representations

SpinQ Technology has turned NMR quantum computing into a desktop appliance [2, 5]. The key engineering step is the replacement of bulky cryogenic NMR magnets with compact permanent magnets that provide a stable homogeneous magnetic field suitable for room-temperature NMR experiments. For the Gemini Lab and Gemini Mini series, the listed proton resonance frequencies are around 27 MHz, corresponding to a magnetic field of roughly 0.64 T. The current educational NMR devices operate at room temperature, with an acceptable ambient operating temperature of 0–40 °C, provided that the temperature change per hour remains below  $\pm 2$  °C. As a result, no liquid helium or nitrogen cooling is required. The devices are essentially maintenance-free and affordable enough for a teaching laboratory. The open case of the SpinQ Gemini Lab is shown in Figure 1.

### 2.1. The Device Family

SpinQ’s educational NMR product line includes compact systems for different teaching and research scenarios. The Gemini Lab is an open-chassis experimental platform supporting one-, two-, and three-qubit experiments through one-spin, heteronuclear two-spin, and homonuclear three-spin systems. It weighs about 18.5 kg and exposes internal modules such as the sample, magnet, shimming, RF, and main-control modules. The Gemini Mini and Gemini Mini Pro are portable



**Figure 1.** Open case of SpinQ Gemini Lab

two-qubit NMR quantum computers weighing about 14.5 kg, with a built-in touchscreen and an Android-based control interface. The current Triangulum II is a three-qubit desktop NMR quantum computer based on a homonuclear three-spin system, weighing about 23 kg and supporting pulse-sequence editing, graphical programming through SpinQuasar, and Python-based hybrid programming through SpinQit. The main characteristics are summarised in Table 1.

**Table 1.** The SpinQ educational NMR quantum-computer family.

| Device                 | Qubits | Qubit carrier (molecule / nuclei)                                                               | Weight   |
|------------------------|--------|-------------------------------------------------------------------------------------------------|----------|
| Gemini Lab             | 1–3    | One-spin / heteronuclear two-spin / homonuclear three-spin systems; interchangeable samples     | ~18.5 kg |
| Gemini Mini / Mini Pro | 2      | Dimethyl phosphite $(\text{CH}_3\text{O})_2\text{POH}$ , $^1\text{H}$ and $^{31}\text{P}$ spins | ~14.5 kg |
| Triangulum II          | 3      | Iodotrifluoroethylene $\text{C}_2\text{F}_3\text{I}$ , three $^{19}\text{F}$ spins              | ~23 kg   |

## 2.2. How Qubits Are Represented: Two and Three Qubits

In the two-qubit Gemini devices the qubits are the  $^1\text{H}$  and  $^{31}\text{P}$  nuclear spins of a single dimethyl-phosphite molecule,  $(\text{CH}_3\text{O})_2\text{POH}$ . Because the two nuclei have different resonance frequencies they can be addressed independently, and the  $J$ -coupling between them provides the two-qubit interaction. A two-qubit register spans the four basis states  $|00\rangle$ ,  $|01\rangle$ ,  $|10\rangle$  and  $|11\rangle$ , so a general state is described

by four complex amplitudes (a  $4 \times 4$  density matrix for mixed states). This is already enough to demonstrate the central phenomena of quantum computing: single-qubit superposition, the preparation of Bell states and entanglement, the controlled-NOT gate, the Deutsch algorithm, and Grover's search over a set of four items.

The three-qubit Triangulum devices use the three  $^{19}\text{F}$  (fluorine-19) nuclear spins of an iodotrifluoroethylene molecule,  $\text{C}_2\text{F}_3\text{I}$ , as the three qubits. a three-qubit register spans eight basis states and is described by eight amplitudes, which substantially enlarges the range of accessible experiments: three-party (GHZ) entanglement, a three-qubit Grover search, the Deutsch–Jozsa algorithm with a larger input, and the three-qubit quantum Fourier transform. Moving from two to three qubits is pedagogically valuable because the exponential growth of the state space, from four to eight amplitudes, becomes concrete rather than merely asserted.

### 2.3. Software and Levels of Access

The devices are driven by SpinQ's graphical control software, SpinQuasar, which supports quantum-computing experiments, NMR spectroscopy, calibration, task management, and example libraries. Current product materials also emphasize SpinQit for Python-based hybrid classical-quantum programming and SpinQLink SDK for programmatic access to circuit-level and physical-layer pulse experiments. In the broader SpinQ software ecosystem, the cloud platform supports OpenQASM 2.0 and Qiskit-compatible [8] workflows. This combination allows students to move between graphical circuit construction, code-based algorithm design, and low-level pulse experiments.

Students compose circuits in a drag-and-drop circuit editor and run them on the real device or on a built-in classical simulator, comparing the two. The software ships with demonstrations of more than ten well-known algorithms, including the Deutsch, Grover and HHL algorithms, and with built-in physics experiments such as Rabi-oscillation observation and the measurement of the  $T_1$  and  $T_2$  relaxation times.

Crucially for education, the platform exposes three levels of abstraction. At the **algorithm level**, the student thinks purely in terms of gates and circuits. At the **gate level**, the mapping of gates onto physical operations becomes visible. At the **pulse level**, the student designs the actual RF pulse sequences that implement the gates, linking the abstract circuit to real signal processing and control engineering. The same problem can thus be revisited at increasing depth as a course progresses.

### 3. Using Quantum Computer in Teaching

At the Faculty of Management Science and Informatics, our intention is to integrate the device into a single-semester course (or a substantial module within one), serving as the practical, laboratory component that complements the lectures on quantum informatics. The aim is not to train students to run production workload, a two- or three-qubit machine cannot do that, but to build physical intuition and to demystify a subject that is otherwise entirely mathematical and remote.

#### 3.1. A Progression of Laboratory Exercises

A natural sequence of laboratory sessions, increasing in conceptual demand, is the following.

- **Single-qubit operations and the Bloch sphere.** Students apply  $X$ ,  $Y$ , Hadamard and phase gates, observe Rabi oscillations as an RF pulse rotates a spin, and connect the abstract gate to a real physical rotation.
- **Decoherence in practice.** By measuring the  $T_1$  and  $T_2$  relaxation times the students see, with real data, why quantum information is fragile, i.e., a concept that simulators tend to hide.
- **Two qubits, superposition and entanglement.** Students prepare Bell states with a Hadamard gate followed by a controlled-NOT, then verify the resulting correlations from the measured spectrum.
- **Quantum algorithms.** The Deutsch and Deutsch–Jozsa algorithms (The Deutsch algorithm can be run on real device, as illustrated in Figure 2 with experimental results from SpinQ Gemini Lab) and Grover’s search are run on the real device, illustrating where quantum parallelism and interference come from.
- **Pulse-level control and NMR spectroscopy.** Advanced students design their own pulse sequences and use the spectroscopy module, linking quantum computing to signal processing, control theory and chemistry.

#### 3.2. Pedagogical Benefits and the Open-Source Ecosystem

The main benefit is tangibility. Students operate a real quantum device, see genuine measurement noise and decoherence, and obtain results that differ from the idealised textbook answer. This prompts productive discussion of gate fidelity, error and the limits of the model. As a consequence, the usual gap between abstract theory and physical hardware is bridged, which tends to raise student



**Figure 2.** Deutsch algorithm on two qubits.

motivation. Because the device is local and self-contained, sessions do not depend on queuing for a remote cloud machine.

A practical and economical approach is to pair the hardware with open-source quantum simulators. The physical device supplies intuition and authentic data for two or three qubits, while open frameworks such as Qiskit [8] let students design and simulate larger circuits that no small NMR machine could execute. Comparing simulated and measured outcomes for the same circuit is itself a valuable exercise, and the use of open, freely available software keeps the course affordable and reproducible. Suitable assessment activities include laboratory reports, a comparison of simulator versus real-device results, and a small final project implementing and running a chosen algorithm.

### 3.3. Limitations

The honest limitations of the platform should be taught explicitly rather than hidden, as they are themselves instructive. The machine offers only two or three qubits and is comparatively slow; it operates on a thermal ensemble using pseudo-pure states rather than a single isolated qubit, so the sense in which it is a quantum computer needs careful explanation; and NMR does not scale to large processors. None of this detracts from its value as a teaching instrument. On the contrary, confronting these constraints gives students a more realistic picture of the present state of quantum technology than a polished cloud interface would.



## 4. Conclusion

Quantum computing is hard to teach precisely because hardware is normally invisible to the learner. a compact, room-temperature NMR quantum computer such as the SpinQ Gemini or Triangulum II turns the subject into something students can touch: real qubits encoded in molecular nuclear spins, real gates implemented by RF pulses, and real decoherence visible in the data. With two- and three-qubit registers, a graphical and programmable software stack offering pulse-, gate- and algorithm-level access, and good integration with the OpenQASM standard and open-source simulators, the platform supports a coherent progression of laboratory exercises within a single university course. We plan to adopt exactly this approach, using the device as the hands-on counterpart to our lectures on quantum information, and we expect it to make an abstract subject markedly more concrete and engaging for students of Faculty of Management Science and Informatics.

## References

- [1] CORY, D. G. – FAHMY, A. F. – HAVEL, T. F.: *Ensemble quantum computing by NMR spectroscopy*, Proc. Natl. Acad. Sci. USA, Vol. **94**, str. 1634–1639 (1997).
- [2] FENG, G. – HOU, S.-Y. – ZOU, H. – SHI, W. – YU, S. et al.: *SpinQ Triangulum: a commercial three-qubit desktop quantum computer*, IEEE Nanotechnology Magazine, Vol. **16**, str. 20–29 (2022), DOI: 10.1109/mnano.2022.3175392.
- [3] FENG, G. – LU, D. – LI, J. – XIN, T. – ZENG, B.: *Quantum computing: principles and applications*, Spin, Vol. **13**, art. 2330001 (2023), DOI: 10.1142/S2010324723300013.
- [4] GERSHENFELD, N. A. – CHUANG, I. L.: *Bulk spin-resonance quantum computation*, Science, Vol. **275**, str. 350–356 (1997).
- [5] HOU, S.-Y. – FENG, G. – WU, Z. – ZOU, H. – SHI, W. et al.: *SpinQ Gemini: a desktop quantum computing platform for education and research*, EPJ Quantum Technology, Vol. **8**, art. 20 (2021), DOI: 10.1140/epjqt/s40507-021-00109-8.
- [6] NIELSEN, M. A. – CHUANG, I. L.: *Quantum Computation and Quantum Information*, Cambridge University Press, 2010, ISBN 978-0-521-63503-5.
- [7] PRESKILL, J.: *Quantum computing in the NISQ era and beyond*, Quantum, Vol. **2**, art. 79 (2018).
- [8] QISKIT CONTRIBUTORS: *Qiskit: an open-source framework for quantum computing* (2019), <https://www.qiskit.org>.
- [9] SUN, Q. – ZHOU, S. – CHEN, R. – FENG, G. – CHEUNG, K. T. – LI, J. – HOU, S.-Y. – ZENG, B.: *From computing to quantum mechanics: accessible and hands-on quantum computing education for high school students*, EPJ Quantum Technology, Vol. **11**, art. 58 (2024), DOI: 10.1140/epjqt/s40507-024-00271-9.
- [10] VANDERSYPEN, L. M. K. et al.: *Experimental realization of Shor’s quantum factoring algorithm using nuclear magnetic resonance*, Nature, Vol. **414**, str. 883–887 (2001).

## Contact addresses

**Ing. Tomáš Sobek**, Department of Informatics, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovakia,  
*e-mail*: `tomas.sobek@fri.uniza.sk`

**doc. Ing. Miroslav Kvaššay, PhD.**, Department of Informatics, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic,  
*e-mail*: `miroslav.kvassay@fri.uniza.sk`

**Zidong Lin**, Shenzhen SpinQ Technology Co., Ltd., 3rd Floor, Block A, Chuanglingtong Technology Building, Shihua Road, Fubao Community, Fubao Subdistrict, Futian District, Shenzhen, China,  
*e-mail*: `zlin@spinq.cn`.

## VYUŽITIE LLM PRI TVORBE TECHNICKÝCH ŠPECIFIKÁCIÍ PRE VEREJNÉ OBSTARÁVANIE

ŠTEFAN TOTH (SK) A JANA TOTHOVÁ (SK)

**Abstrakt.** Príspevok predstavuje praktický návrh riešenia na podporu tvorby technických špecifikácií pre verejné obstarávanie s využitím veľkých jazykových modelov (LLM). Cieľom navrhovaného systému je zjednodušiť a urýchliť prípravu podkladov pri obstarávaní produktov alebo služieb a zároveň podporiť tvorbu všeobecnejších a konzistentnejších technických opisov. Navrhované riešenie vychádza zo spracovania verejne dostupných produktových informácií z webových stránok. Získané textové podklady sú následne spracované pomocou LLM modelu s definovaným promptom, pričom výsledkom je návrh štruktúrovaných technických parametrov a opisov vhodných ako podklad pre elektronický kontraktačný systém alebo ďalšie spracovanie vo forme dokumentov. Príspevok sa zameriava najmä na praktické skúsenosti s návrhom riešenia, možnosti využitia lokálnych open-source modelov aj dostupných cloudových služieb, časovú úsporu pri príprave podkladov a potrebu manuálnej kontroly generovaných výstupov. Výsledkom je ukážka AI-assisted workflow, ktoré môže pomôcť zjednodušiť opakované administratívne a technické činnosti pri príprave verejného obstarávania.

**Kľúčové slová.** Verejné obstarávanie, veľké jazykové modely, technická špecifikácia, Gemini, Gemma.

## UTILIZING LLMS FOR THE CREATION OF TECHNICAL SPECIFICATIONS IN PUBLIC PROCUREMENT

**Abstract.** The paper presents a practical design of a solution to support the creation of technical specifications for public procurement using large language models (LLMs). The aim of the proposed system is to simplify and accelerate the preparation of documents in the procurement of products or services while supporting the creation of more general and consistent technical descriptions. The proposed solution is based on processing publicly available product information from web pages. The retrieved text documents are subsequently processed using an LLM with a defined prompt, resulting in a draft of structured technical parameters and descriptions suitable as inputs for the Electronic Contracting System (EKS) or further processing into documents. The paper focuses mainly on practical experiences with the design of the solution, the possibilities of using local open-source models as well as available cloud services, the time savings in document preparation, and the necessity of human verification of the generated outputs. The result is a demonstration of an AI-assisted workflow that can help simplify repetitive administrative and technical tasks in public procurement preparation.

**Keywords.** Public procurement, large language models, technical specification, Gemini, Gemma.

## 1. Úvod

Verejné obstarávanie v Slovenskej republike je regulované zákonom č. 343/2015 Z. z. o verejnom obstarávaní [7]. Podľa § 42 majú byť technické požiadavky určené tak, aby zabezpečili rovnaký prístup a hospodársku súťaž pre všetkých uchádzačov alebo záujemcov. Odkaz na konkrétneho výrobcu, obchodné označenie, patent alebo typ je prípustný iba v odôvodnenom prípade, keď predmet zákazky nemožno opísať dostatočne presne a zrozumiteľne iným spôsobom; vtedy musí byť doplnený slovami „alebo ekvivalentný“. Samotné odstránenie názvu značky však nediskriminačnosť nezaručuje. Aj presná kombinácia parametrov jedného referenčného produktu môže nepriamo vylúčiť konkurenčné riešenia. Návrh špecifikácie preto musí vychádzať z funkčnej potreby, primeraných hraničných hodnôt a znalosti trhu.

Na Katedre softvérových technológií FRI UNIZA sa podklady pre verejné obstarávanie (opisné formuláre a technické špecifikácie) dlhodobo pripravujú manuálnym prepisom údajov z internetových stránok do textových dokumentov (MS Word). Tento proces je vysoko repetitívny, náchylný na chyby a časovo náročný. Motiváciou pre predstavenú prácu bolo preto tento proces automatizovať pomocou veľkých jazykových modelov (LLM) a vytvoriť nástroj, ktorý by zjednodušil a urýchlil prípravu podkladov pri obstarávaní produktov alebo služieb a zároveň podporil tvorbu všeobecnejších a konzistentnejších technických opisov.

Tento príspevok popisuje možnosti automatizovaného získania textového obsahu z URL požadovaného produktu na obstaranie z internetu a jeho spracovania pomocou LLM. Systém nevytvára právne garantovanú špecifikáciu, ale editovateľný návrh, za ktorý vždy bude zodpovedať spracovateľ podkladov. Článok popisuje aj pracovný postup prototypu s využitím bezplatného cloudového modelu Gemini alebo s využitím lokálneho modelu, ako je napr. Gemma.

## 2. Pracovný postup navrhovaného riešenia

Prototyp navrhovaného systému s názvom **EKS Helper** je implementovaný v jazyku C# a poskytuje webové používateľské rozhranie vytvorené pomocou technológie Blazor. Riešenie je kontajnerizované pomocou Dockeru.

Najdôležitejšou funkciou systému je pridanie produktu s parametrami do zoznamu produktov na obstaranie. Toto je možné buď manuálne (obr. 1) alebo automatizovane (obr. 2). V prípade automatizovaného pridania prebieha proces v nasledujúcich fázach:

1. **Normalizácia URL adresy:** Vložená URL adresa je normalizovaná, odstraňujú sa z nej parametre, ktoré nie sú relevantné (napr. sledovacie a marketingové UTM parametre) a v prípade niektorých vybraných webových stránok sa aj modifikujú parametre (napr. pre Alza.sk sa nastavuje parameter pre zobrazenie karty Parametre).

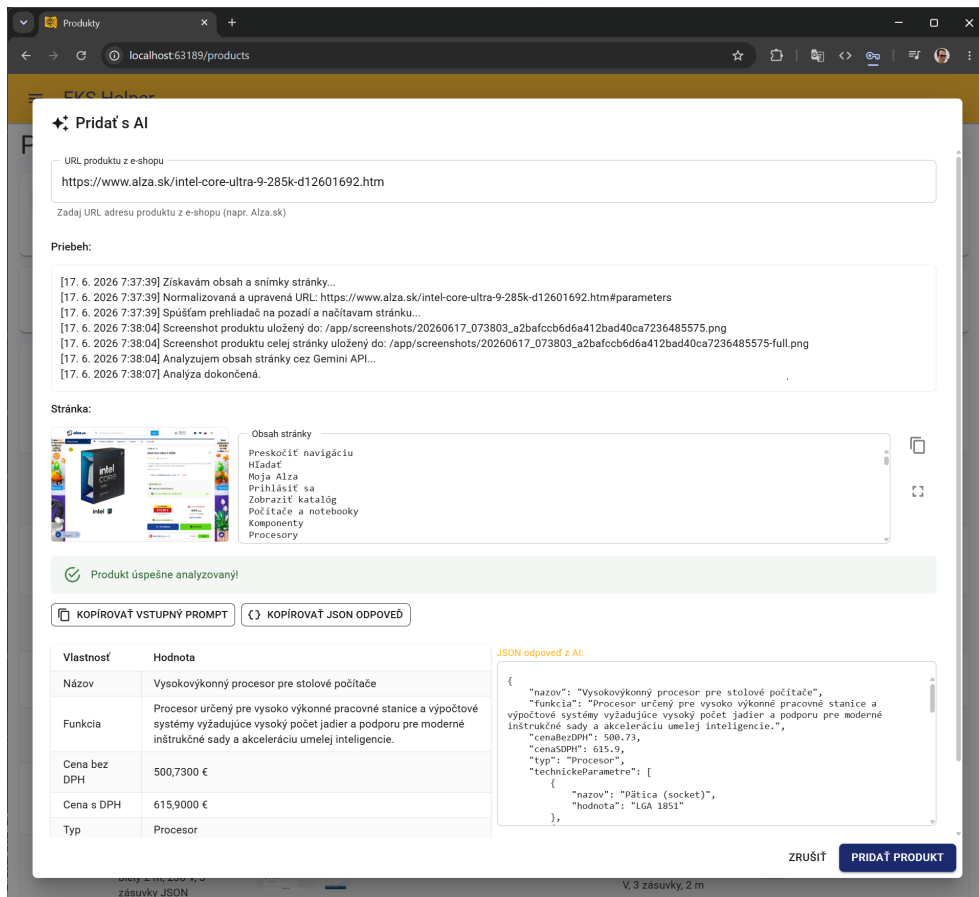
Obr. 1. Dialóg pre manuálne pridanie produktu a úpravu parametrov

Obr. 2. Dialóg pre automatizované pridanie produktu so spracovaním obsahu pomocou LLM

2. **Získanie obsahu:** Na stiahnutie obsahu stránky sa využíva prehliadač Chromium spravovaný knižnicou Playwright [4]. Systém na pozadí simuluje potvrdenie súhlasu so spracovaním cookies, aby bolo možné vyhotoviť kompletný snímok obrazovky (screenshot) celého produktu so zachytením ceny a jeho opisu. Tieto informácie spolu s dátumom a časom získania sa ukladajú ako auditovateľný podklad pre potreby prípravy opisného formulára pre EKS.
3. **Extrakcia textu a analýza pomocou LLM:** Po načítaní sa extrahuje čistý (plain) text bez HTML značiek, ktorý sa následne normalizuje (odstraňujú sa zbytočné viacnásobné prázdne riadky a medzery) a poskytne sa ako dátový vstup do LLM v štruktúre promptu, ktorý je opísaný nižšie. Výstupom bude JSON, ktorý obsahuje názov položky, účel, cenu a zoznam parametrov.
4. **Validácia výstupného JSON:** LLM vráti JSON, ktorého syntaktická validita sa skontroluje. Ide o technickú kontrolu, nie o potvrdenie právnej správnosti alebo primeranosti požiadavky.
5. **Výsledok:** Výsledok analýzy sa zobrazí v prehľadnom webovom rozhraní (obr. 3). Používateľ má možnosť prezrieť si zdrojový text, screenshot, výstupný JSON a z neho získané údaje. V prípade chybného spracovania LLM má možnosť skopírovať prompt a spustiť ho vo svojej preferovanej LLM službe (napr. ChatGPT, Claude, Gemini) s výsledným skopírovaním odpovede vo formáte JSON späť do systému, aby sa parametre pridali. Následne môže používateľ, po pridaní produktu do zoznamu, ručne upravovať parametre.

### 3. Tvorba promptu a integrácia LLM služieb

Kľúčovým prvkom spracovania je presne definovaný systémový prompt, ktorý stavia jazykový model do roly experta na verejné obstarávanie na Slovensku



Obr. 3. Dialóg zobrazujúci výsledok analýzy URL adresy pomocou LLM

a prípravu podkladov pre EKS. Prompt definuje striktnú výstupnú štruktúru JSON. Systémový prompt je uvedený vo výpise 1.

### Výpis 1. Systémový prompt na tvorbu technických špecifikácií

Si expert na verejné obstarávanie na Slovensku a prípravu podkladov pre EKS.  
Z priloženého textového obsahu webovej stránky extrahuj technickú špecifikáciu do štruktúrovaného JSON formátu.

