

Spoločnosť pre otvorené informačné technológie

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



**Zborník príspevkov medzinárodnej konferencie
OSSConf 2011**

**6.–9. júla 2011
Žilina, Slovensko**

Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach

6.–9. júla 2011, Žilina, Slovensko

Organizátori: Miloš Šrámek, FMFI, Univerzita Komenského, SOIT, Bratislava
Michal Kaukič, Žilinská univerzita, SOIT, Žilina
Slavko Fedorik, SOŠ elektrotechnická, Poprad
Lýdia Gábrisová, Žilinská univerzita, Žilina
Karol Grondžák, Žilinská univerzita SOIT, Žilina
Aleš Kozubík, Žilinská univerzita, SOIT, Žilina
Marek Mahut, Red Hat CZ, SOIT, Brno
Tomáš Majer, Žilinská univerzita, SOIT, Žilina
Ľubica Micháľková, Žilinská univerzita, Žilina
Peter Mráz, DRU, SOIT, Kremnica
Miloslav Ofúkaný, Portál Geoinformatika, SOIT, Bratislava
Pavel Stříž, Univerzita Tomáše Bati, Zlín
Ladislav Ševčovič, FEI, Technická Univerzita v Košiciach, SOIT, Košice
Tatiana Šrámková, FEI, Slovenská technická univerzita, SOIT, Bratislava
Peter Štrba, Gymnázium Mikuláša Galandu, SOIT, Turčianske Teplice

Editori: Michal Kaukič
Ladislav Ševčovič
Miloš Šrámek
Pavel Stříž

Vydavateľ: Spoločnosť pre otvorené informačné technológie, Bratislava

ISBN 978-80-970457-1-5

DOI: 10.5300/2011-OSSConf

Sadzba programom pdfL^AT_EX Ladislav Ševčovič

Copyright © 2011 autori príspevkov

Ktokoľvek má dovolenie vyhotoviť alebo distribuovať doslovný opis tohoto dokumentu alebo jeho časti akýmkoľvek médiom 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.

Recenzenti

Bábel, Ján, Ing.
Balážovič, Ľuboš, Mgr., PhD.
Blaško, Rudolf, RNDr., PhD.
Blišťan, Peter, doc., Ing., PhD.
Boháčik, Ján, Ing., PhD.
Boltižiar, Martin, doc., PhD., RNDr., PhD.
Čačková-Talandová, Petra, Ing., Ph.D.
Dolejšová, Miroslava, Ing., Ph.D.
Fodrek, Peter, Ing., PhD.
Grondžák, Karol, doc., Ing., PhD.
Hofierka, Jaroslav, doc., Mgr., PhD.
Homolka, Lubor, Ing.
Ivanič, Jozef, Ing., PhD.
Janech, Ján, Ing., PhD.
Kaňuk, Ján, RNDr., PhD.
Kaukič, Michal, Mgr., PhD.
Klímek, Petr, doc., Ing., Ph.D.
Kozubík, Aleš, RNDr., PhD.
Kubíček, Petr, RNDr., CSc.
Kusendová, Dagmar, doc., RNDr., CSc.
Lehuta, Juraj, Mgr.
Levashenko, Vitaly, doc., Ing., PhD.
Lietz, Wojciech, DVM, PhD.
Mahút, Marek, Mgr.
Ofúkaný, Miloslav, Mgr.
Peško, Štefan, doc., RNDr., PhD.
Petrovič, Pavel, Mgr., PhD.
Rusnák, Pavol, Mgr.
Sadloň, Ľubomír, Mgr., Ing., PhD.
Sarga, Libor, Ing.
Straka, Juraj, Mgr.
Stříž, Martin, Ing.
Stříž, Pavel, Ing., Ph.D.
Ševčík, Jaroslav, Ing.
Ševčovič, Ladislav, RNDr.
Šrámek, Miloš, prof., Ing., PhD.
Šrámek, Rastislav, Mgr.
Zámečník, Peter, Ing.

SPONZORI KONFERENCIE



FAKULTA RIADENIA
A INFORMATIKY
ŽILINSKÁ UNIVERZITA

PARTNERI KONFERENCIE



Obsah

ÚVOD	9
POZVANÍ PREDNÁŠATELIA	11
Jiří Rybička X _Y L ^A T _E X – krok správným směrem	17
Stanislav Slačka Slobodný a otvorený softvér v slovenskom školstve	25
OCENENÉ ŠTUDENTSKE PRÁCE	31
OTVORENÝ SOFTVÉR VO VEDE A VÝSKUME.....	37
Richard Balogh Using an Open Software for Electronics and Robotics	39
Michal Hučko Utilisation of the Open Build Service for a volume data processing library	49
Matúš Chochlík MIRROR – a Portable C++ Reflection Library	57
Ján Janech, Tomáš Bača Principles of DSM Applied in Education and Research with UML .FRI Tool	67
Tomasz Kanik, Małgorzata Domino WEKA – Empirical Study on Applications of Data Mining Techniques in Veterinary Medicine	75
Lucia Pančíková, Katarína Zábovská Prognózovanie v prostredí R	87
Pavel Stríž Program Gambit ve výuce předmětu Rozhodování v riziku a nejistotě	95

Martin Šechný	
Jazyk GDL na spracovanie vedeckých dát	103
Michal Turek	
Využití Scilabu při koordinaci světelných křižovatek	111
Richard Turek	
Možnosti synchronizace linek MHD ve Scilabu	121
OPEN GIS – OTVORENY SOFTVÉR V GIS	131
Miroslava Dolejšová	
Zařazení GIS do výuky předmětu Informatika ve veřejné správě	133
Jaroslav Hofierka, Rastislav Mičaník	
Využitie open source technológií pri vývoji a aplikácii webových GIS riešení na príklade systému GISPLAN 2011	143
Imrich Jakab, Titusz Bugya	
Využitie slobodných softvérov v návrhu koncepcie rozvoja fakulty v oblasti GIS	151
Juraj Kisztner	
Spracovanie dát slovenského katastra pomocou open source aplikácií pre WEBGIS technológie a ich publikovanie 2011	165
Gabriel Petříček, Martin Mahrík	
Publikácia komplexného digitálneho modelu reliéfu prostredníctvom open source technológií 2011	171
T_EX A JEHO PRIATELIA	181
Petra Čáčková	
T _E X a otevřený software v malé firmě aneb Efektivní kancelář	183
Libor Sarga	
Nech T _E X nás být: pohled na krušné začátky sazečské z pohledu nesazečů	189

Pavel Stříž	
Typografie a čárové kódy	199
Pavel Stříž	
Balíček PGFplots: o tvorbě jednoho grafu na zakázku	209
OTVORENÝ SOFTVÉR VO VZDELÁVANÍ A V IT RIEŠENIACH	221
Rudolf Blaško	
L ^A T _E X nie je farba na maľovanie, ale na písanie	223
Rudolf Blaško	
L ^A T _E Xovník	236
Martin Foltin, Michal Blaho, Peter Fodrek	
Otvorený formát textových dokumentov v kancelárii	239
Aleš Kozubík	
Prezentačné materiály v triede beamer	249
Peter Tuhársky	
Použitie free/open-source technológií v samospráve mesta Banská Bystrica – I.	259
Peter Tuhársky	
Použitie free/open-source technológií v samospráve mesta Banská Bystrica – II.	271
ABSTRAKTY	286

Vážení čitatelia,

tento zborník je jedným z výstupov už tretej samostatnej konferencie na Slovensku, ktorá je venovaná slobodnému a otvorenému softvéru a jeho využitiu vo vzdelávaní, vede, ale aj v ostatných oblastiach, kde sa využívajú informačné technológie. Konferencia je jednou z aktivít občianskeho združenia „Spoločnosť pre otvorené informačné technológie“ (SOIT), <http://www.soit.sk>. Podobné podujatia na propagáciu otvoreného softvéru sme organizovali už od r. 2007 (dve špecializované sekcie v rámci medzinárodnej konferencie Aplimat 2007 a 2008 a *Víkend s Linuxom* v Kremnici v lete 2008). Na organizácii tohtoročnej konferencie sa významne podieľali aj partneri z Českej republiky (CSTUG, Liberix) a takisto skladbou účastníkov sa konferencia stala vskutku medzinárodnou.

V celosvetovom meradle, obzvlášť v štátoch EU, pozorujeme veľký nárast používania otvoreného softvéru (OSS). Dokonca aj viaceré komerčné firmy ponúkajú riešenia, kde je základom tento softvér. Veľké firmy v oblasti IT, ako Google, HP, IBM, Red Hat, nielen deklarujú podporu otvoreného softvéru, ale ho aj vyvíjajú a dávajú ďalej k dispozícii vývojárskej komunite. Tým sa výrazne rozširujú možnosti využitia OSS doma, vo vzdelávaní, výskumnej činnosti a tiež aj v softvérových firmách pre nekomerčné i komerčné riešenia.

Napriek týmto faktom, OSS sa dostáva do povedomia verejnosti ťažšie, ako komerčné operačné systémy a softvér. Šíri sa hlavne v menších komunitách, „ústnym podaním“ a osobnými príkladmi blízkych osôb. Existuje mnoho učiteľov, študentov, výskumníkov alebo prosto nadšencov, ktorí OSS vo svojej práci využívajú, nahrádzajú či dopĺňajú ním drahé komerčné riešenia a pritom získavajú cenné skúsenosti, z ktorých by mohli mať úžitok aj ostatní. Problémom je, ako tieto skúsenosti odovzdať. Jednou z príležitostí môže byť aj naša konferencia.

Sme presvedčení, že otvorený softvér sa môže veľmi dobre uplatniť v oblasti výučby a výskumu na všetkých stupňoch škôl. V mnohých prípadoch otvorený softvér v tejto oblasti poskytuje plnohodnotnú náhradu viacerých finančne náročných softvérových produktov (kancelárske balíky, matematický softvér, sieťové služby a administrácia, programy na výučbu rôznych predmetov na základných a stredných školách). Niektoré z týchto otvorených alternatív sú prezentované v materiáloch konferencie (webstránka konferencie a zborník).

Výhody otvoreného softvéru sa neprejavujú len v ekonomickej oblasti, ale majú aj významný dopad na myslenie žiakov, študentov i učiteľov. Podporujú tvorivý prístup k softvéru, ktorý sa hlbavému používateľovi už nejaví ako čierna skrinka, ale umožňuje mu prispôbiť si jeho funkcionality vlastným potrebám. Zároveň majú používatelia veľký vplyv na jeho vývoj, pretože ich ohlasy, želania, kritika a chybové hlásenia pomáhajú vývojárom pri ďalšom zdokonaľovaní a šírení príslušného softvérového systému.

Podujatie *Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach* nie je len konferencia v klasickej, akademickom ponímaní, ale najmä neformálne stretnutie ľudí, ktorých spája záujem o otvorený softvér (jeho tvorbu, využitie, komplexnejšie riešenia na základe existujúceho softvéru). Popri recenzovaných príspevkoch, ktoré sú publikované v tomto zborníku, na podujatí boli prezentované ďalšie možnosti využitia otvoreného softvéru a otvorených informačných technológií. Stručné abstrakty týchto ukážok a prezentácií sú uvedené v závere zborníka.

Súčasťou konferencie bol po prvýkrát aj jednodňový odborný seminár pre učiteľov základných a stredných škôl. Po druhýkrát si v programe konferencie našla svoje miesto aj „Cena SOIT a Liberixu“ o najlepšiu bakalársku a diplomovú prácu týkajúcu sa otvoreného softvéru a ďalších otvorených technológií.

Všetky príspevky, uvedené v tomto zborníku, ako aj prezentácie, ktoré na konferencii odznejú, budú voľne prístupné na stránke podujatia, <http://ossconf2011.soit.sk>. Okrem toho, na tejto stránke nájdete aj odkaz na konferenčné médium (obraz DVD), ktoré okrem spomínaných materiálov konferencie obsahuje distribúciu GNU/Linux Ubuntu s ďalším pridaným softvérom, ktorého sa budú týkať príspevky konferencie.

Táto konferencia je jednou z aktivít SOIT na propagáciu otvoreného softvéru. Sme toho názoru, že čím skôr sa používateľ počítača s otvoreným a slobodným softvérom zoznámi, tým lepšie bude vedieť oceniť jeho prednosti, z ktorých bude neskôr profitovať, či už profesionálne alebo v súkromí. Zároveň sa rozširuje jeho všeobecný rozhľad, keď sa odváži nakuknúť za oponu, mimo javiska, ktoré mu prezentuje svet reklamy a obchodných záujmov softvérových gigantov.

Na podporu a propagáciu alternatívneho vyučovania informatiky a ostatných predmetov na základných, stredných a vysokých školách prostredníctvom bezplatného otvoreného softvéru slúži tiež projekt „Slobodný a otvorený softvér pre školy“ s domovskou stránkou <http://sospreskoly.org>. Cieľom projektu je vytvoriť komunitu záujemcov a používateľov otvoreného softvéru na školách všetkých úrovní. Je pokusom o spojenie tých, ktorí vedia ako na to a rovnako aj tých, ktorí si uvedomujú možnosti, ktoré otvorený softvér pre školstvo a spoločnosť poskytuje, ale nevedia ako a kde začať.

Nič sa neurobí samo od seba, takže tento projekt je príležitosťou aj pre vás. Môžeme postupne budovať vzorové učebne s voľne prístupným edukačným softvérom, pracovať na jeho prípadných úpravách a lokalizácii, vytvárať návody a príručky pre správcov učebne a používateľov, aby sme čoraz viac cítili, že v našich snahách nie sme osamotení. Súčasťou snáh projektu *SOS pre školy* je aj presadzovanie (na úrovni vládnej i regionálnych samospráv) otvoreného softvéru v použití pre školské administratívne a ekonomické účely.

Tento zborník, materiály konferencie, aj samotný fakt, že sa toto podujatie mohlo uskutočniť, sú kolektívnym dielom organizátorov, editorov, autorov príspevkov a prezentácií, recenzentov i všetkých účastníkov. Myšlienka pravidelnej samostatnej OSS konferencie sa mohla realizovať len vďaka príkladnej spolupráci a dobre odvedenej práci všetkých zainteresovaných. Ďakujeme im a dúfame, že sa aj v budúcnosti budeme stretávať na podobných podujatiach a že nás bude čoraz viac, aby naše hlasy nezanikli v lomoze každodenného zhonu a agresívnej reklamy softvérových firiem.

Naša vďaka patrí sponzorom, HP Slovensko, Red Hat a Gista, bez ktorých by sa nemohlo uskutočniť vydanie tohto zborníka. Osobitne treba spomenúť sponzorský dar Slovenskej informatickej spoločnosti, z ktorého sme podporili účasť viacerých účastníkov z radov učiteľov formou čiastočného uhradenia ich nákladov spojených s účasťou.

POZVANÍ PREDNÁŠATELIA

Otvorená vláda na Slovensku?!

Abstrakt: Počas ostatnej dekády sa na celom svete presadzovalo prijatie zákonov umožňujúcich občanom prístup k informáciám verejnej správy. Najvyspelejšie demokracie však prišli so zmenou prístupu od pasívnej transparentnosti (na žiadosť občana) k aktívnemu zverejňovaniu dátových katalógov na internete. Tieto vládne projekty našli okamžite veľkú odozvu v komunite aktivistov, ale aj technológov a programátorov. Dáta sa začali využívať na tvorbu najrôznejších webových aplikácií slúžiacich širokej verejnosti. Prednáška prinesie základnú orientáciu v téme, ako aj v celosvetových a zahraničných iniciatívach za otvorenie vládnych administratív verejnosti, proaktívnom zverejňovaní informácií a zavedení štandardov v tejto oblasti. Predstaví občiansky návrh na zmenu filozofie komunikácie verejnej správy s občanmi na Slovensku a bude informovať o dialógu iniciatívy s vládou.



Foto: denník SME

Zuzana Wienk vedie od roku 2002 občianske združenie *Aliancia Fair-play*, ktoré sa dlhodobo venuje odhalovaniu káuz a verejnej kontrole politickej reprezentácie i verejnej správy. Združenie prevádzkuje službu *datanest.sk*, ktorá zhromažďuje údaje o tom, kam prúdia verejné zdroje, podporuje vznik nových aplikácií s využitím verejných dát a koordinuje snahy o presadenie dátových štandardov na Slovensku. Wienk vyštudovala žurnalistiku a pred založením Aliancie pôsobila ako novinárka v printových a elektronických médiách.

X_YLaTeX – krok správným směrem

Abstrakt: Mnoho uživatelů osobních počítačů má zájem vytvořit kvalitní dokumenty, nejčastěji ze své profesní oblasti, tedy odborné texty s řadou speciálních symbolů. Donedávna byla jednou z mála skutečně použitelných variant sazba v systému LaTeX – avšak vzhledem k jeho historické koncepci vznikala řada obtížně řešitelných problémů zejména s fonty a kódováním vstupu. Příspěvek se zabývá možnostmi a některými zkušenostmi při použití systému X_YLaTeX, jehož vlastnosti mohou výrazně přispět k zachování kvalitní sazby jako ve známém a rozšířeném systému LaTeX, ale s mnoha zajímavými rozšířeními: využití fontů instalovaných v rámci operačního systému, možnosti vstupu znaků v kódování UTF-8 a s tím spojené možnosti sazby v různých jazycích, možnosti matematické sazby a použití speciálních symbolů potřebných pro odborné texty.



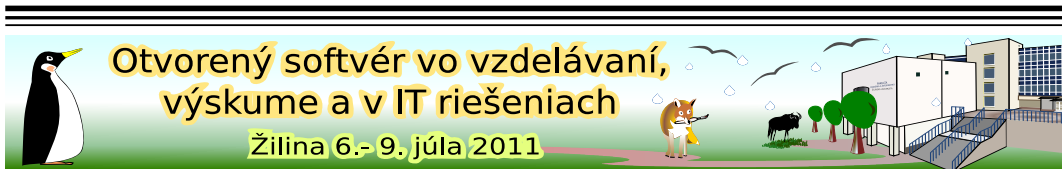
Jiří Rybička se zabývá zpracováním textů na počítači již přes 20 let. Těžištěm jeho zájmu je především sazba publikací systémem \LaTeX . Je členem výboru Československého sdružení uživatelů \TeX u, kde se věnuje zejména začínajícím uživatelům (kniha \LaTeX pro začátečníky, vyšla s podporou tohoto sdružení). Na Mendelově univerzitě se věnuje výuce specializovaného předmětu Zpracování textů počítačem a zabývá se problémy zvyšování formální úrovně závěrečných prací.

Slobodný a otvorený softvér v slovenskom školstve

Abstrakt: Slovenské školstvo prešlo od roku 1989 určité vývojové obdobie v oblasti softvéru na vyučovanie, tvorbu a spracovanie školských informácií. Používal sa proprietárny – licencovaný softvér od rôznych firiem (T602, Microsoft. . .). Tento softvér v začiatkoch podporoval rozvoj a nasadzovanie digitálnych informačných technológií do vzdelávania a bol finančne nenáročný pre školy (bezplatne, za 1 \$). Ďalšou softvérovou možnosťou pre oblasť školských PC a serverov je slobodný – otvorený softvér, ktorý je bezplatný a voľne prístupný pre pedagógov a študentov. V rámci prezentácie , ukážeme jednotlivé moduly slobodného a otvoreného softvéru, ktoré sa používajú v slovenských školách, možnosti získania informácií o realizácii workshopov, vzdelávacích aktivít, akreditovaných vzdelávaní, overovania profesijných kompetencií a využívanie slobodného – otvoreného softvéru na našich školách.



Mgr. Stanislav Slačka vyštudoval MFF UK Bratislava (<http://sslacka.webnode.sk/o-mne/>). Od roku 1987 pracuje v Školskom výpočtovom stredisku Banská Bystrica, v súčasnosti pracovisko Ústavu informácií a prognóz školstva Bratislava. V rámci svojej 24 ročnej školskej praxe participoval na projektoch zavádzania softvérových produktov do vzdelávania (Microsoft Select, Infovek, Slobodný – otvorený softvér a iné). Školské výpočtové stredisko v Banskej Bystrici používa operačný systém Linux od roku 1995. Spravovaním týchto linuxových serverov začínal ako študent strednej školy RNDr. Ján Oravec, v súčasnosti Software Engineer, firmy Facebook Palo Alto, California.



DOI: 10.5300/2011-OSSConf/17

X_YLaTeX – KROK SPRÁVNÝM SMĚREM

RYBIČKA, Jiří (CZ)

Abstrakt. *Mnoho uživatelů osobních počítačů má zájem vytvořit kvalitní dokumenty, nejčastěji ze své profesní oblasti, tedy odborné texty s řadou speciálních symbolů. Donedávna byla jednou z mála skutečně použitelných variant sazba v systému LaTeX – avšak vzhledem k jeho historické koncepci vznikala řada obtížně řešitelných problémů zejména s fonty a kódováním vstupu. Příspěvek se zabývá možnostmi a některými zkušenostmi při použití systému X_YLaTeX, jehož vlastnosti mohou výrazně přispět k zachování kvalitní sazby jako ve známém a rozšířeném systému LaTeX, ale s mnoha zajímavými rozšířeními: využití fontů instalovaných v rámci operačního systému, možnosti vstupu znaků v kódování UTF-8 a s tím spojené možnosti sazby v různých jazycích, možnosti matematické sazby a použití speciálních symbolů potřebných pro odborné texty.*

Klíčová slova. *TeX, X_YTeX, LaTeX, X_YLaTeX, kompatibilita, fonty dokumentu, kódování vstupu, UTF-8, vícejazyčná sazba.*

X_YLaTeX – STEP IN THE RIGHT DIRECTION

Abstract. *Many PC users interested in creating quality documents, most of their profession, i.e. technical texts with number of special symbols. One of the few really usable possibilities is LaTeX – but due to its historical concept there are difficult problems solvable in particular, e.g. the use of various fonts and input encoding. The paper deals with possibilities and some experience in using the system X_YLaTeX whose properties can preserve present quality of LaTeX documents but with many interesting extensions: use of fonts installed in the operating system, the possibility of entering characters in the UTF-8 encoding and with the associated possibility of different languages characters, mathematical and special symbols needed for technical texts.*

Key words and phrases. *TeX, X_YTeX, LaTeX, X_YLaTeX, compatibility, document fonts, input encoding, UTF-8, multilingual typesetting.*

Úvod

Již dlouho slouží zejména pro tvorbu odborných textů mnoha uživatelům systém \LaTeX . S tím je spojena řada specifik. Principy systému definované na začátku 80. let minulého století jsou na jedné straně geniální a v mnoha aspektech trvale použitelné, na druhé straně však s sebou nutně nesou nepřijemné „drobnosti“. Například jednou z nejznámějších vlastností je silná snaha o zpětnou kompatibilitu. Ta ovšem dost často vede k řešení, které je těžkopádné, pro řadu uživatelů nepohodlné a mnohdy i na první pohled nepochopitelné.

I když je například zmíněná snaha o zpětnou kompatibilitu nesporně velmi kladným prvkem, vždy je pravděpodobně potřeba najít vhodný kompromis a rozhodnout se pro optimální řešení.

Systémy postavené na principu \TeX se nacházejí v silné konkurenci jiných přístupů, které mají daleko větší rozšíření. Cílem tohoto článku není zkoumat, proč tomu tak je, příčiny totiž nejsou podstatné. Daleko důležitější je, že konkurenční systémy víceméně určují směry vývoje a uživatelé se tomu přizpůsobují, ať se jim to líbí nebo ne. Opět nebudeme hodnotit, zda je to dobře, do jaké míry je to otázka reklamy, byznysu nebo snad technických prvků.

Postavení systémů, které jsou nyní středem našeho zájmu, tedy systémů volně dostupných a založených na principu \TeX , je zcela jistě nelehké. Je-li naším cílem zvýšit počet uživatelů, kteří je budou *skutečně chtít* využívat, musíme se nutně zamýšlet nad tím, jaké konkurenční výhody tyto systémy vůbec mohou přinášet, dále nad tím, jak tuto skutečnost uživatelům sdělit, a v konečné fázi také nad tím, jak tuto snahu podpořit.

Tento článek by měl být příspěvkem k prvnímu z uvedených prvků – všimneme si možností, které mohou přispět ke zvýšení konkurenceschopnosti systémů postavených na principu \TeX , konkrétně v podání systému X_{\LaTeX} .

Proč X_{\LaTeX} ?

Všeobecně známou silnou stránkou všech systémů postavených na principu \TeX je velká přizpůsobitelnost. Existence nástrojů, jako jsou definice nebo redefinice příkazů a dokonce schopnost změnit vnímání vstupního lexika, poskytuje nepřehledné možnosti vytvářet specifické prostředky pro řešení všemožných potřeb uživatelů. Pomineme-li v tuto chvíli skutečnost, že některá taková rozšíření jsou velmi netriviální a vyžadují skutečně hluboké znalosti mnoha komplikovaných vnitřních vlastností a algoritmů, vzniká tím také do značné míry velmi rozsáhlá a málo přehledná množina realizací, jejíž použitelnost je v některých případech opravdu diskutabilní. Jmenujme například sazbu tabulek – existuje přes 30 balíčků, které se nějak dotýkají nejrůznějších tabulkových prvků. Nejenže ani těch 30 balíčků neřeší všechno, ale některé nelze použít s jinými. Mezi uživateli pak lze vysledovat celou škálu přístupů – od těch, kteří říkají „na všechno už existuje nějaký balíček \LaTeX , jen je potřeba jej najít“, až po ty, kteří zastávají názor „nikdy nepoužívám cizí makra“.

Řekneme-li tedy X_{\LaTeX} nebo X_{\LaTeX} , běžný uživatel pravděpodobně již předem pojme podezření, že se jedná „zas o něco jako další balíček“ nebo specifickou nadstavbu, jejíž použitelnost v běžné praxi českého (slovenského) autora a sazeče nebude nijak

významná, nebo (a to je ještě nepříjemnější) bude nějak měnit již zaběhnuté, ověřené a používané postupy.

Lze však říci, že jako příspěvek ke konkurenceschopnosti systémů tohoto typu představuje X_YL^AT_EX *krok správným směrem*. Autor Jonathan Kew jasně deklaroval jeho základní cíle: vyřešit především dva velké nedostatky dosavadních řešení – umožnit snadné použití všemožných fontů a zpracovávat zdrojový text v kódování UTF-8, a to vše tak, aby to bylo pro uživatele co nejjednodušší.

Zdrojový text

Deklarovaná jednoduchost práce s X_YL^AT_EXem je vyjádřena například i tím, že pro jeho použití při zpracování již existujících textů, u nichž chceme dostat stejný výsledek, není potřeba téměř nic měnit. Stačí připojit balíček `xltxtra` zahrnující v sobě volání dalších potřebných balíčků: zejména `fontspec` a `xunicode`.