#### PRAVIDLÁ:

- Ignoruj marketingový text, zameraj sa iba na technické, fyzikálne a softvérové parametre produktu z obsahu stránky produktu (ak sú v texte podobné produkty, ignoruj ich).
- Odstráň názvy značiek a modelov. Použi všeobecné, nediskriminačné formulácie.
  - napr. "Intel Core i5" -> "procesor min. X jadier alebo ekvivalent"

- napr. "Gorilla Glass" -> "tvrdené ochranné sklo"
- 3. Parametre musia byť:
  - objektívne
  - merateľné (ak je to možné)
  - použiteľné vo verejnom obstarávaní
- 4. Nepoužívaj marketingové frázy.
- 5. Výstup MUSÍ byť validný JSON bez akéhokoľvek textu navyše.
- 6. Každý parameter musí byť formulovaný tak, aby bol použiteľný v-súťažných podkladoch (t.j. nesmie jednoznačne odkazovať na konkrétneho výrobcu).
- 7. Ak je parameter nejednoznačný, preformuluj ho do minimálnej požiadavky (napr. "min .", "max.", "alebo ekvivalent").

#### VÝSTUPNÝ JSON FORMÁT:

```
{
  "nazov": "string",
  "funkcia": "string (max 600 znakov)",
  "cenaBezDPH": "float",
  "cenaSDPH": "float",
  "typ": "string (napr. Klávesnica, Monitor, Zdroj...)",
  "technickeParametre": [
    {
      "nazov": "string (max 600 znakov)",
      "hodnota": "string (max 200 znakov)"
    }
  ],
  "vyhovujePriklad": "string (konkrétny model produktu)"
}
```

#### DOPLŇUJÚCE PRAVIDLÁ:

- "funkcia" má byť krátky opis účelu zariadenia (nie parametre)
- "typ" je typ položky (napr. "Klávesnica", "HDD disk")
- "technickeParametre" obsahuje všetky parametre ako samostatné položky názov a-  
hodnota / charakteristika
- Ak hodnota nie je známa, použi null
- Nepoužívaj HTML ani markdown
- Nepoužívaj vysvetlenia

### 3.1. Cloudové a lokálne modely

Cloudový konektor využíva oficiálne SDK **Google.GenAI** a štruktúrovaný výstup podľa JSON schémy [1]. V čase použitia prototypu systému bol využívaný model **gemini-3.1-flash-lite**. Voľba tohto modelu bola motivovaná najmä tým, že bol k dispozícii zadarmo prostredníctvom bezplatného prístupu (Free Tier). Pre jeho integráciu bolo potrebné získať API kľúč cez platformu Google AI Studio (<https://aistudio.google.com/>). Ako sa ukázalo, hlavnou nevýhodou bezplatnej prevádzky bola nestabilita pri zvýšenej záťaži na tento model. API občas vracalo chybu **The service is temporarily running out of capacity**. Táto občasná nedostupnosť je priamou daňou za bezplatný charakter služby.

z tohto dôvodu bola do systému pridaná možnosť skopírovania promptu a jeho vykonanie používateľom v jeho preferovaných LLM cloudových službách. Ďalšou nevýhodou je, že obsah sa odosiela do cloudu, pričom Google uvádza, že sa v bezplatnej úrovni používa na zlepšenie jeho produktov (iba v platenej verzii sa prompty a odpovede nepoužívajú na zlepšovanie služieb). Avšak, keďže sa posielajú iba obsahy verejných stránok a žiadne citlivé údaje, toto je prijateľné.

Alternatívou k bezplatnému cloudu je lokálna inferencia cez endpoint kompatibilný s OpenAI API. Prototyp možno pripojiť k nástrojom ako napr. **LM Studio** [3] alebo **Ollama** [6]. Počas vývoja bol overovaný model s otvorenými váhami **Gemma 4 31B** s kvantizáciou Q6\_K z rodiny Gemma [2], ktorý sa nám zmestil do GPU s 32 GB VRAM (NVIDIA GeForce RTX 5090). Model bol spustený s kontextom nastaveným na hodnotu 15 000. Nevýhodou okrem vyšších nárokov na lokálny hardvér je aj to, že modely s otvorenými váhami sú často menej presné a menej robustné ako komerčné cloudové modely. Napr. sa stávalo, že tento lokálny model generoval niektoré slová pravopisne chybné v slovenčine alebo v iných jazykoch (napr. v českom jazyku).

Charakteristiky spôsobov nasadenia sú zhrnuté v tabuľke 1.

**Tabuľka 1.** Charakteristiky overovaných spôsobov nasadenia

| Variant           | Prevádzka      | Výhoda                        | Obmedzenie                                                            |
|-------------------|----------------|-------------------------------|-----------------------------------------------------------------------|
| Gemini Flash Lite | cloud          | bezplatný prístup (AI Studio) | občasné výpadky capacity API, prenos dát                              |
| Gemma 4 31B       | lokálna, dense | kontrola nad dátami           | vyššie nároky na lokálny hardvér, občas pravopisné chyby v generovaní |

#### 4. Praktické skúsenosti z vývoja a používania

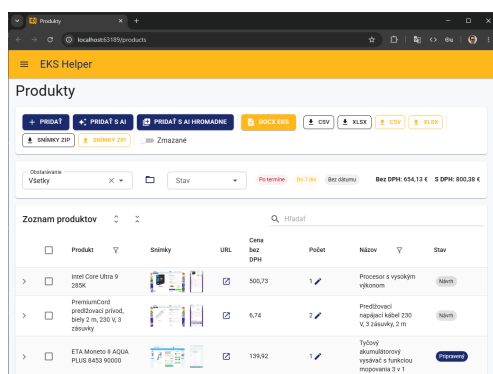
Počas testovania bol systém overovaný na rôznych typoch IT produktov (počítače, klávesnice, monitory) s cieľom posúdiť stabilitu analýzy a časovú úsporu v porovnaní s doterajším čisto manuálnym postupom.

Predbežné praktické testovanie naznačilo výraznú časovú úsporu. Kým manuálna príprava technického popisu jedného produktu (vrátane vyhľadania, čistenia textu, prekladu značiek na všeobecné požiadavky a prepisu do šablóny MS Word) zaberie pracovníkovi spravidla 5 až 15 minút, systém **EKS Helper** vygeneruje prvotný návrh parametrov za menej ako minútu. Najvýznamnejšou prevádzkovou metrikou pre prax je preto celkový čas do vytvorenia a schválenia finálneho dokumentu a počet manuálnych opráv, ktoré musí obstarávateľ vykonať.



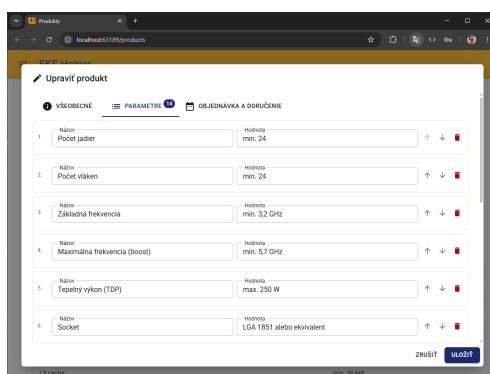
#### 4.1. Prípadová štúdia: Automatizované pridanie produktu

Praktická funkčnosť a proces spracovania sú ilustrované na príklade obstarávania procesora z internetového obchodu Alza.sk (obr. 3). Po zadaní URL adresy referenčného produktu (**Intel Core Ultra 9 285K**) systém vykoná normalizáciu URL adresy, načíta obsah stránky, vyhotoví snímky obrazovky a extrahuje textový obsah. Následne sa spustí analýza pomocou LLM, ktorá vygeneruje štruktúrovaný JSON výstup. Ten sa následne deserializuje do objektu a uloží sa do databázy pomocou EF Core. Získané dáta vrátane cien sú následne načítané do prehľadnej tabuľky vo webovom rozhraní, kde ich môže zodpovedný pracovník skontrolovať a schváliť pre export (obr. 4).



| Produkt                            | Cena bez DPH | Podat | Názov                                        | Stav   |
|------------------------------------|--------------|-------|----------------------------------------------|--------|
| Intel Core Ultra 9 285K            | 505,73       | 1     | Procesor s vysokým výkonom                   | Návrh  |
| PremiumCard                        | 6,74         | 2     | Predčíslovaný nápis na 200 V, 3 zásuvky, 2 m | Návrh  |
| ETA Monitor 8 AQUA PLUS 8433 80000 | 139,02       | 1     | Výkonový akumulátorový výpočet s funkciami   | Export |

Obr. 4. Zoznam produktov na obstarávanie



| Název                        | Hodnota                   |
|------------------------------|---------------------------|
| Počet jadier                 | min. 24                   |
| Počet vlákien                | min. 24                   |
| Základná frekvencia          | min. 3,2 GHz              |
| Maximálna frekvencia (boost) | min. 5,7 GHz              |
| Teplotný výkon (TDP)         | max. 250 W                |
| Socket                       | LGA 1851 alebo ekvivalent |

Obr. 5. Parametre vybraného produktu

#### 4.2. Overenie a ľudský faktor

Bezpodmienečne nutným prvkom celého workflow je ľudská kontrola (**Human-in-the-Loop**). Generovaný JSON slúži výhradne ako predbežný návrh. Používateľ vo webovom prostredí vizuálne overuje a v prípade potreby aj modifikuje parametre (obr. 5) a orientačné ceny oproti pôvodnej stránke a referenčným snímkam. Upravené a schválené parametre je možné exportovať do formátu MS Word (.docx), ktorý sa používa v procese verejného obstarávania na univerzite a následne sa spracúva v systéme EKS [5].

Riešenie stavia na otvorených technológiách a dôraz na technickú suverunitu. Využitie voľne dostupných vývojových prostriedkov a knižníc (.NET, Blazor, MudBlazor, Playwright) spolu s rozhraním kompatibilným s OpenAI API umožňuje prevádzkovať takýto systém lokálne s otvorenými modelmi (Gemma) bez závislosti od komerčného cloudového poskytovateľa.

## 5. Záver

Prototyp **EKS Helper** ukazuje, že LLM možno prakticky zapojiť medzi získanie produktových údajov a odbornú prípravu návrhu technických parametrov. Podstatným výsledkom nie je automatické vytvorenie „správnej“ špecifikácie, ale pracovný postup so štruktúrovaným výstupom, zachovaným zdrojovým vstupom a auditom ľudských zmien.

Cloudové a lokálne modely možno vďaka spoločnému rozhraniu zamieňať. Lokálna inferencia znižuje potrebu odosielať obsah externému API, no prináša zvýšené nároky na lokálny hardvér. Najväčším rizikom zostáva nepodložený parameter alebo neprimeraná hranica, preto je odborné schválenie a kontrola človekom povinnou súčasťou celého procesu.

## Literatúra

- [1] GOOGLE: *Gemini API – Models and structured outputs*, Google AI for Developers, 2026, <https://ai.google.dev/gemini-api/docs/models>.
- [2] GOOGLE DEEPMIND: *Gemma – open models and official variants*, 2026, <https://deepmind.google/models/gemma/>.
- [3] LM STUDIO: *Desktop application for running local LLMs*, 2026, <https://lmstudio.ai>.
- [4] MICROSOFT: *Playwright - Fast and reliable end-to-end testing for modern web apps*, Microsoft, 2026, <https://playwright.dev>.
- [5] MINISTERSTVO VNÚTRA SR: *Elektronický kontrakčný systém (EKS)*, Pravidlá správania a šablóny opisných formulárov, <https://www.eks.sk>.
- [6] OLLAMA: *Get up and running with large language models*, 2026. <https://ollama.com>.
- [7] ZÁKON Č. 343/2015 Z. Z. O VEREJNOM OBSTARÁVANÍ, § 42: časová verzia účinná od 30. mája 2026. <https://static.slov-lex.sk/static/SK/ZZ/2015/343/20260530.html>.

## Kontaktné adresy

**Ing. Štefan Toth, PhD.**, Katedra softvérových technológií, Fakulta riadenia a informatiky, Žilinská univerzita v Žiline, Univerzitná 8215/1, 010 26 Žilina, Slovensko,  
*E-mailová adresa:* [stefan.toth@uniza.sk](mailto:stefan.toth@uniza.sk)

**Ing. Jana Tothová, PhD.**, Univerzitný vedecký park, Žilinská univerzita v Žiline, Univerzitná 8215/1, 010 26 Žilina, Slovensko,  
*E-mailová adresa:* [jana.tothova@uniza.sk](mailto:jana.tothova@uniza.sk)

## PREKLAD KNÍH POMOCOU LOKÁLNYCH VEĽKÝCH JAZYKOVÝCH MODELOV: PRAKTICKÁ SKÚSENOSŤ S EPUB NÁSTROJMI

ŠTEFAN TOTH (SK) A JANA TOTHOVÁ (SK)

**Abstrakt.** Príspevok opisuje praktické použitie veľkých jazykových modelov pri osobnom preklade rozsiahlej EPUB knihy do slovenčiny a motiváciu hľadať otvorené lokálne alternatívy ku komerčným službám. Zameriava sa na nástroje TBL (Translate Books with LLMs) a EPUB Translator, skúsenosti s modelom Qwen 3.6 35B-A3B a hlavný beh s modelom Gemma 4 31B pri preklade 1329-stranovej technickej knihy. Hodnotí čas spracovania, tokeny, spotrebu elektrickej energie, zachovanie štruktúry EPUB a orientačné náklady v porovnaní s cloudovými modelmi. Diskusia uvádza limity prekladu po blokoch, paralelizácie, dávkového spracovania, vzdialených GPU služieb a právne obmedzenia.

**Kľúčové slová.** Veľké jazykové modely, preklad kníh, EPUB, agentický preklad, lokálne modely, Gemma, Qwen, slovenčina.

### BOOK TRANSLATION WITH LOCAL LARGE LANGUAGE MODELS: PRACTICAL EXPERIENCE WITH EPUB TOOLS

**Abstract.** The paper describes the practical use of large language models for personal translation of a large EPUB book into Slovak and the motivation to look for open local alternatives to commercial services. It focuses on TBL (Translate Books with LLMs) and EPUB Translator, reports experience with Qwen 3.6 35B-A3B, and describes the main run with Gemma 4 31B on a 1329-page technical book. It evaluates processing time, token counts, energy consumption, EPUB structure preservation, and indicative costs compared with cloud models. The discussion covers limits of chunk-based translation, parallelization, batch processing, remote GPU services, and legal constraints.

**Keywords.** Large language models, book translation, EPUB, agentic translation, local models, Gemma, Qwen, Slovak.

## Úvod

Pri preklade textov sa v praxi využíva viacero typov nástrojov: od bežných online prekladačov, ako sú Google Translate, Microsoft Translator alebo DeepL, cez profesionálne CAT (Computer-Assisted Translation) nástroje s prekladovými pamätami a glosármi až po cloudové API služby, lokálne open-source riešenia a nástroje založené na veľkých jazykových modeloch.

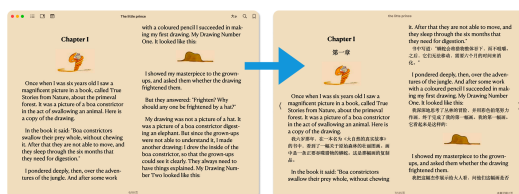
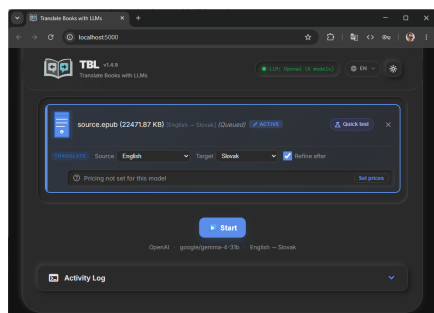
Veľké jazykové modely, napríklad modely z rodiny GPT, Gemini alebo Claude, dokážu lepšie pracovať s kontextom a štýlom, preto môžu pri vybraných úlohách dosahovať lepšie výsledky než bežné prekladače. Pri preklade celej knihy je však

potrebné riešiť rozdelenie textu na menšie časti, zachovanie nadväznosti medzi nimi a následné spojenie preloženého výstupu do výsledného dokumentu. Pri EPUB knihách je navyše dôležité zachovať kapitoly, nadpisy, obrázky, poznámky, odkazy a vnútornú HTML/XML štruktúru.

Pri skúmaní možností pre vlastný preklad nás najprv zaujala webová stránka Translate a Book with AI [7], ktorá po nahratí knihy vo viac ako 20 rôznych podporovaných formátoch poskytuje cenový odhad prekladu, pričom deklaruje zachovanie rozloženia dokumentu pri EPUB alebo DOCX. Jej autor Robin Moisson opisuje, že platformu začal budovať z frustrácie z dostupných možností prekladu a s cieľom vytvoriť jednoduchšiu službu s kvalitnejším výstupom. V časti FAQ uvádza, že používa modely Google Gemini 2.5 Pro, OpenAI GPT-5 a DeepSeek V3 [7]. Po nahratí technickej knihy s 1329 stranami stránka zobrazila náhľad, uviedla 158 476 slov a ponúkla preklad v režimoch Standard za 23,10 €, Pro za 72,40 € a Author za 370,90 €, čo bol pre nás praktický impulz hľadať lokálne open-source alternatívy.

Cieľom príspevku je opísať našu praktickú skúsenosť s lokálnym prekladom tejto rozsiahlej EPUB knihy, použitými nástrojmi, nameranými tokenmi, časom spracovania, orientačnými nákladmi a limitmi takéhoto postupu.

## 1. Nástroje na preklad kníh



**Obr. 1.** Ukážky riešení: vľavo používateľské rozhranie TBL, vpravo dvojjazyčný výstup v nástroji EPUB Translator. Zdroj obrázka EPUB Translator: dokumentácia projektu [8].

### 1.1. TBL (Translate Books with LLMs)

Projekt TBL je webová aplikácia na preklad kníh a dokumentov pomocou lokálnych alebo cloudových modelov. Podporuje formáty EPUB, DOCX a TXT. V dokumentácii sa zdôrazňuje, že nástroj používa inteligentné delenie textu na časti, vie pokračovať po prerušení prekladu a snaží sa zachovať štruktúru dokumentu vrátane EPUB formátovania [4].

Dôležité je, že nástroj podporuje cloudové služby, lokálne prostredia typu Olama aj servery kompatibilné s OpenAI API [4]. V experimente sme využili lokálny server desktopovej aplikácie LM Studio, ktorá umožňuje spustiť lokálne LLM, ako sú modely Qwen alebo Gemma, na vlastnom počítači, pričom API vystavuje na localhoste, štandardne `http://127.0.0.1:1234`.

V experimente bol TBL nasadený lokálne cez Docker kontajner. Použitý bol obraz `ghcr.io/hydropix/translatebookswithllms` a konfiguračný súbor `.env`, v ktorom sa nastavuje najmä model a API endpoint. Webové rozhranie umožňuje nahráť EPUB knihu, zvoliť cieľový jazyk a zapnúť dodatočné vylepšenie prekladu voľbou **Refine after** (obr. 1). Skúsenosti z prvého testovania tohto nástroja sú zhrnuté v praktickej časti.

## 1.2. EPUB Translator

Projekt EPUB Translator od OOMOL Lab sa špecializuje na preklad EPUB kníh a umožňuje aj vytváranie dvojjazyčných EPUB výstupov. Jeho cieľom nie je iba nahradiť pôvodný text prekladom, ale vytvoriť výstup, v ktorom je pôvodný aj preložený text dostupný spolu. Dokumentácia uvádza, že nástroj zachováva formátovanie, obrázky a štruktúru knihy [8].

Nástroj podporuje viacero režimov vloženia prekladu. **REPLACE** nahradí pôvodný text prekladom, **APPEND\_TEXT** vloží preklad priamo za pôvodný text a **APPEND\_BLOCK** vloží preklad ako samostatný blok, čo je odporúčaný režim pre dvojjazyčné čítanie [8].

**Tabuľka 1.** Prehľad vybraných nástrojov na preklad kníh a EPUB dokumentov.

| Nástroj                             | Licencia | Spôsob použitia                                      | Úloha v experimente                                               |
|-------------------------------------|----------|------------------------------------------------------|-------------------------------------------------------------------|
| TBL [4]                             | AGPL-3.0 | Webové rozhranie, Docker, lokálne aj cloudové LLM    | Prvý testovaný nástroj; pri Qwen vykazoval chyby a fallbacky.     |
| EPUB Translator [8]                 | MIT      | Python/API nástroj pre EPUB, lokálne aj cloudové LLM | Hlavný použitý nástroj; lepšie zachoval štruktúru EPUB dokumentu. |
| Ebook Translator Calibre Plugin [2] | GPL-3.0  | Plugin pre Calibre na preklad elektronických kníh    | Praktická alternatíva pre používateľov Calibre.                   |
| bilingual book maker [13]           | MIT      | CLI nástroj na tvorbu dvojjazyčných EPUB kníh        | Príbuzný OSS projekt pre dvojjazyčné výstupy.                     |

Z pohľadu implementácie je zaujímavé, že EPUB Translator umožňuje použiť aj Dual-LLM architektúru, kde sa používa samostatný model na samotný preklad a iný model alebo iné nastavenia na spätné dopĺňanie prekladu do XML štruktúry. Dokumentácia uvádza parametre `translation_llm` a `fill_llm`, čo umožňuje optimalizovať jeden model na jazykovú úlohu a druhý na presnosť práce so štruktúrou dokumentu.

Pre praktické nasadenie je dôležité, že EPUB Translator možno použiť programovo z Pythonu a nastavovať model, endpoint, paralelizáciu aj režim vloženia prekladu. Vďaka tomu bol vhodným kandidátom na hlavný experiment opísaný v ďalšej časti.

### 1.3. Prehľad nástrojov a licencie

Okrem dvoch prakticky skúšaných riešení existujú aj ďalšie projekty, ktoré riešia preklad EPUB alebo širší preklad dokumentov pomocou LLM. Tabuľka 1 sumarizuje ich spôsob použitia, licenciu a úlohu v experimente.

Pri nástrojoch na spracovanie dokumentov treba sledovať licenciu samotného nástroja aj použitého modelu. Pri cloudových API nejde iba o cenu za tokeny, ale aj o limity služby, režim bezplatného alebo plateného používania, uchovávanie dát a prípadné použitie vstupov a výstupov na zlepšovanie služieb. Samostatne treba posudzovať aj autorské práva k prekladanému dielu.

## 2. Preklad EPUB knihy pomocou LLM

Pri preklade knihy nejde o jednoduché volanie modelu s textom „prelož toto“. EPUB súbor je ZIP archív obsahujúci XHTML dokumenty, štýly, obrázky, metadáta a navigačné súbory. Prekladový nástroj preto musí rozbaľiť EPUB, identifikovať textové časti, rozdeliť ich na bloky podľa tokenového limitu, zavolať LLM, vložiť preklad späť do dokumentovej štruktúry a vytvoriť nový EPUB. Pri dlhých knihách je kritické najmä delenie textu: model nemôže dostať naraz 1300-stranovú knihu. EPUB Translator na to používa parameter `max_group_tokens`, predvolene 2600 tokenov, a parameter `concurrency`, ktorým možno riadiť počet paralelných prekladových úloh [8]. V prípade lokálneho modelu je vhodné nastaviť `concurrency` na 1, aby sa predišlo súbežným volaniam a prípadným problémom s pamäťou GPU. Pri cloudových modeloch možno využiť paralelizáciu, čím sa môže urýchliť čas spracovania celej knihy.

Z technického hľadiska je pri nástroji EPUB Translator významné použitie dvoch samostatných Jinja promptov. Prvý prompt `translate.jinja` rieši čistý preklad textu, druhý prompt `fill.jinja` vkladá preklad späť do existujúcej XML štruktúry pri zachovaní značiek, poradia, vnorenia a identifikátorov [8]. Tento dvojkrokový postup je pri EPUB knihách robustnejší než jediné volanie typu „prelož text a zachovaj značky“, pretože znižuje riziko poškodenia dokumentovej štruktúry. Keďže Jinja šablóny sú súčasťou každej požiadavky, ich dĺžka zároveň ovplyvňuje počet vstupných tokenov aj cenu.

## 3. Praktický experiment

V praktickej časti bol použitý technický dokument vo formáte EPUB s rozsahom 1329 strán. Najprv bol skúšaný nástroj TBL s modelom `qwen/qwen3`.

6-35b-a3b, kontextovým oknom približne 15 000 tokenov a Q6\_K kvantizáciou. Pri tomto behu bola veľkosť blokov nastavená na 5000 tokenov, čo mohlo prispieť k fallbackom; menšie bloky sme už nestihli systematicky overiť. Qwen však opakovane nedodržiaval požadovanú štruktúru odpovede alebo neodpovedal korektne, takže výsledok nebol spoľahlivý.

Následné použitie modelu `google/gemma-4-31b` s rovnakým kontextovým oknom a kvantizáciou bolo menej problémové, ale ani tu sa fallbacky úplne nestratili.

Z praktického hľadiska sa preto ako vhodnejšie riešenie ukázal EPUB Translator, najmä pre jeho oddelenie čistého prekladu od spätného dopĺňania textu do XML štruktúry EPUB dokumentu. Hlavný meraný beh bol realizovaný lokálne na počítači s konfiguráciou uvedenou v tabuľke 2.

**Tabuľka 2.** Použité prostredie a nastavenie prekladu.

| Parameter             | Hodnota                                |
|-----------------------|----------------------------------------|
| Operačný systém       | Windows 11                             |
| GPU                   | NVIDIA RTX 5090                        |
| VRAM                  | 32 GB                                  |
| Meraný príkon zostavy | približne 750 W, merané cez UPS APC    |
| Lokálny runtime       | LM Studio/OpenAI-kompatibilné API      |
| Použitý model         | <code>google/gemma-4-31b</code>        |
| Typ dokumentu         | technická EPUB kniha                   |
| Rozsah                | 1329 strán                             |
| Cieľový jazyk         | slovenčina                             |
| Nástroj               | <code>oomol-lab/epub-translator</code> |

Keďže pri nasadzovaní preferujeme Docker a zdrojový kód EPUB Transлятора neobsahoval Dockerfile, pripravili sme vlastný Dockerfile a spustili malý obalový skript v Pythone nad jeho API. Konfigurácia sa čítala z premenných prostredia, takže bolo možné meniť vstup, výstup, model, endpoint, paralelizáciu aj režim vloženia prekladu bez úpravy programu. s výsledkami sme boli spokojní: preklad bol kompletný, štruktúra EPUB dokumentu zostala zachovaná a výstup bol použiteľný na osobné čítanie. Niekoľko chýb však zostalo, napríklad občasné české slová namiesto slovenských, medzery pred guľatými zátvorkami alebo chybné formátovanie zdrojových kódov, čo pravdepodobne súvisí hlavne s použitým lokálnym modelom (obr. 2). Treba tiež poznamenať, že obrázky tento nástroj neprekladá.

Celkovo trval preklad 18 hodín, 1 minútu a 29,6 sekundy. Počas spracovania bol príkon celej pracovnej stanice s GPU podľa merania cez UPS APC približne 750 W. Pri trvaní 18,025 hodiny to zodpovedá orientačnej spotrebe približne 13,5 kWh. Pri modelovej cene elektriny 0,20–0,25 €/kWh by samotná elektrická energia pre tento beh predstavovala približne 2,70–3,38 €. Nižšia hodnota môže zodpovedať realistickému prímeru pre domácnosť, vyššia slúži ako opatrnejší odhad

By default, with the .NET SDK CLI and most code editor-created projects, if you need to reference a NuGet package, you add the reference to the package name and version directly in the project file, as shown in the following markup:

Predvolene pri použití .NET SDK CLI a väčšiny projektov vytvorených v editoroch kódu pridávate referenciu na názov a verziu balíka priamo do súboru projektu, ak potrebujete referovať NuGet balík, ako je znázornené v nasledujúcom výpise:

```
<ItemGroup>
  <PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer"
    Version="10.0.0" />
  ...
</ItemGroup>
```

```
&lt;ItemGroup&gt;
  <PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="10.0.0" /> ...
&lt;/ItemGroup&gt;
```

**Central Package Management (CPM)** is a feature that simplifies the management of NuGet package versions across multiple projects and solutions within a directory hierarchy. This is particularly useful for large solutions with many projects, where managing package versions individually can become cumbersome and error-prone.

**Centrálny manažment balíkov ( CPM )** je funkcia, ktorá zjednodušuje správu verzií NuGet balíkov v rámci viacerých projektov a riešení v hierarchii priečinkov. Toto je obzvlášť užitočné pre veľké riešenia s mnohými projektmi, kde môže byť individuálna správa verzií balíkov náročná a náchylná na chyby.

**Obr. 2.** Ukážka dvojjazyčného výstupu preloženej knihy v Calibre.

s rezervou. Táto suma však nezahŕňa obstarávaciu cenu hardvéru, opotrebenie, čas používateľa ani náklady na prípadné opakované behy.

Z logov a súhrnných štatistík vyplývajú hodnoty v tabuľke 3. Počet strán treba chápať ako orientačný údaj podľa použitého zobrazenia EPUB dokumentu; pri EPUB formáte je presnejšou technickou metrikou počet tokenov, kapitol, XHTML súborov alebo prekladových blokov.

**Tabuľka 3.** Základné vyhodnotenie prekladu jednej vybranej knihy.

| Metrika                            | Hodnota           |
|------------------------------------|-------------------|
| Strán                              | 1329              |
| Celkové tokeny                     | 6 505 090         |
| Vstupné tokeny                     | 2 965 447         |
| Výstupné tokeny                    | 3 539 643         |
| Pomer výstupných/vstupných tokenov | 1,19×             |
| Priemer na stranu                  | cca 4 895 tokenov |
| Vstup na stranu                    | cca 2 231 tokenov |
| Výstup na stranu                   | cca 2 663 tokenov |
| Rýchlosť podľa celkových tokenov   | cca 100,25 tok/s  |
| Rýchlosť podľa výstupných tokenov  | cca 54,55 tok/s   |
| Strán za hodinu                    | cca 73,7          |
| Čas na 1 stranu                    | cca 48,8 s        |



Z technického a výkonového hľadiska ide o veľmi dobrý praktický výsledok: celý EPUB dokument bol spracovaný lokálne, výstup zostal použiteľný na osobné čítanie a priebeh bolo možné vyhodnotiť podľa tokenov, času a spotreby. Toto tvrdenie sa však netýka systematicky meranej jazykovej kvality prekladu, ktorá v tomto článku nebola predmetom podrobného hodnotenia.

#### 4. Orientačné porovnanie ceny

Ak použijeme nameraný počet tokenov ako aproximáciu, vstup predstavuje 2,965447 milióna tokenov a výstup 3,539643 milióna tokenov. Odhad ceny možno počítať vzťahom:

$$\text{cena} = \text{input}_{\text{mil}} \times \text{cena}_{\text{input}} + \text{output}_{\text{mil}} \times \text{cena}_{\text{output}}$$

Takýto výpočet je však iba orientačný, pretože Gemini, GPT, Claude a lokálny Gemma model môžu používať rozdielnu tokenizáciu. Pri modeloch s režimom thinking alebo reasoning môžu navyše pribudnúť interné reasoning tokeny. Gemini pri niektorých režimoch uvádza, že cena výstupu zahŕňa aj thinking tokeny; reasoning modely od OpenAI pracujú s internými reasoning tokenmi navyše. Ceny v tabuľke 4 boli overené 19. 6. 2026 na oficiálnych stránkach s cenníkmi Google Gemini API, OpenAI API a Anthropic Claude API [1, 3, 9].

**Tabuľka 4.** Orientačný odhad ceny pri cloudových modeloch v USD ku dňu 19. 6. 2026.

| Model                                | Cena vstup/výstup       | Odhad ceny           |
|--------------------------------------|-------------------------|----------------------|
|                                      | za 1 mil. tokenov (USD) | prekladu knihy (USD) |
| Claude Haiku 4.5                     | 1,00/5,00               | 20,66                |
| Claude Sonnet 4.6                    | 3,00/15,00              | 61,99                |
| Claude Opus 4.8                      | 5,00/25,00              | 103,32               |
| Claude Fable 5                       | 10,00/50,00             | 206,64               |
| Gemini 2.5 Flash                     | 0,30/2,50               | 9,74                 |
| Gemini 2.5 Pro – do 200 tis. tokenov | 1,25/10,00              | 39,10                |
| Gemini 3.1 Flash-Lite                | 0,25/1,50               | 6,05                 |
| Gemini 3.5 Flash                     | 1,50/9,00               | 36,30                |
| GPT-5.4 nano                         | 0,20/1,25               | 5,02                 |
| GPT-5.4 mini                         | 0,75/4,50               | 18,15                |
| GPT-5.4 – do 272 tis. tokenov        | 2,50/15,00              | 60,51                |
| GPT-5.5 – do 272 tis. tokenov        | 5,00/30,00              | 121,02               |
| GPT-5.5 pro – do 272 tis. tokenov    | 30,00/180,00            | 726,10               |

Pri kurze 1 USD  $\approx$  0,87 € v čase písania článku by to prakticky znamenalo, že pri jednej takejto knihe by najlacnejšie uvedené cloudové riešenie (GPT-5.4 nano) mohlo stáť približne 5,02 USD, teda asi 4,37 €, kým najdrahší uvedený variant (GPT-5.5 pro) by stál 726,10 USD, teda asi 631,71 €. Ak výsledok nie

je potrebný okamžite, ďalšiu úsporu môžu priniesť dávkové (batch) spracovania, kde sú ceny polovičné oproti štandardným volaniam [1, 3, 9]. Napríklad GPT-5.4 nano by v takom prípade vyšiel približne na 2,51 USD (2,18 €), čo vychádza lacnejšie než vypočítaná spotrebovaná elektrická energia pri lokálnom behu.

Popri platených cenách existujú aj obmedzené bezplatné režimy. z porovnávajúcich služieb ponúka v čase písania článku bezplatný API režim (free tier/unpaid services) najmä Google Gemini pre vybrané modely. Treba však upozorniť, že vstupy aj výstupy môžu byť použité na zlepšovanie produktov Google [3], čo môže zahŕňať aj tréning nového modelu. Pri chránených dielach preto môže vzniknúť právne riziko. Pri platených modeloch (paid services) Google uvádza, že vstupné prompty ani odpovede nepoužíva na zlepšovanie produktov, hoci si ich na obmedzený čas ponecháva z bezpečnostných, právnych alebo regulačných dôvodov. Pri používaní viacerých bezplatných modelov od Google sme sa stretli aj s odpoveďou HTTP 503 pri preťažení serverov, čo narúšalo spoľahlivosť spracovania. Pri paralelnom spracovaní môže API vracaať aj chybu 429, pretože má limity na rýchlosť volaní za minútu a za deň. Bezplatná služba je preto vhodná skôr na overenie postupu na krátkej vzorke než ako spoľahlivý spôsob prekladu celého EPUB súboru.

Už spomenutý komerčný odhad Translate a Book with AI pre tú istú knihu bol 23,10 €, 72,40 € alebo 370,90 € podľa režimu. Režim Standard sa tak v orientačnom prepočte nachádza medzi lacnejšími cloudovými modelmi a vlastným volaním výkonnejších modelov. Nie je to však priamo porovnateľné so samotným volaním API, pretože služba zahŕňa aj maržu poskytovateľa a náklady na spracovanie súboru.

Lokálne riešenie nemá priamu cenu za tokeny, ale vyžaduje výkonný hardvér, elektrinu a čas na nastavenie. Dáva preto zmysel najmä vtedy, ak používateľ prekladá veľa kníh, potrebuje súkromie, nechce odosielať obsah do cloudu alebo chce experimentovať s vlastnou prekladovou pipeline.

## 5. Diskusia a limity

Skúsenosť s prekladom rozsiahlej technickej knihy ukazuje, že LLM preklad pre osobné účely je už prakticky použiteľný, nejde však o náhradu profesionálneho prekladateľa. Výsledok treba chápať najmä ako pracovný preklad na čítanie a štúdium, nie ako redakčne pripravený preklad na zverejnenie.

Z praktického hľadiska sa ukazujú tri dôležité kompromisy. Prvým je vzťah medzi časom a cenou. Pri cloudových API možno pri potrebe rýchleho výsledku odosielať viaceré bloky paralelne ako štandardné požiadavky, čo môže výrazne skrátiť celkový čas prekladu. Dávkové spracovanie je zasa vhodné vtedy, keď neprekáža asynchrónne čakanie a cieľom je najmä znížiť cenu. Lokálny beh na jednej pracovnej stanici s jednou grafickou kartou je v tomto smere obmedzený výkonom jedného zariadenia, hoci prináša väčšiu kontrolu nad dátami. Prenájom

GPU inštancie v AI cloude, napríklad cez RunPod, preto netreba chápať ako najlacnejšiu cestu pre jednu knihu, ale skôr ako možnosť vtedy, keď používateľ nemá vlastnú výkonnú GPU alebo chce krátkodobo testovať výkonnejší model bez nákupu hardvéru [10].

Druhým kompromisom je práca s tokenmi. Do vstupu sa nezapočítava iba text knihy, ale aj systémové a Jinja prompty, ktoré určujú správanie modelu pri preklade a spätnom vložení textu do EPUB štruktúry. Kratšie a výstižnejšie šablóny by mohli znížiť počet vstupných tokenov, no prílišné zjednodušenie môže zhoršiť úplnosť výstupu alebo poškodiť štruktúru dokumentu.

Tretím limitom je práca s kontextom. Skúšané nástroje prekladali bloky prevažne nezávisle, preto predpokladáme, že pri beletrii by mohol vzniknúť problém s konzistenciou mien postáv, predmetov, oslovení alebo opakovaných pojmov. Podobný problém rieši aj výskum dokumentového strojového prekladu, ktorý zdôrazňuje význam kontextu, konzistencie a pamäťových mechanizmov pri dlhších textoch [5, 11, 12]. Pomôcť by mohli glosáre, prekladové pamäte, zhrnutia kapitol alebo agentické postupy s pamäťou.

V tomto experimente sa nerealizovalo hodnotenie jazykovej kvality, terminologickej konzistentnosti ani porovnanie viacerých modelov na rovnakej vzorke. Preklad pre osobné čítanie je zároveň iná situácia než zverejnenie preloženej knihy; výstupy z prekladu chránených diel by sa nemali distribuovať bez príslušných práv [6]. Takýto výstup je preto vhodné chápať ako súkromnú pracovnú kópiu, pričom pri zdieľaní, publikovaní alebo inom sprístupnení mimo vlastnej potreby treba samostatne posúdiť licenciu a súhlas nositeľov práv.

## 6. Záver

Lokálne veľké jazykové modely umožňujú vytvárať použiteľné pracovné preklady rozsiahlych EPUB kníh do slovenčiny. V opísanom experimente bolo možné preložiť technickú knihu s rozsahom 1329 strán na jednej pracovnej stanici a získať výstup použiteľný na osobné čítanie. Hlavnou výhodou lokálneho prístupu je súkromie, kontrola nad spracovaním a možnosť experimentovať s vlastnou pipeline; nevýhodou je dlhší čas spracovania a potreba výkonnej GPU. Vzdialená GPU inštancia môže byť doplnkovou možnosťou pri chýbajúcom hardvéri alebo testovaní výkonnejších modelov, nie však automaticky lacnejšou alternatívou k API pri preklade jednej knihy [10].

Hoci sme v tomto príspevku neporovnávali kvalitu výstupu, domnievame sa, že použitie veľkých platených cloudových modelov by mohlo priniesť kvalitnejší preklad a pri paralelnom spracovaní aj kratší celkový čas spracovania. Lokálne riešenie dáva zmysel najmä pri opakovanom preklade, požiadavke na súkromie a technickom experimentovaní.

## Literatúra

- [1] ANTHROPIC: *Claude API pricing; Message Batches*, <https://platform.claude.com/docs/en/about-claude/pricing>; <https://platform.claude.com/docs/en/build-with-claude/batch-processing> [cit. 2026-06-19].
- [2] BOOKFERE: *Ebook Translator Calibre Plugin*, <https://github.com/bookfere/Ebook-Translator-Calibre-Plugin> [cit. 2026-06-19].
- [3] GOOGLE: *Gemini Developer API pricing; Batch API; Gemini API Additional Terms of Service*, <https://ai.google.dev/gemini-api/docs/pricing>; <https://ai.google.dev/gemini-api/docs/batch-api>; <https://ai.google.dev/gemini-api/terms> [cit. 2026-06-19].
- [4] HYDROPIX: *TranslateBooksWithLLMs – README*, <https://github.com/hydropix/TranslateBooksWithLLMs> [cit. 2026-06-18].
- [5] KARPINSKA, M.; IYYER, M.: *Large Language Models Effectively Leverage Document-level Context for Literary Translation, but Critical Errors Persist*. In Proceedings of WMT 2023, s. 419–451, <https://aclanthology.org/2023.wmt-1.41/> [cit. 2026-06-19].
- [6] LITA: *Rozmnožovanie pre osobnú potrebu*, <https://www.lita.sk/rozmnozovanie-pre-oso-bnu-potrebu> [cit. 2026-06-18].
- [7] MOISSON, R.: *Translate a Book with AI; About*, <https://translateabook.com/>; <https://translateabook.com/about> [cit. 2026-06-19].
- [8] OOMOL-LAB: *EPUB Translator – GitHub repozitár a dokumentácia projektu*, <https://github.com/oomol-lab/epub-translator> [cit. 2026-06-18].
- [9] OPENAI: *API Pricing; Batch API*, <https://openai.com/api/pricing/>; <https://platform.openai.com/docs/guides/batch> [cit. 2026-06-19].
- [10] RUNPOD: *GPU Cloud Pricing*, <https://www.runpod.io/pricing> [cit. 2026-06-19].
- [11] WANG, L. et al.: *Document-Level Machine Translation with Large Language Models*. In Proceedings of EMNLP 2023, s. 16646–16661, <https://aclanthology.org/2023.emnlp-main.1036/> [cit. 2026-06-19].
- [12] WANG, Y. et al.: *DelTA: An Online Document-Level Translation Agent Based on Multi-Level Memory*, <https://arxiv.org/abs/2410.08143> [cit. 2026-06-19].
- [13] YIHONG0618: *bilingual\_book\_maker*, [https://github.com/yihong0618/bilingual\\_book\\_maker](https://github.com/yihong0618/bilingual_book_maker) [cit. 2026-06-19].

## Kontaktné adresy

**Ing. Štefan Toth, PhD.**, Katedra softvérových technológií, Fakulta riadenia a informatiky, Žilinská univerzita v Žiline, Univerzitná 8215/1, 010 26 Žilina, Slovensko,  
*E-mailová adresa:* [stefan.toth@uniza.sk](mailto:stefan.toth@uniza.sk)

**Ing. Jana Tothová, PhD.**, Univerzitný vedecký park, Žilinská univerzita v Žiline, Univerzitná 8215/1, 010 26 Žilina, Slovensko,  
*E-mailová adresa:* [jana.tothova@uniza.sk](mailto:jana.tothova@uniza.sk)

## KOLABORATIVNÍ PŘÍPRAVA SBORNÍKU ABSTRAKTŮ POMOCÍ XSLT, T<sub>E</sub>XU A SUBVERSION

ZDENĚK WAGNER (CZ)

**Abstrakt.** Výuka informatiky na základních a středních školách vede k tomu, že studenti umí používat jen kancelářský balík MS Office. Je tedy zcela nepravděpodobné, že by doktorandi v chemických oborech dodali text svého abstraktu vytvořený jiným nástrojem. Grafické editory, které používají, sice umí vygenerovat obrázky ve stupních šedi a podporují i export ve vektorových formátech, ale ve skutečnosti je to RGB, což je pro tisk nepoužitelné. Ukážeme, jak v časové tísní takové příspěvky zpracovat ve spolupráci několika osob, připravit jak sborník v elektronické podobě, tak data, z nichž je tiskárna schopna vytisknout sborník i v šibeničním termínu. Zároveň jsou automaticky vytvořeny soubory pro archivaci příspěvků v systému ASEP.

**Klíčová slova.** Sborník, XSLT, T<sub>E</sub>X, subversion.

## COLLABORATIVE PREPARATION OF PROCEEDINGS OF ABSTRACTS BY USING XSLT, T<sub>E</sub>X, AND SUBVERSION

**Abstract.** The result of education in informatics on schools is that students are able to use MS Office only. It is improbable that Ph.D. students of chemistry could submit the texts of their abstracts created by another tool. Graphical editors, used by them, can export images looking as greyscale and even in vector formats but in fact they are in RGB, which is unusable for printing. We will demonstrate how the proceedings can be prepared in cooperation of several persons in half no time both in the electronic form and in the form usable for printing. Files needed for archivation in the ASEP system are generated automatically.

**Keywords.** Proceedings, XSLT, T<sub>E</sub>X, subversion.

### Úvod

S vývojem techniky se barevný tisk stává stále dostupnějším i kvalitnějším. Použití barvy v tisku přesto představuje vyšší finanční náklady, které mohou být pro vydavatele s malým rozpočtem nepřijatelné. Týká se to především tiskovin, které nemají komerční význam, a jejich produkce je tedy ztrátová. Na druhou stranu roste poptávka po publikacích elektronických. V elektronické publikaci barva cenu nezvyšuje. Pokud publikace vychází v obou formách, je vhodné, abychom nebarevnou verzi pro tisk i barevnou elektronickou verzi generovali ze společného zdroje.

## 1. Spolupráce při tvorbě sborníku

Pokud pracují nejvýše dvě osoby na jednom textu, který je výhradně jejich autorským dílem, pak se mohou dohodnout, kdo bude v dané chvíli na textu pracovat a kdo bude dělat na něčem jiném, pak si role vymění. V takovém případě si mohou text posílat e-mailem. Pokud ale redaktoři mají sestavit sborník z příspěvků od 20 až 40 autorů, přičemž každý autor pošle svůj text v jiný den, a každý z redaktorů pracuje střídavě na stolním počítači v zaměstnání, stolním počítači doma a notebooku, pak to při posílání e-mailem nevyhnutelně vede rychle k tomu, že na žádném z počítačů není aktuální verze celého sborníku. Sdílený disk není řešením. Notebook nemusí mít vždy ke sdílenému disku přístup. Navíc v operačním systému Windows není možné, aby dva uživatelé měli současně otevřen stejný sdílený soubor. Spolupráci je tedy nutno řešit jinak.