Příklad minimálního zdrojového textu:

```
\documentclass{article}
\usepackage[czech]{babel}
\usepackage{xltxtra}
\begin{document}
Příliš žlutoučký kůň úpěl ďábelské ódy.
\end{document}
```

Po překladu překladačem `xelatex` dostaneme výsledné PDF, v němž bude automaticky zvoleno písmo Latin Modern.

Pokusme se přiblížit alespoň některé směry, kterými kráčí X_YL^AT_EX/X_YL^AT_EX.

Směr – fonty

Kdysi bývalo zvykem, že jakýkoliv dokument, který vzešel z T_EXové nebo L^AT_EXové dílny, se pyšnil použitým Knuthovým písmem Computer Modern (byť ve verzi CS nebo ještě nověji LM). K této pýše je „bezsporu“ dobrý důvod: toto písmo je v mnoha aspektech dokonalé a připočteme-li k tomu ještě dokonalé algoritmy sazby, výsledek (zejména v odborných textech s matematickými výrazy) byl skutečně téměř nedostižný (a je i dnes!).

Computer Modern však není jediné použitelné písmo a navíc systémem L^AT_EX nesázíme vždy jen nějakou matematickou perlu. Kvalitní typografický návrh dokumentu tedy počítá s tím, že zvolíme *odpovídající* písmo. Co můžeme vybírat v konkurenčních systémech? Při rozbalení nějaké jejich nabídky s písmy se objeví řádově desítky až stovky položek. Ano, je jistě potřebné ihned dodat, že mnoho z nich představuje nepoužitelný plevel, ale i přesto je samotný proces volby písma velmi jednoduchý a dojde-li k tomu, že pečlivý, znalý a zkušený uživatel si nějaké písmo obstará, do seznamu nabízených písem je dostane prostou kopií do vhodného adresáře.

Kdo z běžných uživatelů někdy ovšem pojal myšlenku, že potřebuje nějaké pěkné písmo do svého dokumentu v L^AT_EXu, obstaral si je a chtěl je použít, pak těžce narazil: nezbývalo než se ponořit do nepřehledné bažiny písmových formátů, národních specifik, různých transformací, konfigurací, adresářových cest a systémových nastavení. Je

zřejmé, že bez hlubokých znalostí, navíc v každém typu distribuce zcela odlišných a závislých na použitém operačním systému, nebylo možné písmo použít. A takové nároky zcela diskvalifikovaly jinak nedostižně kvalitní a do značné míry geniální \TeX ové systémy v očích „běžných“ uživatelů, kteří nemají ambice stát se odborníkem na počítačová písmena, ale „pouze“ chtějí vytvořit třeba pěknou pozvánku na promoci. . .

V tomto kontextu je tedy systém umožňující použít *libovolný* font instalovaný v operačním systému doslova zázrakem. Základem je možnost systému \XTeX , kdy lze pomocí primitivu `\font` připojit libovolný dostupný font zadáním jeho jména. Například zápis

```
\font\ndapis="Minion Pro Bold" at 16pt
\fbbox{\ndapis Příliš náročná firemní výroba větrných elektráren}
```

jednoduše umožňuje vysázet:

Příliš náročná firemní výroba větrných elektráren

Připomeňme, že font Adobe Minion Pro Bold je ve formátu Open Type, na disku se jmenuje MinionPro-Bold.otf a je nakopírován v adresáři spolu s ostatními systémovými fonty. Tato rodina je také součástí volně dostupného Adobe Readeru. Všimněte si, že je automaticky použita ligatura „fi“ a nejsou problémy s národními znaky.

To ale není zdaleka všechno. \TeX ový komfort volby písma, který umožňuje ovládat celý dokument symbolickými příkazy bez detailní specifikace konkrétního písma, je uživateli dostupný pomocí balíčku `fontspec`, jehož autorem je Will Robertson. Nejenže lze nastavit písmovou rodinu, ale také ovládat řezy a stupně standardními \TeX ovými příkazy a ve spolupráci s dalšími balíčky využívat dosavadní způsoby sazby speciálních znaků. Již vytvořené zdrojové texty lze tedy znovu použít.

V přehledu uvedeme základní možnosti balíčku `fontspec`:

`\fontspec` – jedná se o základní příkaz pro výběr fontu. V povinném parametru je potřebné uvést jméno požadovaného fontu, v nepovinném parametru pak lze uvést řadu voleb sloužících k upřesnění výběru a chování fontu. Například zápis `{\fontspec[Scale=0.75]{Lucida Console}Toto je strojopis.}` nastaví strojopisný font Lucida Console a vysází příslušný text v dané skupině: Toto je strojopis.

Po nastavení fontu je možné používat běžné příkazy pro změny řezu, a pokud příslušný řez existuje, je automaticky zapnut. Například v písmu Georgia: Základní řez – *kurzíva* – **tučné** – **tučná kurzíva**.

Volba `Scale` umožňuje změnit originální velikost s daným koeficientem, což například u strojopisného fontu vede k vyrovnanějšímu obrazu s okolním textem.

Kromě konkrétního koeficientu (jak bylo v příkladu použito 0,75) lze použít i inteligentní hodnoty `MatchLowercase` a `MatchUppercase`, které nastavují velikost rovnu minuskové, resp. verzálkové velikosti okolního fontu.

Jednou z výhod písem formátu OTF je řada typografických vymožeností, například variantní číslice. Verzálkové číslice vhodné do tabulek (mají stejné šířky kvůli správnému zarovnání) a minuskové číslice vhodné do běžného textu mohou být zařazeny volbou `Numbers=Uppercase`, resp. `Numbers=OldStyle`.

Například ve fontu Cambria: verzálkové číslice do tabulek 1234567890 a minuskové číslice do textu 1234567890.

Z dalších možných voleb zmíníme `Mapping=tex-text` (umožňuje používat běžné příkazy pro speciální znaky, například dva spojovníky pro pomlčku apod.), `Color` (definuje barvu, zadává se jako trojice hexadecimálních hodnot definujících složky RGB a volitelně čtvrté číslo jako průhlednost), `LetterSpace` (prostrkání, hodnotou je procento stupně písma použité pro prostrkávací mezery), `FakeSlant`, `FakeStretch` a `FakeBold` (mechanické změny tvaru; pomůcka pro modifikaci sklonu, zúžení řezu nebo změnu ductu – tyto změny jsou realizovány jen počítačovou transformací, proto je potřeba jich využívat střízlivě a v odůvodněných případech).

`\setmainfont` – nastavení základního (antikvového) písma dokumentu.
Příkaz má stejnou syntax jako `\fontspec`.

`\setsansfont` – nastavení fontu, který v dokumentu představuje bezserifovou rodinu.
Syntax opět stejná jako `\fontspec`.

`\setmonofont` – nastavení fontu, který v dokumentu představuje strojopisnou rodinu.
Syntax je stejná jako `\fontspec`.

Uvedenými třemi příkazy lze definovat celkové podání dokumentu. Pro změnu rodiny, řezu a stupně fungují běžné \TeX ové nástroje, nastavená písma fungují i v příkazech pro nadpisy atd. Pokud nejsou žádné fonty zavedeny, automaticky se zvolí rodina Latin Modern, reprezentující tradiční pojetí – sazbu dokumentu písmem odvozeným z Knuthova originálu.

`\defaultfontfeatures` – příkaz umožňující nastavit volby, které se pak použijí v každém dalším příkazu nastavujícím konkrétní font. Užitečné je například nastavit `\defaultfontfeatures{Mapping=tex-text}` a tuto volbu pak není nutné uvádět u žádného dalšího příkazu `\fontspec`.

Jak již bylo zmíněno, popsaný balíček `fontspec` je volán z balíčku `xltxtra`. Ten zabezpečuje správnou funkci několika prvků, které jsou přechodem na univerzální kódování vstupu a libovolné fonty porušeny (textové indexy a exponenty, body volitelného dělení, prostředí `verbatim*` a příkaz `\verb*`) a také definuje loga \TeX a další.

Směr – kódování vstupu

V současné době je i v textech primárně psaných v jednom jazyce často potřeba uvádět slova (například vlastní jména) s jiným pravopisem a jiným repertoárem znaků. V tomto směru měly systémy postavené na principu \TeX u vždy rozsáhlé možnosti, všechny zápisy rozličných národních a jiných znaků však byly v původní podobě zapisovány sedmibitovými znaky tabulky ASCII (univerzální způsob dosažitelný na jakémkoliv zařízení). To s sebou samozřejmě nese i určitou těžkopádnost, takže například běžný český text takto nezapisujeme a přirozeně předpokládáme, že národní znaky přímo vidíme v editoru.

Tento způsob práce se s rozvojem konkurenčního programového vybavení stal již standardem. Pro běžného uživatele takových systémů je těžko představitelné, že by například v českém textu zapsal jméno François Didôt jako `Fran\c{c}ois Did\^ot`,

případně jméno azbukou nebo dalšími exotičtějšími jazyky ještě podstatně složitěji. Vyžaduje, aby přímo v pořizovacím programu byly všechny znaky vidět. Zapisuje tedy přímo z klávesnice **François Didôt** nebo **Александр Пушкин** apod. Tuto funkčnost zajišťuje univerzální kódování národních znaků, jakým je UTF-8. Na univerzální kódování přechází většina programů, bylo by tedy velmi svazující, kdyby se T_EXové systémy do této množiny nezařadily.

Důležitost univerzálního kódování vstupu je podtržena ještě i skutečností, že nejde jen o nějaké víceméně exotické národní znaky, které používá jen zlomek uživatelů, ale i o nejběžnější typografické rekvizity potřebné v prakticky jakémkoliv textu.

Porovnejme si alespoň některé možnosti, které byly doposud k dispozici, s možnostmi systému X_YL_AT_EX se vstupem v kódování UTF-8.

Uvozovky – Ty byly dostupné vždy pomocí nějakého příkazu, nejčastěji `\uv{...}`, případně `\clqq` a `\crqq`. Tyto dva příkazy jsou definovány buď ve specifickém stylu **czech**, nebo v české podpoře vložené do stylu **babel**. Úhlové uvozovky byly poněkud problematictější, byly vkládány jako konkrétní znak daného fontu. Díky kódování UTF-8 se můžeme vyhnout jakémukoliv příkazu a přímo zapisovat „**slovo**“, nebo také »**slovo**«. Nevýhodou může být skutečnost, že používaný editor neumí tyto znaky efektivně vkládat přímo z klávesnice, to je však většinou možné snadno překonat. Vhodných editorů, které plně respektují celou znakovou sadu UTF-8, je dostatečné množství.

Pomlčky – Tento znak je pravděpodobně jediný, jehož doposud užívaný zápis je natolik efektivní, že jej není potřeba příliš měnit. Zápis dvou nebo tří spojovníků za sebou vede na ligaturu představující půlčtverčikovou nebo čtverčikovou pomlčku. Tohoto efektu je dosaženo i v novém pojetí (viz **fontspec**), pokud font zavedeme s volbou **Mapping=tex-text**. Kromě toho lze ovšem použít i vložení příslušného znaku s kódem 2013(hex) nebo s kódem 2014(hex). To je výhodné zejména tehdy, pokud editor sám například promění spojovník zapsaný s mezerami za pomlčku.

Měny – Jeden z nejčastějších dotazů poslední doby je „jak prosím zapíšu znak euro?“ a dost trapnou odpovědí bylo „víte, to musíte připojit balíček **eurosym** a pak zapisovat `\euro`“. Uživatel, který klade takový dotaz, je spíše začátečník a předpokládá, když se tento znak jednoduše zapisuje všude jinde, stejně jednoduše to půjde i v tak dokonalých systémech, jaké jsou ty T_EXové. Tento předpoklad je splněn až nyní. Opět jde o zlepšení výsledku, protože příslušný znak je použit z daného fontu, nikoliv z náhradního fontu používaného balíčkem **eurosym**. Příklad: 15 €. Podobně lze využít znaku pro britskou libru a japonský jen: 15 £ a 15 000 ¥.

Násobení – Znak byl dostupný dosud jen jako matematický symbol `\$times$`. Nevýhodou tohoto řešení je zejména skutečnost, že kresba matematického symbolu nemusela dobře korespondovat s kresbou použitého textového fontu. Vezmeme-li v úvahu skutečnost, že se jedná více o textovou než matematickou rekvizitu, je možnost vložení přímého znaku určitě přínosná. Porovnejme matematický symbol: 5× nebo přímý znak daného fontu: 5×

Vybrané matematické symboly – Je možné, že by v matematické sazbě existovalo něco, co by systémy založené na principu T_EXu neuměly nejlépe ze všech? Jak už bylo

vedeno u znaku pro násobení, nejde většinou přímo o matematické výrazy, ale spíše prvky, které se matematickým prostředím vždy řešily, ale nyní to už není nutné a navíc získáváme určité zlepšení. Jedním z takových symbolů je π – ve smyslu konstanty. Tu je podle typografických pravidel potřebné sázet vzpřímeným řezem, ale matematický symbol `\pi` dává kurzívní variantu π . Obdobně symbol μ používaný jako předpona mikro-. Z dalších užitečných znaků jmenujme symbol negace \neg , ochranná známka ® , stupeň $^\circ$, plus minus \pm , exponenty $^1, ^2, ^3$, paragraf \S .

Směr – za hranice českých a slovenských specifik

Již od začátku jsou systémy na principu TeXu vybaveny určitými možnostmi národních specifik (uvedme například široký repertoár akcentů nebo příkaz `\discretionary`), ale přece jen to bylo považováno za nedostačující a pro dosažení kvalitních a prakticky použitelných výsledků byla vždy doplňována rozsáhlá podpora specifická pro české a slovenské uživatele.

Rozhodně nelze říct, že by na tom bylo něco špatného – rozšíření pro kvalitní českou a slovenskou sazbu je velmi propracované a v určitém směru předčilo svými kvalitami původní možnosti, dokonce lze říci, že vůbec umožnilo běžné použití TeXového systému obyčejnými uživateli.

V tomto přístupu nebyla komunita českých a slovenských uživatelů TeXu výjimkou. Podobnými rozšířeními a doplňky byly například vybavovány i instalace pro sazbu polských i německých textů. To s sebou automaticky přinášelo určitou povinnost tyto instalace udržovat a distribuovat, současně to přinášelo jisté obtíže při přenosech zdrojových textů mezi různými takto upravenými instalacemi.

Ukazuje se, že údržba a další rozvoj specifických instalací není nejlepší volbou – možnost využít univerzálního řešení, které je podporováno mezinárodně a u kterého je větší naděje na kontinuitu a další vývoj, je pro národní specifika do budoucna pravděpodobně jedinou schůdnou cestou.

Využití X_YLaTeXu může takovou cestu představovat. Již zmíněné možnosti vstupu zdrojového textu v kódování UTF-8 a možnosti využití nejrůznějších fontů obsahujících národní znaky nejen české a slovenské abecedy, ale i cyrilice apod. jsou toho jistým dokladem. Krokem k univerzálnímu řešení národních specifik sazby je balíček `babel`, který je již důstojnou alternativou specifických národních rozšíření. Kromě toho však lze využít balíček `polyglossia`, o němž se krátce zmíníme.

Balíček `polyglossia`, jehož autorem je François Charette, má v manuálu podtitul „náhrada Babelu v X_YLaTeXu“. Nejedná se však jen o náhradu, ale o nové možnosti využívající plně existujících rozšíření. Balíček zavede pro všechny jazyky použité v dokumentu odpovídající vzory dělení, umožňuje nastavit font při přepnutí jazyka (využívá při tom balíček `fontspec`), nastaví určité typografické konvence (mezerování kolem interpunkce apod.), předefinuje standardní řetězce „kapitola“ „obrázek“ atd., přizpůsobí formátování kalendářního data, případně nastaví i správný směr sazby (využívá balíček `bidi`).

Zavedení balíčku lze provést alespoň dvěma způsoby. První způsob je podobný způsobu balíčku `babel`, tedy `\usepackage[lang1, lang2, ...]{polyglossia}`, ale oproti balíčku `babel` je jazyk uvedený v nepovinném parametru jako *první* brán jako jazyk implicitní.

Druhou možností je zavést balíček `polyglossia` bez volitelného parametru a jednotlivé jazyky aktivovat příkazem `\setdefaultlanguage[volby]{jazyk}` pro implicitní jazyk a příkazem `\setotherlanguage[volby]{jazyk}` pro alternativní jazyky použité v dokumentu.

Přepínání mezi jazyky lze provést pro krátké texty příkazem `\textjazyk{text}`.

Například příkaz `\textrussian{\today}` dává po vysázení vysázení 30 мая 2011 r. nebo `\textgreek{\today}` zase 30 Μαΐου 2011.

Pro delší texty v daném jazyce je k dispozici prostředí s názvem jazyka, například: `\begin{slovak}...\end{slovak}`.

Volitelné parametry u aktivovaných jazyků umožňují různá nastavení a jsou k dispozici pro každý jazyk zvlášť. Jejich podrobný seznam je uveden v manuálu. Protože sazba exotických jazyků není hlavním cílem většiny našich uživatelů, nebudeme se rozsáhlým možностям balíčku `polyglossia` v těchto oblastech věnovat.

Závěr

Systém $X_{\text{q}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ představuje bezesporu nástroj, který poskytuje uživatelům $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ významná ulehčení při realizaci typografického návrhu dokumentů, rozšíření možností a řešení drobných, ale nepříjemných problémů.

Podobně jako výstupní formát PDF znamenal významné přiblížení $\text{T}_{\text{E}}\text{X}$ ových nástrojů světu konkurenčních systémů počítačové sazby, je i $X_{\text{q}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ systémem, jež posunuje vývoj podobným směrem.

Z praktických zkušeností vyplývá, že s novými přístupy jsou spojeny i dílčí nedostatky – ne všechno funguje, jak je popisováno v manuálech, ne všechno je zcela transparentní a zcela jistě se musíme smířit se změnami, které nezaručují stoprocentní zpětnou kompatibilitu.

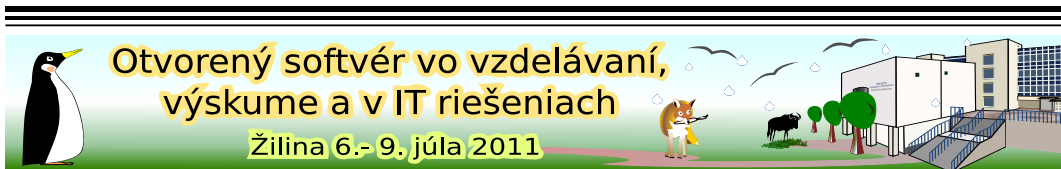
Seznámení s možnostmi $X_{\text{q}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u může uživateli přinést významný efekt. Pokud k tomu přispěje i tento článek, byl jeho cíl bezezbytku splněn.

Kontaktní adresa

Doc. Ing. Jiří Rybička, Dr.,

Ústav informatiky Provozně ekonomické fakulty Mendelovy univerzity v Brně,

Zemědělská 1, 613 00 Brno, e-mail: rybička@mendelu.cz



DOI: 10.5300/2011-OSSConf/25

SLOBODNÝ A OTVORENÝ SOFTVÉR V SLOVENSKOM ŠKOLSTVE

SLAČKA, Stanislav, (SK)

Abstrakt. Slovenské školstvo prešlo od roku 1989 určité vývojové obdobie v oblasti softvéru na vyučovanie, tvorbu a spracovanie školských informácií. Používal sa proprietárny – licencovaný softvér od rôznych firiem (T602, Microsoft . . .). Tento softvér v začiatkoch podporoval rozvoj a nasadzovanie digitálnych informačných technológií do vzdelávania a bol finančne nenáročný pre školy (bezplatne, prípadne za 1 \$). Ďalšou softvérovou možnosťou pre oblasť školských PC a serverov je slobodný – otvorený softvér, ktorý je bezplatný a voľne prístupný pre pedagógov a študentov. V rámci prezentácie ukážeme jednotlivé moduly slobodného a otvoreného softvéru, ktoré sa používajú v slovenských školách, možnosti získania informácií o realizácii workshopov, vzdelávacích aktivít, akreditovaných vzdelávaní, overovania profesijných kompetencií a využívanie slobodného – otvoreného softvéru na našich školách.

Kľúčové slová. Slobodný a otvorený softvér, školstvo.

1 Úvod

Slovenské školstvo prešlo za posledných 20 rokov určité vývojové obdobie v oblasti používania počítačov a informačno-komunikačných technológií. Niektorí si pamätáme obdobie používania 8 bitových počítačov typu PMD, PP-01, ktoré využívali na prácu jednoduchý Basic. Následne v školstve sa objavili na krátku dobu PP-06, ktoré znamenali nástup počítačov typu PC. Začiatkom 90-tych rokov do školstva nastupujú PC-AT a používanie operačného systému MS DOS. Tieto počítače pracovali samostatne a vznikla potreba vytvoriť sieťové prepojenia. V tejto oblasti sa objavuje v roku 1994 operačný systém Linux, ktorý nahrádza finančne náročný UNIX. Bol vytvorený ako slobodný softvér, vývojový model open source. Jedna z prvých najstarších spravovaných verzií Linuxu bola distribúcia Slackware.

V slovenskom regionálnom školstve, v roku 1995, Školské výpočtové stredisko v Banskej Bystrici ako jedno z prvých na Slovensku inštaluje Linux na svojich serveroch a využíva tento operačný systém na zabezpečenie pripojenia do Internetu, správu elektronickej pošty, nastupujúcu službu hypertextových informačných zdrojov a vyhľadávaní. S nástupom Infoveku v roku 1998 sa otvára pre slovenské základné a stredné školy brána Internetu a školy

začínajú využívať informačno-komunikačné technológie (IKT) vo vzdelávacom procese. Používanie IKT prináša so sebou aj nároky na užívateľský softvér, pripojenie škôl do Internetu. Infovek pokračuje v distribúcii softvéru na školy, ktoré začali realizovať Školské výpočtové strediská formou Selectovej zmluvy. Školy získavajú najmä kancelársky balík na prácu s textom, tabuľkami a prezentáciami.

V tomto období sa aj v oblasti užívateľského softvéru v regionálnom školstve predstavuje slobodný a otvorený softvér. Ako jedným z prvých je program Audacity na nahrávanie a spracovanie zvuku. Fakulta prírodných vied Univerzity Pavla Jozefa Šafárika v Košiciach pod vedením Doc. Ľubomíra Šnajdera realizuje prípravu učiteľov základných a stredných škôl na využívanie IKT vo vyučovaní v module nahrávanie a spracovanie zvuku, s podporou Audacity.

V oblasti spracovania obrazu bol na školy distribuovaný licencovaný softvér Zoner Callisto. Tento softvér bol v niektorých školách nahradený Open source softvérom GIMP, alebo programom Inkscape. Základné a stredné školy používajú vo veľkom počte na vyučovacom procese kancelársky balík Office. Za pomoci kancelárskeho balíka pripravujú svojich žiakov na prácu s textom, tabuľkami, učia ich vytvoriť prezentácie a na obchodných akadémiách aj prácu s databázami. Ak chceme naučiť žiakov prácu s kancelárskym balíkom, je potrebné, aby si žiak mohol aj v domácom prostredí odskúšať používanie, respektíve vyriešiť úlohu, alebo sa pripraviť na hodinu, preto niektoré školy prechádzajú na používanie kancelárskeho balíka Open Office. Tento programový balík obsahuje textový editor, tabuľkový procesor, prezentačný softvér, databázu, spracovanie obrázkov. Open Office prešiel určitým vývoje- vým obdobím. V súčasnosti sa používa vo veľkej miere verzia 3.2x.

2 Anketa o využívaní slobodného a otvoreného softvéru

V mesiaci máj v dňoch od 23.5 do 27.5.2011 bola realizovaná anonymná elektronická anketa o využívaní slobodného a otvoreného softvéru v slovenskom školstve. Tejto anonymnej ankety sa zúčastnilo 138 administrátorov a správcov počítačových učební základných a stredných škôl Slovenska. Na otázky odpovedali anonymne, výberom z 3 možností. Vytvorenie ankety, zber údajov a spracovanie výsledkov bolo realizované za podpory formulárového prostredia Google dokumentov.

Pôsobisko respondentov

	Počet	Percent
na základnej škole	63	48%
na strednej škole	65	50%
v školskom zariadení	3	2%

Tabuľka prezentuje zloženie účastníkov anonymnej ankety. Úvodom je potrebné si položiť otázku. Bola táto vzorka dostatočná, reprezentatívna na realizovanie anonymnej elektronickej ankety z ktorej je možné robiť nejaké analýzy – závery? Ponuku na aktívne zapojenie sa

do anonymnej elektronickej ankety dostali všetci administrátori – správcovia počítačových učební na Slovensku, ktorí sú členmi diskusnej skupiny Infoveku. Anketa sa uskutočnila s aktívnym využitím elektronickej diskusnej skupiny správcov počítačových učební základných a stredných škôl Slovenska.

Počuli ste už o slobodnom a otvorenom softvéri?

Odpoveď	Počet	Percent
Áno	122	93%
Nie	6	5%
Neviem sa vyjadriť	2	2,00%

Na túto otázku odpovedalo 93 percent áno. To znamená, že administrátori počítačových učební v slovenskom regionálnom školstve majú informácie o slobodnom a otvorenom softvéri.

V nasledujúcej otázke sme zisťovali, používanie slobodného a otvoreného softvéru na základných a stredných školách.

Prezentované výsledky odpovedí v tabuľke sú veľmi zaujímavé. 82 percent základných a stredných škôl používa slobodný a otvorený softvér vo svojej škole. Z tabuľky vidíme, že 17 účastníkov ankety čo je 13 percent nepoužíva slobodný a otvorený softvér vo svojej škole. Anketa bola nastavená tak, aby títo účastníci nepokračovali v ankete, pretože odpovedať na zostávajúce otázky by bolo pre nich zbytočné. Možno v budúcnosti, ak sa bude realizovať podobná anonymná elektronickej anketa, by bolo dobré zistiť používanie vo vzdelávacom procese a používanie v riadiacom procese. Ďalej si dovoľím prezentovať výsledky podľa využívania slobodného a otvoreného softvéru vo vzdelávacom procese, spracovaní školskej agendy.

Používate slobodný a otvorený softvér vo vašej škole, školskom zariadení?

Odpoveď	Počet	Percent
Áno	108	82%
Nie	17	13%
Neviem sa vyjadriť	6	5%

Najviac základné a stredné školy z oblasti slobodného a otvoreného softvéru používajú program Audacity. Z tabuľky vidíme, že viac ako 56 percent škôl používa tento program. Rozšírenosť Audacity je priamo úmerná dĺžke používania, vzdelávania učiteľov na prácu s programom. V rámci anonymnej elektronickej ankety nebol robený prieskum používania slobodného a otvoreného softvéru v oblasti spracovania videa.

Používate na vyučovanie, spracovanie zvuku program Audacity?

Odpoveď	Počet	Percent
Áno	73	56%
Nie	37	28%
Neviem sa vyjadriť	3	2%

Na druhom mieste je kancelársky balík Open Office, ktorý využíva 50 percent účastníkov anonymnej ankety, vo vyučovacom procese a na spracovanie školskej agendy.