Řešením tohoto typu spolupráce je použití verzovacího systému. A když už verzovací systém k dispozici máme, využijeme ho i k jiným účelům. Píšeme tak společné grantové projekty, vědecké články a další dokumenty. Texty často obsahují důvěrné informace. Nemůžeme tedy použít veřejné služby, centrální repozitář musí být na počítači, ke kterému si můžeme fyzicky přijít a ověřit, kdo má k počítači přístup, zda a jak jsou data šifrována apod. Verzovací systém musí být snadno instalovatelný člověkem, který nemá IT jako svoji hlavní pracovní činnost, a musí být snadno pochopitelný pro uživatele-neprogramátory. A právě takovým systémem je subversion [1]. Instalace do webového serveru Apache [2] s protokolem WebDAV [3] je velmi snadná a založení nového repozitáře na serveru je otázkou méně než půl minuty.

## 2. Konverze textu

Doktorandi mají na výběr, zda svůj abstrakt vytvoří v editoru MS Word, nebo použijí  $\text{\LaTeX}$ . Prakticky všichni volí první variantu,  $\text{\LaTeX}$  byl použit jen výjimečně. I Linux má kancelářské balíky, postupem času se jich v distribuci vystřídalo několik. Linuxový kancelářský balík vždy nabízel export do  $\text{\LaTeX}$ u, a to v několika úrovních, které se lišily tím, které z rysů ošklivosti byly zachovány v  $\text{\LaTeX}$ ovém kódu, který vůbec nerespektoval pravidla strukturního značkování. Protože textový soubor z kancelářského balíku je sada souborů XML [4], napsal jsem si vlastní transformační styl využívající XSLT 2.0 [5] pro použití s procesorem Saxon [6]. Jeho instalace nemusí být snadná pro uživatele-neprogramátory, proto jsem celý nástroj provozoval i jako webovou službu. Soubor ve formátu SXW, nyní ODT, byl nahrán pomocí webového formuláře na server a ten zobrazil výsledek, který stačilo zkopírovat do textového editoru a uložit. Cílem tohoto nástroje bylo odstranění formátování a zachování pouze základních stylů, jimiž je tučné písmo, kurzíva, exponenty, indexy, matematické symboly. Nástroj byl vytvořen v době, kdy  $\text{\TeX}$  ještě neuměl nativně pracovat v Unicode (a balíček

*inputenc* nativní práci v Unicode nenahrazuje), umí nástroj konvertovat písmo devanāgarī tak, aby s pomocí balíčku Velthuis Devanagari<sup>1</sup> bylo možno sázet dokumenty v sanskrtu i v hindštině, maráthštině, asamštině a dalších jazycích. Tato možnost se však ve sborníku chemické konference nevyužije, proto příslušné přepínače příkazového řádku nebylo možno ve webovém formuláři aktivovat.

Konverzní nástroj nepřevede úplně vše. Navíc třeba *slovo* lze v kancelářském editoru označit kurzívou mnoha způsoby, takže i výstup po transformaci může vypadat různě:

- `\textit{slovo}`
- `\textit{s}\textit{l}\textit{o}\textit{v}`
- `\textit{sl}\textit{ovo}`
- `\textit{s}\textit{lov}\textit{}`

Variant, s nimiž jsem se setkal, je mnohem více. Normalizace pomocí nástrojů XSLT není jednoduchá, pohodlnější je následné zpracování krátkým programem v jazyku Perl [7].

Transformační nástroj nezkonvertuje vše, vždy je nutná ruční úprava. Tabulky je nutno přepsat ručně, po konverzi máme jen texty bez jakéhokoliv formátování. Obrázkům se budeme věnovat v další kapitole.

Transformace XSLT převede celý odstavec do jednoho dlouhého řádku, což je pro použití v  $\text{\TeX}$  nevýhodné. Pokud je v textu chyba,  $\text{\TeX}$  zobrazí číslo řádku a nedefinované makro. První odstavec této kapitoly má v mém editoru 18 řádků. Pokud by to byl výsledek transformace a po ručních úpravách bych jednou sáhl o klávesu vedle a místo `\cite` napsal `\cute`, na dlouhatánském řádku by se to hledalo nesnadno. Podobně verzovací systémy zpracovávají soubory po řádcích. Předpokládejme, že jsem provedl *commit*, pak jsem objevil překlep v makru a znovu soubor odeslal do repozitáře. Později si chci zobrazit rozdíl mezi těmito revizemi. Příslušný příkaz mi sdělí, že k drobné změně došlo někde v dlouhatánském řádku a nebude snadné zjistit, že jsem při opravě nahradil *u* písmenem *i*. Proto zmíněný program v jazyku Perl přeformátuje odstavce tak, aby sestávaly z více řádků, jejichž délka nepřesahuje 80 znaků.

### 3. Zpracování obrázků

Nástroj popsany v kapitole 2 nekonvertuje obrázky, ale zcela je ignoruje. Obrázky je nutno *nějak* získat a vložit ručně. Obrázek se často dá získat tak, že se v LibreOffice smaže kompletní text, ponechá se jen obrázek a vytiskne se do PDF. Výsledný soubor se pak ořízne programem *pdfcrop*. I MS Office umí vložit vektorový obrázek a LibreOffice vektorový charakter zachová. Problém je s bitmapovými obrázky. MS Office je totiž přepočítá na nízké rozlišení, které je pro tisk

---

<sup>1</sup>Balíček již řadu let nemá vlastní web ani repozitář a je vyvíjen přímo v  $\text{\TeX}$  Live. Ve skutečnosti však má status *deprecated* a je zachován pouze pro zpracování starých dokumentů.

nepoužitelné. Navíc některé grafické nástroje umí vygenerovat vektorový obrázek (EPS, PDF, SVG), ale do MS Wordu to umí vložit jen jako bitmapu, která pak bude mít malé rozlišení. Autorům je tedy nutno sdělit, že kromě vložení do dokumentu potřebuju i obrázky v samostatných souborech.

Obrázek pro tisk musí být ve stupních šedi, ale v elektronické verzi může být barevný. Autor tedy musí buď dodat dvě varianty téhož obrázku se stejnými rozměry, nebo nechat přípravu nebarevné verze jako úlohu pro sazeče. V balíčku pro sborník příslušné konference máme definice:

```
\newif\ifZWcolor
\DeclareOption{color}{\ZWcolortrue}
\ifZWcolor
  \def\suffix{color}
\else
  \def\suffix{gray}
\fi
```

Autor může chtít v textu barevnou fotografii. Soubor, který jsme od autora dostali, pojmenujeme `fotografie-color.jpg`. Pro tisk fotografii odbarvíme a soubor pojmenujeme `fotografie-gray.png`. Balíček *graphicx* si umí příponu domyslet, takže fotografii vložíme do dokumentu příkazem:

```
\includegraphics{fotografie-\suffix}
```

Podmíněný příkaz `\ifZWcolor` můžeme použít i pro jiné účely, třeba pro zapnutí barvy v tabulce.

Mnohé programy vytvářejí obrázky vypadající jako šedé, ale ve skutečnosti jsou barevné v prostoru RGB a splňují podmínku  $R = G = B$ . Takový obrázek je však pro tisk nepoužitelný a je nutno jej převést na stupně šedi. Podrobně se práci s barvou věnuje přednáška pro CSTUG [8], jejíž záznam je volně dostupný.

Nejjednodušší je práce s bitmapovým obrázkem. Převod na stupně šedi zvládne každý dobrý grafický editor a výsledek uložíme ve formátu PNG. Musíme dohlédnout na to, aby obrázek neobsahoval průhlednost, protože specifikace PDF/X pro tisk ji zakazují.

Při práci s vektorovou grafikou chceme vektorový charakter zachovat. K úpravám použijeme program QPDF [9] a již zmíněný Perl [7], který můžeme použít přímo z příkazového řádku<sup>2</sup> bez nutnosti psaní programu. Nejprve si obrázek ve formátu PDF převedeme na lidsky čitelnou verzi a najdeme si operátory *rg* a *RG*, abychom si ověřili, že skutečně máme soubor v barevném prostoru RGB. Provedeme to těmito příkazy:

```
qpdf --qdf rgb.pdf rgb.qdf
perl -ne 'print if /\b rg\b/i' rgb.qdf
```

V jednodušším případě  $R = G = B$  provedeme konverzi příkazem:

```
perl -pe 's/(\S+)\s+\1\s+\1\s+r(g)\b/$1 $2/i' rgb.qdf >gray.qdf
```

---

<sup>2</sup>Uvedené příkazy fungují v systémech kompatibilních s POSIX, nemusí fungovat ve Windows.



Pokud je obrázek barevný, musíme barvu přepočítat na stupně šedi. V literatuře je několik vzorců s různými číselnými hodnotami, nejčastější je:

$$Gr = 0,3R + 0,59G + 0,11B, \quad (1)$$

Číselné hodnoty nemusíme dodržet. Nejde o věrnost, ale o rozlišitelnost objektů. Numerické hodnoty tedy můžeme upravit, aby výsledek byl vhodný pro čtenáře. Příkaz s použitím hodnot v rovnici (1) pak vypadá takto (vše musí být na jednom řádku bez mezery mezi `\b` a vykřičníkem):

```
perl -pe 's!(\S+)\s+(\S+)\s+(\S+)\s+r(g)\b
!(int(300*$1+590*$2+110*$3)/1000)." ".$4!ei' rgb.qdf >gray.qdf
```

Změny, provedené konverzí na stupně šedi, si raději zkontrolujeme a potom upravíme výsledný soubor na PDF:

```
diff -c rgb.qdf gray.qdf
fix-qdf <gray.qdf >gray.pdf
```

Někdy obsahuje PDF komplikované struktury, které pomocí jednoduchých regulárních výrazů odbarvit nelze. Pak programem Ghostscript [10] obrázek v-rastrujeme v rozlišení 600 dpi, uložíme jako nekomprimovaný TIFF, grafickým editorem převedeme na stupně šedi a uložíme ve formátu PNG.

#### 4. Příprava souborů pro ASEP

Kromě PDF pro tisk a barevné elektronické varianty potřebujeme ještě soubory, které je nutno archivovat v systému ASEP [11]. Při duální publikaci má ISBN jen tištěná verze. Obálka, která je i pro tisk barevná, je v samostatném souboru, blok obsahuje titulní stránku a tiráž. Počet stran bloku musí být dělitelný čtyřmi. Volitelně tedy vkládáme patitul s logem ústavu, aby tato ryze technická podmínka byla dodržena. V elektronické verzi nemáme obálku, ale jen titulní stránku a tiráž, která však nemá ISBN. Počet stran není technicky omezen. Znamená to, že paginace se v tištěné a elektronické verzi liší. V systému ASEP evidujeme jen tu verzi, která má ISBN, tedy tištěnou. Bohužel jsem nedostal odpověď na otázku, zda mám připravit jen čisté soubory s jednotlivými abstrakty, nebo mám k nim doplnit obálku a stránku s vydavatelským záznamem a tiráží, kde je uvedeno ISBN. Pro jistotu vytvářím obě verze. Tištěná verze ovšem nemá obálku ve stejném rozměru, při skládání tedy použiju obálku z elektronické verze, což je vždy první strana.

Podle požadavků Komise pro vědeckou výchovu má být abstrakt pro sborník nejvýše dvoustránkový. Ve sborníku tedy máme jednostránkové i dvoustránkové abstrakty a jejich oddělení tedy vyžaduje složitější logiku. Texty všech abstraktů máme v podadresáři `texty` a do sborníku je načítáme makrem `\itext`, které je definováno takto:

```
\def\pdfname#{\gdef\zw@lbl}\pdfname{ }
\def\itext#1{\pdfname{#1}\input{texty/#1}}
```

Každý abstrakt je vložen do prostředí `Abstrakt`, které funguje jako skupina. Můžeme tedy definovat lokální makra vhodná jen pro jeden abstrakt a neovlivnit zbytek sborníku. Prostředí definuje vzhled nadpisu, ale důležité je, že zapisuje návěští, která využijeme při generování samostatných PDF. Příslušná část definice tohoto prostředí je:

```
\newenvironment{Abstrakt}[3]{\clearpage
...
\ifcat$\zw@lbl$\typeout{*** No PDF name set ***}\else
\label{B-\zw@lbl}\fi
}{\ifhmode \par \fi \ifcat$\zw@lbl$\else \label{E-\zw@lbl}\fi
\clearpage \pdfname{}}
```

Stránka s vydavatelským záznamem a tiráží má návěští `CopyrightPage`. Informace o návěštích L<sup>A</sup>T<sub>E</sub>X zapisuje do souboru `*.aux` do makra `\newlabel`. Můžeme je tedy načíst a využít k vytvoření samostatných PDF. Použijeme k tomu program v jazyku Perl, jehož kompletní kód s vynechanou dokumentací ve formátu POD je uveden níže:

```
#!/usr/bin/perl
use utf8;
use Getopt::Long;
use File::Path qw(make_path);
use Pod::Usage;
use strict;

our %blabels;
our ($input, $cover, $nocover, $withcover);
die "Wrong options.\n" unless GetOptions
    'input=s' => \$input,
    'cover=s' => \$cover,
    'nocover=s' => \$nocover,
    'withcover=s' => \$withcover,
    'help|?' => sub {
        pod2usage(1);
    },
    'man' => sub {
        pod2usage(-exitstatus => 0, -verbose => 2);
    };
die "Input is mandatory.\n" unless $input;
die "Cover is mandatory.\n" unless $cover;
die "Nocover directory is mandatory.\n" unless $nocover;
die "Withcover directory is mandatory.\n" unless $withcover;

$nocover .= '/' unless $nocover =~ /\$/;
$withcover .= '/' unless $withcover =~ /\$/;
```

```

our $pdfcat = 'pdf-stapler cat';
our $cvpage = "$cover.pdf 1";
our $pdf = "$input.pdf";

open(IN, "$input.aux") or die "Cannot open the AUX file.\n";

foreach (($nocover, $withcover)) {
    print "Making $_\n";
    make_path($_) unless -d $_;
    die "Cannot make directory $_" unless -d $_;
}

our $cpr = 2; # compatibility

while (<IN>) {
    if (/newlabel\{CopyrightPage\}\S*\{(\d+)\}/) {
        $cpr = $1;
    }
    if (/newlabel\{([EB])-(\S+)\}\.*\{(\d+)\}/) {
        print "*** $1 ... $2 ... $3 ***\n";
        $blabels{$2} = $3 if $1 eq 'B';
        if ($1 eq 'E') {
            my $first = $blabels{$2};
            if ($first) {
                my $pages = $first == $3 ? $first : "$first $3";
                system "$pdfcat $pdf $pages $nocover$2.pdf";
                system "$pdfcat $cvpage $pdf $cpr $pages $withcover$2.pdf";
            }
        }
    }
}

close IN;

=encoding utf8

=head1 NAME

pdfs.pl: Make single PDF files

...

```

Ve smyčce `foreach` vytvoříme adresáře pro obě varianty souborů, ve smyčce `while` čteme informace o návěštích a generujeme soubory.

## 5. Závěr

Popsaný pracovní postup ukazuje, jak lze práci z velké části automatizovat a připravit sborník pro tisk velmi rychle. Zkušenost z 15 let spolupráce v redakci mě přesvědčila, že má smysl vysvětlovat doktorandům, že obrázek poslali v nevyhovujícím formátu. Doktorandi si to většinou pamatují a v dalším roce to udělají na první pokus správně. Postupem času se z doktorandů stanou školitelé, kteří to svým doktorandům vysvětlí sami, a počet nevyhovujících obrázků již klesl k nule. Přesto se někdy stane, že domluva s doktorandem je složitá a je rychlejší, když sazeč obrázek překreslí sám.

## Reference

- [1] APACHE SUBVERSION, <https://subversion.apache.org/>.
- [2] APACHE, HTTP Server Project, <https://httpd.apache.org/>.
- [3] WebDAV Resources, <http://www.webdav.org/>.
- [4] W3C: Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C Recommendation, 26 November 2008, <https://www.w3.org/TR/xml/>.
- [5] W3C: XSL Transformations (XSLT) Version 2.0 (Second Edition). W3C Recommendation, 30 March 2021 (Amended by W3C), <https://www.w3.org/TR/xslt20/>.
- [6] KAY, M.: Saxon documentation. Saxonica, <https://www.saxonica.com/documentation12/index.html>.
- [7] PERL: The Perl Programming Language, <https://www.perl.org/>.
- [8] WAGNER, Z.: Barevný text a barevné ilustrace v tisku i elektronických publikacích. Valné shromáždění CSTUGu, Brno, 2025, <https://www.cstug.cz/informace/zpravy/2025-11-01-valna-hromada-2025/>.
- [9] QPDF, <https://qpdf.sourceforge.io/>.
- [10] GHOSTSCRIPT: <https://ghostscript.com/>.
- [11] ASEP: Evidence výsledků vědecké práce v AV ČR, <https://asep-portal.lib.cas.cz/>.

## Kontaktní adresa

**Ing. Zdeněk Wagner, CSc., Praha,**

*E-mailová adresa:* [zdenek.wagner@gmail.com](mailto:zdenek.wagner@gmail.com), <https://www.zdenek-wagner.eu>.

## **ÚPRAVA FOTOGRAFIÍ V DARKTABLE, OD PLÁNOVÁNÍ AŽ K VELKOFORMÁTOVÉMU TISKU**

ZDENĚK WAGNER (CZ)

**Abstrakt.** Ze svého pohledu nepovažuji za nutné, abych používal jen software s otevřeným zdrojovým kódem, ale je pro mě nezbytné, aby software spolehlivě fungoval v Linuxu. Pokud má ještě navíc otevřený zdrojový kód, je to vítaný bonus. Darktable je takovým programem pro zpracování fotografií především v raw formátu, a to pomocí nedestruktivních úprav. Programy, které používají nedestruktivní úpravy, se používají zcela jinak než většina běžných grafických editorů. Zde ukážeme základní metodiku práce včetně ukázek některých pokročilých postupů, ale také zmíníme, jak je nutno snímky pořídit, aby byly zpracovatelné až k případnému velkoformátovému tisku.

**Klíčová slova.** postproces, fotografie, darktable.

### **POSTPROCESSING PHOTOS IN DARKTABLE, FROM PLANNING TO LARGE FORMAT PRINTING**

**Abstract.** From my point of view it is not necessary to use OpenSource software only. However, the software must reliably work in Linux. OpenSource in such a case is useful additional feature. Darktable is such a program for postprocessing photos especially in the raw format using non-destructive operations. Such programs are used in a different way than the majority of graphical editors. We will show the basic methodology including a few advanced procedures but we will mention as well how the photos should be acquired to make even the large format printing possible.

**Keywords.** postprocess, photography, darktable.

## **Úvod**

Internet je zaplaven spoustou nekvalitních fotografií, počínaje sociálními sítěmi, konče zpravodajskými servery. Ani zpravodajským serverům nezáleží na tom, zda se na snímky někdo podívá, ale jde jim jen o rychlost. V tvorbě vizuálního obsahu se navíc hodně prosazuje umělá inteligence. Pokud fotograf chce, aby se lidé na jeho díla podívali víckrát než jednou, nebo si je dokonce koupili, pak nevyhnutelně musí něčím zaujmout, dobrou kompozicí, vystižením okamžiku, zobrazením emocí a vysokou technickou kvalitou. Vysvětlení uměleckých atributů fotografie patří do zcela jiného příspěvku na jiné konferenci. Zde se budeme zabývat jen tím, co lze udělat až dodatečně v počítači pomocí software s otevřeným zdrojovým kódem, tedy technickou kvalitou. V postprocesu nelze opravit vše, je nutné fotografii správně pořídit, tomu se budou věnovat úvodní kapitoly.

Stále se vedou diskuse o tom, zda smí fotograf své snímky dodatečně upravovat. Nikomu však nevadí, že obrazy malířů nevystihují realitu, že malíři často malují objekty, které nikdy neexistovaly. A stejné je to s fotografií. V technické fotografii chceme zaznamenat objekt přesně do nejmenších detailů. Pokud realitní agentura nabízí k prodeji či pronájmu nemovitost, pak je nutno fotograficky zobrazit vše tak, jak to vypadá. Stejně tak na e-shopu musí být zboží zobrazeno věrně, jinak by to zákazník reklamoval, protože dostal něco jiného, než si objednal. Ve výtvarné fotografii ale přesná shoda s reálným předmětem není důležitá, je třeba na snímek pohlížet jako na obraz vytvořený jinou technikou, než jakou používají malíři.

V době, kdy fotografie vyžadovala drahou techniku a zpracování v temné komoře představovalo komplikovaný fotochemický proces, byli vyhledáváni a dobře placeni portrétní fotografové. Když si rodiče nechali nafotit své děti, případně celou rodinu, určitě chtěli na fotografiích vypadat hezky. Retuš tedy byla součástí práce portrétního fotografa. Krajinaři zase potřebovali ztmavit oblohu, protože lidské oko zvládne větší rozsah jasů než citlivá fotografická vrstva, digitální čipy jsou na tom ještě hůř. Retuš a barevné úpravy tedy nepřišly až s digitální technikou, ale jsou to techniky stejně staré jako fotografie sama. Práce s retušovacím štětcem a neostrými maskami byla samozřejmě mnohem složitější než současné používání těchto metod pomocí softwarových nástrojů.

V následujícím textu budou popsány některé techniky úpravy fotografií. Každý pak musí sám usoudit, pro který typ fotografie se příslušné techniky hodí a zda je smí ve svém díle využít.

## 1. Příprava

Každý technický postup má svá omezení. I špatná fotografie se dá někdy zachránit, ale jsou hranice, za které jít nelze. Do obrazu už nedostaneme to, co tam není, tedy pokud si fotografii toho předmětu nepořídíme samostatně a nepoužijeme fotomontáž. Nelze také zvýšit rozlišení. Nástroje AI takovou možnost sice nabízejí, ale informaci, která chybí v obrazových datech, si ta inteligence nějak domyslí. Je otázka, zda si ji domyslí tak, jak si to fotograf představoval. Příprava tedy určuje, kolik času budeme muset následně strávit postprodukcí a zda dosáhneme požadovaného výsledku.

### 1.1. Plánování

Fotografická práce začíná už doma plánováním. Ujasníme si, kdy, kde a co budeme fotografovat. Záleží na tom, zda budeme fotografovat v ateliéru, kde je veškerá technika ihned dostupná, nebo budeme v terénu a vše si musíme přinést. Podle žánru si zvolíme objektivy, Základním pravidlem je, že neuděláme žádný snímek objektivem, který jsme nechali doma ve skříni. Na druhou stranu, na některé snímky stačí mobil, někdy je užitečný dron, pokud se v dané lokalitě smí létat.