Sumárny výsledok je zaujímavý, pretože kancelársky balík Open Office obsahuje minimálne 5 modulov, textový editor, práca s tabuľkami, tvorba prezentácií, databáza, grafický editor, ktoré sa vyučujú na základných a stredných školách a aplikácie sú vo veľkej miere rovnaké ako licencovaný MS Office. Napriek tomu skončili až na 2. mieste v ankete.

Používate Open office ako kancelársky balík na vyučovanie, spracovanie školskej agendy?

Odpoveď	Počet	Percent
Áno	65	50%
Nie	47	36%
Neviem sa vyjadriť	2	1%

V prvej trojke sa umiestnil produkt Hot potatoes – horúce zemiačky.

Tento slobodný softvér používa viac ako 30 percent škôl, ktoré sa zapojili do anonymnej ankety. Softvér sa používa na tvorbu interaktívnych testovacích modulov. Základné informácie sa nachádzajú na <http://hotpot.uvic.ca/> Hot potatoes je jeden z novších softvérov, ktorý sa objavil v ponuke pre školstvo. Softvér veľmi rýchlo získava svoje miesto v slovenskom regionálnom školstve.

Používate programový balík Hot Potatoes- (horúce zemiačky) na tvorbu interaktívnych vzdelávacích materiálov?

Odpoveď	Počet	Percent
Áno	41	31%
Nie	69	53%
Neviem sa vyjadriť	3	2%

(http://www.pastelka.sk/deti_horucezemiacki.html, <http://www.edu2.wbl.sk/Vystupy-z-Hot-Potatoes.html>).

Nasleduje otvorený elearningový portál Moodle s takmer 29 percentami.

Elearningová platforma, alebo digitálny vzdelávací obsah je finančne nenáročná a dáva školám možnosť komplexného využitia digitálnych technológií vo vyučovacom procese.

Využívajú sa tu prostriedky a nástroje, ktoré žiaci používajú v bežnom živote pri vzájomnej komunikácii (email, chat, diskusná skupina).

Používate elearning Moodle?

Odpoveď	Počet	Percent
Áno	38	29%
Nie	74	56%
Neviem sa vyjadriť	2	2%

Záverečná tabuľka z anonymného elektronického prieskumu prezentuje používanie operačného systému Linux v školskom prostredí. 23 percent zástupcov škôl v anonymnej elektronickej ankete uviedli, využívanie operačného systému Linux.

Tento výsledok anonymnej elektronickej ankety je veľmi zaujímavý a motivuje k určitým analýzám, odporúčaniam.

Využívate ako operačný systém na školských počítačoch Linux?

Odpoveď	Počet	Percent
Áno	30	23%
Nie	83	63%
Neviem sa vyjadriť	1	1%

Účastníci elektronickej ankety v záverečnej otázke doplnili ešte zoznam otvoreného softvéru, ktorý je im známy, prípadne ho používajú. Najviac účastníkov uviedlo internetový prehliadač Firefox, nasledovali grafické editory Gimp, Inkscape, program na prácu s elektronicou poštu Thunderbird. Ďalej to bol program pre deti Tuxpaint, programovací jazyk Free pascal, ktorý sa stal programovacím jazykom v súťaži stredoškôľakov ZENIT programovanie. Ďalšími často používanými boli IrfanView, Joomla a iné. Môžeme konštatovať, že v našom regionálnom školstve je odpovedajúca informovanosť o slobodnom a otvorenom softvéri. V základných, stredných školách, prípadne v školských zariadeniach sa používa tento softvér aj vo vzdelávacom procese.

3 Záver

Dôležitým faktorom pri využívaní slobodného a otvoreného softvéru v regionálnom školstve je informovanosť o jednotlivých produktoch, možnosť odskúšať si ich, ponuka odpovedajúcich školení a vzdelávacích aktivít. V tejto oblasti Školské výpočtové stredisko v Banskej Bystrici distribuovalo na všetky stredné školy banskobystrického kraja za podpory SOIT live DVD EDUubuntu. Školy si mohli tento systém nainštalovať do školských počítačov a otestovať si možnosti otvoreného softvéru.

Školské výpočtové stredisko v Banskej Bystrici realizovalo v spolupráci so Štátnym Inštitútom odborného vzdelávania v roku 2006 aktivity, kde učiteľom bol predstavený program na spracovanie zvuku Audacity. Následne si odskúšali úpravu, spracovanie zvuku s podporou Audacity. Vzdelávanie absolvovali učitelia zo stredných odborných škôl celého Slovenska.

Veľmi prospešnou aktivitou v oblasti prezentácie slobodného a otvoreného softvéru je projekt „Ďalšie vzdelávanie učiteľov informatiky“. V rámci tohto vzdelávania sa učitelia zoznámili s otvoreným softvérom, prakticky si odskúšali jednotlivé moduly a prezentovali formou elektronického bloku svoje skúsenosti a názory. Prikladám linky na niektoré odkazy:

<http://miroslavvojtek.blog.sme.sk/c/245305/0-slobodnom-softveri-na-hodinach-informatiky.html>

<http://ludmila-dvui.blogspot.com/2010/01/slobodny-otvoreny-softver-3dg1-zadanie.html>

V súčasnej dobe Školské výpočtové stredisko v Banskej Bystrici ponúka v rámci vzdelávacích aktivít v zmysle zákona 317/2009 Z.z. akreditované vzdelávania a overenia profesijných kompetencií získaných výkonom pedagogickej a odbornej činnosti alebo sebvzdelávaním (§35 ods.6 Zákona 317/2009 Z.z.) v jednotlivých moduloch Open Office (Write, Calc, Impress, Base), alebo základy práce s Open Office.org. Viac informácií sa nachádza na adrese www.svsbb.sk. Prvých 20 učiteľov stredných a základných škôl zo Slovenska úspešne absolvovalo akreditované overenie profesijných kompetencií v moduloch základy práce s OpenOffice.Org, textový editor Writer a získali akreditovaný certifikát. Títo učitelia nadobudli poznatky a skúsenosti s používaním slobodného a otvoreného softvéru sebvzdelávaním a individuálnym štúdiom.

Záverom je potrebné povedať, že aj v našom školstve sa postupne rozvíja využívanie slobodného a otvoreného softvéru vo vzdelávaní a riadiacom procese. Ak sa vytvoria podmienky na používanie slobodného a otvoreného softvéru v prepojení s dobrým licencovaným softvérom, vytvorí sa tu priestor na adekvátny rast a vzájomné zdravé konkurenčné prostredie v oblasti softvéru. V tomto prostredí pre učiteľov a žiakov nášho školstva bude vytvorený priestor, aby sa v školskej praxi transformovala do reálnej podoby teória: „Technológie informačnej spoločnosti ponúkajú spôsob, aby učitelia menej učili a žiaci viac pochopili“.

Stanislav Slačka (Mgr.),

ÚIPŠ – Školské výpočtové stredisko, Tajovského 25,

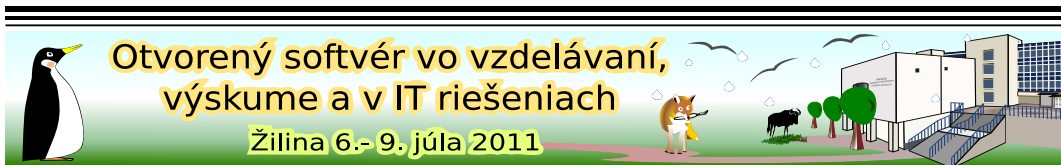
975 73 Banská Bystrica

stanislavslacka@gmail.com

Facebook:www.facebook.com/stanslacka

<http://sslacka.webnode.sk>

OCENENÉ ŠTUDENTSKÉ PRÁCE



CENA ZA ŠTUDENTSKE PRÁCE

SOIT (SK) a Liberix (CZ)

Spoločnosť pre otvorené informačné technológie a spoločnosť Liberix vyhlásili súťaž o najlepšiu bakalársku a diplomovú prácu súvisiacu s otvoreným softvérom a ďalšími otvorenými technológiami. Cieľom súťaže bolo stimulovať záujem o používanie OIT na školách, pretože sú to práve školy, na ktorých je propagácia progresívnych technológií najúčinnnejšia. Vypísaním Ceny chcú organizátori vysloviť uznanie pedagógom, ktorí sa touto formou venujú propagácii OIT.

Cieľovou skupinou Ceny boli slovenské a české univerzity so zameraním najmä na informačné technológie. Udeľovanie Ceny SOIT a Liberixu je súčasťou konferencie Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach. Posudzovateľmi prác boli Jakub Ondrušek, Jozef Šiška, Karol Pauchly, Michal Fojtík, Michal Páleník, Milan Moravčík, Milo Ofúkaný, Miloš Šrámek, Pavel Petrovič, Pavol Lupták, Pavol Rusnák, Peter Mráz, Peter Štrba a Rastó Šrámek.

1 Laureáti Ceny

Do súťaže za rok 2011 bolo zaslaných 15 prác. Posudzovatelia rozhodli o prvých troch miestach a laureátoch Ceny takto:

1. cena

Generovanie a spracovanie backtrace záznamov (Bakalárska práca)

- Autor: Michal Toman
- Vedúci: Marek Grác
- Škola: Masarykova Univerzita, Brno

Abstrakt. Práca sa zaoberá problémom vzdialeného generovania backtrace záznamu v operačnom systéme Fedora a popisuje implementáciu aplikácie Retrace Server, ktorá túto službu poskytuje.

Práca bola vypracovaná v spolupráci s Red Hat Czech, s. r. o.

1. cena

Podpora snapshotu a rollbacku pro konfigurační soubory v distribuci Fedora (Bakalářská práce)

- Autor: Jan Včelák
- Vedúci: Aleš Smrčka
- Škola: Vysoké učení technické v Brně, Fakulta informačních technologií

Abstrakt. Tato práce se zabývá analýzou možností zálohování a verzování konfiguračních souborů pod operačním systémem GNU/Linux. Na základě zjištěných skutečností je navržen, implementován a otestován nový nástroj pro distribuci Fedora. Jeho cílem je zachovat výhody existujících řešení a odstranit jejich nedostatky. Také přináší nové užitečné funkce, jako je synchronizace se vzdáleným serverem a možnost integrace do jiných nástrojů pro správu.

Práce bola vypracovaná v spolupráci s Red Hat Czech, s. r. o.

2. cena

Databázový manager pro IDE CodeLite (Diplomová práce)

- Autor: Peter Janků
- Školitel: Michal Bližňák
- Škola: Univerzita Tomáše Bati ve Zlíně, Fakulta Aplikované Informatiky

Abstrakt. Cílem diplomové práce je vytvořit multiplatformní aplikaci pro práci se serverovými a souborovými databázemi. Aplikace umožní manipulaci s daty a strukturou databází na základě SQL dotazů, vizualizaci a editaci databázové struktury pomocí ERD diagramů a generování tříd jazyka C++ pro přístup k databázovým objektům.

2 Prihlásené práce

Okrem ocenených prác, boli do súťaže prihlásené tieto práce (bez poradia):

Lukáš Bezdička: *Prezentačný režim pro PDF prohlížeč evince*

- Školiteľ: RNDr. Jan Kasprzak
- Škola: Masarykova univerzita Fakulta informatiky, Brno

Marek Filip: *Kalibrácia fotoaparátov a kamier, princípy a otvorený softvér*

- Školiteľ: Mgr. Michal Kaukič CSc.
- Škola: Žilinská univerzita v Žiline

Zoltán Harsányi: *Nástroj na vytváranie mashup aplikácií*

- Školiteľ: doc. Ing. Viera Rozinajová, PhD.
- Škola: FIIT STU

Václav Jureček: *Návrh open-source školního informačního systému s důrazem na specifika open-source projektů*

- Školiteľ: Ing. RNDr. Barbora Bůhnová, Ph.D.
- Škola: Masarykova Univerzita, Brno

Ivan Juriga: *Programovanie robotov LEGO v Linuxe*

- Školiteľ: Ing. Michal Kvasnica PhD.
- Škola: STU Bratislava, Fakulta chemickej a potravinárskej technológie

Peter Margetiak: *Systém pre voľby v lokálnych komunitách*

- Školiteľ: Mgr. Michal Kaukič, CSc.
- Škola: Žilinská univerzita v Žiline

Adrián Pekár: *Optimalizácia zhromažďovacieho procesu nástroja BasicMeter*

- Školiteľ: Ing. Juraj Giertl, PhD.
- Škola: Technická univerzita v Košiciach

Petr Kolář: *Svobodný software a jeho možnosti při výuce na FM VŠE*

- Školitel: Ing. Pavel Pokorný
- Škola: VŠE Praha, Fakulta managementu v Jindřichově Hradci

Jitka Polednová: *Geografické informační technologie ve výuce*

- Školitel: Ing. Pavel Pokorný
- Škola: VŠE Praha, Fakulta managementu v Jindřichově Hradci

Vladimír Smataník: *Rozšíření možností programu Geogebra o nástroje pro výučbu pravde-
podobnosti a statistiky*

- Školitel: RNDr. Aleš Kozubík, PhD.
- Škola: Fakulta riadenia a informatiky, Žilinská univerzita v Žiline

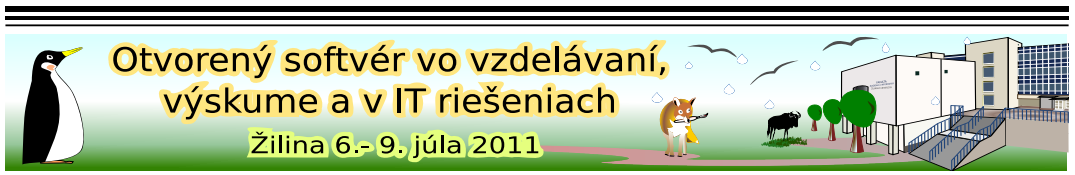
Matúš Valo: *Archivácia živých videokonferenčných prenosov v prostredí systému EVO*

- Školitel: Ing. Miroslav Biňas, PhD.
- Škola: Technická univerzita v Košiciach

Ľubomír Vasek: *Mapy pre turistov v open source prostredí*

- Školitel: prof. Ing. Miloš Šrámek, PhD.
- Škola: Univerzita Komenského, Fakulta matematiky, fyziky a informatiky

OTVORENÝ SOFTVÉR VO VEDE A VÝSKUME



DOI: 10.5300/2011-OSSConf/39

USING AN OPEN SOFTWARE FOR ELECTRONICS AND ROBOTICS

BALOGH, Richard, (SK)

Abstract. *In our paper we describe several applications used for the design of embedded systems and robots and also for teaching purposes in Robotics. We used the gEDA suite for the design of new electronic controller of the mobile robot. We also mention programming environments Arduino and Processing for the robot programming. We describe the use of powerful computational tool Octave for advanced data analysis in mobile robotics. The same software we use in education of Robotics at the Faculty of Electrical Engineering and Information Technology (FEI) STU in Bratislava. All the mentioned software we use on Windows XP, although they are also available for other platforms.*

Key words and phrases. *Computer aided design, simulation software, programming.*

1 Introduction

In the area of the mobile robotics it is often necessary to design the electronic control and auxiliary circuits, sensors and actuators. In this paper we show how we designed a new controller for mobile robot Acrob [1] using an open design system gEDA. Improved robot was used to create maps of the environment using ultrasonic distance sensors. For the subsequent analysis and visualization of measured data, we use an open source software GNU/Octave. The mobile robot is used also for teaching Robotics at the FEI STU. For its programming we use an Arduino environment and for visualization a Processing language. All applications are used in Windows XP platform.

2 gEDA – CAD for electronics

To design a new control unit for the mobile robot, we used several programs from the gEDA package [2]. It is a project that has produced and continues working on a full GPL'd suite and toolkit of Electronic Design Automation tools. These tools are used for electrical circuit design, schematic capture, simulation, prototyping, and production. Currently, the gEDA project offers a mature suite of free software applications for electronics design, including schematic capture, attribute management, bill of materials (BOM) generation, netlisting into

over 20 netlist formats, analogue and digital simulation, and printed circuit board (PCB) layout.

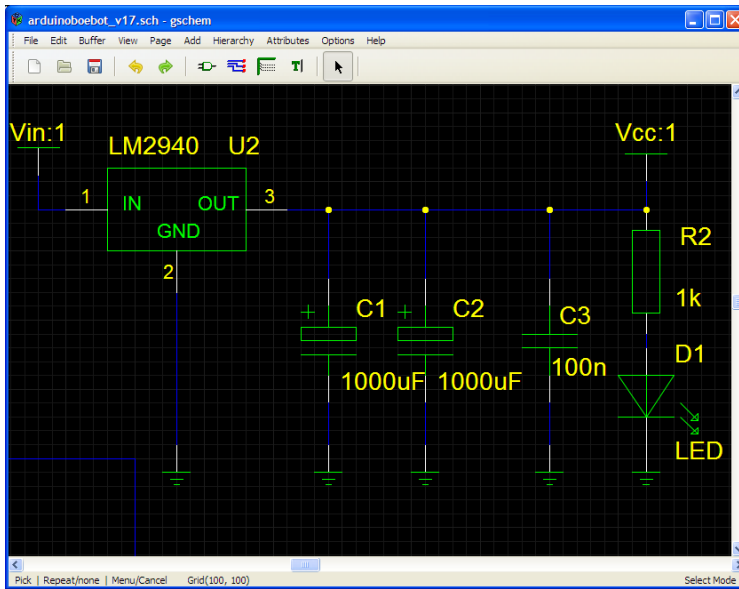


Fig. 1: Schematic capture editor gschem

At the beginning of the project we need to design diagrams using the gschem program. It is a program which creates a graphic form of the connection diagram (Fig. 1). We used version 1.6.0.20091004 compiled for Windows. The application offers many pre-installed component libraries and with an ability to add missing ones. Very helpful was also the shared component database available at the <http://www.gedasymbols.org/> (see Fig. 2). Then we were able to connect the components with electric connections and provide them with the necessary attributes.

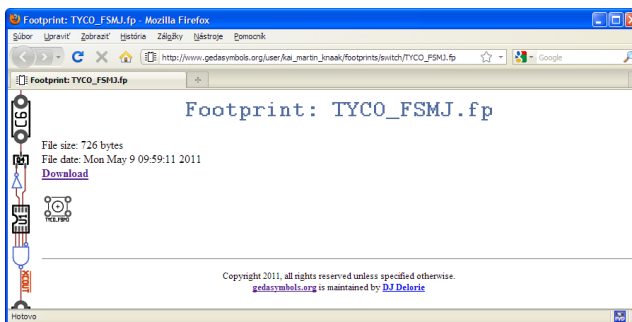


Fig. 2: On-line component database at the gedasymbols.org

After the error-checking of the design according to the design rules, we can generate a list of components and assign various parameters to them, e.g. the case information for the purpose of the printed circuit board design as well as the ordering code for component purchase. Then we continued with the layout design of the printed circuit board with the pcb program from the gEDA suite (version 20091103 on win32). It is an interactive printed circuit board editor for the X11 window system. PCB includes a rats nest feature, design rule checking, and can provide industry standard RS-274-X (Gerber), NC drill, and centroid data (X-Y data) output for use in the board fabrication and assembly process. PCB offers high end features such as an auto-router and trace optimizer. For some less-common components we had to prepare a description file with their case dimensions and pinout. Those *footprints* have been added to the library. Most commonly used elements are already a part of the installed libraries. Newly created components can be shared in the above-mentioned database at the gedasymbols.org.

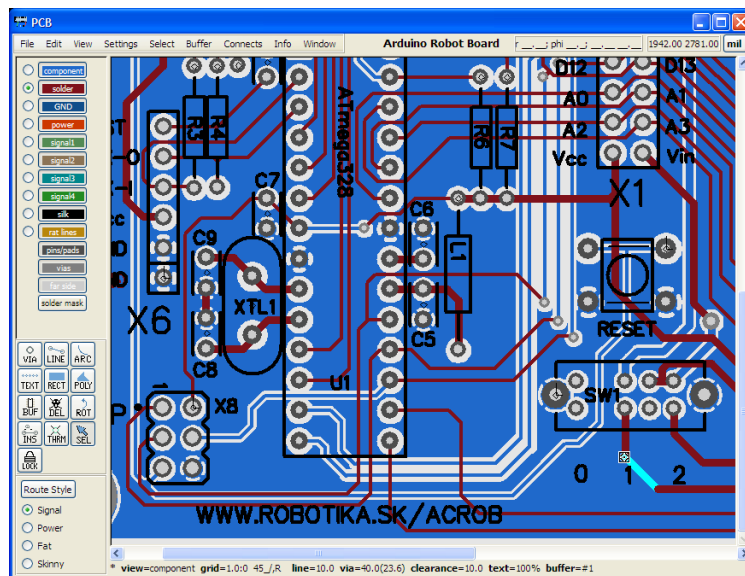


Fig. 3: Printed circuit board editor pcb

Design of the board starts with definition of the size and deployment of components, whose positions are important (eg. connectors at the edges, mounting holes, switches, etc.). Then we can import the list of interconnections generated by a schematic editor. These appear as rubber nets – *rat-nets*. It is very helpful during the placement of remaining components, because we can see whether it is not in place with too much connections, which can make interconnection impossible. When all components are on their place we start to interconnect them in several layers of the printed circuit board. Although the program offers also the autorouter, its results were unsatisfactory for us, so we placed all tracks by hand (Fig. 3). The

final design phase includes the labelling, texts placement and other auxiliary information for the technological process. Then finally the board can be sent into the production.

The production was without problems, a manufacturer did not request any corrections or additional information and printed circuit boards were made in an outstanding quality. Assembled control unit is on Figure 4.

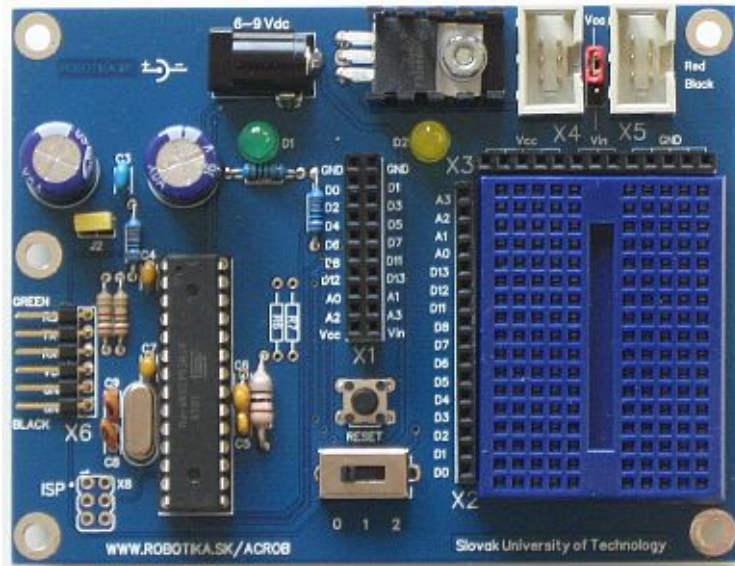


Fig. 4: Assembled and populated board

3 Arduino and Processing

Important part of the design process was also the decision about the possibility of its programming. Based on our positive experiences with an avr-gcc compiler [5] for Atmel processors, we decided to use popular environment Arduino. It allows an easy programming in C++ and provides the wide users and developers community, so almost for any common task there are already libraries available.

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments [3].

A sample program for a mobile robot is shown in Fig. 5. Its task is to measure the distance ultrasonic sensor and to send the measured value to the master system using the serial line (converter to USB was used). Received data can be monitored directly in a terminal window, which is part of the Arduino environment. When we want to visualize data, it is good idea to use a more specialized program. In our case it was almost ready Peter Dainty's project

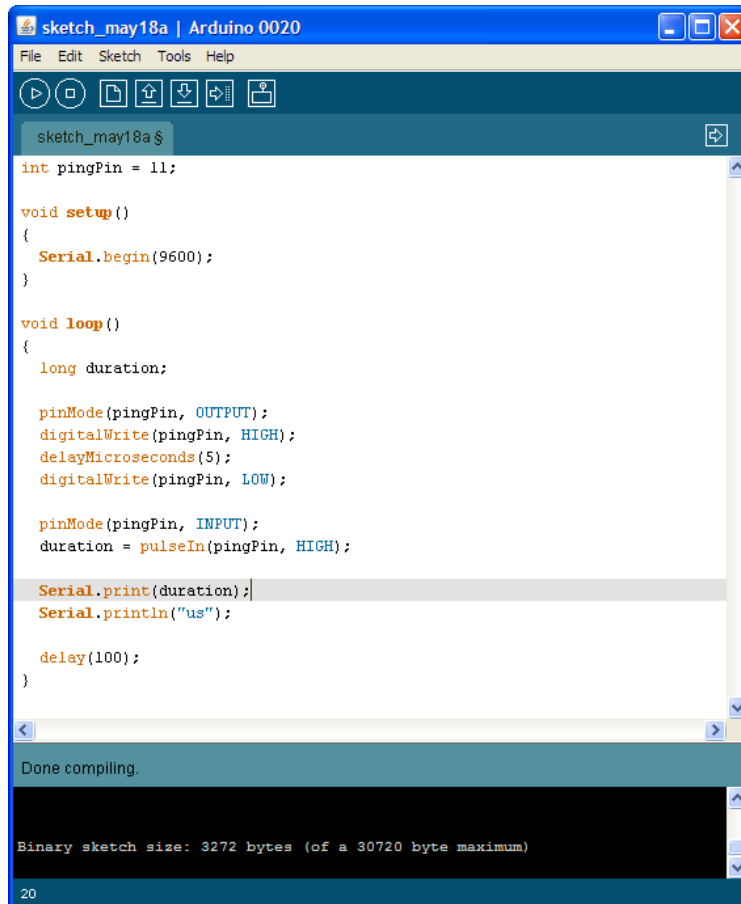


Fig. 5: Arduino development environment

RadarScreen [6], which was written in Processing [7]. Since the author also provided source code, it was no problem to add and modify it according to our requirements. We added a drop-box with a choice of serial port, on which we expect the measured data. These are then displayed real time on the screen (Fig. 6). The finished project was made available to the students who monitored their own data in the labs.

Processing is an open source programming language and environment for people who want to create images, animations, and interactions. Initially developed to serve as a software sketchbook and to teach fundamentals of computer programming within a visual context [7].