Slovo *fotografie* pochází z řečtiny a znamená *kresba světlem*. Musíme tedy uvážit, zda je v dané lokaci správné množství světla. Málo světla totiž vyžaduje vysokou hodnotu ISO, ale to je vykoupeno větším šumem. Odstranění šumu však vede ke ztrátě ostrosti. Pro fotografování při nedostatku světla proto potřebujeme systémový blesk, nebo dokonce studiová světla a příslušné stativy.

## 1.2. Jak velký formát je velký?

Tato otázka není dobře položena, měli bychom se ptát, jak velkou fotografii můžeme kvalitně vytisknout. U zobrazovacích zařízení se pod pojmem rozlišení obvykle rozumí jen absolutní počet pixelů bez ohledu na fyzickou velikost. Rozlišení tedy určíme tak, že z Pythagorovy věty vypočítáme počet pixelů na úhlopříčce a vydělíme délkou úhlopříčky v palcích. Nejstarší monitory měly rozlišení 72 dpi<sup>1</sup>, novější 96 dpi. Monitor ovšem vysoké rozlišení nepotřebuje. Bod na obrazovce může svítit nejen naplno, ale máme k dispozici 256 úrovní jasu pro každý barevný kanál (RGB). Máme tedy k dispozici  $256^3 = 2^{24}$  barevných tónů. Naproti tomu tiskový bod má jen dvě úrovně, buď vytištěn je, nebo není. Intenzita barvy se snižuje použitím rastru. Plocha je rozdělena na malé čtverce a intenzita barvy je dána tím, jaká část plochy je vyplněna tiskovými body. I lidské oko má určitou rozlišitelnost, takže stačí, aby body a čtverce byly dostatečně malé, a oko si je spojí do plochy. Pro tisk fotografií v běžné knize (formát A5 nebo B5) potřebujeme rozlišení 300 dpi. Na větší fotografie se ovšem díváme z větší vzdálenosti. Na střední formát, tedy např. A2, stačí rozlišení 250 dpi, na ještě větší formát můžeme použít 200 dpi.

Jak velkou fotografii tedy můžeme vytisknout? Výšku a šířku v pixelech nám dokáže zobrazit mnoho různých nástrojů. Jeden *inch* je 25,4 mm. Pokud tedy výšku a šířku vydělíme deseti, dostaneme velikost v milimetrech při rozlišení 254 dpi. Pokud chceme přibližnou velikost v milimetrech pro 300 dpi, vydělíme počty pixelů dvanácti, při vydělení osmi dostaneme rozměry pro 200 dpi.

## 1.3. Nastavení fotografického přístroje

Při prohlížení archivu se řada fotografií a fotografek ptá: „Kdy a kde jsem to byl(a)?“ Odpovědi na tuto otázku jsou informace v EXIF, které mohou obsahovat i polohové informace. Mobilní telefony vkládají polohové údaje automaticky, pokud to v nastavení není zakázáno. K digitálním kamerám často existuje mobilní aplikace, např. pro Nikon je to SnapBridge. Pomocí této aplikace lze přenášet jak časové údaje včetně časového pásma, tak polohové údaje, jak je ukázáno na obrázku 1, pokud není aktivní režim *v letadle* a *připojení k síti* je zapnuto. Grafické programy pak umí ukázat polohu na mapě, ale je možno přechíst polohu i programem z příkazového řádku, jak je vysvětleno na webové stránce [1]. Pokud nahradíme text `deg` znakem `°`, můžeme polohu na mapě vyhledat i bez grafické

<sup>1</sup>dots per inch

aplikace. EXIF obsahuje údaje, které jsou užitečné pro archiv fotografa, ale může prozradit důvěrné informace [2].



Obrázek 1. Párování s aplikací SnapBridge pro Nikon

Během postprocesu je možno odstranit mnoho různých chyb, některé snadno, některé nesnadno, ale jsou dvě chyby, které odstranit nelze, ztrátu barev a neostrost.

Problematika barev je komplikovaná [3] a nebudeme ji zde rozebírat. Důležité je vědět, že barvy ztrácíme jak při přexpozici, tak při podexpozici. Automat, který měří expozici a barevné vyvážení, předpokládá, že scéna je neutrálně šedá. V krajinářské fotografii se často stává, že obloha vyjde čistě bílá bez barev a bez kresby. V takovém případě je nutno korekcí expozice snímek mírně podexponovat. Zesvětlením v postprocesu bychom nezískali zpět barvu ani kresbu, ale jen šedou plochu. Podobně může nastat opačná situace, stíny jsou černé bez barvy a bez kresby. Snímek tedy musíme mírně přexponovat. *Mírně* ale neznamená o 1/3 EV, při záznamu do *raw* tak malá změna nehraje roli. V extrémních světelných podmínkách můžeme ztrácet barvu a kresbu ve světlech i ve stínech. Pak musíme použít přechodový filtr nebo HDR (high dynamic range).

Neostrost může mít několik příčin. Nebudeme vysvětlovat, proč automat někdy zaostří na nevhodnou vzdálenost, a pomineme prozaickou příčinu, kdy fotograf(ka) zapomene automatické zaostřování zapnout a nevšimne si toho. Významnými příčinami neostrosti je však malá hloubka ostrosti a pohybová neostrost. Malá hloubka ostrosti je způsobena příliš malou clonou. Objektiv musíme zaclonit tak, aby celý hlavní objekt byl ostrý. Ve špatných světelných podmínkách to ale vede k prodloužení expozičního času na takovou hodnotu, že dojde k pohybové neostrosti, buď tím, že přístroj neudržíme, nebo tím, že se příliš rychlý děj rozmáže.

Po dlouhé zkušenosti jsem dospěl k tomu, že mám zapnutou automatickou regulaci citlivosti ISO, což je vidět i na levé části obrázku 1. Automatika udržuje přednastavenou hodnotu ISO = 100 a změní ji pouze v případě, kdy expoziční čas



vychází tak dlouhý, že se nedá udržet, nebo při nastavení pevného expozičního času vychází clona mimo technické možnosti objektivu. Automatické ISO nedává smysl při fotografování statické scény, kdy je fotografický přístroj na stativu. Pak chceme zaclonit podle požadované hloubky ostrosti a udržet nízkou citlivost kvůli malému šumu, ale dlouhá expozice nevádí. Nesmíme ji aktivovat při dlouhých expozicích s použitím ND filtrů a při použití studiových zábleskových světel.

## 2. Postproces

Samotné pořízení snímku je první částí úspěchu, ale bez následného zpracování by výsledek nebyl plnohodnotný. Už jsme zmínili, že v určitých situacích snímek záměrně podexponujeme, nebo přexponujeme. To však není jediný důvod k následným úpravám.

### 2.1. Proč upravujeme fotografie?

Místo odpovědi také začneme otázkou. V první řadě seriálu Fargo [4] Lorne Malvo říká (pokud si to pamatuju dobře): „Víte, proč je oko nejcitlivější na zelenou barvu? Až najdete odpověď na mou otázku, budete znát odpověď i na tu svou.“ Pokud se ptáme, jestli správně *vidí* citlivá vrstva fotografického filmu, snímací čip digitálního přístroje, nebo lidské oko, pokládáme si špatnou otázku. Nic ze zmíněných alternativ nevidí dobře ani špatně, vidí jinak. Oko slouží k jinému účelu než technické prostředky. Proto chceme upravit snímek, aby vypadal tak, jak by ho vidělo lidské oko v okamžiku pořízení.

Otázkou je použití retuše. V reálném životě se člověk na vše podívá krátce a mnoha detailů si nevšimne, zatímco na fotografii se může dívat delší dobu. Proto je vyretušování rušivých objektů opodstatněné.

Základní úpravy provádíme vždy, a to i u snímků, na nichž zdánlivě nezáleží. I takové snímky ale mohou pokazit renomé fotografa, proto neupravené snímky ukazujeme nejvýše jako didaktický materiál.

Fotografie patří do výtvarného umění. Úpravami tedy můžeme dosáhnout jistého výtvarného záměru, kdy vytvoříme obrazový záznam něčeho, co vlastně reálně neexistuje.

### 2.2. Nedestruktivní úpravy

Tradiční grafické editory zasahují přímo do upravovaného obrázku. Představme si, že z obrázku vymažeme text. Pak upravíme vyvážení bílé, v části snímku změníme jas, odstraníme šum. Nyní si uvědomíme, že jsme vlastně text smazat nechtěli, chceme ho vrátit zpět a zachovat ostatní úpravy, ale to nejde. Funkce *undo* neumí selektivně vrátit jednu operaci, vrací vždy jen tu naposled provedenou a udržuje omezený počet operací, které lze ještě odstranit. A pokud soubor uložíme,

zavřeme grafický editor, vypneme počítač a k úpravě snímku se vrátíme následující den, funkci *undo* již nemáme. Každá operace má za následek destrukci, která se postupně stává nevratnou. Možná někde máme zálohu...

Nedestruktivní úpravy do původního obrázku nezasahují. Grafický editor, který pracuje tímto způsobem, si v postranním souboru<sup>2</sup> (nebo v databázi) ukládá parametry jednotlivých operací. Můžeme např. snímek oříznout, upravit vyvážení bílé, doostřit, odstranit šum. Nyní si všimneme, že se nám vyvážení bílé nelíbí, můžeme se tedy k té operaci vrátit a upravit její parametry. Doostření nám do obrázku přidalo nevhodné detaily, které nelze potlačit ani odstraněním šumu, proto celou operaci doostření odvoláme. I ořez je zaznamenán jen pomocí parametrů, můžeme tedy vrátit zpět část, kterou jsme odřízli. Fotografii můžeme chtít ve dvou variantách, barevné a černobílé. Každá z nich potřebuje minimálně jinou úpravu jasu. U nedestruktivního editoru můžeme mít k jednomu *raw* souboru více souborů postranních, které jsou ve srovnání s obrazovými daty velmi malé, podle počtu a složitosti provedených operací od 5 KiB do 15 KiB, zatímco *raw* soubor pro zrcadlovku a bezzrcadlovku *full frame* vyžaduje 55 MiB, pro středoformátový přístroj více než 100 MiB. Vytváření postupných záloh při práci s destruktivním editorem tedy vyžaduje obrovský diskový prostor, což ale u nedestruktivního editoru nepotřebujeme. Navíc se k fotografii můžeme vrátit i po měsících a stále máme k dispozici postranní soubor. A v okamžiku, kdy jsme s úpravami spokojeni, exportujeme fotografii do vhodného formátu pro sdílení a tisk, obvykle JPG. Ostatně tímto způsobem jsem připravoval i ukázky pro svoji přednášku. Levý panel na obrázcích v kap. 3 ukazuje historii úprav, které jsem dočasně odvolal.

Zatím jsme popsali nesporné výhody nedestruktivních úprav, ale má to i nevýhody. Editor pracuje s původními daty *raw* souboru a stále rostoucím počtem úprav v postranním souboru. Při každé drobné změně musí přepočítat celý obrázek. Znamená to, že pro pohodlnou práci potřebujeme výkonný počítač.

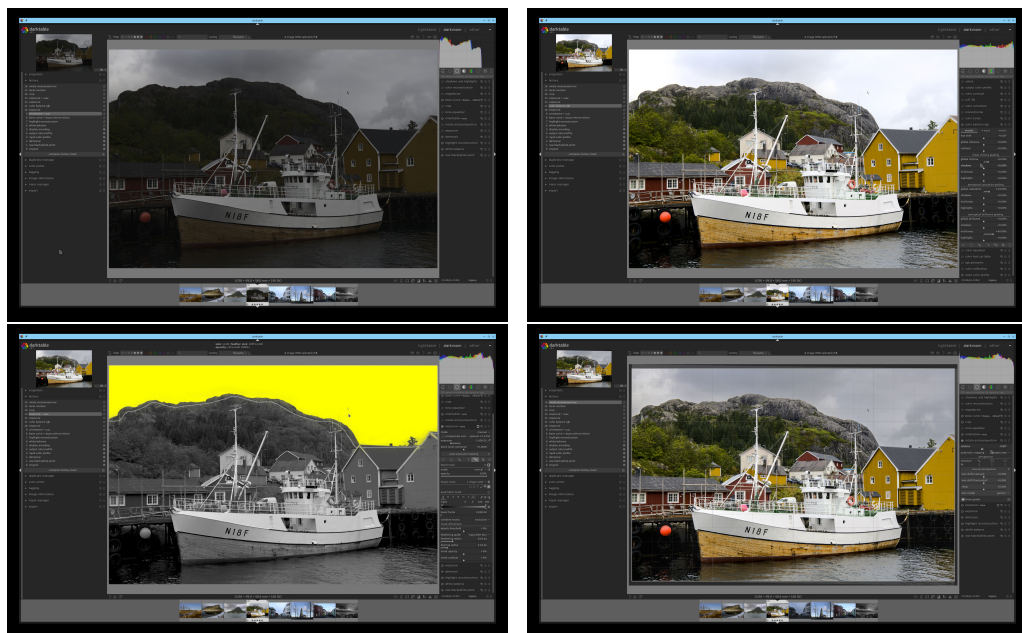
### 2.3. Struktura programu Darktable

Darktable [5] má obdobnou strukturu jako Lightroom od firmy Adobe. Můžeme importovat snímky již uložené na disku, ale můžeme je též načíst přímo z digitálního přístroje pomocí protokolu PTP<sup>3</sup>. Snímky musí být importovány, abychom s nimi mohli pracovat. K dispozici máme dvě komponenty, prosvětlovací pult (*lighttable*) a temnou komoru (*darkroom*). Názvem a pracovním postupem tedy program připomíná tradiční fotografické pracoviště. Pomocí prosvětlovacího pultu můžete kontrolovat náhledy, přidávat hvězdičky (maximálně pět) a označovat snímky barevnými štítky. Nabízí se šest štítků a na rozdíl od jiných programů

---

<sup>2</sup>anglicky *sidecar file*

<sup>3</sup>Picture Transfer Protocol



Obrázek 2. Ukázka masky pro ztmavení oblohy

obrázek může mít několik barevných štítků současně. Označením si tedy vybereme snímky, které má smysl upravovat. Také si zde můžeme zkopírovat postranní soubor, abychom pro snímek mohli vytvořit více variant. Vlastní úpravy pak *vyvoláváme* v temné komoře. Po návratu k prosvětlovacímu pultu vidíme již snímky upravené.

Dostupná dokumentace ovšem není vhodná pro začátečníky. Naštěstí je řada užitečných videonávodů na YouTube. Nejrozsáhlejší je kolekce videí od Bruce Williamse [6]. V době psaní tohoto textu jich už je více než 160 a sahají daleko do minulosti. Od té doby se ovládání programu změnilo, takže staré díly již nemusí být aktuální.

### 3. Ukázky

Cílem tohoto textu není výuka používání Darktable, protože těch možností je mnoho. Navíc lze často téhož výsledku dosáhnout různými způsoby. Každý uživatel se musí dopracovat ke své metodě, kterou považuje za nejpohodlnější. Předvedeme si tedy jen několik základních postupů, které by měly sloužit jako ochutnávka.

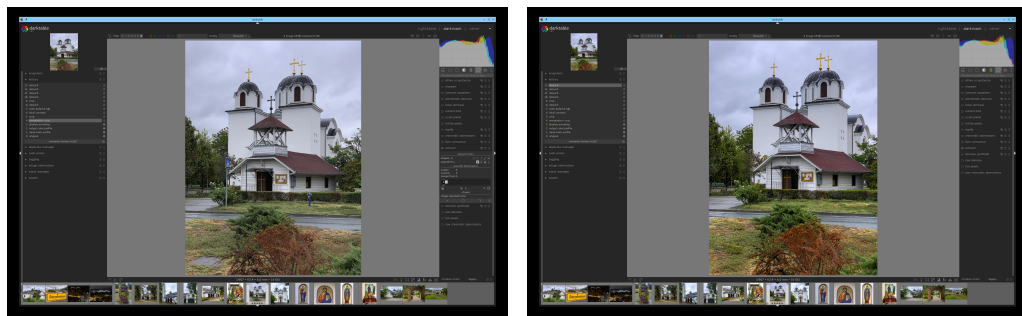
Na obrázku 2 je loď v norském Nusfjordu. Vlevo nahoře je stav, jak jsme snímek exportovali z přístroje. Snímek byl záměrně podexponován, aby zůstala kresba v obloze. Podexpozici naznačuje i histogram. Vpravo nahoře je stav po zesvětlení.



**Obrázek 3.** Ukázka automatické korekce perspektivy, vyvážení bílé

Nevypadá to špatně, ale obloha ztratila dramaticčnost. Vytvoříme si tedy kopii modulu *exposure* a vyznačíme si masku metodou *drawn and parametric*. Obrázek vlevo dole ukazuje, jak jsme pomocí několika bodů zakreslili křivku tak, aby zasahovala trochu mimo oblohu. Pak jsme si masku nechali zobrazit jako žlutou plochu a parametricky jsme vybrali jen světlejší část. Tím se maska automaticky posunula tak, že v ní nejsou zahrnuty hory, komín domu, strom, antény na lodi a dokonce ani letící racek. Takto vymaskovanou plochu ztmavíme. Nakonec jsme ještě srovnali horizont tak, aby domy i lodní stěžně byly svislé.

Obrázek 3 ukazuje večerní záběr ulice v Reykjavíku. Nebyl pořízen objektivem typu *tilt shift*, proto se domy kácejí. Obrázek vlevo nahoře je původní stav. Vpravo nahoře vidíme v pravém panelu otevřené parametry modulu *rotate and perspective* a ikonkou vpravo v horní řadě jsme nechali načíst strukturu. Automat označil linie, které by podle jeho názoru měly být vodorovné a svislé. Dolní řada ikonek nabízí tři možnosti korekce, svislou, vodorovnou a úplnou. Použili jsme úplnou a na obrázku vlevo dole je vidět, že výsledek musí být oříznut. Výhodou nedestruktivních úprav je, že akci můžeme odvolat, pokud nejsme s výsledkem spokojeni. Můžeme také zvolit jinou metodu korekce. Přísně matematická korekce někdy nevypadá dobře, ale můžeme ručně upravit parametry, které vypočítal automat. Vpravo dole je ukázáno, že výsledný obrázek bylo nutno zesvětlit, ale i posunout vyvážení bílé, protože automat se nechal večerním světlem oklamat.



Obrázek 4. Retuš fotografie z iPhone XS

Obrázek 4, na němž je, psáno gajicí<sup>4</sup>, „Hram Svete Petke u Petrovaradinu“ (Srbsko), ukazuje, že i fotografie pořízené mobilním telefonem je někdy vhodné upravit v Darktable. Obvykle není nutno zasahovat do barevnosti, nejčastějšími zásahy jsou ořez a retuš. Mobilní aplikace sice úpravy nabízejí, ale jsou vždy destruktivní. Při retušování vždy používám režim *healing*. Zpracování obrázku 4 bylo provedeno v několika etapách. Odstranění kanálu bylo snadné, jen jsem ho označil kruhem a odnesl do trávy. S hydrantem to bylo těžší, musel jsem ho retušovat malými elipsami po částech. A ještě víc práce bylo s dopravní značkou. Ve složitějších případech, než byl tento, je nutno použít *wavelets*. Tím jsem na jiné fotografii dokázal modelce zahojit odřené koleno. Postup není snadný k vysvětlování v psaném textu, dobře to vysvětluje s živou ukázkou v jednom ze svých videí Bruce Williams [6].

#### 4. Závěr

V textu jsou předvedeny nejčastěji prováděné úpravy, a to z pohledu fotografa, který neumí kreslit. S tím souvisí práce s maskami, které si Darktable umí dopočítat, ale i způsob retušování, kdy uživatel vlastně jen označí část s rušivými objekty a jinou část, z níž má Darktable dopočítat vzor pro odstranění poruch. Například odstranění nečistoty na snímacím čipu je záležitostí jednoho kliknutí, což zabere zlomek sekundy. Ukázky jsou zobrazeny jako snímky obrazovky, přestože texty na postranních panelech nejsou zcela čitelné (jsou malé i na velkém monitoru), ale ukazují výhody nedestruktivní editace.

Při práci s Darktable jde hlavně o zvyk a získání praxe, kdy si uživatel vytvoří vlastní pracovní postup. Časově náročný je počáteční výběr snímků, které má smysl dále upravovat. Vlastní vyvolávání v temné komoře by mělo být snadné. Já osobně raději trávím čas při fotografování v terénu, ne při zpracování u počítače. Typicky mi trvá zpracování fotografie asi půl minuty, ale při nutnosti složitější

<sup>4</sup>Chorvatská abeceda, kterou podle vzoru Jana Husa vytvořil v roce 1835 chorvatský jazykovedec Ljudevit Gaj.

retuše to trvá déle. V extrémních případech trvá retuš i půl hodiny, ale takovou fotografii mám nejvýše jednu ročně. Vždy se snažím, abych fotografii pořídil takovým způsobem, že náročné úpravy v počítači nebudou vyžadovány. Je to v podstatě otázka kompozice, světla, expozice a odnesení předmětů, které by v záběru kompozičně vadily.

## Reference

- [1] WAGNER, Z.: Testy digitálních fotopřístrojů a příslušenství, <http://icebearsoft.euweb.cz/dicamtests/>
- [2] BISCHOFF, P.: EXIF metadata privacy: A picture is worth a thousand data points, <https://www.comparitech.com/blog/vpn-privacy/exif-metadata-privacy/>
- [3] HSL and HSV, Wikipedia, the free encyclopedia, [https://en.wikipedia.org/wiki/HSL\\_and\\_HSV](https://en.wikipedia.org/wiki/HSL_and_HSV)
- [4] HAWLEY, N.: Fargo (seriál), Season 1, 2014.
- [5] DARKTABLE, <https://www.darktable.org/>
- [6] WILLIAMS, B.: Bruce Williams Photography. YouTube, <https://www.youtube.com/@audio2u>

## Kontaktní adresa

**Ing. Zdeněk Wagner, CSc.,**

*E-mailová adresa:* [zdenek.wagner@gmail.com](mailto:zdenek.wagner@gmail.com), <https://www.zdenek-wagner.eu>.

## **STATISTICKÁ DATA V DIGITÁLNÍM VĚKU: STRATEGIE A NÁSTROJE ČSÚ**

**Petra KUNCOVÁ (CZ), Helena KOLÁČKOVÁ (CZ)**

ČSÚ je jedním z nejvýznamnějších poskytovatelů dat v České republice. Hlavním přístupovým bodem je webová prezentace ČSÚ, která prošla v roce 2024 zásadní změnou struktury i grafiky. Data poskytovaná ČSÚ mají velmi důležitou vlastnost – jsou metadatově popsaná a splňují standardy požadované v rámci evropského statistického systému. Data nabízí ČSÚ v různých podobách – od jednoduchých html tabulek až po celé databáze, a to vše v různých formátech včetně formátu otevřených dat. Relativně novou formou poskytování údajů je Geoportál ČSÚ, který nabízí nejen zajímavé vizualizace, ale také nový Datový portál GIS – novou platformu pro diseminaci datových sad s geograficko-statistickým obsahem v souladu s otevřenou formální normou (opendata).

Základním posláním Českého statistického úřadu je sběr, zpracování a poskytování informací o vývoji Česka. V oblasti sběru dat upřednostňuje moderní statistika využívání administrativních zdrojů, v současnosti jich využívá přes 100, čímž snižuje zátěž respondentů. Nově jsou integrována data z pokladních systémů řetězců nebo z rezervačních portálů. Důležitým zdrojem dat stále zůstávají i statistická zjišťování u respondentů. Všechna data jsou přísně chráněna podle zákona č. 89/1995 Sb., přičemž ochrana důvěrnosti statistického údaje má vždy přednost před jeho zveřejněním. Údaje jsou poskytovány zásadně v agregované formě, aby nebylo možné přímo či nepřímo identifikovat konkrétní subjekty.

Z pohledu uživatelů je zásadní dostupnost dat prostřednictvím různých distribučních kanálů. Hlavním zdrojem informací je portál [csu.gov.cz](https://csu.gov.cz), který je strukturovaný tak, aby pokud možno všichni uživatelé našli hledané informace. Mezi klíčové nástroje donedávna patřila Veřejná databáze (VDB), což je tradiční prezentační databáze se statistickými daty ze všech okruhů. Moderním nástupcem VDB je DataStat – nová vícedimenzionální databáze umožňující přístup přes API, parametrické dotazování a pokročilé filtrování. Statistický geoportál je nová platforma sloužící k vizualizaci demografických a ekonomických dat na mapových podkladech. Další neméně důležitou komponentou portálu [csu.gov.cz](https://csu.gov.cz) je Katalog produktů, jenž sdružuje všechny typy statistických výstupů a produktů, včetně těch, které jsou dostupné ve strojově čitelném formátu <https://csu.gov.cz/databaze-a-aplikace>.

Významným posunem v oblasti dostupnosti dat je rozvoj otevřených dat. ČSÚ publikuje datové sady v rámci Národního katalogu otevřených dat, a to včetně metadat, klasifikací a číselníků. Otevřená data jsou dostupná nejen ke stažení, ale také prostřednictvím aplikačních rozhraní (API), což umožňuje jejich přímé napojení na analytické

nástroje a informační systémy. Tento přístup zásadně rozšiřuje možnosti jejich využití v oblasti datové vědy a automatizovaného zpracování. V katalogu je evidováno cca 80 otevřených datových položek, pokrývajících ekonomické, demografické i geografické domény <https://data.gov.cz/pro-uÅivatele/otevřená-data/>, <https://opendata.gov.cz/informace:start>, <https://csu.gov.cz/otevrena-data-v-katalogu-produktu-csu>. Pro akademickou a IT komunitu mají otevřená data zásadní význam: umožňují kvantitativní analýzy, tvorbu ekonometrických modelů, práci s časovými řadami i pokročilé analytické workflow. Strojová čitelnost a standardizace datové struktury umožňují integraci do výzkumných prostředí (R, Python, MATLAB) i informačních systémů. ČSÚ podporuje využívání svých otevřených dat pravidelnou účastí na hackathonech veřejné správy, na kterých vznikají nové zajímavé a užitečné aplikace. Stále se rozšiřující produkce otevřených dat ČSÚ tak posiluje spolupráci mezi všemi jejich stávajícími i budoucími uživateli. <https://csu.gov.cz/otevrena-data-v-katalogu-produktu-csu>

Specifickou oblast představuje poskytování mikrodat pro vědecké účely. Přístup k těmto datům je přísně regulován s ohledem na ochranu důvěrnosti. Data jsou poskytována pouze oprávněným institucím pro konkrétní projekt na základě smluvního vztahu a za podmínky dodržení bezpečnostních pravidel. Výzkumníci pracují s daty, která neumožňují přímou identifikaci jednotky, přičemž výstupy podléhají kontrole, aby nedošlo k identifikaci jednotlivých subjektů.

Významnou roli v celém systému hrají také statistické registry a klasifikace, které umožňují standardizaci dat a jejich srovnatelnost v čase i mezi jednotlivými regiony. Tyto nástroje jsou nezbytné pro kvalitní analýzu a interpretaci statistických informací. Díky nim je možné provádět komplexní analýzy a porovnání v čase i prostoru.

ČSÚ taktéž plní nezastupitelnou roli v organizaci voleb, kde odpovídá za technické zpracování, sumarizaci výsledků a rozdělení mandátů. Další stěžejní agendou je Sčítání lidu, domů a bytů (SLDB), což je rozsáhlé šetření opakující se každých 10 let. V roce 2021 probíhalo sčítání primárně on-line, což umožnilo realizaci této velké akce v době pandemie Covidu. Úřad rovněž podporuje statistickou gramotnost skrze vzdělávací projekty jako EduStat nebo Minisčítání pro zvýšení statistické gramotnosti mládeže.

Datové zdroje ČSÚ tak představují ucelený systém, který reflektuje současné požadavky na dostupnost, kvalitu a bezpečnost informací. Jejich význam bude nadále růst s ohledem na rostoucí potřebu datově podloženého rozhodování a rozvoj digitálních technologií v oblasti zpracování dat. Kombinace otevřeného přístupu k agregovaným datům, moderních nástrojů pro jejich zpracování a jasně definovaných pravidel pro práci s citlivými údaji umožňuje jejich široké využití v akademické sféře, veřejné správě i komerčním sektoru. Klíčovým trendem do budoucna je další rozvoj otevřených dat, automatizace přístupu a integrace statistických zdrojů do digitálních ekosystémů.

## DATASTAT – PREZENTAČNÍ DATABÁZE ČSÚ

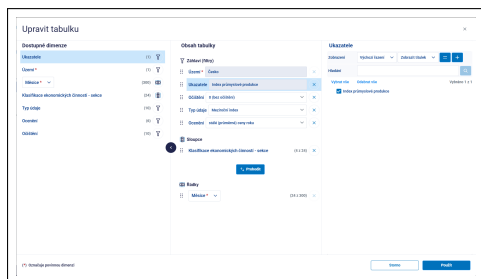
Martin ČERNÝ (CZ)

DataStat se v roce 2026 ujímá role hlavní prezentační databáze Českého statistického úřadu a nahrazuje tím dosavadní Veřejnou databázi ČSÚ. Nová webová aplikace přináší





případně setřídění seznamu tabulek, pro opakované využití. V zobrazené tabulce jsou vizuálně odlišené názvy dimenzí, které jsou umístěné do sloupců a do řádků. Dimenze umístěné jako filtry nad tabulkou umožňují rychlý výběr jiné položky, což se okamžitě projeví aktualizací zobrazených dat.

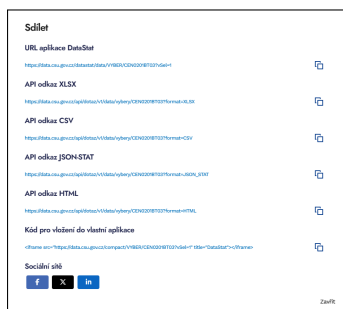


Obr. 3. Vlastní úprava tabulky.

položek zvolené dimenze. Uživatel může položky filtrovat pomocí textového vyhledávání, hromadně označovat, měnit řazení položek, zvolit variantu zobrazení textových popisků a/nebo kódů, přepínat varianty dimenze, případně nastavit výchozí úroveň pro rozbalení hierarchie. Finální struktura zobrazené tabulky v hlavičce a legendě, i nabídka položek ve filtrech nad tabulkou, je pak vytvořená automaticky na základě použitého nastavení.

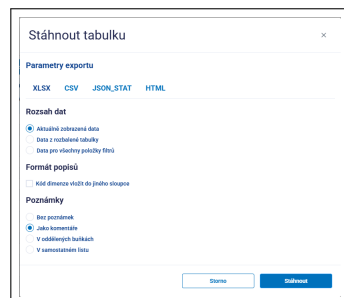
Tabulku je možné zobrazit na celou obrazovku, nebo přepnout do zobrazení zvoleného typu grafu, případně kartogramu. Zobrazený graf či kartogram je možné ještě upravit pomocí vlastních nastavení, které jsou pro dané zobrazení k dispozici. DataStat umožňuje kromě online zobrazení dat také jejich stažení do několika formátů, konkrétně do XLSX, CSV, JSON-STAT a HTML. Pro každý z těchto formátů je možné ještě upřesnit několik parametrů: rozsah dat, formát popisků a poznámky k údajům.

Zobrazenou tabulku, resp. její datový obsah lze sdílet pomocí vygenerovaného odkazu nebo vložit do vlastní aplikace.



Obr. 5. Sdílení tabulky.

Vlastní úprava tabulky se provádí v samostatném okně, které se otevírá ikonou Upravit tabulku. V levé části okna je přehled všech dostupných dimenzí dané sady s počtem položek v jednotlivých dimenzích a s možností rychlého umístění dimenze do některé z pozic v tabulce. Prostřední část okna zobrazuje aktuální uspořádání a obsah tabulky s možností rychlé výměny dimenzí ve sloupcích a řádcích. Právý panel umožňuje výběr konkrétních



Obr. 4. Nastavení exportu dat.

Pro přihlášené uživatele je k dispozici funkce Uložit, která aktuálně zobrazenou tabulku uloží do samostatné složky Moje v tematickém stromu DataStatu. V dialogovém okně pro uložení je možné zadat vlastní název tabulky, zvolit režim aktualizace obsahu dat a nastavit zasílání upozornění při aktualizaci dat.

Při nastavení notifikace je možné zvolit i formát a další parametry exportu dat, takže uživatel bude aktualizovaná data v požadovaném souboru dostávat rovnou do emailu, a nebude muset ani spouštět webový prohlížeč a data stahovat z prezentace ručně.

Dôležitým prínosom DataStatu je také možnosť strojového získavania štatistických dát pomocou aplikačného rozhraní, tzv. API. Základní informace pro využití této funkcionality, včetně několika příkladů, je zveřejněn na adrese <https://csu.gov.cz/zakladni-informace-pro-pouziti-api-datastatu>.

Podrobná technická dokumentace je k dispozici na těchto adresách:

- Přehled <https://data.csu.gov.cz/api/katalog/v1/swagger-ui/index.html>.
- Výběr dat <https://data.csu.gov.cz/api/dotaz/v1/swagger-ui/index.html>.

Štatistická data, ktorá jsou zveřejňovaná prostřednictvím databáze DataStat, jsou zároveň poskytována jak otevřená data, a to díky automatické evidenci v Národním katalogu otevřených dat (NKOD). Zde jsou datové sady poskytované z DataStatu nabízené ve třech distribucích: CSV, JSON a datová služba.

Český štatistický úřad plánuje obsah databáze DataStat postupně rozšiřovat tak, aby byla pokud možno všechna zveřejňovaná štatistická data zpřístupněna uživatelům se všemi výše uvedeným možnostmi.

## OTEVŘENÁ DATA ČESKÉHO ŠTATISTICKÉHO ÚŘADU

Martin ČERNÝ (CZ)

Pojem otevřená data je v České republice definován v § 3a odst. 5 zákona č. 106/1999 Sb. takto: „Otevřenými daty se pro účely tohoto zákona rozumí informace zveřejňované způsobem umožňujícím dálkový přístup v otevřeném a strojově čitelném formátu, jejichž způsob ani účel následného využití není povinným subjektem, který je zveřejňuje, omezen a které jsou evidovány v národním katalogu otevřených dat.“

Za otevřená data lze tedy považovat taková data, která

- jsou dostupná na internetu ke stažení bez technických a jiných překážek (registrace, omezení počtu přístupů...),
- jsou strojově čitelná,
- používají standardy s volně dostupnou specifikací,
- jsou zpřístupněna za jasně definovaných podmínek užití dat bez omezení,
- jsou registrována v Národním katalogu otevřených dat (NKOD).

Poslední uvedená podmínka je specifická pro české prostředí, ale jasně odlišuje oficiální otevřená data od jiných volně přístupných a strojově čitelných dat.

Otevřená data díky uvedeným požadavkům splňují i obecné principy FAIR:

- Vyhledatelnost otevřených dat je zajištěna jejich registrací do NKOD a vyplněním všech povinných položek metadat.
- Dostupnost je zajištěna jejich vystavením na internetu.
- Interoperabilita je zajištěna využíváním otevřených formátů, strojovou čitelností, standardů.
- Opětovná využitelnost je pak zajištěna vhodnými (neomezujícími) podmínkami užití.

Klíčovým prvkem v celém ekosystému otevřených dat je Národní katalog otevřených dat (NKOD), který slouží jako centrální evidence datových sad veřejné správy. Datové sady představují množinu souvisejících dat, vztahujících se k jednomu věcnému tématu. Samotná data jsou poskytována v podobě tzv. distribucí datové sady, což jsou konkrétní datové zdroje, typicky soubory ke stažení nebo služby API.

Katalogizační záznam v NKOD o konkrétní datové sadě obsahuje příslušná metadata, např. název datové sady, poskytovatele datové sady, informace o periodicitě aktualizace, územním pokrytí, apod. Informace o dostupných distribucích jsou uvedeny u každé datové sady a jejich součástí je vždy i datový formát, ve kterém jsou data daná distribuce k dispozici.



**Obr. 6.** Národní katalog otevřených dat jako centrální rozcestník.

Český statistický úřad (ČSÚ) patří mezi nejvýznamnější poskytovatele otevřených dat v České republice. V Národním katalogu otevřených dat má aktuálně zaregistrováno přes 1600 datových sad, které obsahují široké spektrum statistických, geografických i administrativních dat. Součástí datových sad evidovaných v NKOD jsou také metadata zahrnující statistické ukazatele, číselníky a vazby mezi nimi.

ČSÚ zajišťuje zveřejňování otevřených dat pomocí několika prezentačních systémů, které jsou zaměřeny na specifický charakter dat.

Prvním z těchto systémů je DataStat, hlavní prezentační databáze ČSÚ, která obsahuje oficiálně publikovaná statistická data ze všech tematických okruhů. Otevřená data jsou poskytována v podobě souborů ke stažení, a to ve formátu CSV a JSON-STAT. Tento formát JSON byl zvolen s ohledem na jeho využívání dalšími statistickými úřady i díky jeho srozumitelnosti a snadné použitelnosti. Tyto vlastnosti podporuje i detailní dokumentace s příklady a dostupné nástroje a knihovny pro různé programovací jazyky na stránce <https://json-stat.org>.

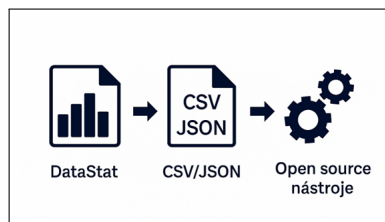
DataStat nabízí kromě webového uživatelského rozhraní pro vlastní interaktivní výběr a stahování dat také API rozhraní, které poskytuje informace o všech zveřejněných datových sadách evidovaných v Národním katalogu otevřených dat. Toto rozhraní je sice primárně určené pro registraci datových sad do NKOD, ale může být efektivně využito i dalšími uživateli a jejich aplikacemi. Prakticky se jedná o dvě jednoduché služby:

- Seznam datových sad <https://data.csu.gov.cz/opendata/katalog>.
- Metadata datové sady, např. <https://data.csu.gov.cz/opendata/sady/CEN0101D>.

Pro uživatele, kteří si nechtějí stahovat kompletní obsah datové sady, ale chtějí si vybrat jen některé ukazatele nebo položky, je nejhodnější využít webové rozhraní DataStatu. V něm lze nejprve vyhledat konkrétní datovou sadu, zobrazit výchozí tabulku a pomocí funkce „Upravit tabulku“ vytvořit vlastní datový výběr. Tento výběr je následně nutné po přihlášení (po předchozí registraci uživatelského účtu) uložit. Při ukládání je

Národní katalog otevřených dat funguje pro uživatele jako centrální webový rozcestník pro vyhledávání existujících datových sad s možností pokročilé filtrace podle různých kritérií. NKOD obsahuje pouze metadata o datových sadách s odkazy na konkrétní distribuce. Samotná data ve formě souborů ke stažení nebo API služeb jsou poskytována prostřednictvím publikačních systémů jednotlivých poskytovatelů otevřených dat.

Český statistický úřad (ČSÚ) patří mezi nejvýznamnější poskytovatele otevřených dat v České re-



**Obr. 7.** Otevřené datové sady z Data Statu.

možné nastaviť režim aktualizácie dát: predĺženie časovej rady s pevným počátkom alebo automatický posun se zachovaním posledných  $N$  dostupných období. Po uložení výberu DataStat vygeneruje unikátny identifikátor, ktorý lze pomocou funkcie „Sdílet“ využíť např. pro strojové stahování dat v požadovaném formátu.

Dalším prezentačním systémem ČSÚ poskytujícím otevřená data je Geoportál ČSÚ, který zpřístupňuje datové sady s geograficko-statistickým charakterem prostřednictvím Datového portálu GIS [<https://geodata.csu.gov.cz/portal/apps/sites/#/homepage>].

Webové rozhraní nabízí možnost interaktivního vyhledávání a filtrování datových sad podle různých kritérií. Pro zvolenou sadu jsou zobrazeny informace o dostupných formátech a územní vymezení dat na mapě.

Součástí řešení Datového portálu GIS je také API rozhraní se seznamem datových sad [<https://geodata.ab.czso.cz/as/odsdatovyportalapi/api/lod/catalog>], které se využívá především pro automatickou aktualizaci NKOD, ale je veřejně dostupné.

Mezi další prezentační systémy poskytující otevřená data patří volební portál na adrese [www.volby.cz](http://www.volby.cz). Otevřená data jsou zde nabízena v samostatné sekci webové prezentace a jsou uspořádána podle jednotlivých druhů voleb a roků jejich konání.

Pro potřeby automatické evidence otevřených dat z volebního portálu a dalších publikačních systémů ČSÚ do NKOD bylo zveřejněno ještě jedno API rozhraní [<https://geodata.csu.gov.cz/as/odslkodapi/api/lod/catalog>].

Toto API rozhraní bude v dohledné době obsahovat také informace o otevřených datových sadách poskytovaných novým prezentačním systémem pro metadata ČSÚ. Ten by měl být spuštěn v podobě webové aplikace během roku 2026.



Obr. 8. API odkazy pro vlastní uložený výběr v DataStatu.

## OPEN SOURCE NÁSTROJE PRE AUTOMATIZÁCIU PRÍPRAVY 3D MODELU ĽUDSKÉHO TEĽA

Samuel GOLIAN (SK), Marek KLIMO (SK)

Príspevok predstavuje doplnok vyvinutý pre open-source nástroj Blender, ktorý automatizuje prípravu 3D modelov ľudského tela pre použitie vo webovej aplikácii. Doplnok je napísaný v programovacom jazyku Python s využitím rozhrania bpy a umožňuje redukciu geometrickej siete 3D modelov, redukciu rozlíšenia textúr a automatické vytváranie viacerých úrovní detailu (LOD). Výstup je určený pre webovú aplikáciu Fri3DBody slúžiacu na výučbu anatómie a fyziológie v rámci študijného programu Biomedicínska informatika na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline.

## **TERAPEUTICKÁ BIBLE: STUDIJNÍ BIBLE S POZNÁMKOVÝM APARÁTEM, VYSÁZENÁ S POMOCÍ OPTEXOVÉHO MAKRA OPBIBLE**

**Šimon KONEČNÝ (CZ)**

Komplexní sazečské projekty jako studijní Bible nemusejí trvat měsíce, nýbrž jen pouhé minuty. Poznámkový aparát lze použít s kterýmkoliv s mnoha různých překladů Bible. Ukážeme si mimořádně efektivní alternativu k tradiční sazbě z Wordu do InDesignu a některé z jejích výrazných předností. Více viz <https://phos.club/>

## **GEMINI: PRVNÍ ČESKÝ PŘEKLAD BIBLE POŘÍZENÝ S POMOCÍ AI**

**Šimon KONEČNÝ (CZ)**

Na světě je první český překlad Bible pořízený umělou inteligencí. Aby mohl plnohodnotně fungovat, vyžadoval důkladnou lidskou revizi. Právě v této korigované podobě se stává jedním z klíčových překladů, na kterých názorně předvedeme výkon a flexibilitu makra OpBible. Více viz <https://phos.club/>

## **GUARDIANS: OD CTF PLATFORMY K REALISTICKÝM INCIDENTOM**

**Dominik DVORSKÝ (CZ)**

Prednáška predstaví súťaž Guardians od Binary Confidence ako praktický formát na rozvoj zručností v oblasti incident response, threat huntingu a digitálnej forenzej analýzy. Vysvetlí, na akej platforme súťaž beží, ako sa prepája s CyberHeroes Academy a pridá aj zákulisné príbehy a ukážky vybraných útokov.

## **AKÉ NÁSTROJE POUŽÍVAME PRI ORGANIZÁCII NAMAKANÝCH PODUJATÍ**

**Mirek BIŇAS (SK)**

Všetko sa začalo podujatím Namakaný deň v roku 2017 a jednoduchou webovou stránkou. Z nadšeneckej akcie pre študentov informatiky na Technickej univerzite v Košiciach sa postupne stalo podujatie pre stovky stredoškôľakov a ich učiteľov. Odvtedy sme zorganizovali viac ako 50 online webinárov, viaceré prezenčné workshopy a publikovali niekoľko vzdelávacích materiálov. Za každým podujatím však stojí množstvo organizačnej práce – od plánovania programu, komunikácie s účastníkmi a prednášajúcimi až po propagáciu či vyhodnocovanie spätnej väzby. V príspevku predstavím open source nástroje a voľne dostupné online služby, ktoré nám tieto činnosti pomáhajú efektívne zvládať a ktoré sa nám v praxi najviac osvedčili.

## **DRYDOCK – U-BOOT PRO LINUXOVÉ TELEFONY**

**Petr HODINA (CZ)**

DryDock je špeciálne sestavený U-Boot bootloader zaměřený na mainline linuxové telefony. Cílem projektu je snížit bariéru vstupu pro vývojáře kernelu a distribucí tým, že do bootloaderu přináší funkcionality typu emulace sériové konzole po USB, DFU, Fastboot, Mass storage a dále PXE a TFTP boot.

## **GEMINI AKO POMOCNÍK PRI VÝUČBE ÚVODU DO INFORMAČNEJ BEZPEČNOSTI PRE VIACERÉ ŠTUDIJNÉ PROGRAMY NARAZ NA EUBA**

**Eva RAKOVSKÁ (SK)**

Informačná a kybernetická bezpečnosť sa dnes stávajú kľúčovou súčasťou firemnej kultúry. V súčasnosti nejde len o záležitosť IT oddelení a data governance. Úzko súvisia s riadením podnikových procesov (process governance), ako aj so zosúladením interných pravidiel s legislatívou, reguláciami a výrobnými štandardmi. Práve preto je nevyhnutné odovzdávať základy kybernetickej bezpečnosti aj študentom neinformatických odborov. Na Ekonomickej univerzite v Bratislave na Fakulte hospodárskej informatiky pripravujeme študentov v programoch: Data Science, Dátová analytika a kontroling, Hospodárska informatika, Informačný manažment, Aktuárstvo, Účtovníctvo a audítorstvo. Všetci títo absolventi sa v praxi stretnú so záväznými európskymi reguláciami – či už ide o smernicu NIS2, nariadenie o digitálnej prevádzkovej odolnosti (DORA), ochranu osobných údajov (GDPR), elektronické peňaženky (eIDAS) alebo najnovší AI Act. Rovnako dôležitá je pre nich orientácia v priemyselných štandardoch (ISO/IEC 27001, PCI DSS) a metodických rámcoch (NIST, CIS).