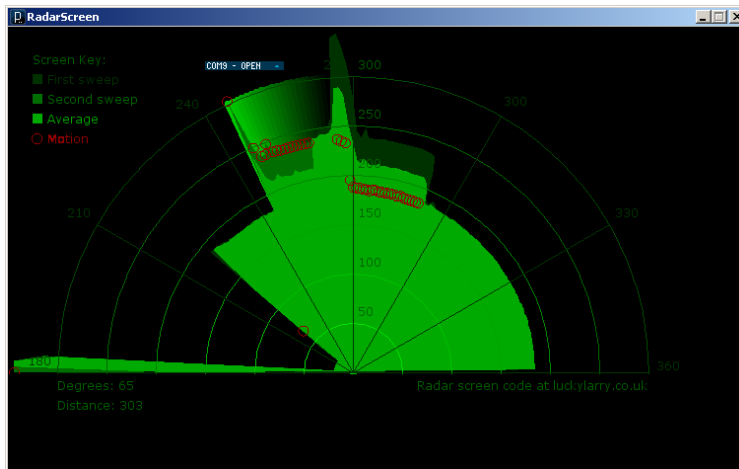


Fig. 6: Radar Screen window

4 GNU/Octave

For the measured data analysis it is obvious to use a Matlab – a high-level language and interactive environment that enables you to perform computationally intensive tasks. Unfortunately, the price for Matlab doesn't not allow its common deployment in schools. Moreover, the academic license does not allow its use for research. Therefore, we choose the GNU/Octave, even working with it is often more difficult than with Matlab.

GNU/Octave is an interpreted language, primarily intended for numerical calculations [8]. Provides functions for the numerical solution of linear and non-linear problems and to implement other numerical experiments. It also provides extensive graphics capabilities for data visualization and manipulation. Octave language syntax tries to follow the syntax introduced in Matlab, so most programs are easily portable. Octave is normally used through its interactive command line interface, but it can also be used to write non-interactive programs. Less experienced users can use the graphical front-end QtOctave [9] for otherwise console application.

We already mention that work with Octave is sometimes more difficult than with the Matlab. From our point of view, the main reason is the lack of huge repository of libraries known from the Matlab environment and in our case also the more difficult installing process in the Windows OS.

The example below illustrates the use of this package for calculations and displaying of measured data in robotics. The graph in Fig. 7 shows a map generated by a mobile robot based on multiple ultrasonic distance sensor measurements and assigning certain probabilistic value for the area reached by an acoustic sensor cone. Individual measurements are then combined based on recursive form of Bayesian theorem for the conditional probability for each square of the occupancy grid. The calculations are based on ultrasonic sensor model described

in [10] and adjusted to our sensor properties. We used Octave version 3.3.1991 and its front-end QtOctave 0.10.1. Scripts were tested and worked both in an Octave and Matlab environments.

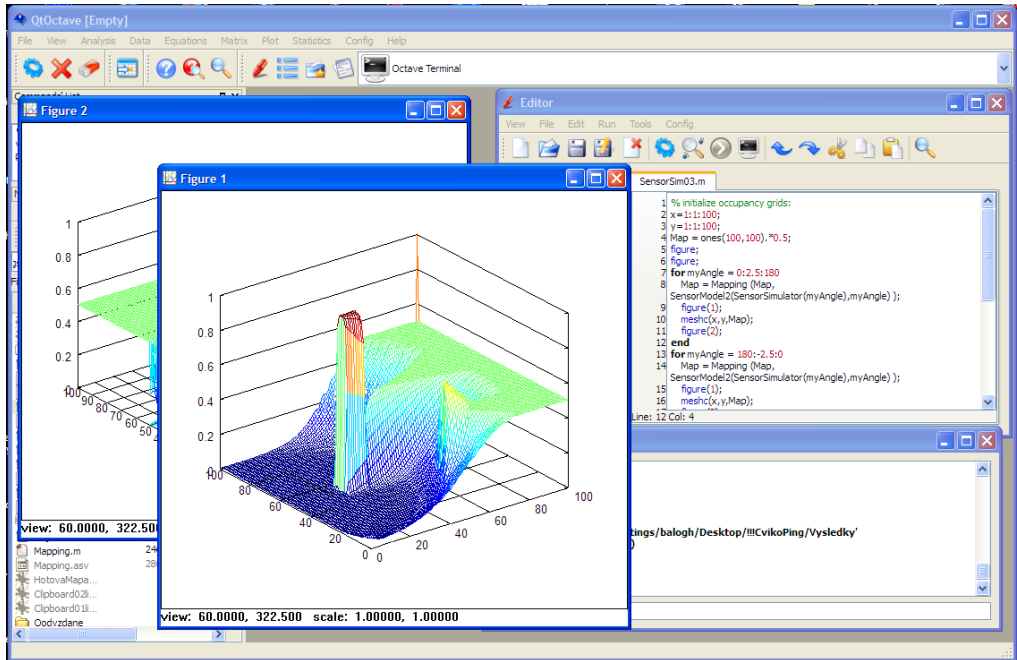


Fig. 7: Radar Screen

5 Educational use

All results of our research are used also in teaching. The mobile robot Acrob is used for the laboratory exercises at the Robotics course in the 3rd year of the bachelor studies. Students write their programs in the Arduino environment and we provide them with pre-compiled visualisation applications written in Processing. For the data analysis students prefer to use the Matlab (when available). To produce reports of measurements they use an OpenOffice when working in the classroom (because another package was not available). Surprisingly there were no problems for students to use it. For homework students prefer to use a MS Office, so documents sometimes have problems with a portability. From the perspective of a teacher the biggest advantage of open source software is the opportunity to offer the software for students wishing to penetrate deeper into the issues. Last, but not least there is a time saving lack of the administration processes associated with the commercial applications (license management, public tenders, registering, updating, etc.).

6 Conclusion

Although the author of the article is not an uncritical admirer of an open source software, coincidentally in our project we used only this kind of applications. Especially in our geographic region, its price is one of the biggest advantage. On the other hand, there is more complicated installation of such products and solving dependencies between them, which is more time-consuming. Fortunately, in an academic area many of these problems can be solved by the students within the projects. Even the first (bachelor's) degree students can at least explore the possibilities and smooth the paths for further steps in the effective usage of such programs.

For the CAD software, we found no major problems standing in the way of their routine use. The largest investment in this type of program depends in the rich component libraries. Even a small error in e.g the dimensional description of single component can result in a deterioration of the whole product. Therefore, precise and proven libraries are valued and highly appreciated. After establishing a reliable and proven libraries the work is easy and enables some automation and also the interoperation with other systems, since the data format is open too.

For the Arduino environment we encountered some problems and we can see several opportunities for further improvement of its environment. An Octave environment holds the basic compatibility with the popular Matlab, but some less common requirements are difficult to deal with. In particular, it is a missing equivalent to the graphical environment Simulink known from the Matlab. Also, we were unable to read data in real time from the serial interface, although probably in a different operating system it would work.

All around, the open software applications are a promising alternative worth to try, especially in an academic area, offering cheap and open solutions.

Acknowledgement

Publication of this paper was supported by the grant VEGA 1/0177/11.

References

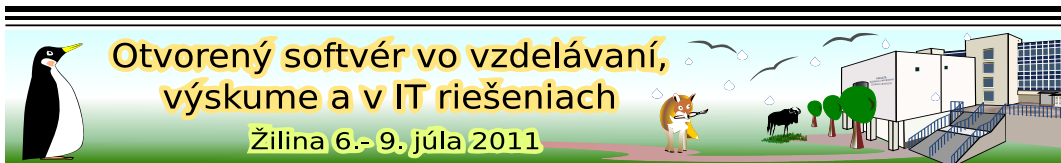
- [1] BALOGH, R.: *Acrob – an Educational Robotics Platform*, AT&P Journal Plus, November, 2010. Vol. 10 (2), pp. 6-9. Available online: http://www.atpjournals.sk/buxus/docs//casopisy/atp_plus/plus_2010_2/atp_plus_2_2010_pre_web.pdf
- [2] gEDA project. Available online: <http://www.gpleda.org>
- [3] Arduino. Available online: <http://www.arduino.cc/>
- [4] BANZI, M.: *Getting Started with Arduino*. O'Reilly, 2009.

- [5] GCC, the GNU Compiler Collection. Available online: <http://gcc.gnu.org>
- [6] DAINTY, Peter: *Arduino + Processing: Make a Radar Screen's Visualise Sensor Data from SRF-05*. Available online: <http://luckylarry.co.uk/arduino-projects/>
- [7] Processing - an open source programming language. Available online: <http://www.processing.org/>
- [8] GNU Octave Project. Available online: <http://www.octave.org>
- [9] ROSADO, Pedro Luis Lucas: *QtOctave version 0.10.1 Project page*. Available online: <https://forja.rediris.es/projects/csl-qt octave/>
- [10] MURPHY, Robin: *An Introduction to AI Robotics*. The MIT Press, November 2000. ISBN 0-262-13383-0.

Contact

Richard BALOGH,

Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Ilkovičova 3,
812 19 Bratislava, Slovakia, richard.balogh@stuba.sk



DOI: 10.5300/2011-OSSConf/49

UTILISATION OF THE OPEN BUILD SERVICE FOR A VOLUME DATA PROCESSING LIBRARY

HUČKO, Michal, (SK)

Abstract. *Building and installing of software can be a tedious process. This is especially true if the software has large number of dependencies or it's highly configurable. Ensuring problem-free distribution of such software to wide range of platforms or distributions can be rather complicated. In the paper we describe our experience with utilisation of the Open Build Service for binary distribution of a library for volume data processing. The library contains implementation of filtering operations on various hardware or software platforms like OpenGL, CUDA or OpenCL.*

Key words and phrases. *OBS, Open Build Service, volume data processing.*

VYUŽITIE SLUŽBY OPEN BUILD SERVICE PRE KNIŽNICU NA SPRACOVANIE OBJEMOVÝCH DÁT

Abstrakt. *Kompilácia a inštalácia softvéru býva často komplikovaný proces. Špeciálne v prípade softvéru s veľkým počtom závislostí na iných knižniciach, prípadne softvéru so širokými možnosťami konfigurácie. Docieliť bezproblémovú distribúciu takéhoto softvéru na väčšie množstvo platforiem, resp. distribúcií je pomerne pracné. V článku popisujeme využitie služby Open Build Service na binárnu distribúciu knižnice a nástrojov na spracovanie objemových dát. Knižnica obsahuje implementáciu filtrovacích operácií pre rôzne hardvérové a softvérové platformy, akými sú OpenGL, CUDA, či OpenCL.*

Kľúčové slová. *OBS, Open Build Service, spracovanie objemových dát.*

1 Introduction

Delivering an application to a user can be a non-trivial process. This is especially true if the application has software or hardware dependencies that are not common. Either the application is distributed as a source code which needs to be built or compiled to a package or executable installer. The first option gives higher demands on a user of the application when all the dependencies have to be preinstalled and build process needs to be configured by the user in respect to his/her hardware and software. In the case of binary distribution all the tedious work lies on the packager lifting the burden of compilation from the end-user.

Still there is lot of work to be done to deliver all configurations of the software to all possible target operating systems. Luckily, this can be automated with certain tools or services – for Linux based operating systems the openSUSE’s Open Build Service (OBS) aims to provide the necessary tools and infrastructure. We prospected utilisation of the OBS for distribution of the f3d library – a collection of tools for processing of volumetric data (e.g. CT or MRI scans). As the f3d library takes advantage of special hardware like graphics cards, we explored possibilities of using CUDA or OpenCL under OBS.

2 The f3d library

The f3d acronym stands for ‘format three-dimensional’ [1] and actually is also a name of the file format used by the library. Purpose of the library is to provide tools for volumetric data processing and analysis. In the following text we’ll explain motivation for such library, it’s organisation and implementation.

2.1 Volumetric data

The f3d library is aimed at working with volumetric data used in medicine. These include data from computed tomography (CT), magnetic resonance imaging (MRI) and various other acquiring methods. All the scanning devices create not just a single image, but a volume of voxels (volume elements). One can imagine that the data consist of a stack of common 2D images called slices. In fig. 1 there are example images from CT and MRI. Dependent on the resolution of the used scanner and size of the scanned subject the measured data size ranges from tens of MiB to of couple of GiB. For example, data of dimensions of 1024 cubed having values stored in single precision floating point numbers has 4GiB.

2.2 Data processing

Similar to the case of 2D images also for volumetric data there exist methods which aim at increasing quality of the data or removing artifacts caused by measuring device. For example, if the data is noisy we might want to smooth it in order to avoid errors in subsequent analysis. Operations implemented in the library include: distance transform, edge detection, Gaussian filtering (smoothing), enhancement of tubular structures, thresholding, watershed transform and others [2] (examples are shown on fig. 2).

The previously mentioned operations can be classified in three groups – point, local and global operations. The first two groups require for processing of one voxel either no or only a local neighbourhood of the voxel. Global operations generally might visit any voxel in the data in order to process a single voxel of the volume. When common dimensions of the data are taken into account this yields a great number of computations which – especially in the case of local operations – are identical and rather primitive. When processing of one voxel is independent of the processing of other voxels we have a task which can be easily parallelised. This makes utilisation of SIMD (single instruction – multiple data) capabilities of recent processors and graphics cards an apparent optimisation.

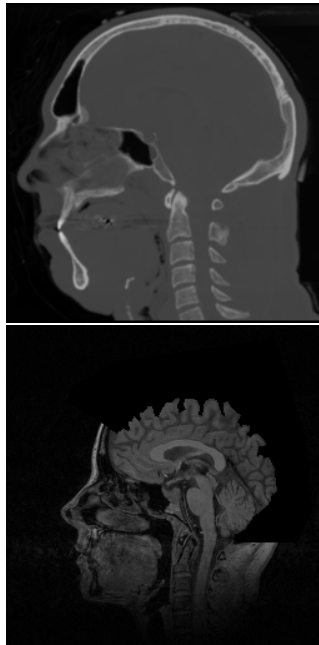


Fig. 1: CT (left) and MRI (right) sample image

2.3 The library organisation

The f3d suite consist of couple of parts. Each part is written in C/C++ and is managed by the CMake build system. Libraries can be built either as static ones or as shared (we use shared variants for the packaging). Executables (tools) might be in the form of a C++ application or a shell script. We now introduce all the parts and specify their dependencies.

f3dformat Written in pure C, contains functions for reading and saving f3d files.

Depends on zlib

Output libf3dformat.so*

f3dla Basic liner algebra classes used in the processing operations.

Depends on none

Output libf3dla.so*

f3dview Viewer application for the f3d data. Displays slices of the volume in the x, y and z axes.

Depends on f3dformat, wxWidgets

Output f3dview (executable)

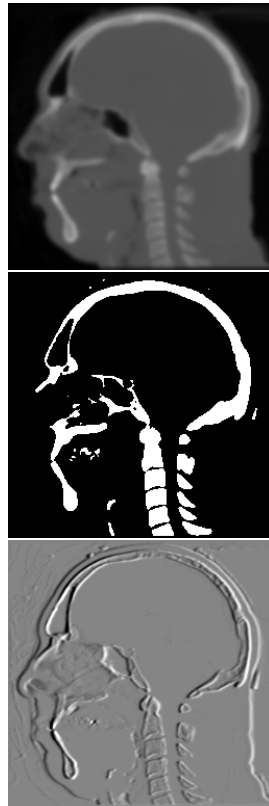


Fig. 2: From left to right – blurred image, thresholded image, gradient image

f3dfilter A library containing implementations of various filters for different software and hardware architectures.

Depends on yasm, Mesa, CUDA, OpenCL, GSL

Output libf3dfilter.so*, libf3dfilterCPU.so.*, libf3dfilterSSE.so.*,
libf3dfilterOpenGL.so.*, libf3dfilterCUDA.so.*, libf3dfilterOpenCL.so.*

f3dclass Collection of tools (executables) executing various filter operations on the volume data.

Depends on f3dformat, f3dla, f3dfilter, zlib

Output libf3dclass.so*, various filter executables named f3d*

The base part of the f3d project is the f3dformat library. It provides functions for reading from and writing to f3d files. External projects which use f3d file format and other f3d packages which need access to f3d files use this library.

For simple inspection of the volume files the `f3dview` application can be used. It uses `wxWidgets` as a GUI framework and of course is dependent on `f3dformat` library.

The rest of the executables come from `f3dclass` part of the project. Some of the operations are implemented directly in the `f3dclass` library while rest of the operations are implemented in the `f3dfilter` libraries. This includes operations which can be optimised by utilising the SIMD processing capabilities of modern processors or graphics cards. SIMD stands for “single instruction – multiple data”. For example, modern processors with SSE instructions can perform single operation (i.e. multiplication) simultaneously on four pairs of numbers instead of just one. Similarly, today graphics cards offer possibilities of general parallel computing having multiple computing units. Operations like separable or non-separable convolution can be easily parallelised.

Currently, in addition to basic CPU code, there are implementations of filters for processors with SSE instructions and graphics cards using OpenGL and CUDA (nVidia only). OpenCL versions of certain filters also exist supporting both processors and graphics cards. Because each variant has different hardware and software requirements, it is essential to allow installation of only certain types of filters (i.e. no CUDA on a system with an AMD graphics card). This of course requires checking of the available implementations on-the-fly.

As it was shown earlier, the `f3dfilter` part of the project creates one `f3dfilter` library and number of `f3dfilterXXX` libraries – one for each implementation variant (XXX denotes the variant). The general library contains no implementation of the filters at all, it contains manager classes instead. A manager class probes for the available shared libraries containing hardware dependent filter implementations and returns list of those which were found. Upon request it creates instance of the filtering class and returns it to the caller. Therefore, to successfully filter a volume with some operation the `f3dfilter` library and at least one `f3dfilterXXX` library having hardware implementation of the requested operation is required.

3 Open Build Service (OBS)

The Open Build Service¹ (only recently renamed from openSUSE Build Service) is an environment and a service provided by openSUSE/Novell for building, packaging and distribution of software. Originally developed at openSUSE to be used for building add-on packages, it evolved to an open-source build tool supporting also non-SUSE distributions. It can be either installed on private servers or used as a service at <https://build.opensuse.org/>, where anyone can register and build packages in his/her own home project. All this makes OBS an ideal solution for packaging needs. Therefore, we employed OBS in `f3d` building and distribution noticing weak-spots of the process in respect to the `f3d` library.

We used the service of <https://build.opensuse.org> for building by registering and creating an `f3d` subproject in our home project². Because our main goal was distribution of the `f3d` package to potential users, we wanted to provide packages for as many distributions as possible. After some experiments we narrowed the supported distributions to openSUSE,

¹<https://build.opensuse.org>

²<https://build.opensuse.org/project/show?project=home:granxarixia:f3d>

Fedora, Debian and Ubuntu. On other distributions we had problems with meeting some requirements which in our case was the CMake build system and the wxWidgets library.

Earlier, f3d was distributed only as a source and it was necessary to build it by hand. For users having previously no experience with the library it was often a tedious process, consisting of installing requirements, configuring the project with CMake and selecting the desired variants of the f3dfilter library. When a problem arose it was often too complicated to solve it quickly. Now, all problems are solved by the packager when sources are uploaded to the OBS and the user needs only to install the built packages.

3.1 Problematic requirements

In addition to the earlier mentioned requirements which were not met on some distributions we had also problems with certain hardware variants of the f3dfilter library. The SSE version of the library has simple requirements – it requires only an assembler, which is easy to satisfy. Also no special hardware drivers are necessary. The OpenGL variant requires graphics drivers which provide OpenGL implementation. This requirement can be easily met with the Mesa package which provides open-source implementation of the OpenGL standard. For execution of the filter it may be necessary to have proprietary drivers installed to achieve decent computational times or even run the application because of the missing OpenGL extensions. However, for packaging purposes it's important that the library can be built without the need of any proprietary software and that the produced package has clean and easily met dependencies.

The problematic technology in respect to packaging are OpenCL and CUDA. The latter is a proprietary technology of nVidia – one requires a proprietary SDK from nVidia to build the software. The SDK is not available in the common distribution repositories. Another problem lies in the availability of the proprietary drivers. Although there exists repository with these drivers for openSUSE (and possibly other distributions) it is hosted by nVidia and therefore inaccessible to OBS. This prohibits us from mirroring the needed packages and providing a full dependency tree to users of our repository.

Similar situation to CUDA is with the OpenCL technology. Though the specification is free and managed by the Khronos³ group, implementations are proprietary. For the consumer hardware AMD provides an OpenCL implementation supporting their processors and graphics cards, nVidia supports their graphics cards and only recently Intel released the Linux version of their OpenCL implementation supporting Intel processors. However, we observed that the code compiled with nVidia's SDK successfully ran on a computer with no nVidia hardware or software having AMD components and AMD OpenCL libraries installed. This shows a possibility to have an open-source library like Mesa which would give a basic OpenCL support. If such library would be created, it would be possible to build and package software using OpenCL. On the client side dependencies would be easily met. Such library would not need to implement the whole functionality as for the utilisation of OpenCL one has to choose hardware platform from the list of supported platforms. With no proprietary

³<http://www.khronos.org/>

drivers empty list could be returned, prohibiting the application from continuing. This would create a similar situation as in case of OpenGL when software using the graphics library can be distributed practically on every system, however runs only when appropriate drivers are installed. No direct dependency on non-open-source software exists.

3.2 The provided packages

Because of the previously mentioned problems our repository currently provides only CPU, SSE and OpenGL variants of the f3dfilter library. The user is free to select which packages he/she wants according to his/her needs. For example, there's no need in installing the OpenGL variant of the filters on a machine without a graphics card and with no X server.

4 Conclusions

In the paper we presented the f3d library for volume data processing and our experiences with the usage of the OBS service for it's building, packaging and distribution. We mentioned observed problems with proprietary software on which parts of our library is dependent. With the perspective of OpenCL replacing CUDA in the future and possible creation of open-source OpenCL implementation (containing management code only) we hope to deliver full-featured f3d suite. Still we greatly simplified distribution of the f3d library and tools supporting major Linux distributions in comparison to previously required manual building of the software.

Acknowledgement

This work was supported by the Science Grant Agency (VEGA) under contract No. 1/0631/11.

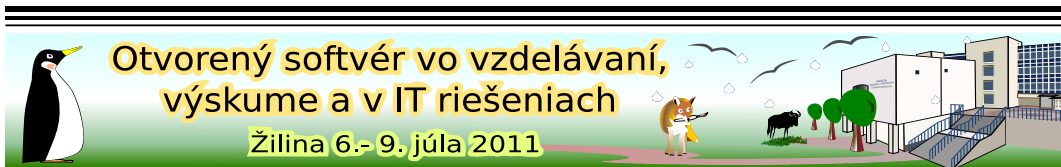
References

- [1] ŠRÁMEK, M. – DIMITROV, L. I.: *f3d – a file format and tools for storage and manipulation of volumetric data sets* : 1st International Symposium on 3D Data Processing, Visualization and Transmission, pages 368-371, 2002
- [2] ŠRÁMEK, M. – DIMITROV, L. I. – STRAKA, M. – ČERVEŇANSKÝ, M.: *The f3d tools for processing and visualization of volumetric data* : Journal of Medical Informatics and Technologies, pages MIP-71-MIP-79 2004

Kontaktná adresa

Michal HUČKO,

Katedra aplikovanej informatiky FMFI UK v Bratislave, Mlynská dolina,
842 48 Bratislava, michal.hučko@fmph.uniba.sk



DOI: 10.5300/2011-OSSConf/57

MIRROR – A PORTABLE C++ REFLECTION LIBRARY

CHOCHLÍK, Matúš, (SK)

Abstract. *Reflection and reflective programming can be used as valuable tools in a wide range of tasks such as for implementation of serialization, marshalling, remote procedure calls, scripting, automated GUI-generation, implementation of several distinct design patterns, etc. C++ as one of the most prevalent programming languages however, for various reasons, lacks a standardized reflection facility. In this paper we present Mirror - a portable library adding reflection to C++. This library supports functional style static compile-time reflection and meta-programming and also provides an object-oriented run-time layer for dynamic reflection.*

Key words and phrases. *Reflection, reflective programming, meta-programming, design-pattern implementation.*

MIRROR – PRENESITEĽNÁ REFLEXNÁ KNIŽNICA PRE C++

Abstrakt. *Reflexia a reflexívne programovanie, sú užitočné nástroje, použiteľné v rôznych situáciách, napríklad pri implementácii serializácie, marshallingu, vzdialeného volania procedúr, podpory pre skriptovanie, automatizovaného generovania grafických používateľských rozhraní, atď. C++ ako jeden z najrozšírenejších programovacích jazykov, nemá štandardizovanú podporu pre reflexiu. V tomto článku prezentujeme Mirror – prenesiteľnú reflexnú knižnicu pre C++. Táto knižnica poskytuje funkcionálne rozhranie pre compile-time reflexiu a objektovo-orientované rozhranie pre run-time reflexiu.*

Kľúčové slová. *Reflexia, reflexívne programovanie, meta-programovanie, implementácia návrhových vzorov.*

1 Motivation

There is a wide range of computer programming tasks that involve the execution of the same algorithm on a set of types defined by an application or on instances of these types, calling of free or member functions in an uniform manner, conversion of data between the language's intrinsic and an external formats for the purpose of implementing the following:

- serialization or storing of persistent data in a custom binary format or in XML, JSON, XDR, ASN1, etc. and deserialization or construction of class instances from external data representation formats (like those listed above), from the data stored in a relational database, from the data entered by a user through a GUI
- automatic generation of a relational schema for application data and object-relational mapping (ORM)
- support for scripting and remote procedure calls (RPC) / remote method invocation (RMI)
- object inspection via a GUI and object access via web services (WS)
- visualization of data and the relations in the data
- implementation of some design patterns
- conceptual description and documentation of a software system

There are several approaches to the implementation of such systems. The most straightforward and also usually the most error-prone is manual implementation. Many of the tasks listed above are inherently repetitive and basically require to process programming language constructs (types, structures, containers, functions, etc.) in a very uniform way that could be easily transformed into a meta-algorithm.

While it is acceptable (even if not very advantageous) for example, for a design pattern [9] implementation to be made by a human, writing RPC/RMI related code is a task much better suited for a computer. Other situations like reference manual and documentation creation, require some human work, but can be partially automated.

This leads to the second, heavily used approach, preprocessing and parsing of the program source text by a (usually very specific) external program (documentation generation tool, interface definition language compiler for RPC/RMI, web service interface generator, a rapid application development environment with a form designer, etc.) resulting in additional program code, which is then compiled into the final application binary.

This approach has several problems, first it requires the external tool which may not fit well into the build system or may not be portable between platforms or be free; second, such tools are task-specific and many of them allow only a limited, if any, customization of the output.

Another way to automate these tasks is to use reflection, reflective programming, meta-programming [1] and generic programming [2], as shown for example in [11], for the implementation of the factory design pattern, though the method described therein is somewhat limited, because it requires that the constructed types or classes have a default constructor. The same approach is taken by several other reflection facilities, with the same limitations. Other examples of design pattern implementation with the help of reflection can

be found in [16, 17, 33]. Reflection can also be employed to perform other tasks listed above as shown by projects like [20, 28, 29, 32] and others.

2 Reflection – introduction and related work

The term reflection refers to the ability of a computer program to observe and possibly alter its own structure and/or its behavior. This includes building new or altering the existing data structures, doing changes to algorithms or changing the way the program code is interpreted [10]. It is a particular kind of *metaprogramming*.

The advantage of using reflection is in the fact that everything is implemented in a single programming language, and the human-written code can be closely tied with the customizable reflection-based code which is automatically generated by compiler meta-programs.

Support for reflective programming is most common in high-level languages, often using a virtual machine, an interpreter or another run-time environment, like JAVA, C#, CLOS or Smalltalk [3, 4, 7] and less common or limited in lower-level statically typed languages like Objective C, Lisp or Scheme [10]. Today, C++ as one of the most popular [25, 27] multi-paradigm programming languages, lacks a direct support for reflection. One of the attempts for standardized reflection in C++ was made in [15], but the work seems to be on hold recently [14]. There are several custom-built reflection systems, with varying degree of introspective and reflective capabilities, using various approaches.

The OpenC++ [26] is a programming toolkit that allows writing meta-level source-to-source transformation programs according to a meta-object-protocol (MOP) [4, 16] at the program preprocessing/parse-time. The meta-level program is compiled by C++ compiler linked with an OpenC++ add-on. Unfortunately only some compilers are supported which makes the approach non-portable. Publications [18, 19] introduce a similar source-to-source translator named FOG, adding support for aspect-oriented programming and compile-time reflection to C++. More common are run-time reflection systems like, the CERN's Seal C++ reflection system [14], or those described in [7, 8, 12, 13, 21, 30]. There are also several static reflection facilities with some run-time features, like the one described in [31].

3 Mirror library

3.1 Introduction and history

The Mirror library is a non-intrusive and portable compile-time introspection and reflection library for the C++ language, developed by the author. It provides reusable meta-data reflecting C++ constructs like namespaces, types, classes, their base classes, member variables, class templates and their template parameters, etc., and implements an extensive set of meta-programming tools for the traversal and usage of these meta-data in compile-time functional algorithms. Furthermore it provides several high-level tools, like the factory generator utility, described in greater detail in [6]. It adheres to the three principles of

well-designed reflective facilities [3]: *encapsulation*, *stratification* and *ontological correspondence*. There are two versions of the library, both available for download under the Boost Software License (accessible at the following URL: http://www.boost.org/LICENSE_1_0.txt).

The first version [22], implemented in C++98, which is portable across all platforms having a conforming C++ compiler. One of the basic usages of this library - object serialization and marshalling in parallel applications using the MPI [24], is described in [5]. However, the development of this version has been discontinued and its use is deprecated.

The second version [23], which is currently being developed, is using multiple new features from the recent ISO C++ standard (nicknamed C++11). This version removes several design deficiencies discovered during the usage of the previous version and in addition contains also a run-time layer, built on top of the compile-time meta-objects, providing a polymorphic object-oriented interface for their usage in run-time reflective algorithms. Recently there has also been added a third - object-oriented compile-time layer. This ternary interface makes the library applicable in a wide range of scenarios. It can be downloaded from SourceForge.net at: <http://sourceforge.net/projects/mirror-lib/files/>.

3.2 Architecture

Mirror uses a hierarchic architecture where each layer provides services to the layers above.

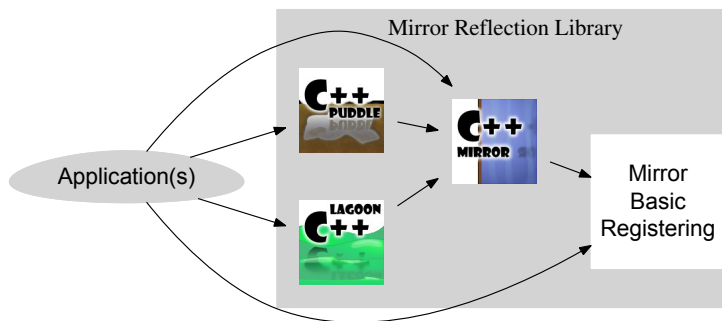


Fig. 1: Architecture of the Mirror library

- *Registering and basic meta-data*: Since standard C++ provides only a very limited set of meta-information to build upon, the basic programming language constructs like namespaces, types, classes, variables, functions, etc. need to be registered before they can be reflected. However Mirror tries to make the process of registering simple by providing a set of user-friendly registering macros and has the native and many of the other common types, classes, templates and namespaces pre-registered. This is the lowest layer and is used by the application only to register its components.

- *Functional compile-time layer* - **Mirror**: Built on the basic registered meta-data. Suitable for generic meta-programming similar for example to the standard `type_traits` library. Allows to write compile-time meta-programs which allow the compiler to generate efficient program code based on the meta-data provided by reflection.
- *Object-oriented compile-time layer* - **Puddle**: Based on the functional layer, it provides an object-oriented interface which is more suitable for certain programming tasks and also provides some run-time features, but is still inherently compile-time allowing for extensive optimization by the compiler.
- *Object-oriented run-time layer* - **Lagoon**: Based on top of the compile-time layer, it provides a run-time polymorphic interface, more suitable for run-time reflective programming, allowing the use of the provided meta-data in a dynamic manner dependent on other data available only at run-time. One of its disadvantages is the performance penalty induced by virtual function calls and the inability of the compiler to inline such calls in many cases together with long compilation times.

3.3 Goals and Features

The library is developed with the following goals in mind:

- *Reusability*: The meta-data provided by Mirror is reusable in many situations and for many different purposes.
- *Flexibility*: Mirror and the additional layers built on top of it allow to access the provided meta-data both at compile-time and run-time in a functional and object-oriented manner depending on the application needs.
- *Encapsulation*: Mirror and the additional layers provide interfaces for easy access to program meta-data.
- *Stratification*: Mirror is non-intrusive and separates the meta-level from the base-level constructs it reflects.
- *Ontological correspondence*: The meta-level facilities correspond to the ontology of the base-level C++ language constructs which they reflect.
- *Completeness*: Mirror tries to provide as much useful meta-data as possible, including various specifiers, iteration of namespace members and much more.
- *Ease of use*: Although Mirror allows to do very complicated reflective metaprogramming, simple things are kept simple.
- *Cooperation with other librares*: Mirror can be used with the introspection facilities provided by the standard library and other libraries.

3.4 High-level utilities

As briefly mentioned above, Mirror, besides the meta-objects reflecting various language constructs, also provides several high-level utilities built on top of the basic meta-data.

3.4.1 Factory generator

The factory generator utility allows to automate the implementation of the factory design pattern for types which are reflectible by the Mirror library. By factory we mean here a class, which can create instances of a *product* type, but does not require that the caller chooses the manner of the construction nor supplies the required parameters directly in the C++ intrinsic data representation.

Such generated factory examines the input which can be provided in an external data representation like XML, JSON, a simple scripting language, a dataset which results from a query to a relational database, a GUI, etc., selects the constructor that matches the available input data, converts the data from the external representation and calls the constructor. It may even construct some of the arguments recursively by the means of other, nested factories.

The process how the factories are generated from the meta-data provided by Mirror and a set of user-specified templates is described in [6]. Example of a GUI created by factory which was generated by the Mirror's factory generator for a simple *tetrahedron* class is shown on Figure 2.

3.4.2 Invoker generator

The invoker generator allows to create invokers which are able to call arbitrary reflectible functions, in the same manner as the generated factories call constructors. This feature is however still experimental.

3.5 External tools

The manual registering process can be tedious and error-prone in some situations. Although the registering macros use auto-detection and many things do not have to be specified explicitly, some changes in the base-level classes, like adding or removing of a member variable, constructor or a whole class, etc. require changes in the registering code.

If no special tweaking in the reflection of the base-level constructs (like hiding certain members or constructors of a class or specifying of getter/setter functions for a member variable) is required, then support for automatic reflection is desirable. To provide this support is the aim of the *MAuReEn* (Mirror Auto-Reflection Engine) project also developed by the author, available at <http://gitorious.org/maureen>.

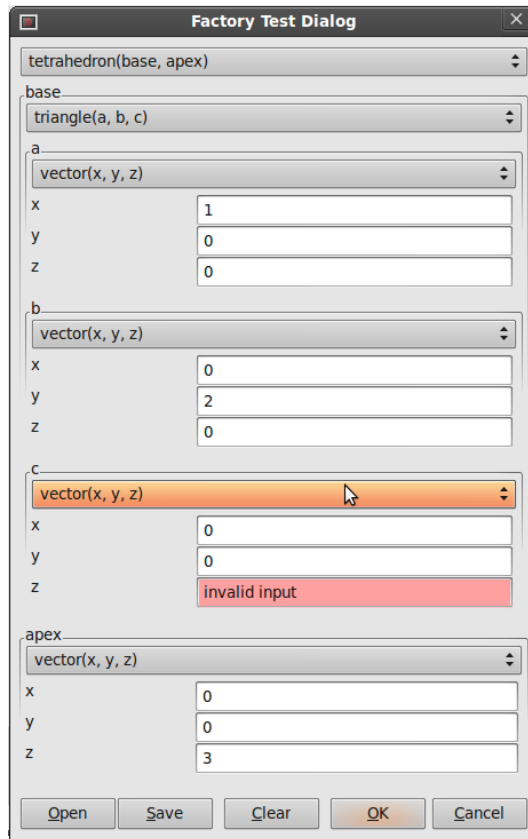


Fig. 2: Example of a GUI created by a factory generated by the Mirror's factory generator

4 Conclusion and future work

The Mirror library is still in development and there are several features that are planned for future releases, including but not limited to the following:

- Refactoring of the existing facilities for registering and reflection of free and class member functions with proper support for function overloads.
- An abstract factory generator that would combine the concrete factories generated by the existing factory generator utility into a single abstract factory.
- An object manipulator generator that would allow generating manipulators working on existing instances of various types. Such manipulators could perform tasks like GUI object inspection, serialization, etc. in a similar manner as the generated factories are used for object instantiation.

- An additional semantic-layer that would allow to tie conceptual data to the reflected program components, allowing the visualization of application's structure and data, generation of more user-friendly GUI/Web interfaces, access for agent oriented systems and automated reasoning.

Acknowledgement

This publication is the result of the project implementation: *Centre of excellence for systems and services of intelligent transport*, ITMS 26220120028 supported by the Research & Development Operational Programme funded by the ERDF.



References

- [1] ABRAHAMS, D. – GURTOVOY, A.: *C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond*. Addison-Wesley Professional, 2004
- [2] ALEXANDRESCU, A.: *Modern C++ Design: Generic Programming and Design Patterns Applied*. Addison-Wesley Professional, 2001.
- [3] BRACHA, G. – UNGAR, D.: *Mirrors: Design Principles for Meta-level Facilities of Object-Oriented Programming Languages*. In Proceedings of the 19th ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages and Applications, pages 85 104, 2003.
- [4] CHIBA, S.: *A Metaobject Protocol for C++*. In Proceedings of the ACM Conference on Object-Oriented Programming Systems, Languages, and Applications, 1995.
- [5] CHOCHLÍK, M.: *Support for object-oriented parallel programs for grids and clusters*. In Proceedings of 2nd International Workshop on Grid Computing for Complex Problems, Bratislava, Slovakia, pages 88-96, 2006.
- [6] CHOCHLÍK, M.: *Generating Object Factory Classes with the Mirror Reflection Library*, Journal of Information, Control and Management System, Vol. 8, No. 2, ISSN 1336-1716, 2010.

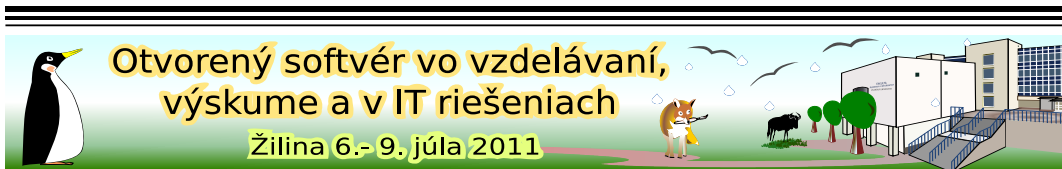
- [7] CHUANG, T-R. – KUO, Y.S. – WANG, C-M.: *Non-intrusive object introspection in C++*. Software-Practice and Experience, issue 32, pages 191-207, 2002.
- [8] DEVADITHYA, T. – CHIU, K. – LU, W.: *C++ Reflection for High Performance Problem Solving Environments*. In Proceedings of the 2007 spring simulation multi-conference - Volume 2, Norfolk, Virginia, USA, pages: 435-440, 2007.
- [9] GAMMA, E. – HELM, R. – JOHNSON, R. – VLISSIDES, J.: *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, ISBN 0-201-63361-2, 1995.
- [10] KIRBY, G.N.C.: *Reflection and Hyper-Programming in Persistent Programming Systems*, Ph.D. Thesis, University of St Andrews, 1992.
- [11] KOVACS, R.: *Creating Dynamic Factories in .NET Using Reflection*. <http://msdn.microsoft.com/en-us/magazine/cc164170.aspx>, downloaded: April, 22nd 2010.
- [12] MADANY, P. W. – ISLAM, N. – KOUGIOURIS, P. – CAMPBELL, R. H.: *Reification and Reflection in C++: An Operating Systems Perspective*. 1992.
- [13] MADINA, D. – STANDISH, R. K.: *A system for reflection in C++*. In Proceedings AUUG 2001: Always on and Everywhere, 2004.
- [14] ROISER, S. – MATO, P.: *The Seal C++ Reflection system*. CERN, Geneva, Switzerland.
- [15] STROUSTRUP, B.: *XTI An Extended Type Information Library*. http://lcgapp.cern.ch/project/architecture/XTI_accu.pdf, downloaded: April, 30rd 2011.
- [16] TANTER, E. – NOYÉ, J. – CAROMEL, D. – COINTE, P.: *Partial Behavioral Reflection: Spatial and Temporal Selection of Reification*. OOPSLA'03, Anaheim, California, USA, 2003.
- [17] TATSUBORI, M. – CHIBA, S.: *Programming Support of Design Patterns with Compile-time Reflection*. In OOPSLA'98 Workshop on Reflective Programming in C++ and Java, Vancouver, Canada, 1998.
- [18] WILLINK, E. D. – MUCHNICK, V. B.: *Weaving a Way Past the C++ One Definition Rule*. In European Conference on Object Oriented Programming, Lisbon, June 14, 1999.
- [19] WILLINK, E. D.: *Preprocessing C++ : Meta-Class Aspects*. In Proceedings of the Eastern European Conference on the Technology of Object Oriented Languages and Systems, TOOLS EE 99, Blagoevgrad, Bulgaria, June 1999.

- [20] *A C++ reflection-based data dictionary*. <http://sourceforge.net/projects/crd/>, May, 15th 2011.
- [21] *Metaclasses and Reflection in C++*. <http://www.vollmann.ch/pubs/meta/meta/meta.html>, April, 28th 2011.
- [22] *Mirror C++ reflection library documentation (C++98 version)*. <http://svn.boost.org/svn/boost/sandbox/mirror/doc/html/mirror.html>, downloaded: May, 15th 2011.
- [23] *Mirror C++ reflection library documentation (C++11 version)*. <http://kifri.fri.uniza.sk/~chochlik/mirror-lib/html/>, downloaded: May, 15th 2011.
- [24] *MPI: A Message-Passing Interface Standard*. MPI Forum, 2003: <http://www.mpi-forum.org/docs/mpi1-report.pdf>, downloaded: April, 23rd 2010.
- [25] Ohloh.net: *compare languages tool*. <http://www.ohloh.net/languages/compare>, May, 10th 2011.
- [26] *OpenC++*. <http://opencxx.sourceforge.net>, downloaded: May, 10th 2011.
- [27] *Programming Language Popularity*. <http://langpop.com/>, downloaded: May, 10th 2011.
- [28] *Property Set Library (PSL)*. <http://sourceforge.net/projects/psl/>. downloaded: May, 15rd 2011.
- [29] *QxOrm library*. <http://sourceforge.net/projects/qxorm/>. downloaded: May, 15rd 2011.
- [30] *Reflection for C++*. <http://www.garret.ru/cppreflection/docs/reflect.html>, downloaded: April, 30th 2011.
- [31] *Static reflection in C++ using minimal repetition*. <http://www.enchantedage.com/cpp-reflection>, downloaded: April, 30rd 2011.
- [32] *Template Reflection Library*. <http://sourceforge.net/projects/trl/>, downloaded: May, 15rd 2011.
- [33] *The Visitor Pattern*. <http://www.oodesign.com/visitor-pattern.html>, downloaded: April, 30nd 2011.

Kontaktná adresa

Matúš CHOCHLÍK (PhD.),

Katedra informatiky FRI ŽU v Žiline, Univerzitná 8215/1,
010 26 Žilina, Matus.Chochlik@fri.uniza.sk



DOI: 10.5300/2011-OSSConf/67

PRINCIPLES OF DSM APPLIED IN EDUCATION AND RESEARCH WITH UML .FRI TOOL

JANECH, Ján, (SK); BAČA, Tomáš (SK)

Abstract. *We would like to introduce how principles of DSM can be used in education and research. According to traditional understanding of DSM it is an approach that increases productivity of software development by bringing in higher level of abstraction. What we are doing is usage of one DSM tool for teaching several standard visual modeling languages at our university. When doing research DSM allows us to create a specific language and a model in that language that is easier to understand, than a set of mathematical formulas would be. It can be afterwards processed automatically to facilitate experiments. At the University of Žilina we are developing a tool called UML .FRI, which is optimised for mentioned kinds of usage.*

Key words and phrases. *DSM, education, research, UML .FRI, modeling, open source.*

UPLATNENIE PRINCÍPOV DSM VO VYUČOVANÍ A VÝSKUME S NÁSTROJOM UML .FRI

Abstrakt. *V tomto príspevku predstavíme využitie princípov DSM vo vyučovaní a výskume. Tradične je DSM chápané ako prístup, ktorý zvyšuje produktivitu vývoja softvéru vďaka zavedeniu vyššej miery abstrakcie. My sa však snažíme o použitie jedného DSM modelovacieho nástroja na vyučovanie viacerých štandardných vizuálnych modelovacích jazykov. Pri výskume nám DSM umožňuje vytvorenie špecifického jazyka, ktorý popisuje problém jednoduchšie, ako napríklad matematický model. Automatizované spracovanie následne uľahčuje experimentálnu činnosť. Na Žilinskej univerzite preto vyvíjame nástroj UML .FRI, ktorý je optimalizovaný práve na takýto spôsob použitia.*

Kľúčové slová. *DSM, vyučovanie, výskum, UML .FRI, modelovanie, opensource.*

1 Introduction

At the Faculty of Management Science and Informatics of the University of Žilina there is a mandatory subject for all students at masters degree lasting for three semesters. Its goal is to teach teamwork, thus they are grouped to teams, in which they are creating larger projects.

Some of these projects have become generational, because every year a group of new students join them to learn and eventually take over the work of their older colleagues. [5]

At the one of the oldest projects students are developing a CASE¹ tool called UML .FRI². According to the name one can assume that the project focuses on UML³, which used to be true few years ago. The project was started with an idea to create a replacement for expensive proprietary UML tools, which were used at the faculty that time. The UML .FRI has already partially fulfilled the goal by being published as open source under the GPL⁴ and by being actively used in education of several computer science subjects. [2] Section 5 summarizes, where UML .FRI has already been used.

Capabilities of the tool have been improved over the time. It now supports DSM⁵, which makes it easy to add a new visual modeling language into the tool. Section 2 describes known ways of usage of DSM in commercial sphere, where it increases productivity of software development process. However we have found out quite different ways of use for it – two main activities of universities which are education and research.

When doing research, one often has to create some models. However, researcher often has to choose between easy-to-understand graphical model and its written form, which can be processed by computer. If graphical model was easy for computers to read, it would combine good features of both approaches. You will find an example in section 3.

DSM has advantages in education as well. However the UML .FRI tool has to meet several special requirements that would not have been so important if it had been meant for commercial sphere only. The main advantage of DSM in education is in replacement of variety of CASE tools by one so that students do not have to learn how to use every one of them. They can focus just on modeling instead. On the other hand the tool has to support modeling language according to the standard. Further description of this topic is in section 4.

2 Common usage of DSM

DSM is a methodic for various kinds of system development, most often software products. Nowadays it is mostly used in commercial sphere. The advantage of this approach is in increased productivity of development by heightening level of abstraction during modeling. Designer does not have to specify so much technical details and can focus more on the problem itself.

Software development with DSM is separated into two phases. During the first one a “domain expert” is required. That is a person who understand the area – domain – for which the designed system is addressed. His task is to create a definition of DSML⁶, that enables to create as accurate and as understandable model as possible. The definition of the language

¹Computer Aided Software Engineering

²Homepage of the project is <http://umlfri.org/>

³Unified Modeling Language

⁴GNU General Public License, see <http://www.gnu.org/licenses/gpl-2.0.html>

⁵Domain Specific Modeling

⁶Domain Specific Modeling Language

is called metamodel⁷. During the second phase a “normal designer” creates a model of the system in DSML with DSM tool. [8]

The ability to generate most of source codes of final product from model makes DSM to be a real contribution. Under certain circumstances it is also possible to generate up to 100% of codes. To achieve it the domain expert has to create not only metamodel but also domain specific code generator that will use component library. However this approach is profitable for large projects only, because the creation of a new metamodel, specific code generator and library would cost a lot of effort and time. [3, 4, 9]

3 DSM in research

Similar to software development, in research it is a common practice to create various models. Model can describe some problem, phenomenon, state of observed system or conditions the solution must satisfy. Many forms of models are used in practice, however the most common two categories are graphical and textual forms. [6]

Graphical representation of a model is convenient for being quite easy-to-understand. One can easily imagine how the system would be like in reality. When an experiment is being done with the model one can usually watch its progress and eventually verify its results. Minor problem of this representation is its loss of legibility when the model grows larger. Major problem is difficulty with reading and processing it by computer program. This presents quite a disadvantage nowadays.

On the contrary a model written in textual form is usually quite easy to read by computer. It can also contain a lot of information without much loss of its legibility for humans. It still requires certain effort to understand the model and imagine its features.

If one decided to use DSM in research, roles of domain expert and normal designer would most likely overlap. It would be caused by number of requests for features of DSML, because a researcher would usually satisfy with only very small set of item types. Another cause would be also difference in lifecycle of the metamodel. In software development the phase of metamodeling and designing are quite strictly separated. In research however the metamodel evolves on the fly according to current needs and partial results of experiments, with only little need for backward compatibility.

For an illustration imagine a simulation of traffic in road network, for which the road network is an important input parameter of the simulation. One would be trying to observe some patterns between changes of shape of the road network and changes in behavior of the traffic. With certain abstraction road network can be modeled as a graph composed of edges representing roads and nodes representing crossroads. Edges and properties can have some properties according to needs of the experiment.

Graph by its definition is represented by a set of nodes and a set of edges. An example of such a representation is at Fig. 1. It is written in textual form and is quite easy to process

⁷Metamodel is composed of description of every item that can be used in the language. They can be of various types of elements and connections. Description of an item includes its graphical notation, what types of information it can hold and how it can be joined with other items into larger formations.

```

nodes = {A: (1, 0), B: (0, 1), C: (2, 1), D: (0, 3), E: (2, 3)}
edges = {(A, B): 3, (A, C): 3, (B, C): 4, (B, D): 4, (C, D): 6,
        (C, E): 4, (D, E): 4}

```

Fig. 1: Example of model in textual form

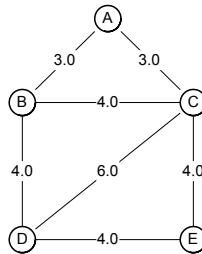


Fig. 2: Example of model in graphical form

by computer program. On the other hand an experimenter is most likely to imagine or rather draw a diagram of the graph similar to the one at Fig. 2. To modify the graph one would make modifications do the diagram and afterwards to the textual form.

A logical assumption is that if one created metamodel for modeling road networks and designed its shape, one would get easy-to-understand model, which is already somehow processed by computer program. Therefore one would only need to find a way how to get it out of the DSM tool so that one can use it for the simulation.

UML .FRI satisfies this requirement with its public API⁸. It is possible to create and use plugins⁹ that extends functionality of the application. One can therefore create one's own routine for automated processing or modification of created model. [1, 7]

4 DSM in education

DSM can be advantageous for education as well. We firstly focus on requirements of school environment on CASE tools. Usually it is not the point to explain how to use a tool. On the contrary teachers are trying to minimize the time needed to introduce it to students and use it only as an aid for teaching.

That is true not only for CASE tools. Since first school years pupils are using specialized software to better understand topic of the subject. They use programs to learn alphabet in gamesome form or to attend an excursion to stars. Later on at high school with specialization already chosen they encounter software for simulation of electrical circuits, with

⁸Application Programming Interface

⁹An extension in form of executable code

CAD¹⁰ systems for modeling of machines, with mathematical solvers, IDEs¹¹ at lessons of programming. . .

Anyway the aforementioned stays true. A teacher tries to use tools only as an aids. With complexity of the tool however extends the time students require to learn how to use it properly. It is therefore interesting idea to search for similar areas and group them into one tool so that students can use it during the whole time at school.

There is another significant advantage of such a solution and that is price. It would be saving if school acquired licenses only for one application and was able to use it in several subjects instead of purchasing licenses for many of them. [2]

UML .FRI was being developed as aid for education from the beginning and is trying for unification of visual modeling CASE tools. At basics of algorithmisation lessons Flow Charts are being used; at object oriented modeling lessons UML class diagrams are being used; at lessons of analysis and design of applications all UML diagrams, BPMN¹² and others are being used. . .

It is achieved by the main principle of DSM tools, which is that the designer of metamodel and developer of the application are two separate people. Thus the tool itself does not support in any modeling language. Instead it is possible to add a new metamodel which enables modeling in any kind of visual language. Important practice in UML .FRI is also that metamodel is completely separated from project (one model). Compared to software development, where every project is created for its own domain, in education the whole class is creating similar projects in common modeling language. [5]

Such an usage of DSM differs from the common commercial one mainly by modeling often in standard languages (UML, E-R¹³, BPMN. . .). These however already have their notation¹⁴ standardized. To avoid confusion of students and to teach them the language properly, it is not possible to comply to graphical abilities of the tool as one would not hesitate to do when creating a new language.

UML .FRI solves this problem by its robust graphical engine. Compared to other DSM tools UML .FRI enables to define not only simple geometrical shapes but also quite complex items presented in standard languages.

There is one more possibility, how to improve education with UML .FRI, although it is not directly related to DSM. As mentioned at the end of section 3, one can extend functionality of application with a plugin. It is therefore possible to create an interactive tutorial, which provides student with guidance and points out on mistakes he has made. Another example would be simulation of designed model (for example of Petri net) which would verify whether the student created correct model or not.