Vysvetliť základné technické pojmy a širšie súvislosti vo veľkých skupinách (50 až 100 študentov z rôznych študijných programov) je pedagogicky mimoriadne náročné. V rámci seminárov musíme pokryť širokú škálu tém: základná terminológia a riadenie rizík, následky zanedbania bezpečnosti a reakcia na incidenty, typy útokov, základy kryptografie a certifikačné authority, elektronický vs. digitálny podpis, technické riešenia na ochranu OS, databáz, cloudu a sietí. Keďže takto široký rozsah sa nedá efektívne študovať priamo z rozsiahlej literatúry a právnych noriem, je nutné informácie sumarizovať a prepájať.

V tomto procese sa ako neoceniteľný pomocník ukazuje využitie veľkých jazykových modelov. Každý z dostupných modelov vyniká v niečom inom a nie je jednoznačné, ktorému dať prednosť. DeepSeek-R1 je výborný na hĺbkové logické uvažovanie. Claude 3.5 Sonnet dominuje v oblasti programovania a kódovania. Google Gemini sa ukazuje ako najvhodnejší nástroj na prácu s rozsiahlymi dokumentmi (ako sú NIS2, DORA či Zákon o kybernetickej bezpečnosti). Vďaka tomu je Gemini ideálnym asistentom pri príprave prednášok a skrátených študijných materiálov pre rôznorodé publikum. Prompty je možné zadávať s presnými špecifikáciami ako napríklad: „Chystám prednášku na tému ‘Detekcia ransomvéru pomocou strojového učenia’ pre študentov Data Science a Hospodárskej informatiky. Navrhni mi štruktúru 90-minútovej prednášky. Pre dátových vedcov zahrň prehľad algoritmov (napr. Random Forest) na analýzu sieťového správania. Pre

hospodárskych informatikov doplní biznisovú časť: finančné straty pri výpadku a vplyv ransomvéru na reputáciu firmy. Na záver priprav 3 diskusné otázky do pléna.“ Pomocou takýchto zadaní dokáže vyučujúci rýchlo získať kvalitný základ pre prednášky a overené zdroje, od ktorých sa môže ďalej odraziť.

Pri počte 50 až 100 študentov v skupine je detailná manuálna kontrola všetkých priebežných zadaní takmer nemožná. Ako optimálny model sa osvedčila tvorba spoločnej Wiki v prostredí LMS Moodle, doplnená o ústnu prezentáciu vybranej témy. Ďalšou povinnosťou študentov je aktívne zapájanie sa do diskusných fór, ktoré rozvíjajú témy z Wiki. Aj pri kontrole týchto diskusií nachádza Google Gemini svoje uplatnenie. Dokáže promptne porovnať dve študentské odpovede z hľadiska ich nadväznosti, identifikovať zhody či odlišnosti v argumentácii, posúdiť štruktúru a analyticky vyhodnotiť, ktorý príspevok je kvalitatívne lepší. Výrazne tak šetrí čas pri pedagogickom hodnotení.

Počas semestra je mimoriadne efektívne zadávať študentom rôzne typov testov. Ide buď o autokorektívne testy na samoštúdium (kde majú študenti možnosť opakovania toho istého testu a získajú správne odpovede aj s vysvetlením), alebo o ostré testy, ktoré sa započítavajú do finálneho hodnotenia. Pri oboch typoch je kľúčové mať k dispozícii rozsiahlu databázu otázok a ich alternatív. Najväčšou výzvou pre vyučujúceho býva vymyslieť dostatočné množstvo relevantných, no nesprávnych možností (distraktorov).

V tejto fáze sa Google Gemini ukazuje ako ideálny pomocník, ktorý dokáže navrhnúť logické a uveriteľné nesprávne odpovede. Pri takomto zadaní je však potrebné presne špecifikovať nielen samotnú otázku, ale aj požadovanú hĺbku vedomostí a presný počet správnych možností. Príprava testov úzko nadväzuje nielen na prednášky, ale aj na študentské Wiki a diskusné fóra. Aby sme študentov motivovali čítať tieto materiály, stačí do Gemini vložiť krátky odstavec textu z Wiki s promptom na vytvorenie otázky a správnych odpovedí priamo na základe tohto textu. Takto vygenerované úlohy slúžia na priebežné overovanie vedomostí, ale aj ako základ pre finálnu skúšku.

Pri veľkom počte študentov, potrebuje učiteľ dostatok jednoznačne sformulovaných zadaní aj pre otvorené otázky na skúšku. Gemini ich často dokáže naformulovať zrozumiteľnejšie než človek. Podobne ako pri diskusných príspevkoch, aj pri hodnotení otvorených odpovedí na skúške môže Gemini asistovať. Dokáže porovnať študentské odpovede so vzorom a navrhnúť adekvátne bodovanie. Aby bol výsledok spoľahlivý, prompt musí obsahovať: presnú hodnotiacu stupnicu; vzorovú správnu odpoveď, kontext (napr. z akého študijného programu študenti sú, aké témy prebrali a do akej hĺbky). Bez dodatočných kritérií máva Gemini tendenciu hodnotiť príliš zhovievavo. Správne nastavený model však pomáha vyučujúcemu nájsť rovnováhu medzi subjektívnym pohľadom a spravodlivosťou.

Najväčšia výhoda kombinácie nástrojov LMS Moodle (Wiki, diskusie) a modelu Gemini spočíva v schopnosti promptne reagovať na zmeny. Kybernetická a informačná bezpečnosť sú dynamické oblasti, v ktorých je neustále sledovanie trendov a vyhľadávanie príkladov z praxe časovo aj mentálne vyčerpávajúce. AI prináša do výučby flexibilitu a umožňuje so študentmi rozoberať aktuálne témy v reálnom čase. Netreba však zabúdať na riziká. Využívanie AI si vyžaduje neustálu obozretnosť.

1. Ide o dôslednú kontrolu, preto všetky vygenerované materiály a testové otázky musí učiteľ pozorne prečítať a overiť v relevantných zdrojoch.

2. Stále je nezastupiteľná rola učiteľa a hodnotenie otvorených odpovedí a diskusií je vždy výsadou a osobnou zodpovednosťou učiteľa, nie umelej inteligencie.



Zodpovednosť za finálne rozhodnutia a známky zostáva na človeku. Každé hodnotenie je v konečnom dôsledku do istej miery subjektívne, no veľké jazykové modely nám slúžia ako vhodný nástroj na to, aby sme sa priblížili k objektívite hodnotenia. Po dvoch rokoch experimentovania s využívaním Gemini v povinne voliteľnom predmete Úvod do informatickej bezpečnosti pre viac študijných programov sa ukazuje, že kombinácia vytvárania stručnejších študijných materiálov obohatených o konkrétne príklady, incidenty a riešenia v podnikoch, spolu s dynamikou študentských diskusií a Wiki, dodáva výučbe novú dynamiku a študenti sa aktívnejšie zapájajú na seminároch. Z pohľadu učiteľa si však seminár vyžaduje neustále vzdelávanie sa a sústredenie na diskutované a prezentované témy.

## **CYBERGAME: FENOMÉN, KTORÝ HLADÁ NOVÉ HVIEZDY KYBERBEZPEČNOSTI**

**Adam GAJDOŠÍK (SK), Miroslava MICHALÍKOVÁ (SK)**

Od nuly k svetovému renomé za päť rokov. Cesta originálnej slovenskej značky a ľudia, ktorí za ňou stoja. CyberGame otvorila svet kybernetickej bezpečnosti širokej verejnosti. Aj napriek atraktívnemu názvu však zostáva náročnou programátorskou súťažou, ktorej autorský developerský tím tvoria profesionáli z Národného centra kybernetickej bezpečnosti. Odborným garantom CyberGame je Národný bezpečnostný úrad. Hrajú juniory, študenti, učitelia, programátori, profesionálni bezpečáci. Vzniká excelentná základňa pre hľadanie talentov a budovanie juniorského reprezentačného tímu Slovak Cyber Team. CyberGame registruje hráčov z vyše 50 krajín a v piatom ročníku pritiahli tri hracie platformy viac ako päťtisíc účastníkov. CyberGame obsahuje viac ako sedemdesiat úloh z oblastí: malvérová, forenzná a OSINT analýza, kryptografia a ofenzívna bezpečnosť. Stále pribúdajú úlohy inšpirované reálnou praxou a skutočnými incidentmi, takže atraktivita súťaže rastie a hra zároveň slúži ako tréningová platforma pre profesionálov.

## **A COMPARATIVE STUDY OF TRANSFORMER MODELS FOR SLOVAK DAY-AHEAD MARKET ELECTRICITY PRICE FORECASTING**

**Matúš REMEŇ (SK)**

Accurate electricity price forecasting is essential for short-term trading, portfolio optimization, and risk management in day-ahead electricity markets. This task is particularly challenging in smaller and highly interconnected bidding zones, where prices are affected by nonlinear market dynamics, cross-border constraints, volatility, price spikes, and increasingly frequent negative prices. This paper evaluates the performance of selected transformer-based time-series forecasting models for Slovak Day-Ahead Market prices. Five architectures are compared: Temporal Fusion Transformer, PatchTST, Crossformer, iTransformer, and TimeXer. The models are assessed for both deterministic point forecasting and probabilistic quantile forecasting.

## HACK TO LEARN, LEARN TO HACK

**Lukáš NAĎ (SK)**

Open-source nástroje sú základom etického hackingu — od Kali Linuxu, cez Metasploit a Nmap, až po moderné nástroje s prvkami umelej inteligencie. Prednáška predstaví tento ekosystém a na príkladoch z praxe — vrátane skúseností s vedením Siemens Healthineers Cybersecurity Academy pre stredoškolských študentov — ukáže, že útočné myslenie nie je len doménou špecialistov, ale univerzálnym prístupom k riešeniu problémov, ktorý obohacuje prácu každého vývojára, učiteľa aj výskumníka. Účastníci odídu s konkrétnymi odporúčaniami, ako začať svoju cestu v ofenzívnej bezpečnosti pomocou voľne dostupných nástrojov a platforiem.

## NAHLIADNIME ZA OPONU WIKIPÉDIE

**Matej GROCHAL (SK)**

Odpovede na niektoré otázky o fungovaní Slovenskej Wikipédie (s možnosťou Q n A).

## FOTOVOLTIKA V DOMÁCNOSTI: AUTOMATIZÁCIA A ZDIELANIE

**Michal PÁLENÍK (SK)**

Fotovoltické panely sú bežnou súčasťou koloritu vidieka. Dokážu výrazne pomôcť s nákladmi na elektrinu, spoľahlivosťou dostupnosti elektriny či s pomocou rodinným príslušníkom. Na získanie plného potenciálu využitia fotovoltiky je nutné presunúť spotrebu z iných energetických zdrojov na elektrinu, automatizovať využívanie elektriny v čase, keď je lacná, využívať rôzne tarify a v prípade finančnej dostupnosti využívať fyzickú baterku. Ukážeme si základné možnosti automatizácie cez HomeAssistant. Druhým spôsobom, ktorý zvyšuje využitie panelov, je zdieľanie elektriny medzi domácnosťami. Predstavíme si základné podmienky na sfunkčnenie zdieľania cez EDC, možnosti nastavenia v prípade viacero domácností, finančné prínosy zdieľania a výhody plynúce zo zapojenia viacero dodávateľských aj odberateľských domácností.

## (R)EVOLÚCIA ŠKOLSKÝCH PROGRAMOVACÍCH NÁSTROJOV

**Zuzana TKÁČOVÁ (SK)**

Prednáška ponúka stručný prehľad aktuálnych a modernizovaných programovacích nástrojov využiteľných vo výučbe informatiky. Zameriava sa na to, ako sa dnes mení spôsob vstupu do programovania a aké nové možnosti prinášajú súčasné nástroje pre žiakov aj učiteľov. Účastníci získajú orientáciu v tom, čo je dnes vo svete školských nástrojov nové, prepracované alebo znovu objavené v modernej podobe. Cieľom je pomôcť učiteľom zorientovať sa v aktuálnom ekosystéme a vybrať nástroje, ktoré dávajú zmysel v dnešnej výučbe informatiky.

## VIBECODING – CESTA OD KÓDU K DIGITÁLNEMU PRODUKTU

**Zuzana TKÁČOVÁ (SK)**

Vibe coding nie je náhrada programovania. Je to posun v tom, ako o ňom premýšľame. Tato prednáška prináša vibe coding ako moderný prístup k vývoju digitálnych produktov, kde sa programovanie stáva súčasťou širšieho procesu – návrhu, experimentovania, rýchleho prototypovania a iterácií. Ukážeme si, ako môže učiteľ informatiky viesť žiakov od prvotnej myšlienky až po funkčný prototyp, pričom AI nástroje slúžia ako katalyzátor nápadov, zrýchlenie vývoja a priestor na experiment. Cieľom nie je písať viac kódu. Cieľom je vidieť celý systém. A práve v tom sa mení informatika z „programovania úloh“ na tvorbu digitálnych produktov s reálnym dopadom.

## FORĎEJO A DECENTRALIZOVANÝ GIT: JAK SE ZBAVIT ZÁVISLOSTI NA VELKÝCH KORPORÁTECH

**Petr VELYČKO (CZ)**

Pro většinu lidí jsou v dnešní době slova „git“ a „GitHub“ téměř synonyma. Když se řekne „nahraj to na git“, málokoho napadne, že adresa repozitáře nemusí začínat <https://github.com>. Na přednášce si však ukážeme, že se dá vydat i jinou cestou, zprovoznit „vlastní“ analog GitHubu (těmto službám se tradičně říká Git Forges) zvládne takřka každý. Nejednoho uživatele také potěší, že při tom nemusíme dělat kompromisy ani u těch pokročilejších možností, jako například GitHub (Forĝejo) Actions.

## MAPOVANIE POMOCOU YOLO: OPEN-SOURCE PRACOVNÝ POSTUP PRE DETEKCIU OBJEKTŮV POMOCOU YOLO A QGIS

**Jakub ŠPERKA (SK), Renata ĎURAČIOVÁ (SK),  
Tibor LIESKOVSKÝ (SK)**

Príspevok prezentuje open-source pracovný postup pre automatizovanú detekciu objektov z geopriestorových dát s využitím modelov YOLOv26 a YOLOv11 (You Only Look Once). Dôraz je kladený na integráciu metód počítačového videnia do prostredia GIS a na reprodukovateľnosť celého procesu spracovania. Metodika je demonštrovaná na dvoch typoch dát: digitálnych modeloch reliéfu (DMR) odvodených z leteckého laserového skenovania (LLS) so špecializovanou vizualizáciou pre detekciu archeologických mohýl a ortofotomozaikách pre detekciu bazénov. Vstupné rastre sú transformované na dlaždice (640×640 pixelov), pričom anotácie sú generované z geopriestorových vektorových dát a konvertované do kompatibilného tvaru pre tréning modelov YOLO.

Tréning a inferencia prebiehajú s využitím open-source nástrojov (Ultralytics YOLO, Python knižnice ako Rasterio a GeoPandas), pričom detekcia na veľkoformátových dátach je realizovaná pomocou „sliding window“ prístupu. Výsledky sú transformované späť do geopriestorového formátu (GeoPackage), čo umožňuje ich priame využitie v GIS

analýzach a tvorbe výstupov. Navrhnutý postup demonštruje, že open-source riešenia poskytujú škálovateľný a flexibilný prístup k detekcii rôznych typov objektov v geopries-torových dátach.

Táto práca bola podporená Agentúrou na podporu výskumu a vývoja na základe zmluvy č. APVV-22-0151 a projektom Vedeckej grantovej agentúry MŠVVaM SR a SAV, VEGA č. 1/0626/25.

## **OBČANSKÁ VĚDA VE SVĚTE OPEN GIS**

**Jakub TROJAN (CZ)**

Občanská věda (citizen science) prožívá v posledních letech výrazný rozmach, přičemž klíčovou roli v tomto procesu hrají nástroje otevřeného GIS. Příspěvek se zaměřuje na průsečík těchto dvou komunit: jak open-source geoinformační technologie demokratizují vědecký výzkum a umožňují veřejnosti aktivně přispívat k tvorbě i analýze geografických dat. Diskutovány budou rovněž výzvy spojené s kvalitou dat generovaných občanskými vědci a metody jejich validace v prostředí open GIS. Příspěvek zhodnotí potenciál citizen science pro akademický výzkum i pro potřeby veřejné správy – zejména v kontextu regionálního rozvoje. Cílem je inspirovat účastníky k využití synergií mezi open GIS komunitou a komunitou citizen science a ukázat, že otevřený software a otevřená data nejsou jen technickým nástrojem, ale i prostředkem pro demokratizaci vědy a posílení občanské participace ve výzkumu.

## **CYBERGAME OČAMI HRÁČA – ČO MI DALA KYBERNETICKÁ SÚŤAŽ A ČO MÔŽE DAŤ VAŠIM ŠTUDENTOM**

**Ivan ORAVEC ml. (SK)**

O CyberGame sa dá rozprávať z mnohých strán. Ja vám poviem, aké to je hrať. Na tohtoročnom CyberGame som skončil na 8. mieste v slovenskom poradí a prednáška je osobnou výpoveďou o tom, čo ma do súťaže dostalo, čo ma tam držalo, kde som narazil na strop a čo som objavil až v záverečných kolách. Pozrieme sa aj na to, ako generatívna AI mení dynamiku súťaže – nie ako hrozba, ale ako nová premenná, s ktorou musia rátať hráči aj tvorcovia výziev. Ak hľadáte inšpiráciu pre svojich študentov alebo sami zvažujete vyskúšať niečo podobné, táto prednáška je pre vás.

## **THE LUMI-Q VLQ IN THE EUROPEAN ECOSYSTEM**

**Branislav JANSÍK (CZ)**

The talk will present the LUMI-Q consortium and the VLQ quantum computer in the broader European HPC ecosystem, outlining its composition, properties, and technical characteristics and principles of operation. We will highlight access routes available for Slovak users, special focus will be given to integration with Karolina and other EuroHPC

supercomputers, enabling execution of hybrid quantum-classical workloads. Further, quantum computing application domains will be discussed and examples of quantum algorithms executed on the VLQ machine will be presented.

## MAPATHONY V UNIVERZITNÍM PROSTŘEDÍ

**Tereza SRNEČKOVÁ (CZ)**

Mapathony představují formu zapojení veřejnosti do aktivit občanské vědy, která kombinuje prvky neformálního vzdělávání, dobrovolnictví a práce s geodaty. Předložený příspěvek vychází z probíhající studie založené na sérii šesti mapathonů realizovaných v univerzitním prostředí na Univerzitě Tomáše Bati ve Zlíně. Výzkum sleduje různé aspekty účasti, včetně motivace, vnímaného přínosu, bariér a následného zapojení účastníků do mapování, a to prostřednictvím dotazníkového šetření doplněného o sledování aktivity účastníků po skončení akce. Předběžné výsledky poukazují na faktory, které mohou ovlivňovat motivaci účastníků a jejich ochotu pokračovat v mapování i po skončení akce a identifikují vybrané bariéry spojené s účastí a zapojením se. Příspěvek přispívá k diskusi o roli mapathonů v univerzitním prostředí a nabízí další poznatky využitelné pro jejich optimalizaci.

## PROSTOROVÁ ANALÝZA KRIMINALITY: OD DAT K POCITU BEZPEČÍ

**Michal PLÁNKA (CZ)**

Přednáška se zaměřuje na prostorovou analýzu kriminality s důrazem na rozdíl mezi registrovanou a latentní kriminalitou. Zatímco registrovaná kriminalita vychází z oficiálních statistik (např. data Policie ČR prezentovaná prostřednictvím mapových portálů), latentní kriminalita odráží subjektivní vnímání bezpečnosti obyvatel, které lze zachytit pomocí pocitového mapování. Cílem je ukázat, jak se tyto dva pohledy na bezpečnost v prostoru liší, kde se naopak překrývají a jaké faktory mohou tyto rozdíly ovlivňovat. Součástí přednášky bude také práce s otevřenými daty o kriminalitě a trestech odnětí svobody poskytovanými Ministerstvem spravedlnosti a jejich využití v prostředí GIS. Představeny budou základní analytické přístupy a vizualizační techniky, které umožňují lépe porozumět prostorovým vzorcům kriminality a podpořit rozhodování v oblasti veřejné bezpečnosti, urbanismu či regionálního rozvoje.

## EAGIFS – YET ANOTHER CONTENT MANAGEMENT SYSTEM WITH A FEW TWISTS

**Matúš CHOCHLÍK (SK)**

EAGifs is a versioned file management system implemented in modern Python. It allows to store files of various sizes and formats, both text-based and binary in filesystem repositories. The repositories can be cloned and distributed among multiple hosts and synchronized by using secure communication mechanisms like SSH and serve as a simple

back-up solution. It allows to organize files into groups store them in encrypted form and to attach simple textual tags (like hash-tags) and key-value attributes, like the title of a document, the genre of a media file or name of the artist or author. It also implements automated extraction of some meta-data embedded in some file formats. The stored files can be searched and manipulated using both very simple and more complex criteria and then number of version stored for each file can be set either by a repository-wide setting or individually. It also provides a FUSE filesystem driver that allows to mount a repository into customizable directory and file hierarchies.

## **NA ČO NÁM JE CSIRT.SK?**

**Ján DOVJAK (SK), Dávid POLAKOVIČ (SK)**

Krátko predstavenie jednotky CSIRT.SK a služieb, ktoré poskytuje.

## **BEZPEČNÝ VÝVOJ KÓDU V DOBE AI**

**Ján DOVJAK (SK), Dávid POLAKOVIČ (SK)**

Na čo si dať pozor pri vývoji kódu, ako pri tom bezpečne používať AI, a podobne.

## **TURISTICKÁ MAPA PRE TÝCH, KTORÍ OČAKÁVAJÚ VIAC AKO LEN OBYČAJNÚ MAPU**

**Filip CIRBUS (SK)**

Už je to dávno, čo ľudia začali používať papierové mapy. Aj v dnešnej dobe sa nájdú ľudia, ktorí na papierovú mapu nedajú dopustiť. Vyskúšali ste už ale online mapy? Poznáte už všetky? Chcete od digitálnej mapy viac ako vám aktuálne ponúka? Príďte spoznať mapový portál freemap.sk a jeho funkcie, ktoré vám chýbajú, alebo ste ani nevedeli, že ich potrebujete!