¹⁰Computer Aided Design

¹¹Integrated Development Environment

¹²Business Process Model and Notation

¹³Entity-Relationship

¹⁴Visual representation of items

5 Where UML .FRI has already been deployed

Nowadays UML .FRI is being used mainly as education aid at Faculty of Management Science and Informatics at University of Žilina. It has been used in subjects that does not require full support of UML or are using different modeling languages. More precisely:

- *Basics of Informatics 1,2* – Already replaced subjects, where basics of algorithmisation and structured programming were taught. Algorithms were modeled in Flowcharts.
- *Object Oriented Programming* – Also canceled subject that followed after Basics of Informatics 2. Students were taught basics of OOP and were using UML class diagrams in UML .FRI.
- *Informatics 1,2* – These (and few other) subjects replaced Basics of Informatics and Object Oriented Programming with Object First approach. Students are taught basics of modeling in UML with aid of UML .FRI.

Besides students can and individually are creating models in UML, BPMN, E-R and DFD¹⁵ for their course works, bachelor and master's thesis. Also employees of the faculty are sometimes in role of students and those that attended Oracle Academy last year modeled E-R with UML .FRI. [6]

It has been also used as an aid for experimental research recently. The example from section 3 is real system that is just being developed.

6 Conclusion

Even though the principles of DSM were created for increasing productivity of software development compared to development with GPML¹⁶ tools, their usage can be nowadays found also in education and research. The proof is a tool UML .FRI.

UML .FRI is being developed with a goal to be included in education. It is intended to replace several various tools for modeling UML, E-R, Flowcharts, DFD... Thus a lot is taken into account how metamodels are written. Maximum of attention is paid to notation of items so that students does learn standard languages properly and are not confused when they start working with another tools.

UML .FRI has already been successfully deployed in education of subjects that required simpler shapes of items. UML class diagram was used in already canceled subject Object oriented programming. Besides it Flowcharts were used in also canceled Basics of Informatics 1, 2. After being replaced by new subjects Informatics 1, 2 that are being taught with methodic Objects first, the education continued with aid of UML .FRI. Students and employees of the faculty use it also elsewhere than just on these subjects.

There are also ways on how to use UML .FRI in research. Thanks to its ability to support modeling in arbitrary language it is an ideal candidate for creating simulation models.

¹⁵Data Flow Diagram

¹⁶General Purpose Modeling Language

References

- [1] Bača, T.: Zásuvné moduly v nástroji UML .FRI. In: *Otvorený softvér vo vede, vzdelávaní, výskume a v IT riešeniach*. vol. 1, pp. 9–14. Vedecko-technická spoločnosť pri Žilinskej Univerzite (2009), ISBN: 978-80-89276-16-5
- [2] Bača, T., Juríček, M., Odlevák, P.: Uml .fri – case tool development. *Journal of Information, Control and Management Systems* 6(1), 145–150 (2008)
- [3] Cook, S., Jones, G., Kent, S., Wills, A.C.: *Domain-Specific Development with Visual Studio DSL Tools*. Pearson Education, Inc., Boston (2007), ISBN: 978-0-321-39820-8
- [4] DSM Forum: DSM case studies and examples, <http://www.dsmforum.org/cases.html>
- [5] Janech, J.: UML/DSM nástroj UML .FRI. In: *Otvorený softvér vo vede, vzdelávaní, výskume a v IT riešeniach*. vol. 1, pp. 9–14. Vedecko-technická spoločnosť pri Žilinskej Univerzite (2009), ISBN: 978-80-89276-16-5
- [6] Janech, J., Bača, T.: DSM nástroj UML .FRI. In: Kříž, P. (ed.) *Objekty 2009*. pp. 84–93. Univerzita Hradec Králové, Fakulta informatiky a managementu (2009), ISBN: 978-80-7435-009-2
- [7] Janech, J., Bača, T.: Architecture of plugin management system in UML .FRI tool. In: Merunka, V., Vajgl, M. (eds.) *Objekty 2010*. pp. 41–52. Katedra informatiky a počítačů, Přírodovědecká fakulta, Ostravská univerzita v Ostrave (2010), ISBN: 978-80-7368-899-8
- [8] Kelly, S., Tolvanen, J.P.: *Domain-Specific Modeling*. John Wiley and Sons, Inc., New Jersey (2008), ISBN: 978-0-470-03666-2
- [9] MetaCase: Nokia case study (2007), http://www.metacase.com/papers/MetaEdit_in_Nokia.pdf

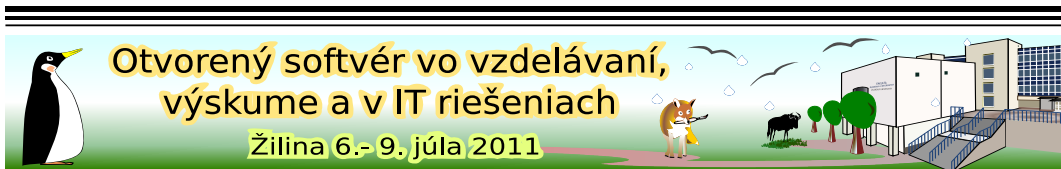
Kontaktná adresa

Ján JANECH (Ing., PhD.),

Katedra softvérových technológií, FRI ŽU v Žiline, Univerzitná 1,
010 01 Žilina, jan.janech@kst.uniza.sk

Tomáš BAČA (Ing.),

Katedra softvérových technológií, FRI ŽU v Žiline, Univerzitná 1,
010 01 Žilina, tomas.baca@kst.uniza.sk



DOI: 10.5300/2011-OSSConf/75

WEKA – EMPIRICAL STUDY ON APPLICATIONS OF DATA MINING TECHNIQUES IN VETERINARY MEDICINE

KANIK, Tomasz, Ing. (PL); DOMINO, Małgorzata, MVDr. (PL)

Abstract. *This work has demonstrated the potential use of data mining techniques to obtain generic criteria for an individual anaesthetic type selection. The main framework of the data mining approach, some of the core modelling methods and the data mining results are briefly described and assessed. We try to focus on the characteristics of medical data and discuss how data miners deal with them. Thus, it is highly expected that data mining methods will find interesting patterns from data sets and be important for veterinary medicine research and practice.*

Key words and phrases. *Data Mining, Veterinary Medical Data, Injectable Anaesthetic Regimen, Classification Rules, Decision Trees.*

WEKA – EMPIRICKÁ ŠTÚDIA O APLIKÁCII TECHNIK DOLOVANIA DÁT VO VETERINÁRNEJ MEDICÍNE

Abstrakt. *V práci je demonštrované možné využitie techník hĺbkovej analýzy dát na vytvorenie generických kritérií pre individuálny výber typu anestézie. Sú opísané princípy hĺbkovej analýzy dát, ďalej hlavné modelovacie metódy a získané výsledky, spolu s ich vyhodnotením. Dôraz je kladený na charakteristiky medicínskych dát a rozbor metód na získanie informácií pomocou techník hĺbkovej analýzy. Dá sa očakávať, že tieto techniky dokážu vyextrahovať zaujímavé súvislosti z dát a môžu byť významným prínosom pre veterinárny výskum a prax.*

Kľúčové slová. *Hĺbková analýza dát, klasifikačné pravidlá, veterinárne dáta, injekčná anestézia, rozhodovacie stromy.*

1 Introduction

Choice of general anaesthesia regimen is a common problem for veterinarians. The decision to use a particular drug and dose should be adjusted to the patient's general condition, type of treatment and its duration. Obtaining detailed medical history data and conducting additional tests (including blood tests) is sometimes difficult on field research. Then the doctor must rely on the poor information from the interview, physical examination and general cardiovascular examination shape prior to surgery. Properly selected type of anaesthesia (in situ – an infusion) allows safe, painless surgery and reduces the risk of postoperative complications.

Considerably easier for the veterinarian in treatment planning on field conditions, even more lifesaving perform treatment in an emergency, would be to create a decision tree for the selection of anaesthetic regimen. Numerous literature data often relate to the situation experimentally. Provide detailed data on the mechanism of action and effects of the active substance on the body. However, data is often difficult to interpret or references directly to the clinical case. The veterinarian, anaesthetizing patient in an emergency, must take a quick decision here and now. The decision suited to the current status of the patient. With a reference tool to facilitate the selection and continuation of general anaesthesia on the parameters of the overall research, veterinarian could be easier, better and safer carried out treatments on field conditions.

The question is: whether it is possible to construct a decision tree for regimen of anaesthesia choose? The answer may bring Data Mining, a relatively new field of the applied computer science and mathematics.

What is a Data Mining

“Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner. In many cases, the statistical methods are not new and data mining software products may even share the same underlying algorithms used in statistical software sold by the same vendor. For example, regression calculations may be performed with identical code. On the other hand some data mining products also contain algorithms less traditionally a part of statistical analysis software and methods e.g. neural networks” [1].

Data Mining, like any mathematical tool, requires adequate knowledge to effectively use and exploit its potential. Unfortunately, theoretical knowledge alone is not helpful in real problems solving, especially when analysing a large amount of data. Here you can easily use a computer, and software that lets you easily and intuitively calculate and verify the hypothesis. In this article we used open source software WEKA and additional plug-in *Forecast Tool*.

Now it only remains to formulate the data-mining's problem and solve it with available mathematical and computing tools.

What is the WEKA?

WEKA is short name of the Waikato Environment for Knowledge Analysis too. It is available as Java source code at Waikata University page [2]. It is a full, industrial-strength implementation of essential data mining techniques. It includes illustrative code and working implementations of machine learning methods too. It offers clean, spare implementations of the simplest techniques, designed to aid understanding of the mechanisms involved. It also provides a work-bench that includes full, working, state-of-the-art implementations of many popular learning schemes that can be used for practical data mining or for research. Finally, it contains a framework, in the form of a Java class library that supports applications that use embedded machine learning and even the implementation of new learning schemes.

2 Data Mining Problems

The basic problem is the choice of aesthetic regimen based solely on the general examination data available at the time of patient transfer to surgery. Another important aspect is to identify relationships between the applied regimens of general anaesthesia, the scheme of anaesthesia's maintain and patient's monitored parameters. On that base, we need to create decision trees that allow the identification of cases in which the selected patterns may be used. The last problem is the possibility of prediction patient's parameters based on real-time parameters monitoring.

3 Available Medical Data

Two types of data are available in medical veterinary domain. The one is a data-set acquired by medical experts, which is collected for a special research topic [3]. These data have the following characteristics: (1) The number of records is small. (2) The number of attributes for each record is smaller, compared with the number of records. (3) There is no attributes with missing values. We refer to this type of data as P-dataset (prospective data). The analysis of those data is called prospective analysis in veterinary because data collection is triggered by the generated hypothesis. Statistical analysis was usually applied to these data-sets [4].

The other type is a data-set retrieved from experiments with general anaesthesia described by medical veterinary literature. These data is stored in a file with a specific research purpose. Usually, these data include laboratory tests, although researchers in medical informatics are discussing how to store medical image, and physical examinations as electronic patient records [5]. We refer to this type of data as R-dataset (retrospective data). These data from literature have the following characteristics: (1) The number of records is large. (2) The number of attributes for each record is large (3) Many missing values will be observed. The analysis of those data is called retrospective analysis in veterinary because data will be

analysed after data collection. Those data will lose any good features which prospective data holds and even statistical techniques do not perform well.

During research basic data from general examination were collected. It was the temperature inside the body **RT** (*Rectal Temperature*), *Heart Rate* **HR**, *Respiratory Rate* **RR**, and the examination of glucose concentration in the blood cells **BS** (*Blood Sugar*). These are the basic parameters monitored during induction and maintenance of general anaesthesia, which reflecting its efficacy and safety. Information about the patient's condition before anaesthesia help classifying the animal for surgery, selecting proper anaesthesia regimen, and provides a base for patient's monitoring during surgery. A significant decrease in heart rate, respiratory rate and temperature is evidence of weakening the patient's vital signs, caused by excessively deep anaesthesia (with a large dose of the drug), regimen mismatched or by a pathological body's condition (deteriorating general condition). Unchanged maintenance of the anaesthesia regimen, in this case, may lead to further deterioration of patient's general condition or even decease. Thus, surgical or pharmacological action is required.

The significant increase in heart rate and respiratory rate, indicates the body's defensive reaction – the mobilization of forces, as a response to pain (too shallow anaesthesia) or changing conditions, e.g. hemodynamic (tearing of blood vessels). In this case anaesthesia regimen need to be revised and medical activities (increasing the flow of fluid in the drip, verifying reasons for increasing values of the parameters) leading to patient's stabilization have to be done too. Extremely high values of monitored parameters indicate persistence of the process, which also constitutes a threat to life.

For the safety of general anaesthesia, it is important that the monitored parameters fluctuated within normal limits. The objective of the veterinary doctor leading anaesthesia is a selection of such a regimen (for anaesthesia induction and maintain) that allow to have an influence on patient's parameters and general condition.

4 WEKA in action

The P-dataset and the R-dataset submitted for analysis were stored in spreadsheets, which were then exported to a CVS format and supplemented with additional descriptions of the attributes, thus create a ARFF's file standard. This conversion was not necessary, because the data could be directly imported into the program, but created data files (flat-data file) will be used for further research. Environment WEKA can also work with databases and internet sources of data, which is a big improvement in distributed computing.

After reading the data by the WEKA environment we have available pre-processing panel. It is used for initial preparation of data by the application of different filters, classification, conversion and other transformations on records. In order to prepare the data from P- and R-dataset following methods were used: *Add* – an instance filter that adds a new attribute to the dataset; *Discretize* – an instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes; *AddExpression* – an instance filter that creates a new attribute to a mathematical expression applying the existing attributes; *MathExpression* –

which modify numeric attributes according to a given expression; *Times series delta* – an instance filter that assumes instances form time-series data and replaces attribute values in the current instance with the difference between the current value and the equivalent attribute value of some previous (or future) instance. In addition, R-dataset used *ReplaceMissingValues*, which replaces all missing values for nominal and numeric attributes in a dataset with the modes and means from the training data.

Next step is to determine which groups of cases can be considered when analysing the data. For this purposes we move to the clustering panel and choose one from among the possible algorithms. Base on they specification, algorithm Cobweb have been chosen. It implementing the *Classit* clustering algorithms, which separate patients into categories (groups), for which we will find a relationship in their monitored parameters. Clustering algorithm result can be read in clear text form, or visualized as a graph or chart. The result (categories) is applied on dataset by pre-processing panel. We use *AddCluster* for separation the numerical values into a specific compartments. This function work as a filter that adds a new nominal attribute representing the cluster assigned to the instance each specified clustering algorithm.

Decision tree is created by Classification panel, which is the next step. Here, among the many available algorithms, we choose the one that best captures the nature of the expected classification result. For solving the problem of decision-making we considered between algorithms *LMT* (class for generating a pruned or unpruned C4.5 Decision Tree), *M5P* classifier for building ‘logistic model trees’, which is classification tree with logistic regression functions at the leaves) and *J48* (class for generating a pruned or unpruned C4.5 Decision Tree). The result of applying the chosen classifier will be tested according to the options that are set by clicking in the Test options box. There are four test modes: *use training set* – the classifier is evaluated on how well it predicts the class of the instances it was trained on; *supplied test set* – the classifier is evaluated on how well it predicts the class of a set of instances loaded from a file; *cross-validation* – the classifier is evaluated by cross-validation, using the number of folds that is entered in the Folds text field; *percentage split* – the classifier is evaluated on how well it predicts a certain percentage of the data which is held out for testing. The amount of data held out depends on the value entered in the percent’s field. During the tests, the algorithm J48 with training set mode obtain better results in the classification correctness than any of other algorithms. That why we choose it for a future data analysis. The WEKA calculations show that the choice of anaesthesia can be classified with very high probability. It is good base for the next steep – anaesthesia maintenance chooses.

This step will create a decision tree, which will help in choosing a anaesthesia’s maintain regimen. For this purpose, the data contained in the P-dataset was adjusted in the Preprocessing panel, to indicated nominal value of the deviation of the patient vital signs in relation to known and diagnosed cases of R-dataset. Functions used in pre-processing were AddExpression, MathExpression and Discretize. For the purposes of statistical analysis standard deviation was used and the result is formulated in the text form, as M – for maintenance

according to the scheme: metadomidyna, ketamine, butorphanol; X – for maintenance of the schema: xylazine, ketamine, butorphanol; NON – if clear decision can not be take. Depending on comparison of the patient's revised parameters with the standard parameters and diagnosed cases from R-dataset, a positive result (M or X) or negative (NON) were recorded in the additional column of P-dataset table. The created table was used to prepare a decision tree for choosing an anaesthesia's maintain regimen.

The last step was to try to predict changes in patient parameters base on information collected during the surgery. For this purposes WEKA addition Forecast was used. Time series analysis is the process of using statistical techniques to model and explain a time-dependent series of data points, in our case patient's parameters. Time series forecasting is the process of using a model to generate predictions (forecasts) for future events based on known past events. Time series data has a natural temporal ordering – this differs from typical data mining/machine learning applications where each data point is an independent example of the concept to be learned, and the ordering of data points within a data set does not matter [6].

Prediction model for analysis need frequent measurement of patient's parameters, that why we used data from R-dataset. In three cases, used to analysis, the entry of selected parameters followed with a frequency of 5 minutes and the time and manner of conducting general anaesthesia corresponded to the data in P-dataset. On the panel Forecast as analysed parameters HR, RT, RR, BS were selected and as an identifier the time of measurement was chosen. Confidence intervals were set on 95% and Perform evaluation had been chosen, which tells the system to perform an evaluation of the forecaster using the training data. In Base learner panel algorithm SMOReg was chosen, which implements Alex Smola and Bernhard Scholkopf's sequential minimal optimization algorithm for training a support vector regression model. This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default [7]. The results of the algorithm can be seen in a text form, together with additional information, or as a graph of parameter changes in time.

5 Results

In this research we have explored a decision tree as an anaesthesia regimen model. We have conducted experiments on J48 Classifier using WEKA tool to achieve the accuracy of performance in attaining the detection of the anomalies. A decision-tree model is built by analysing training data (220 reference cases in R-dataset) and the model is used to classify unseen data (44 cases from P-dataset). Classification on test set is necessary to get a reliable value of accuracy of the generated model. J48 generates decision trees, the nodes of which evaluate the existence or significance of individual features, e.g., heart rate, respiration rate. Following a path from the root to the leaves of the tree, a sequence of such tests is performed resulting in a decision about the appropriate class of anaesthesia. The decision trees are constructed in a top-down fashion by choosing the most appropriate attribute each time.

An information-theoretic measure is used to evaluate features, which provides an indication of the “classification power” of each feature. Once a feature was chosen, the training data was divided into subsets, corresponding to different values of the selected feature, and the process is repeated for each subset, until a large proportion of the instances in each subset belong to a single class.

Program as result show text output with detailed accuracy by class. We try to describe it using Tab. 2 as an example. The first two columns are the TP Rate (True Positive Rate) and the FP Rate (False Positive Rate). For the first level where ‘anaesthesia=MDT’ TP Rate is the ratio of MDT cases predicted correctly to the total of positive cases. There were 0 instances correctly predicted as MDT, and 20 instances in all that were MDT. So the TP Rate = $16/20 = 0.8$. The FP Rate is then the ratio MDT cases of incorrectly predicted as XYL cases to the total of MDT cases. 0 MDT instances were predicted as XYL and there were 20 MDT cases in all. So the FP Rate is $0/20 = 0$. The next two columns are terms related to information retrieval theory. When one is conducting a search for relevant documents, it is often not possible to get to the relevant documents easily or directly. In many cases, a search will yield lots results many of which will be irrelevant. Under these circumstances, it is often impractical to get all results at once but only a portion of them at a time. In such cases, the terms recall and precision are important to consider. Recall is the ratio of relevant documents found in the search result to the total of all relevant documents. Thus, higher recall values imply that relevant documents are returned more quickly. A recall of 30% at 10% means that 30% of the relevant documents were found with only 10% of the results examined. Precision is the proportion of relevant documents in the results returned. Thus a precision of 0.75 means that 75% of the returned documents were relevant. Lastly, the F-measure is a way of combining recall and precision scores into a single measure of performance. The formula for it is: $(2 * recall * precision) / (recall + precision)$. In the context we described, these measures are important elements to consider when studying the performance of a certain model in the domain of informational search and retrieval. In our example, such measures are not very applicable, because the recall in this case just corresponds to the TP Rate, as we are always looking at 100% of test dataset and precision is just the proportion of MDT and XYL cases in the test sample [8].

The final segment of the text output produced by the program is an entropy evaluation measures. We try to describe it using Tab. 1 as an example. The first line shows the number and percentage of cases that were correctly predicted. The second line shows the number and percentage of cases that the classifier predicted incorrectly. The third line shows the kappa statistic, which measures the agreement of predictions with the actual class. This statistic is not very informative as it can have low values even when there are high levels of agreement as in the case above. In general, Kappa statistics are only appropriate for testing whether agreement exceeds chance levels, i.e. that predictions and actual classes are correlated. Since classifiers are designed and intended to be correct in their predictions, the Kappa statistic is not very helpful. It will usually find that predictions and actual classes are correlated and even a weak classifier will tend to show a correlation between the two.

6 Summary

The empirical results show that general anaesthesia choice model (No. 1) have 90.91% of correctly classified instances on a tested P-dataset. Average specificity is 0.109 (FP-rate) and sensitivity (TP-rate) is 0.909. That indicate that decision tree works well with real test data. The results also show that testing time and training time of the classifiers are slightly good. Detailed results of algorithm evaluation are shown in Tab. 1, detailed accuracy by model class is in Tab. 2, the resulting tree is illustrated in Fig. 1.

The second result show that general anaesthesia maintenance's choice model (No. 2) have about 97.73% of correctly classified instances on a tested P-dataset. Average specificity is 0.008 (FP-rate) and sensitivity (TP-rate) is 0.977. That indicate that decision tree works well with real test data, same as it was in previous model result. Detailed results of algorithm evaluation are shown in Tab. 1, detailed accuracy by model class is in Tab. 3, the resulting tree is illustrated in Fig. 2.

The final calculation performed in the WEKA was an attempt to verify the prediction ability of patient's parameters in the next 10 minutes. Comparison with the real, measured data show a maximum deviation of RT at 1%, HR at 9%, RR of 8% BS at 12%. Sample results are shown in Fig. 3.

7 Conclusion

The goals obtained by WEKA seems to be correct and may be used for further study in this field. High correlation between WEKA date base and proper selection of anaesthetic regimen indicates proper decision tree construction and its usefulness in practice. As stated in this report our working hypothesis need to be verified by larger number of cases and all single anomalies have to be excluded. This would help to improve the decision algorithm. A patterns of maintaining general anaesthesia in terms of patient's parameters abnormalities need to be evaluated as well and some of negative leaves of decision tree will have to be exchange by different aesthetic drugs doses to maintain a surgery level of anaesthesia.

It seems to be very interesting to use a software as a prognostic algorithm in medicine. Unfortunately it requires larger number of dates to make correction to the model [9]. This goal would be archived by using particular medical devices.

Data mining applications in veterinary medicine can have tremendous potential and usefulness. However, the success of this hinges on the availability of clean clinical data. In this respect, it is critical that the veterinarians consider how data can be better captured, stored, prepared, and mined. Possible future directions include the standardization of clinical vocabulary and the sharing of data across institutions to enhance the benefits of veterinary data mining applications [10].

Tab. 1: Summary of evaluation on test set

Result	Model 1	Model 2
Correctly Classified Instances	40 (90.91%)	43 (97.73%)
Incorrectly Classified Instances	4 (9.09%)	1 (2.27%)
Kappa statistic	0.8136	0.9652
Mean absolute error	0.0909	0.0121
Root mean squared error	0.2132	0.0778
Relative absolute error	26.91 %	4.56 %
Root relative squared error	52.40 %	21.57 %
Coverage of cases (0.95 level)	100 %	100 %
Mean rel. region size (0.95 level)	45.46 %	21.36 %
Total Number of Instances	44	44

Tab. 2: Detailed accuracy by class for first tree model

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0	0	0	0	0	0	NON
1	0.200	0.857	1	0.923	0.950	XYL
0.800	0	1	0.800	0.889	0.950	MDT
0.909	0.109	0.922	0.909	0.908	0.950	Weighted Avg.

Tab. 3: Detailed accuracy by class for second tree model

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.947	0	1	0.947	0.973	0.998	NON
1	0	1	1	1	1	X
1	0.030	0.917	1	0.957	0.997	M
0.977	0.008	0.979	0.977	0.977	0.998	Weighted Avg.