## **OD BÁDATELSKÝCH AKTIVÍT K OBČIANSKEJ VEDE: MOBILNÉ APLIKÁCIE A OTVORENÉ DÁTA V ENVIRONMENTÁLNEJ VÝCHOVE**

**Katarína SKOKANOVÁ (SK)**

Bádateľské aktivity sú v slovenských školách čoraz rozšírenejšou pedagogickou praxou – učitelia ich poznajú, využívajú a rozvíjajú. Výskumná otázka v bádateľskom vyučovaní slúži hlavne žiakovi; v občianskej vede slúži viac spoločnosti – a jej odpoveď môže ovplyvniť ochranu prírody aj verejné rozhodovanie. Analýza príspevkov z konferencie ŽiVEKo 2025 ukazuje, že 22

## STÁRNOUCÍ VĚDEC (KLIMATOLOG) V NOVÉ INFORMAČNÍ DOBĚ

**Radim TOLASZ (CZ)**

Výsadou stáří je možnost, a dokonce i povinnost, ohlížet se do minulosti a hodnotit. Dnes už nikdo nepochybuje, že neznalost historie nám přináší opakování již známých chyb. Poslouchejme tedy starší a zkušenější. Dnes se do vyššího věku dostávají pamětníci začátků překotného rozvoje programování, osobní výpočetní techniky a počítačů v kapse, které již ne úplně přesně označujeme jako telefony. Tento vývoj dramaticky změnil i klimatologii z vědy o minulosti na vědu o aktuálním stavu a budoucím vývoji podnebí. Z papírových výkazů jsme se postupně přes vysílačky, dálnopisy a pevné linky dostali k online datovým úložištím, které aktualizujeme minimálně každou hodinu, častěji i každých 10 minut nebo každou minutu. Od tužky a kalkulačky jsme se přes Fortran, Basic, C++ dostali ke skriptovacím jazykům JavaScript, R, Python a dalším. Potřebujete dnes pracovat s klimatologickými daty? Ukážeme si jak na to. Datové zdroje jsou dostupné ve formě OpenData, prezentace bývá standardně v tabulkách, grafech nebo mapách a jednoduché aplikace nám připraví AI.

## OPEN SOURCE APLIKÁCIE V HUMANITÁRNOM MAPOVANÍ

**Patrik BRIGANT (SK)**

Zaujímalo Vás niekedy aké nástroje sa používajú pri mapovaní oblastí zasiahnutými epidémiami či prírodnými katastrofami? Predstavíme si tie, ktoré používajú Lekári bez hraníc a podobné neziskové humanitárne organizácie zaoberajúce sa vytváraním podkladovej mapy OpenStreetMap v miestach postihnutými týmito nešťastiami.

## PTT SOFTWARE: JAK SE PLÁNUJE MHD

**Jaromír ŠULC (CZ)**

Organizace městské hromadné dopravy je velmi komplexní činnost, pod kterou spadá plánování linek, jízdních řádů, vytížení vozidel i pracovní rozpis řidičů. Není snadné dát vše dohromady tak, aby celek fungoval optimálně a přitom každé zpoždění či výpadek nezpůsobily totální chaos. Na přednášce se podíváme na obecný proces všech činností, které dnes dopravní podnik provozující MHD musí udělat, aby autobusy každé ráno vůbec vyjely. Dále co musí všechno udělat, aby o svém provozu dokázal informovat cestující. U části procesů si ukážeme, jak se dají matematicky popsat a algoritmicky řešit, jak musí vypadat vstupní data, aby tato automatizace vůbec byla možná a jak s daty pracovat, aby zbytečně nespádala do kategorie „big data“. Dle zájmu posluchačů se dotkneme teorie grafů, genetických algoritmů nebo lineárního programování.

## **OTVORENÁ VEDA A AKTUÁLNE VÝSLEDKY PRIESKUMOV CVTI SR NA VYUŽÍVANIE OTVORENÉHO SOFTVÉRU V SLOVENSKOM AKADEMICKOM PROSTREDÍ**

**Zuzana STOŽICKÁ (SK)**

Centrum vedecko-technických informácií SR už od roku 2017 prostredníctvom Kontaktnej kancelárie pre open access aktívne propaguje princípy otvorenej vedy na Slovensku, tvorí a prekladá vzdelávacie materiály, organizuje podujatia a podporuje dobrú prax vo všetkých aspektoch otvorenej vedy, od vedeckého publikovania, cez otvorené vzdelávacie zdroje, výskumné dáta, otvorené IT nástroje, zodpovedné hodnotenie vedy, až po občiansku vedu. V súlade s Národnou stratégiou pre otvorenú vedu na roky 2021-2028 sme v roku 2025 uskutočnili pilotný prieskum využívania otvorených IT nástrojov na Univerzite Komenského v Bratislave a v roku 2026 celoslovenský prieskum ekosystému otvorenej vedy, ktorý mapuje úroveň povedomia, postojov a aplikovania otvorených postupov v slovenskom akademickom prostredí. Budeme diskutovať o najvýznamnejších výsledkoch týchto prieskumov a ich implikáciách pre nasledujúcu stratégiu rozvoja otvorenej vedy na Slovensku.

## **STAV A PROBLÉMY ZVEREJŇOVANIA OTVORENÝCH ÚDAJOV V PODMIENKACH ÚZEMNEJ SAMOSPRÁVY**

**Vladimír JENDRISEK (SK)**

Napriek tomu, že otvorené údaje majú moc odbúravať byrokráciu a zlepšovať služby pre občanov, v slovenskej samospráve tento potenciál zatiaľ nie je využitý. Štúdia Najvyššieho kontrolného úradu SR z roku 2025 odhalila nelichotivý stav: zatiaľ čo väčšie mestá a VÚC sa snažia, u malých obcí zverejňuje využiteľné dáta menej ako 1

Hoci Slovensko prijalo medzinárodné záväzky už v roku 2011, reálny pokrok je pomalý a nesystematický. Ak chceme modernú a transparentnú samosprávu, musíme začať vnímať dáta ako strategickú prioritu, nie ako administratívnu záťaž.

## **KVALIFIKOVANÉ ELEKTRONICKÉ PODPISY BEZ MÁGIE A OPEN-SOURCE AUTOGRAM(Y)**

**Marek CELUCH (SK)**

Obrázok podpisu v PDF nie je najdôveryhodnejším spôsobom autorizácie dokumentu. V EÚ máme vďaka nariadeniu eIDAS zhodu na kvalifikovaných elektronických podpisoch, ktoré majú veľmi vysokú mieru dôveryhodnosti. Takéto podpisy vieme na Slovensku vytvárať pomocou občianskeho preukazu všetci a zadarmo. Aby to bolo aj jednoduché, v Slovensko.Digital sme vyvinuli open-source podpisovač Autogram, neskôr pribudla mobilná verzia a najnovšie sa tešíme z podpisového portálu, ktorý celý proces združuje na jednom mieste. Kvalifikovaný elektronický podpis je v skutočnosti pomerne jednoduchá



kryptografická operácia, no v praxi okolo nej vzniká veľa nedorozumení. V tejto prednáške si ukážeme, čo sa reálne deje, keď dokument podpisujeme elektronicky – od hashu dokumentu cez podpis privátnym kľúčom uloženým v čípe „QSCD“ až po uloženie do niektorého z formátov XAdES, PAdES, CAdES.

Vysvetlíme si, prečo niektoré podpisy nazývame kvalifikované, akú právnu silu majú podľa nariadenia eIDAS, na čo sa dajú použiť a ako sa takéto podpisy overujú pomocou certifikátov a „dôveryhodných zoznamov“. Stručne predstavíme aj open-source projekty, ktoré v Slovensko.Digital vyvíjame na podpisovanie dokumentov – Autogram, Autogram v mobile a Autogram Portal. Ukážeme si, ako tieto aplikácie využívajú knižnicu DSS, a povieme si, ako sa do nich aj vy môžete zapojiť.

## HOW VECTOR DATABASES WORK AND WHAT IS IT GOOD FOR?

Jiří ČINČURA (CZ)

Vector databases are the last hot topic. But what is a vector database? How it relates to LLMs? Why even traditional relational databases are gaining vector search capabilities?

## THE RABBIT HOLE OF BUILDING A GPS-SYNCHRONIZED TIME SERVER

Jiří ČINČURA (CZ)

Accurate and reliable time underpins everything from secure communications to distributed systems, yet most of us take it for granted. What starts as a seemingly simple idea – „Why not just build my own Stratum 1 time server?“ – quickly turns into a deep dive through layers of hardware quirks, signal integrity issues, and the unexpected joys of GNSS modules. In this talk, we'll explore the journey of building a GPS-synchronized NTP/NTS server: from choosing receivers and handling PPS signals, tuning operating systems, and ensuring long-term stability. Along the way, you'll hear about pitfalls, lessons learned, and the surprising complexity behind something as „simple“ as keeping the right time.

## POPULARIZACE VĚDY | PÁTEČNÍCI

Vítězslav ŠKORPÍK (CZ)

Jak se setkat s budoucím prezidentem, nejlepším českým zpěvákem, hvězdou Youtube scény, či bývalým předsedou Akademie věd? Veďte přednáškový cyklus a nebudete mít problém! Pátečníci je zažité, byť neoficiální, označení účastníků pravidelných přednášek organizovaných Popularizační sekcí Českého klubu skeptiků Sisyfos. Tradici pátečních přednášek o vědě a společnosti zavedl už v roce 2000 fyzik Pavel Vachtl a historicky a myšlenkově se Pátečníci odkazují na stejnojmennou prvorepublikovou skupinu kolem prezidenta Masaryka a spisovatele Čapka. Moderní Pátečníci se schází na Přírodovědecké

fakultě UK a mají na kontě celou řadu úspěchů. Dosáhli například na přednášku s milionem zhlédnutí na YouTube, přednášeli u nich světoznámí egyptologové, astronomové, či politici.

## **VIBE CODING V PRAXI: AKO STAVIAM APLIKÁCIE, KTORÉ MAJÚ TISÍCE POUŽÍVATELOV SÁM, BEZ TÍMU**

**Martin GÉČ (SK)**

Vibe coding je zmena paradigmy v tom, ako sa dnes softvér stavia. Prednáška vychádza z reálnych skúseností: autor samostatne vyvinul a prevádzkuje aplikáciu Rejoy (wellness, iOS/Android/web) s viac ako 10 000 používateľmi – predtým bez AI, dnes s AI ako vývojovým tímom, ktorý má na starosti okrem vývoja aj marketing a analytiky. Na konkrétnych príkladoch ukáže, ako Claude Code a Codex menia architektúru produktu a firmy. Od prototypov aplikácií cez nasadenie až po autonómnych AI agentov. Pozornosť bude venovaná aj limitom AI: čo stále vyžaduje ľudský dohľad a kde sú bezpečnostné riziká autonómnych agentov.

## **INOVACE PRAKTICKÉ VÝUKY – VIRTUÁLNÍ LABORATOŘE NETLAB**

**Jaroslav BURČÍK (CZ), Marcel POLÁČEK (CZ), František KÝN (CZ),  
Kristián TICHOTA (CZ)**

NetLAB představuje revoluční přístup k praktické výuce a výzkumu v oblasti informačních technologií. Díky vzdálenému přístupu nabízí studentům i vědcům snadný přístup a prostor pro výuku, simulaci, návrh a testování moderních scénářů nejen ze světa sítí, ale i kyberbezpečnosti a operačních systémů.

## **SROVNÁNÍ TECHNOLOGIÍ DIGITÁLNÍ SAZBY Z UŽIVATELSKÉHO HLEDISKA: LATEX3, LUALATEX A MARKDOWN**

**Vítek STARÝ NOVOTNÝ (CZ)**

Přednáška představí trojici technologií digitální sazby: LaTeX3 jako dokumentový formát a programovací jazyk (expl3), LuaTeX jako typografický stroj s vestavěným programovacím jazykem Lua a Markdown jako značkovací jazyk a preprocesor pro HTML a další výstupní formáty.

Tyto technologie budou následně porovnány z pohledu uživatele. Jaké jsou výhody psaní dokumentů v LaTeXu oproti Markdownu a naopak? Kdy se při programování vyplatí použít jazyk Lua a kdy expl3? Existují i dnes důvody, proč místo LuaTeXu použít starší TeXové stroje? Co jsou PE gramatiky, čím se liší od regulárních výrazů a k čemu se hodí? Jak lze při sazbě dokumentů v jazyce Markdown uplatnit princip nedestruktivních úprav známý z grafických a zvukových editorů?

Prednáška vychádza z viac než deseti let autorovy práce na projektoch súvisiacich s LaTeXem 3 a Markdownem ve školství, ve firmách i v oblasti svobodného softwaru. Je určena pro zájemce o přípravu elektronických dokumentů a typografické programování bez ohledu na předchozí zkušenosti. Posluchači se seznámí se základními principy práce s jazyky LaTeX a Markdown a s programováním v explu 3 a Lua. Získají také přehled o tom, kdy je vhodné jednotlivé technologie použít a jak mezi nimi volit v typických situacích při přípravě dokumentů a typografických nástrojů.

## **ATLAS DEMÄNOVSKÝCH JASKÝŇ – SLAM LIDARY, GNSS, THERION, CLOUDCOMPARE, KOMPLEXNÉ MODELY A TVORBA PUBLIKÁCIE**

**Pavel HERICH (SK)**

Ako sa dá zmapovať 500 jaskýň v jednej doline? Akú úlohu v tom zohrávajú mobilné (SLAM-based) lidary a ako obstáli superlacné resp. kvalitné GNSS RTK systémy? Predstavíme možnosti slovenského softvéru Therion pre tvorbu speleologickej dokumentácie, vytváranie komplexných modelov jaskýň s povrchom a tiež možnú budúcnosť jaskynného mapovania.

## **NOBELOVY CENY**

**Vítězslav ŠKORPÍK (CZ)**

Proč Nobelovu cenu (NC) obdrželo tak málo žen, a ještě méně Čechů a Slováků? Proč není astrofyzika fyzikální obor? A jsou Nobelovy ceny ve skutečnosti politická bitva? Nobelovy ceny, které se udělují od roku 1901, jsou v oceňovaných oborech považovány za nejvyšší možnou poctu. Přesto je jejich historie plná omylů a opomenutí. Velké vášně vyvolalo nedávno udělení NC za fyziku klimatologům a počítačovým vědcům. Jsou tato ocenění v pořádku? A jaké obory se v rámci fyziky oceňují? Proč se neudělují NC za geofyziku, či biofyziku, ale za kvantovou fyziku ano?

## **PROČ A JAK DOSTAT VLASTNÍ R-KOVÉ KÓDY NA CRAN?**

**Matúš MACIAK (CZ)**

This presentation provides a brief overview of the process of developing an R package, illustrated through the example of ProfileLadder — the package which embeds stochastic, data-driven, and functionally-based reserving methods for claims reserving in insurance business. The talk follows the path from the methodological research, theoretical justification, and personal R scripts, to a fully structured package submitted to CRAN (Comprehensive R Archive Network), highlighting key steps such as code organization, development, testing, deployment, and version control. The talk concludes with reflections on the practical challenges and benefits of making software publicly available,

offering a short case study and a few lessons learned for researchers considering creating their own R packages.

## LINUX NA TELEFONU

David HEIDELBERG (CZ)

Co kdyby váš telefon byl skutečně váš? Linux na mobilech není jen technologický experiment, ale snaha o návrat kontroly, soukromí a otevřenosti do zařízení, které nosíme každý den v kapse. V této přednášce se podíváme na současný stav linuxových mobilních distribucí, komunitu, která za nimi stojí, i na to, proč má smysl se o tuto alternativu zajímat.

## MÔŽU ZA TO ČERTI, A JEHO ŽENA!

Slavko FEDORIK (SK)

Krátke zhrnutie mojej cesty vizualizácie dát, ktorá začala pri matplotlib a skončila... pri matplotlib, alebo prečo som prešiel k pygal a nakoniec sa i tak vrátil k matplotlib. A tiež ako generujem mapové diagramy, že to vôbec nie je až také ťažké, ako to spočiatku vyzeralo.

Celé ako ukážka, že vizualizácia dát je aj pre ľudí, ktorí nie sú vedcami ani geografmi, stačí chcieť dáta vizualizovať. A že sa pri tom môže stretnúť s problémami, ktoré by vôbec problémom byť vôbec nemuseli, keby... A to všetko s trochou kódu v Pythone...

## SEZNÁMENÍ S PANOPTEXUM | PANDOC WRITER FOR OPTeX

Jan MARTÍNEK (CZ)

PanOpTeXum je projekt, který přidává do Pandocu podporu formátu OpTeX. Pandoc umí převádět dokumenty mezi desítkami formátů – Markdown, HTML, Word, reStructuredText a mnoho dalších – ale OpTeX dosud chyběl. PanOpTeXum to řeší vlastním writerem v jazyce Lua, takže uživatelé OpTeXu mohou konečně využít celý ekosystém Pandocu a snadno zpracovat dokumenty z libovolného vstupního formátu přímo v OpTeXu. Ukážeme si jak projekt použít a zkombinovat tím výhody různých formátů.

## TEX A METAPOST POD ZEMOU: TVORBA MÁP JASKÝŇ V PROGRAME THERION

Martin BUDAJ (SK)

V príspevku ukážeme, prečo je na tvorbu máp jaskýň žiaduca flexibilita MetaPostu a plain TeXu. Aj pri týchto nástrojoch narážame na ich limity, ktoré sa však (väčšinou) dajú prekonať. Na príkladoch predstavíme, ako sú definované mapové znaky a ako sú implementované napr. unicode texty, tieňovanie a priehľadnosť v MetaPoste alebo ako

sa skladá komplexná mapa v plain TeXu s podporou správy farieb a georeferencovaných PDF.

## **DOPRAVNÉ LABORATÓRIUM KATEDRY ŽELEZNIČNEJ DOPRAVY**

**Lumír PEČENÝ (SK)**

Katedra železničnej dopravy disponuje moderným špecializovaným dopravným laboratóriom, ktoré je z hľadiska technického a technologického vybavenia jedinečným zariadením svojho druhu nie len na Slovensku, ale aj v celoeurópskom priestore.

Dopravné laboratórium bolo v priebehu rokov 2022-2024 kompletne zrekonštruované a zmodernizované. Počas týchto troch rokov boli realizované stavebné úpravy ako dopravného laboratória, tak aj príslušných odborných učební, v ktorých prebiehala výučba študentov študijného programu železničná doprava.

## **JAK SE DĚLÁ KOSMONAUTIX**

**Dušan MAJER (CZ)**

Přednáška, která se nevěnuje kosmonautice samotné, ale má za úkol seznámit účastníky s naším webem kosmonautix.cz, jeho historií, tvorbou, ale hlavně má nabídnout pohled za oponu – jak vznikají články a jak k nim jako redakce přistupujeme.

## **OPEN-SOURCE SMEROVAČ NA BEŽNE DOSTUPNOM HARDVÉRI**

**Eubor JURENA (SK)**

Témou prednášky bude predstavenie možností postavenia vlastného smerovača na bežne dostupnom hardvéri pomocou open-source softvéru. Aké sú limity dnešných CPU, sieťových kariet a je možné smerovať line-rate  $n \cdot 10\text{Gbps}$  v Linuxe?

## **BALÍČEK RCZECHIA**

**Jindra LACKO (CZ)**

Předmětem příspěvku je úvod do problematiky prostorových dat – v čem jsou stejná jako ostatní a jak se liší – v kontextu statistického programovacího jazyka R. Ukážeme si, jak a kde prostorová data hledat a najít, a představíme několik technik přínosných pro statistickou praxi: tvorbu přesvědčivých vizuálů (kreslení map) a přípravu vstupů pro další analýzu, jako jsou distanční matice a matice sousedství. Techniky si ukážeme na reálných datasetech z oblasti geomarketingu, dopravního modelování a výzkumu dlouhodobosti.

## **AKO SME VYTVORILI NÁSTROJ PRE VIZUALIZÁCIU SFLOW (OPEN SOURCE)**

**Blažej KRAJŇÁK (SK)**

Když jde o parsování, ukládání a vizualizaci síťových telemetrických dat v sítích o rychlosti stovek gigabitů, mnoho open-source nástrojů přestává být dostačujících. Tato prezentace popisuje, jak jsme vytvořili lehký, ale výkonný interní nástroj pomocí stacku GoFlow2 – Clickhouse – Grafana.

## **VÝUKA ZÁKLADŮ ANALÝZY DAT: EXCEL, RSTUDIO A AI JAKO VZÁJEMNĚ SE DOPLŇUJÍCÍ NÁSTROJE**

**Martina LITSCHMANNOVÁ (CZ)**

Schopnost pracovat s daty patří mezi klíčové kompetence současného vzdělávání. V praxi se však často setkáváme s otázkou, jak studenty do analýzy dat uvést tak, aby byl přechod od jednoduchých nástrojů k pokročilejším metodám přirozený a srozumitelný. Přednáška představí přístup, který využíváme k výuce základů analýzy dat v předmětech Základy statistiky (bakalářské studium) a Pravděpodobnost a statistika (magisterské studium). Ve výuce kombinujeme tři prostředí – Excel, RStudio a nástroje umělé inteligence.

Excel slouží jako přirozený vstupní bod pro práci s daty – studenti se v něm učí základní operace, práci s tabulkami, jednoduché výpočty a základní vizualizaci. Následně je ukázáno, jak lze stejné typy úloh (a posléze i složitější úlohy) řešit v prostředí RStudio, které umožňuje reprodukovatelnou analýzu, efektivní práci s daty a tvorbu kvalitních grafů. Třetí složkou jsou nástroje AI, které mohou pomoci při generování kódu, vysvětlování postupů, hledání chyb nebo při interpretaci výsledků. V přednášce budou prezentovány konkrétní příklady výukových aktivit, které ukazují, jak tyto tři nástroje vhodně kombinovat. Pozornost bude věnována také tomu, jak AI využívat tak, aby podporovala porozumění postupům, nikoli pouze mechanické získávání výsledků. Cílem přednášky je ukázat, že propojení běžně používaného tabulkového procesoru, statistického softwaru a nástrojů umělé inteligence může výrazně usnadnit vstup studentů do světa analýzy dat a zároveň rozvíjet jejich kritické a analytické myšlení.

Vydanie tohto zborníka bolo podporené grantom KEGA 010ŽU-4/2026 „Matematika ako jazyk: myslenie, diskurz a porozumenie pomocou Open Source nástrojov“.

**OTVORENÝ SOFTVÉR VO VZDELÁVANÍ, VÝSKUME  
A V IT RIEŠENIACH**

Zborník príspevkov medzinárodnej konferencie OSSConf 2026,  
konanej 1. júla – 3. júla 2026 v Žiline

Prvé vydanie 2026

Elektronická sadzba programom pdf $\text{\LaTeX}$  Rudolf Blaško

Vydala Žilinská univerzita v Žiline

Tlač EDIS – vydavateľstvo UNIZA

Náklad 100 ks

Rozsah 238 strán

ISBN 978-80-554-2211-4