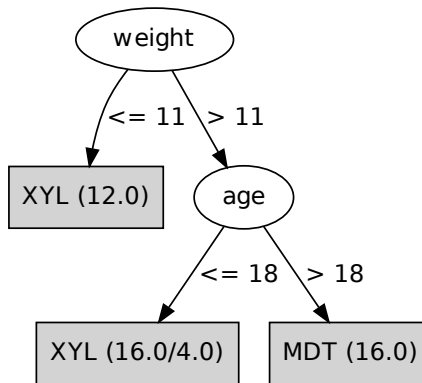


Fig. 1: The decision tree for a general anesthesia regimen

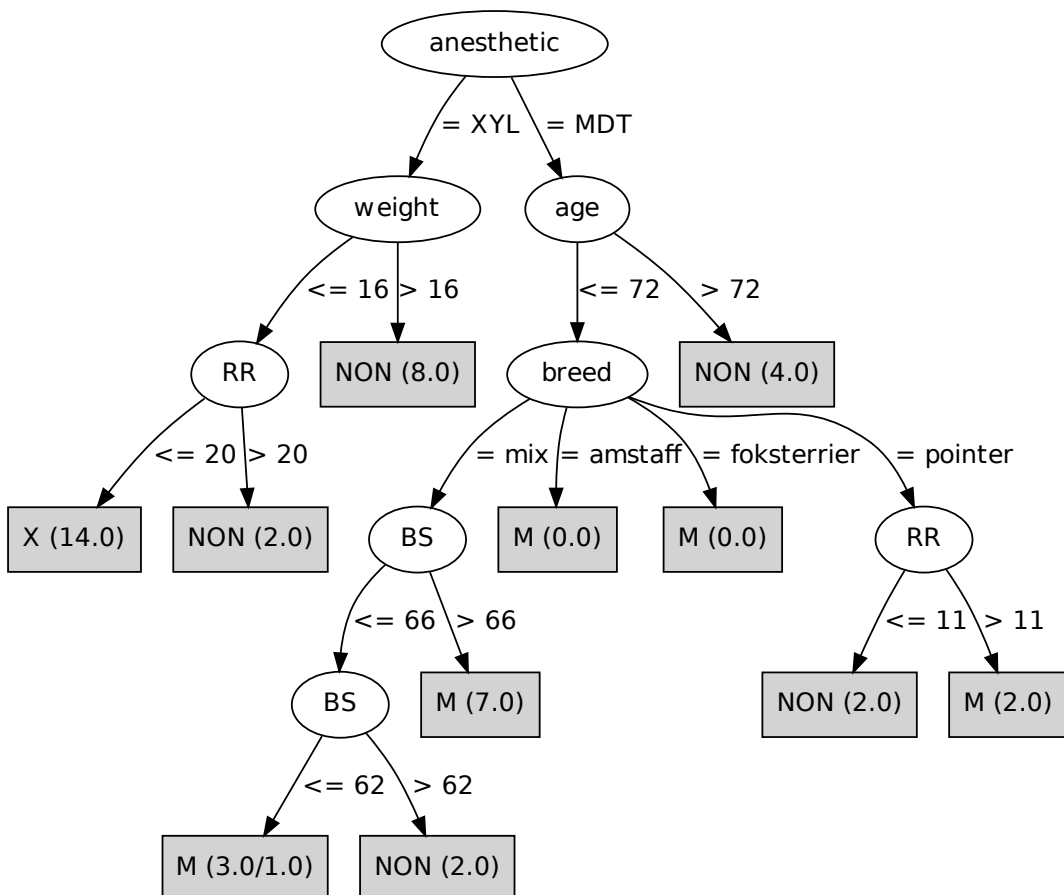


Fig. 2: The decision tree for a maintaining general anaesthesia regimen

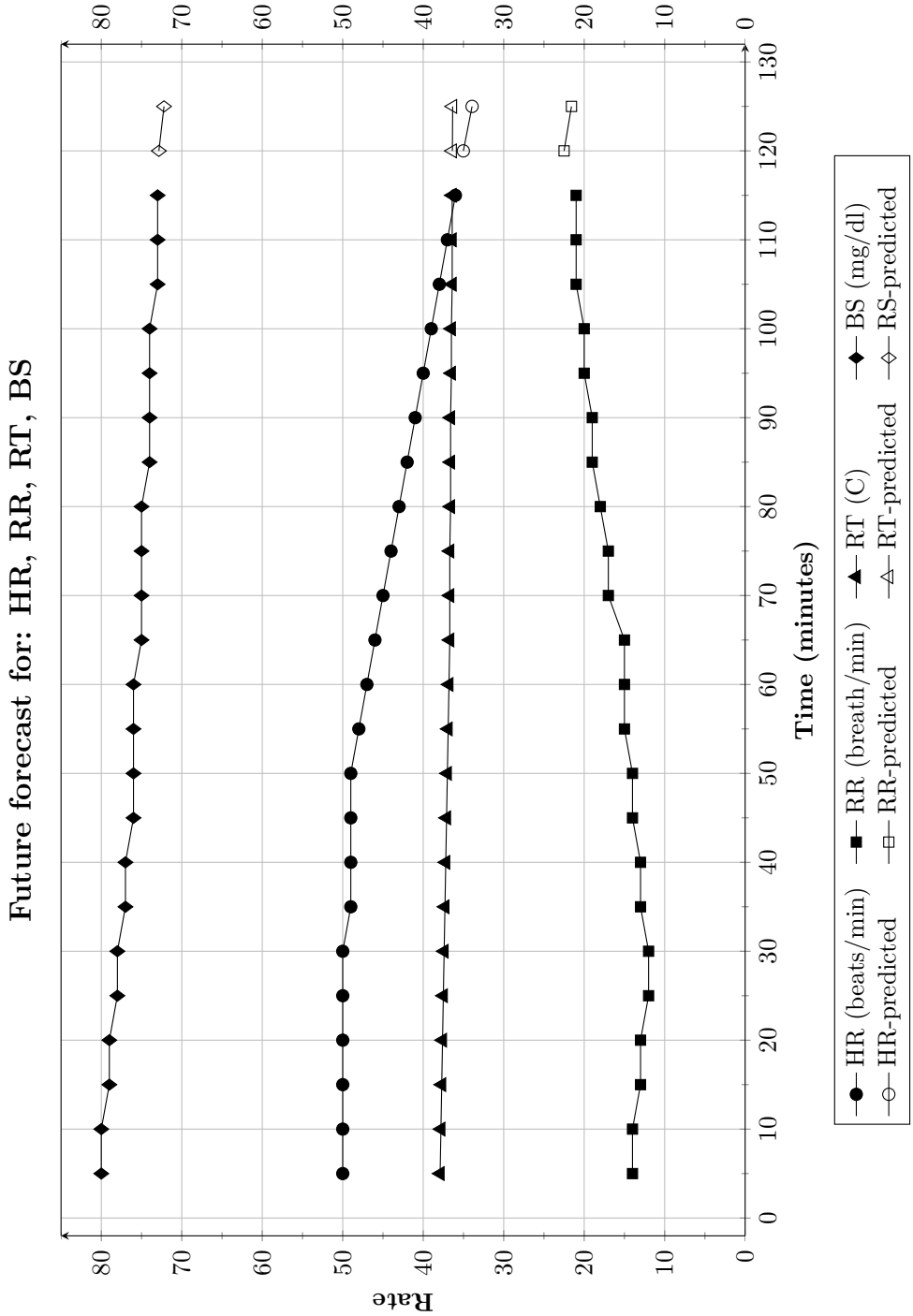


Fig. 3: Patient's parameters prediction in the next 10 minutes

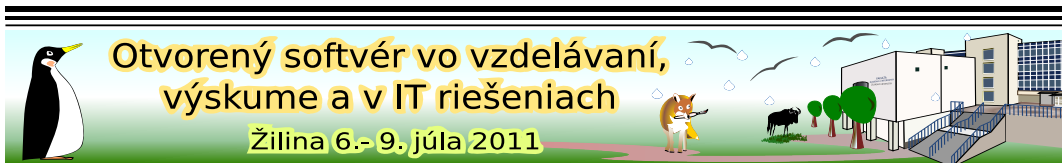
References

- [1] HALL, M. – EIBE, F. – GEOFFREY, H. – PFAHRINGER, B.: *The WEKA Data Mining Software: An Update*. New York : ACM, 2009. doi:10.1145/1656274.1656278
- [2] Machine Learning Group at University of Waikato: *WEKA*.
<http://www.cs.waikato.ac.nz/ml/weka>; accessed May 30, 2011.
- [3] DOMINO, M.: *Porównanie bezpieczeństwa i kosztów dwóch modeli infuzyjnego znieczulenia ogólnego do kastracji psów i kotów*. Warsaw: Veterinary Journal, Volume 20, Issue 164, 2010.
- [4] ALTMAN, D.: *Practical Statistics for Medical Research*. Chapman & Hall/CRC, 1991. ISBN 978-0-412-27630-9.
- [5] BEMMEL, J. H. – MUSEN, M. A. (editors): *Handbook of Medical Informatics*. Springer-Verlag, Heidelberg, 1997. No. of pages: 621+XL. ISBN 3-540-63351-0. doi:10.1002/(SICI)1097-0258(19980630)17:12<1416::AID-SIM883>3.0.CO;2-M
- [6] Pentaho Community: *Time Series Analysis and Forecasting with Weka*.
<http://wiki.pentaho.com/>; accessed May 30, 2011.
- [7] FARQUAD, A. H. – RAVI, V. – RAJU, S. B.: *Support vector regression based hybrid rule extraction methods for forecasting*. Expert Systems with Applications: An International Journal, Volume 37, Issue 8, August, 2010. doi:10.1016/j.eswa.2010.02.055
- [8] Machine Learning Group at University of Waikato: *Weka Data Miner*.
<http://wekadocs.com/>; accessed May 30, 2011.
- [9] Fieselmann, J. – Hendryx, M. – Helms, C. – Wakefield, D.: *Respiratory rate predicts cardiopulmonary arrest for internal medicine inpatients*. Journal of General Internal Medicine, 1993. doi:10.1007/BF02600071
- [10] PAYTON F. C.: *Data mining in health care applications*. IGI Publishing Hershey, PA, USA, 2003. ISBN 1-59140-051-1.

Contact addresses:

Tomasz KANIK (Eng.),
kanikt@gmail.com

Małgorzata DOMINO (VMD),
Zakład Rozrodu Zwierząt, Andrologii i Biotechnologii,
Katedra Chorób Dużych Zwierząt z Kliniką,
Wydział Medycyny Weterynaryjnej,
Szkoła Główna Gospodarstwa Wiejskiego,
ul. Nowoursynowska 100, 02 797 Warszawa,
malgorzata_domino@sggw.pl



DOI: 10.5300/2011-OSSConf/87

PROGNÓZOVANIE V PROSTREDÍ R

PANČÍKOVÁ, Lucia, (SK); ZÁBOVSKÁ, Katarína, (SK)

Abstrakt. V článku sa zaoberáme jednou z ponúkaných možností softvérového riešenia štatistického modelovania. Je ňou „jednoducho R“, prostredie určené pre štatistické výpočty a vytváranie grafov. Príspevok pojednáva o možnostiach aplikovania ekonometrických a štatistických metód, o možnostiach predikcie hodnôt ekonomických časových radov.

Kľúčové slová. Voľný softvér, prostredie R, krátkodobý časový rad, predikcie.

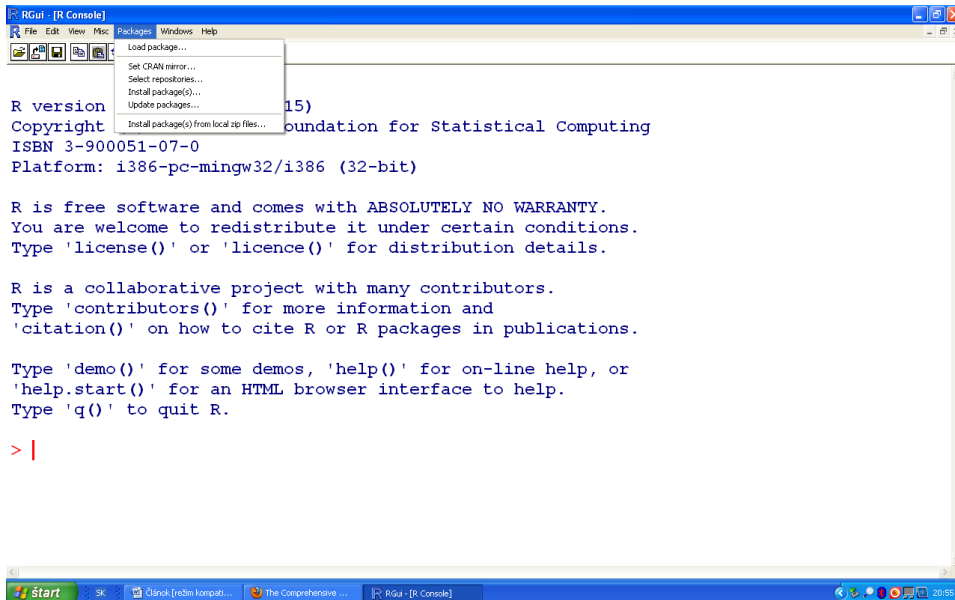
FORECASTING IN R ENVIRONMENT

Abstract. This article deals with one of offered potentialities software solution to statistical modelling. It is simple R, environment for statistical calculations and graphics. The paper refers on econometric and statistical modelling applications possibilities and predictions of economic time series values.

Key words and phrases. Open source, R environment, short time series, predictions.

1 Úvod

Programátori z Univerzity v Aucklande na Novom Zélande Ross Ihaka a Robert Gentleman pomenovali nové softvérové prostredie a objektovo orientovaný programovací jazyk jednoducho R. Aby sa v nás prebudila zvedavosť? Čo sa môže skrývať za tak „banálnym“ názvom a „nezaujímavou“ vstupnou obrazovkou (pozri obrázok 1)? Z nášho pohľadu jednoznačne sloboda. „Každý má možnosť vidieť, študovať a zmeniť zdrojové kódy, opravovať ich chyby a aktívne sa podieľať na vývoji napr. formou pripomienok v elektronických konferenciách a hlásení o chybách (bug reports) pre vývojárov“ (Kaukič [9]). Aj zdanlivú náročnosť práce v tomto prostredí potlačí radostný pocit, že máme príležitosť nesúhlasit' s obsahom funkcie, ktorú vytvoril niekto iný, zaoberajúci sa zhodnou alebo príbuznou problematikou, a naopak úžasný pocit vlastnej jedinečnosti nad vytvorením vlastnej funkcie. K tvorcom jazyka, inšpirovaného jazykom S, v komerčnej verzii softvéru S Plus, sa pridali ďalší „zapálení“ ľudia. Vytvoril sa tím spolupracovníkov nazývaný „the Core Team“. Neustále sú vytvárané a dopĺňané ďalšie verzie tohto prostredia, nové balíky a knižnice.

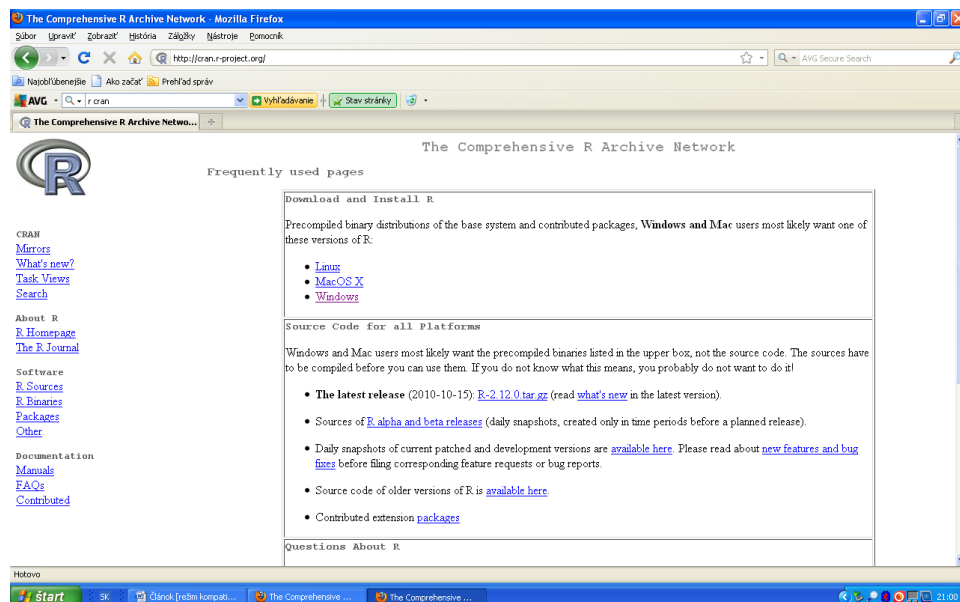


Obr. 1: Prostredie R

2 Prognostické aplikácie

V súčasnosti je vyvinutá verzia 2.12.1. Obrázok č. 2 zobrazuje vstupnú ponuku na stránke <http://cran.r-project.org/> alebo jej českej verzii www.r-project.cz, kde si užívateľ dokáže stiahnuť nielen samotný softvér, a to či už pod Windowsom alebo Linuxom, ale rovnako i balíky a manuály, uľahčujúce prácu začiatočníkom i pokročilejším. Stránka ponúka množstvo ďalších informácií, dotýkajúcich sa práce s týmto „Open Source“. Popri širokej škále ponúknutých funkcií pre zložité matematické operácie, štatistické i grafické výstupy, treba vyzdvihnúť reálnu možnosť optimalizácie a komplexnosti riešenia zvolenej problematiky. Tieto skutočnosti sú podporené práve výberom celého radu vstavaných funkcií a zároveň možnosťou vytvorenia si vlastnej, „šitej na mieru“, vhodnej pre konkrétnu aplikáciu.

Analýza, modelovanie a prognózovanie ekonomických dát predstavuje širokú oblasť aplikovanej štatistiky. Naša pozornosť je sústredená najmä na modelovanie vzťahov mikro a makroekonomických veličín a predikcie ich krátkodobých časových radov. Jedným zo špecifik krátkodobých radov je predpoklad prítomnosti sezónnej zložky, resp. otázka identifikácie štatisticky významnej alebo niekoľkých významných periód. Táto skutočnosť je jedným z určujúcich faktorov výberu predikčnej metódy. Medzi prostriedky identifikácie sezónnosti sa zaraďujú tak grafické metódy, ako i napríklad výpočet výberových autokorelačných koeficientov, periodogram, metódy dekompozície, v neposlednom rade skúsenosť

Obr. 2: Obrázok stránky <http://cran.r-project.org/>

analytika, ktorá ho vopred „navádza“ na významnosť predpokladanej dĺžky periódy daného časového radu.

Zo všeobecne používaných metód prognózovania krátkodobých časových radov v doméne časovej vyberáme nasledujúce:

- Regresná analýza, kauzálne ekonometrické modely – založené na predpoklade prítomnosti reálneho vzťahu (závislosti) hodnôt ekonomických veličín v čase, opísaného matematickou funkciou. Sezónne modely sú často s „minimálnou“ zložitou modelované pomocou regresných modelov s trendovou zložkou a umelými vysvetľujúcimi premennými, popisujúcimi práve sezónnosť.
- Metódy exponenciálneho modelovania – adaptívne metódy, pričom najväčšia váha je prikladaná časovo najmladším dátam, za použitia premenlivých parametrov pre každé časové obdobie. Pri predpoklade časových radov so sezónnou zložkou sa využíva Holt-Winterovo exponenciálne vyrovnávanie s tromi vyrovnávajúcimi konštantami, reagujúcimi na trendovú a sezónnu zložku časového radu.
- ARIMA (Autoregressive Integrated Moving Average) modely, resp. ich sezónna reprezentácia SARIMA (Seasonal Autoregressive Integrated Moving Average) – založené na predpoklade vnútornej závislosti hodnôt časového radu. Identifikujú mieru závislosti skúmanej veličiny od jej predchádzajúcich hodnôt a od časovej závislosti náhodnej veličiny procesu [3]. Predpokladom výberu a identifikácie konkrétneho modelu je otázka stationarity i sezónnosti.

- (d) ARCH (Autoregressive Conditionally Heteroskedastic), resp. GARCH modely a ich modifikácie – založené na predpoklade premenlivého rozptylu, teda rozptylu hodnôt podmieneného predchádzajúcimi hodnotami časového radu. Táto skupina modelov reaguje na niekoľko podstatných obmedzení pri modelovaní ARIMA modelov, a sice: linearity AR modelov, predpoklad o nemennosti rozptylu náhodnej veličiny a podmienku stacionarity časového radu, nereálnu pre finančné dáta, aktuálnu oblasť ich aplikácie.
- (e) Kalmanova filtrácia, modely stavových premenných – využitie Kalmanových rekurzívnych procedúr na prognózovanie časových radov zaťažených sezónnosťou.

Modelovanie časových radov vo frekvenčnej doméne využíva prostriedky spektrálnej analýzy. Časový rad je v tomto prípade popísaný súčtom sínusových a kosínusových kriviek s rôznymi amplitúdami a frekvenciami.

R ponúka možnosť kombinácie prostriedkov identifikácie sezónnosti tak v doméne časovej, ako i frekvenčnej. Kombinácia prostriedkov modelovania časových radov v jednotlivých fázach – identifikácii, kvantifikácii, verifikácii i prognózy vytvára prostredie na vysokokvalitné riešenia. Nestretávame sa s obmedzeniami v podobe chýbajúcich balíkov alebo chybných funkcií, pretože užívateľ má priestor na vytvorenie vlastných, alebo modifikácie existujúcich funkcií. Ak hovoríme o predikciách krátkodobých ekonomických radov alebo finančných vysokofrekvenčných dát, samotné „R-ko“ ponúka vo svojich vstavaných balíkoch množstvo funkcií vychádzajúcich z teoretických predpokladov pomenovaných metód. Zdrojové kódy funkcií, naprogramované používateľmi, sú k dispozícii (verejné) ostatným. Nevýhodou môže byť skutočnosť, že pri riešení určitej úlohy sa stretne s rôznymi funkciami, pretože každý tvorca môže mať odlišný pohľad na to, čo má byť výstupom. V tomto prípade treba ale povedať na obranu „R-ka“, že práve poznanie jednotlivých zdrojových kódov, ako i programovacieho jazyka umožňuje modifikáciu, prípadne naprogramovanie úplne iného postupu. Nasledujúci obrázok zobrazuje časť zdrojového kódu, vytvoreného z dôvodu potreby zmeniť výpočet hraníc významnosti autokorelačných koeficientov, ako ukážka vytvorenia vlastnej funkcie:

```
Bartlett <- function(data,max.lag=NULL)
{
num=length(data)
if (is.null(max.lag)) max.lag=ceiling(10+sqrt(num))
if (max.lag > (num-1))
  stop("Number of lags exceeds number of observations")
bar.for=array(NA,max.lag)
acf.data=acf(kvant.rad,max.lag, plot=FALSE)\$acf[-1]
acf2=acf.data\^{}2;
for(i in 1:max.lag){
  bar.for[i]=((1+2*sum(acf2[1:i-1]))/length(data)) }
result=bar.for;
```

```
return(result) }
#ACF
grafACF=function(data, max.lag=NULL, values=FALSE)
{
  num=length(data)
  if (is.null(max.lag)) max.lag=ceiling(10+sqrt(num))
  if (max.lag > (num-1))
    stop("Number of lags exceeds number of observations")
  higher=2*sqrt(Bartlett(data,max.lag));
  LAG=1:max.lag/frequency(data)
  under=-higher;
  acf.data=acf(data,max.lag, plot=FALSE)\$acf[-1]
  maxim=range(acf.data)
  plot(LAG,
  type="n",ylim=c(-max(maxim),max(maxim)),xlim=c(1,max.lag),
  xlab="Posun (K)",ylab="ACF")
  abline(h=0,lty=1,col=1)
  lines(higher,lty=2,col=4)
  lines(-higher,lty=2,col=4)
  lines(LAG*frequency(data),acf.data,type="h")
  if(values){
  result{\textless}-list(acf=round(acf.data,3));
  return(round(acf.data,3)) }
  }
grafACF(new.rad,30)
```

Zdrojový kód vybranej funkcie (Zdroj: [6])

Pôvodné hranice, obsiahnuté v „R-ku“, nezodpovedali výpočtu podľa Bartletta, uvádzaného napríklad v [3], na základe vzťahu pre výpočet rozptylu autokorelačného koeficientu.

Pri tomto teste platí, že ak vypočítaná hodnota štatistiky je väčšia ako kritická hodnota Studentovho rozdelenia na zvolenej hladine významnosti a so stanoveným počtom stupňov voľnosti, zamietame nulovú hypotézu o nevýznamnosti daného autokorelačného koeficientu. Následne sú dané hranice zakreslené do korelogramu.

Takýmto spôsobom, za predpokladu znalosti programovacieho jazyka R, prípadne iného programovacieho jazyka, si používateľ dokáže vytvoriť vlastnú funkciu. Rovnako nie je problém optimalizácia vybranej časti procesu modelovania (napr. verifikácia) alebo predikcie časových radov ekonomických, finančných dát. Programovacie prostredie R je plne kompatibilné s inými softvérmi, spomeňme napríklad SPSS alebo Statisticu. Vďaka celosvetovému rozšíreniu nielen vrámci akademickej obce sa stalo významným prostriedkom riešenia špecifických problémov. Obsahuje najnovšie štatistické metódy, ktoré mnohokrát

v komerčných softvéroch zabudované nie sú, alebo sú ich nadstavbou. Vysokokvalitné grafické možnosti i už spomínané výhody tohto prostredia rozširujú škálu využitia daného štatistického softvéru. Nemenej dôležitým faktorom jeho spoznania a využívania je určite literatúra, odvolávajúca sa na aplikácie práve v R-ku, treba povedať, že najmä zahraničná. Z tohto dôvodu pracujeme so študentmi na vytváraní manuálov pre potreby vyučovaných predmetov i potreby výskumných úloh v oblasti štatistického prognózovania.

3 Záver

Každému výberu softvéru predchádza fáza porovnávania, hľadania potrebných balíčkov a funkcií. Zároveň so sebou prináša výhody i nevýhody pre jednotlivé aplikácie, vyplývajúce z ponúkaných možností vlastného spracovania. Jednoznačným prínosom zoznámenia sa a využívania nového softvérového prostredia je ale hľadanie kreatívneho riešenia, „uchope- nie príležitosti byť prvý“, nositeľom čoho by mali byť práve učitelia a študenti vysokých škôl. Cieľom tohto príspevku bolo aspoň čiastočne odpovedať na otázku: Prečo použiť pri štatistickom prognózovaní R?, a to bez kritiky alebo zatracovania širokej škály vysokokvalitných komerčných štatistických softvérov.

PodĎakovanie

Tento príspevok vznikol s podporou projektu VEGA 1/0667/10 „Soft techniky v modelovaní a predikcii ekonomických a finančných časových radov“.

Literatúra

- [1] CRAWLEY, J., M.: *The R Book*. Wiley, 2007. ISBN: 978-0-470-51024-7
- [2] KALINA, M. – BACIGÁL, T. – SCHIESSLOVÁ, A.: *Základy pravdepodobnosti a matematickej štatistiky*. Bratislava: STU, 2010. ISBN 978-80-227-3273-4
- [3] MARČEK, D. – MARČEK, M.: *Analýza, modelovanie a prognózovanie časových radov s aplikáciami v ekonomike*. Žilina: EDIS – vydavateľstvo ŽU, 2001. ISBN 80-7100-870-2
- [4] MARČEK, M. – PANČÍKOVÁ, L. – MARČEK, D.: *Ekonometria a soft computing*. Žilina: EDIS – vydavateľstvo ŽU, 2008. ISBN 978-80-8070-746-0
- [5] PARADIS, E.: *R for Beginners*. 2005
- [6] POVAŽANEC, P.: *Optimalizácia softvérového riešenia prognózovania časových radov ekonomických dát*. Diplomová práca. Žilina: Fakulta riadenia a informatiky ŽU, 2010

- [7] <http://www.r-project.org/>
- [8] <http://www.r-project.cz/>
- [9] KAUKIČ, M.: *Možnosti a perspektívy využitia open source softvéru vo výučbe a výskume na vysokých školách*, Aplimat 2007, Proceedings, Part IV: Open Source Software in Research and Education, Bratislava, 2007
- [10] <http://www.revolutionanalytics.com/why-revolution-r/what-is-open-source-r.php>
- [11] <http://zoonek2.free.fr/UNIX/48R/all.html>
- [12] http://cms.jcmf.cz/archiv_svoc/2009/www/prace/4/Lanik.pdf

Kontaktná adresa

PANČÍKOVÁ, Lucia, (Ing, PhD.),

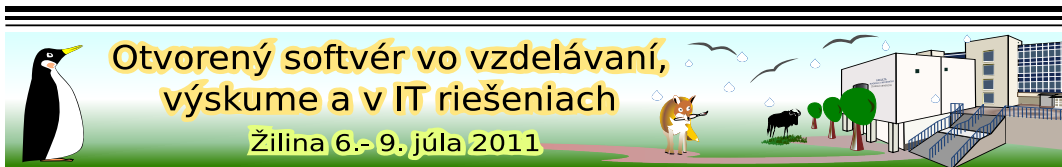
Katedra makro a mikroekonomiky, FRI ŽU v Žiline,

Univerzitná 8215/1, 010 26 Žilina, lucia.pancikova@fri.uniza.sk

ZÁBOVSKÁ, Katarína, (Ing, PhD.),

Katedra makro a mikroekonomiky, FRI ŽU v Žiline,

Univerzitná 8215/1, 010 26 Žilina, katarina.zabovska@fri.uniza.sk



DOI: 10.5300/2011-OSSConf/95

PROGRAM GAMBIT VE VÝUCE PŘEDMĚTU ROZHODOVÁNÍ V RIZIKU A NEJISTOTĚ

STRÍŽ, Pavel (CZ)

Abstrakt. V tomto příspěvku si představíme Open Source Software pojmenovaný Gambit, <http://www.gambit-project.org/>, který je využíván ve výuce v předmětu Manažerské rozhodování v riziku a nejistotě na Fakultě managementu a ekonomiky Univerzity Tomáše Bati ve Zlíně. Autor poukáže na ukázky z teorie her i z rozhodovacích stromů.

Klíčová slova. Gambit, teorie her, rozhodovací strom, hry proti přírodě.

THE USE OF GAMBIT SOFTWARE IN THE DECISION-MAKING UNDER RISK AND UNCERTAINTY COURSE

Abstract. In this article the author introduces Open Source Software called Gambit, downloadable at <http://www.gambit-project.org/>, which is used in the Managerial Decision-Making Under Risk and Uncertainty course at Faculty of Management and Economics, Tomas Bata University in Zlín, The Czech Republic. The author also points out at several game theory problems as well as a few problems which can be stated and drawn as decision tree situation.

Key words and phrases. Gambit, Game Theory, Decision Tree, Games Against Nature.

1 Nabídka úvodem

Učiním vám nabídku! Zahráli byste si proti autorovi článku kámen-nůžky-papír, když možnosti volí náhodně? A co kdyby volil zarputile kámen častěji (40 %) než nůžky a papír (30 %) a svou strategií by nehodlal změnit?

2 První kroky v Gambitu

Abychom se dokázali racionálně rozhodnout, využijeme program Gambit [1], který umí řešit situace teorie her (první autorova nabídka), i rozhodovací strom (druhá nabídka), včetně zahrnutí pravděpodobnostního hráče, to je role, do které se autor příspěvku sám uvrhl.

Program Gambit se dá stáhnout z oficiálních stránek, www.gambit-project.org, verze z roku 2007 je obdobná verzi 2010, pro Linux, Mac OS X i Microsoft Windows. Pro Windows zvolte verzi z roku 2007, neb binárky nejsou plně připravené u verze 2010 a i při využití Cygwinu nebo MinGW bude čtenář zápasit s grafickým uživatelským prostředím (nutnost doinstalovat wxWidgets) a nemožností užití jednoho z algoritmů na 64bitových strojích. Pokud si tyto přepychové věci odpustíte, stačí po rozbalení (za předpokladu, že máte již nainstalované gcc-c++, make atd.) dát:

```
$ ./configure --disable-enumpoly --disable-gui
$ make
$ make install
```

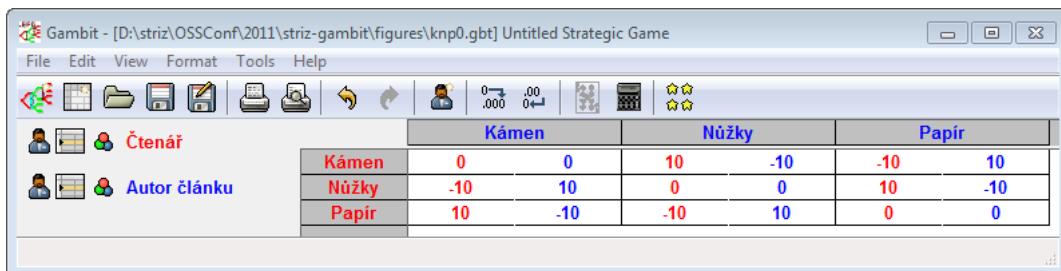
Instalace verze 2007 na nativních Windows je přímočará. Budete mít k dispozici grafické uživatelské rozhraní i algoritmy spustitelné jako exe soubory z příkazové řádky.

Uživatelé pracující na Linuxu či Mac OS X na tom budou z pohledu instalace lépe.

2.1 Řešení problému z teorie her

Vyřešíme si naši první situaci, což je dle klasifikace hra dvou racionálně uvažujících hráčů s nulovým součtem (co první získá, druhý ztrácí a vice versa). Spustíme si Gambit a dáváme File→New→Strategic game. V levé části si přidáme jeden řádek a jeden sloupec díky ikonkám ‚Add a strategy for this player‘. Dvojklikem změníme popisky řádků a sloupců na Kámen, Nůžky a Papír. Nyní si ohodnotíme jednotlivé možnosti her. Když vyhraji dáváme kladnou hodnotu, např. +10, když je remíza tak nula, a když prohrajeme tak −10. Červená pole jsou prvního hráče, modrá pak druhého.

Tabulka (dvojmatice) je symetrická podle hlavní osy, v každém řádku, sloupci i dvojbuňce je součet výher nula. Dostáváme:

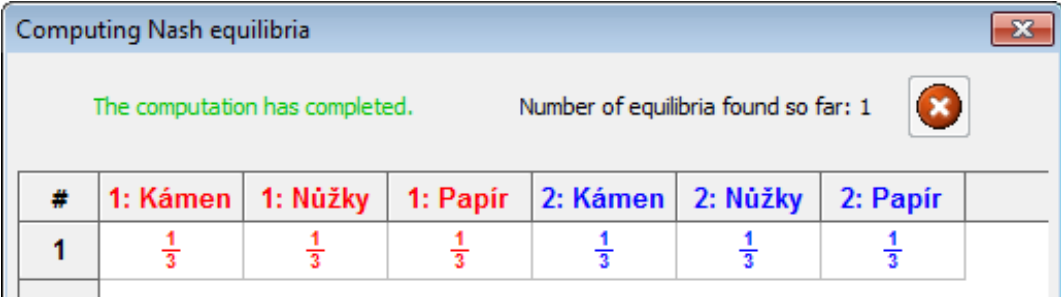


The screenshot shows the Gambit application window with a menu bar (File, Edit, View, Format, Tools, Help) and a toolbar. The main area displays a strategic game matrix for a game with three strategies: Kámen (Rock), Nůžky (Scissors), and Papír (Paper). The matrix is symmetric across the main diagonal. The rows are labeled 'Čtenář' (Reader) and 'Autor článku' (Author of the article). The columns are labeled 'Kámen', 'Nůžky', and 'Papír'. The payoff values are: (Kámen, Kámen) = 0, (Kámen, Nůžky) = 10, (Kámen, Papír) = -10, (Nůžky, Kámen) = -10, (Nůžky, Nůžky) = 0, (Nůžky, Papír) = 10, (Papír, Kámen) = 10, (Papír, Nůžky) = -10, (Papír, Papír) = 0.

	Kámen	Nůžky	Papír
Čtenář	0	10	-10
Autor článku	-10	0	10
	Kámen	Nůžky	Papír
Kámen	0	10	-10
Nůžky	-10	0	10
Papír	10	-10	0

Po vyřešení (druhá ikonka zprava s názvem ‚Compute Nash equilibria of this game‘) s předvolenými možnostmi ‚Compute all Nash equilibria‘ a ‚with Gambit’s recommended method‘ získáváme:

To je řešení naší hry. Každý z hráčů by měl jednotlivé možnosti (řádky, sloupce) volit náhodně a se stejnou pravděpodobností. Po stisknutí ‚OK‘ v levé části pod jmény uvidíme dvakrát ‚Payoff: 0‘, tedy u této hry není vítězů ani poražených v dlouhém období.



#	1: Kámen	1: Nůžky	1: Papír	2: Kámen	2: Nůžky	2: Papír
1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

2.2 Řešení rozhodovacího stromu

Pokud se však naše volby liší a strategie je neměnná, lze hru sehrát jako hru proti přírodě. K tomu budeme potřebovat aparát rozhodovacích stromů, ten Gambit také umí.

V Gambitu si naklikneme File→New→Extensive game. Uvidíme jeden černý uzel. Na ten klikneme pravým tlačítkem myši a volíme ‚Insert move‘, změníme ‚2‘ na ‚3‘ a dáváme ‚OK‘. Objeví se nám tři nové černé uzly. Vybereme horní pravým tlačítkem na myši a volíme ‚Insert move‘. Navolíme ‚Insert move for the chance player‘ v první rolovací šipce. I zde změníme dvojku na trojku a potvrdíme ‚OK‘. Dvojklikem na červenou a zelenou ‚1‘ můžeme nastavit názvy strategií (Label) a i pravděpodobnosti (Probability) u přírody (Chance). Pole volíme dvojklikem levého tlačítka myši, volby poté potvrdíme ‚OK‘.

Pravým tlačítkem myši volíme uzel u Nůžek červeného hráče, ‚Insert move‘. Měníme na: ‚Insert move for the chance player‘, ‚at information set 1 (3 actions, 1 member node)‘, dáváme ‚OK‘. Stejně to zrealizujeme u posledního uzlu červeného hráče.

Poslední fáze je vyplnit cenové štítky, v programu Gambit označeno jako ‚(u)‘ (angl. utility). Budeme jich potřebovat devět v úplně pravé části stromu. Postupně je vybereme dvojklikem a připsujeme výhry a prohry prvního a druhého hráče. Ve finále dostaneme:

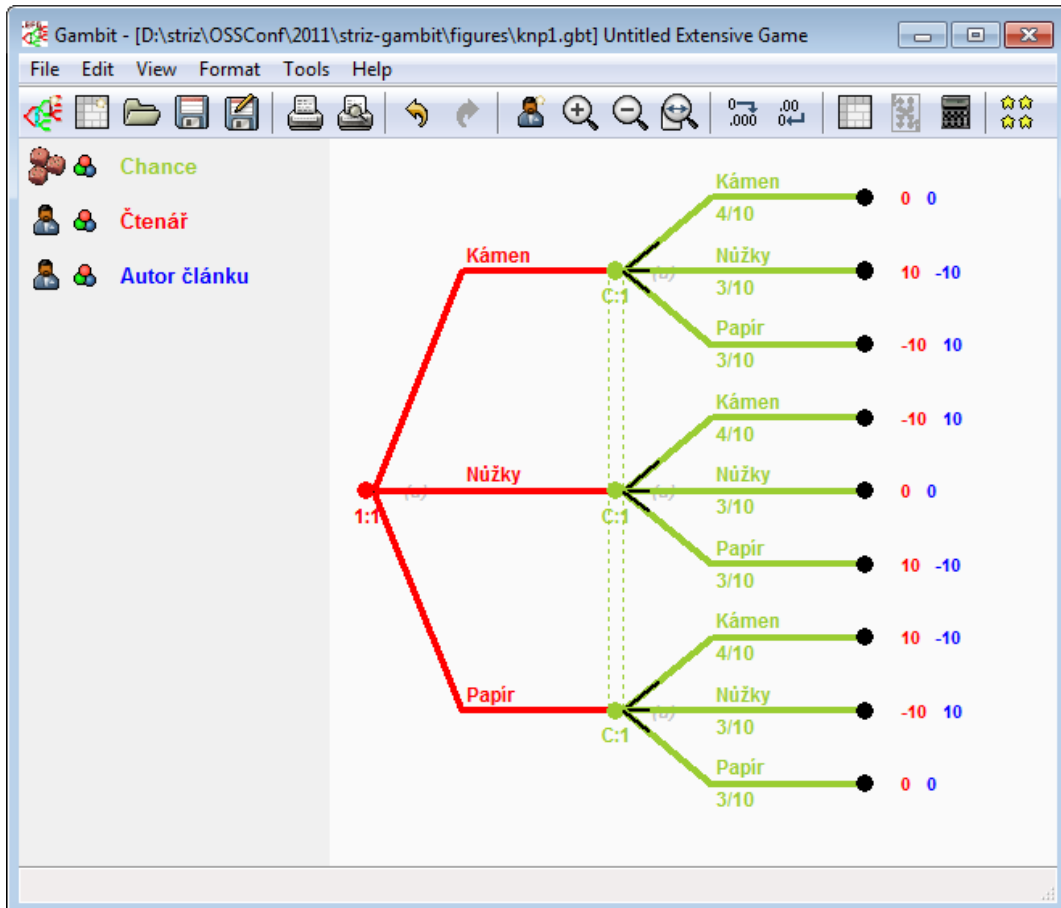
Po vyřešení získáváme optimum, volit vždy Papír. Opět v levé části pod jmény hráčů uvidíme výhru/prohru. Nyní by si první hráč polepšil z nuly na jednotku výhry a druhý hráč (který se de facto hry racionálním způsobem neúčastní) ztrácí jednotku výhry.

3 Pro náročnější čtenáře

3.1 Věžňovo dilema

Pokud je čtenář obeznám s touto disciplínou aplikované matematiky, tak jej známý problém věžňova dilema nepřekvapí. Je to hra s nenulovým součtem a má jednu Nashovu rovnováhu (oba vězni si věří a zapírají) a jedno Paretoovo optimum (vězni si nevěří a druhého podvedou).

Nepřekvapí nás tedy, že mezi 140 ukázkami (podadresář /games/ ve verzi 2007; podadresář /contrib/games/ ve verzi 2010) tento problém nalezneme. Jedná se konkrétně o soubor pd.nfg (angl. Prisoner's Dilemma). Po otevření a vyřešení získáváme:



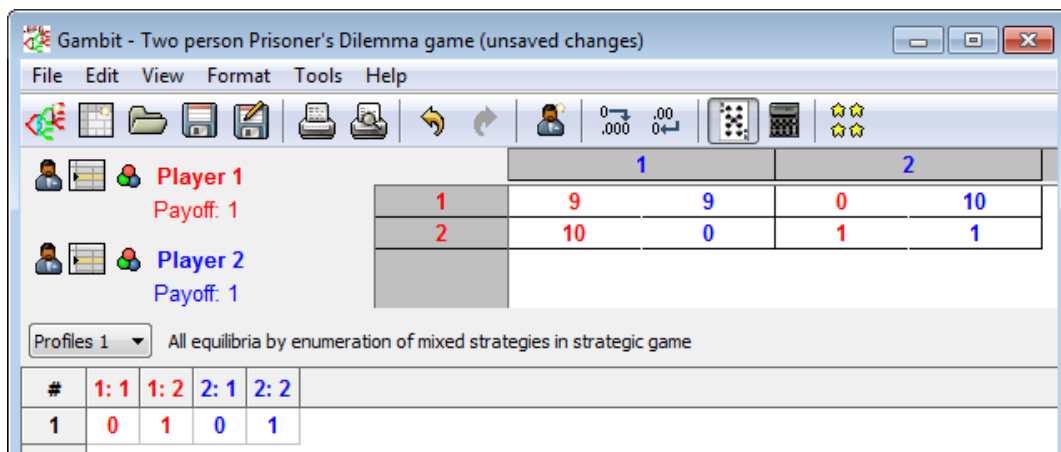
Computing Nash equilibria

The computation has completed. Number of equilibria found so far: 1

#	1: 1	1: 2	1: 3	2: *
1	0	0	1	1

3.2 Gambit z příkazové řádky

Příjemné zjištění je fakt, že lze daný problém vyřešit z příkazové řádky, z libovolného ze tří operačních systémů. Vstupem je textový nešifrovaný soubor `pd.nfg`, který má jasně definovanou strukturu, která je popsána v manuálu.



```
$ cat pd.nfg # Linux a Mac OS X
```

nebo

```
> more pd.nfg & rem Microsoft Windows
```

```
NFG 1 R "Two person Prisoner's Dilemma game" { "Player 1" "Player 2" }
```

```
{ { "1" "2" }
```

```
{ "1" "2" }
```

```
}
```

```
""
```

```
{
```

```
{ "" 9, 9 }
```

```
{ "" 10, 0 }
```

```
{ "" 0, 10 }
```

```
{ "" 1, 1 }
```

```
}
```

```
1 2 3 4
```

Řešení získáme spuštěním jednoho z algoritmů, např. na Linuxu a Mac OS X, z adresáře /contrib/games/:

```
$ gambit-enummixed <pd.nfg
```

Na Microsoft Windows 7 např. takto:

```
C:\Program Files (x86)\Gambit>gambit-enummixed.exe <games/pd.nfg
```

Výstupem získáváme (NE značí v angl. Nash equilibrium):

Compute Nash equilibria by enumerating extreme points
Gambit version 0.2007.12.04, Copyright (C) 2006, The Gambit Project
Enumeration code based on lrslib 4.2b,
Copyright (C) 1995-2005 by David Avis (avis@cs.mcgill.ca)
This is free software, distributed under the GNU GPL

NE,0,1,0,1

3.3 Výhody a nevýhody

Shrneme-li, tak můžeme říci, že

- + Program Gambit pracuje v grafickém režimu i z příkazové řádky.
- + Vstupní soubory `efg` a `nfg` jsou nešifrované, tedy snadno generovatelné.
- + Informační banner u výstupu se dá vypnout parametrem `-q`.
- + V grafickém režimu snadno získáme dominantní řádky a sloupce.
- + Verze 2007 běží na všech verzích Windows, se kterými autor přišel do styku.
- + Gambit obsahuje sadu 140 ukázek, řada z nich je z odborných knih.
- + Dokumentace programu je přehledná.
- + Při uložení výsledku v grafickém režimu se stává součástí `efg` či `nfg` souboru pod tagem `analysis`.

Na druhé straně však:

- Nevýhodou zůstává, že z příkazové řádky nelze získat matici výher hráčů, ty by se musely dohledat (řešení v čisté strategii) či dopočítat (řešení ve smíšené strategii).
- Osoba neznalá teorie her může snadno narazit na situaci, kdy vybraný algoritmus zkolabuje. Pak zkolabuje i program Gambit či daný spustitelný soubor. Není to uživatelsky přívětivě řešené v grafickém režimu pod Windows a v řadě případů je potřeba ukončit proces až přes Správce úloh.
- Verze 2010 zatím neobsahuje podporu pro nativní Windows a nepodporuje 64bitovou platformu pro algoritmus `enumpoly`. Kompilace verze 2010 s grafickou podporou (`wxWidgets`) je pod Cygwinem na Windows 7 64bit téměř nemožná, pro neprogramátora je to nad lidské síly.
- Ačkoliv je to zmíněno v cílech autorů programu, tak Gambit v současnosti nepodporuje kooperativní hry, podporuje jen nekooperativní hry a hry proti přírodě.

4 Nabídka místo Závěru

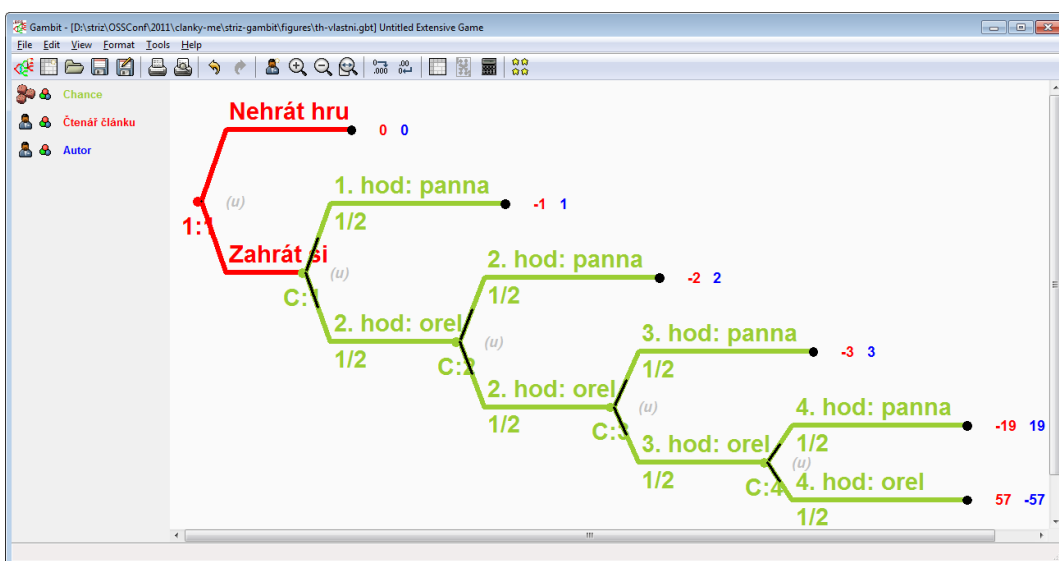
Závěrem opět jedna autorská nabídka ke hře s nulovým součtem, kterou kantor dává svým studentům ve výuce, a která snad příjemně začátečnické partii do rozhodovacích stromů, nám necht' poslouží na ověření si znalostí práce s Gambitem.

Pokud se hry nezúčastníte, nic nevyhraje ani nic nezískáte. Pokud do hry půjdete (červený hráč), jedná se o hod korunovou mincí (z České republiky) dle následujících pravidel. Hra je smyšlená, ve výši výhry a prohry netřeba hledat hlubší smysl.

Nehrajete-li, nic nezískáte a nic neztratíte (0 Kč). Hází se mincí, maximálně čtyřikrát, pokud čtyřikrát po sobě padne orel, autor článku platí účastníku konference 57 Kč. V opačném případě je vypláceno jemu. U prvního hodu při panně korunu a hra končí (při orlu pokračuje), u druhého dvě, u třetí tři koruny a u posledního, je-li panna, celých 19 korun.

Pokud se s autorem článku na OSSConf2011 v Žilině potkáte, máte možnost požadovat výhru, nebo naopak zaplatit autorovi prohru, s odkazem na tento článek. ☺

Zde je formálně zapsaná hra – hodnoty pro 2. hráče jsou postupně 0; 1; 2; 3; 19 a –57, pro prvního hráče jsou s opačným znaménkem:



Abych netrpělivého čtenáře článku nenechal na pochybách, na další straně je řešení.

S teorií her se můžeme setkat v ekonomii, politice, právních sporech [2–4] a ostatně i ve filmu, kdy jeden ze zločinců (vězňovo dilema) si odpykává trest za svého partáka, který ho podrazil, a pak se po propuštění z vězení mstí.

Ovšem nechodme daleko, zájemce necht' shlédne film 'Čistá duše – A Beautiful Mind', který je o Johnu Nashovi, průkopníku v teorii her a nositeli Nobelovy ceny za ekonomii z roku 1994 (vedle Reinharda Seltena a Johna Harsanyi). Klíčový moment filmu a jeho netradiční řešení, které ovlivnilo celou ekonomii, je znám pod termínem bitva pohlaví (angl. Battle of Sexes či Bach or Stravinsky – BoS).

Problematiku rozhodovacích stromů potkáme ve hrách jako jsou go, šachy, poker i bridž (angl. double-dummy problems, částečně lze použít i na single-dummy problems).

Poděkování

Vznik tohoto článku byl částečně podpořen ESF projektem číslo CZ.1.07./2.2.00/15.0104. Za pročetění první chaotické verze rukopisu autor děkuje kolegovi Martinu Juráskovi.


Řešení hry proti autorovi

Krása úlohy zmíněné na předchozí straně spočívá v tom, že při velkém opakování takové hry je autor článku skutečně ztrátový, a to jedna koruna česká na hru. Tu si zaslouží každý čtenář, který by do hry šel, případně student, který rozhodovací strom správně ohodnotil a do hry vstoupil. V Gambitu si navíc můžeme nakliknout jednotlivé uzly, které jsou po vyřešení numericky ohodnoceny.

V reálném životě to však znamená, že autor článku by takovou hru hrál s každým zájemcem jen jednou a je tu sportovní šance, že čtyřikrát po sobě autorovi článku orek nepadne a na svých ztrátových 57 Kč si pěknou řádku her počká, tedy většina účastníků OSSConf2011 by naopak musela autorovi platit, i když jen položky korunové.

Autor článku je vždy ochotný uznat svou prohru v dlouhodobém horizontu a onu korunu každému účastníkovi/studentovi uhradit, ať už z pohledu sportovního nebo kantorského. A kdo nehraje, nevyhraje ani nemá šanci přijít si ke ztrátě!

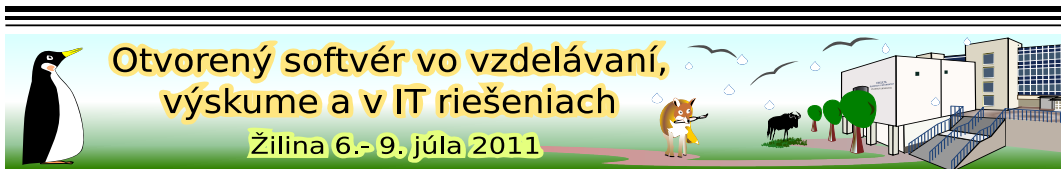
Literatura

- [1] MCKELVEY, R. D. – MCLENNAN, A. M. – TUROCY, T. L. (2010). **Gambit**  *Gambit: Software Tools for Game Theory*, Version 0.2010.09.01. Dostupné na serveru: <http://www.gambit-project.org/>
- [2] CHOBOT, M. – TURNOVCOVÁ, A.: *Modely rozhodovania v konfliktných situáciách a za neurčitosti*, Alfa, Bratislava, 1980.
- [3] MAŇAS, Miroslav. *Teorie her a její aplikace : Učebnice pro studenty VŠE*. 1. vyd. Praha : Státní nakladatelství technické literatury, 1991. 278 s. ISBN 80-03-00358-X.
- [4] PITEL, J. a kol. *Ekonomicko-matematické metody*. Bratislava, Příroda, 1988.

Kontaktní adresa

Pavel STRÍŽ (Ing., Ph.D.),

Tomas Bata University in Zlín, nám. T. G. Masaryka 5555,
760 01 Zlín, Czech Republic, striz@fame.utb.cz



DOI: 10.5300/2011-OSSConf/103

JAZYK GDL NA SPRACOVANIE VEDECKÝCH DÁT

ŠECHNÝ, Martin (SK)

Abstrakt. GNU Data Language (GDL) je jazyk na spracovanie vedeckých dát a zároveň prostredie na spúšťanie programov v tomto jazyku. GDL je slobodný softvér kompatibilný s komerčne licencovaným Interactive Data Language (IDL). GDL je platformovo nezávislé prostredie a využíva iné dostupné nainštalované knižnice a aplikácie. Jazyk GDL umožňuje spracovávať vstupy z klávesnice, dátové súbory, obrázky a dokáže vizualizovať dáta tabuľkami, grafmi, obrázkami. GDL je efektívny pri numerickej analýze dát, vektorovej reprezentácii, použití matematických funkcií a procedúr. Tento nástroj je vhodný pre široké použitie vo vede, výskume, aj ako alternatíva k známym matematickým a vizualizačným nástrojom.

Kľúčové slová. GDL, IDL, vedecké dáta, programovanie, vizualizácia.

GDL LANGUAGE FOR SCIENTIFIC DATA PROCESSING

Abstract. GNU Data Language (GDL) is a language for scientific data processing and also the environment for launching programs in that language. GDL is a free software that is compatible with commercially licensed Interactive Data Language (IDL). GDL is a platform-independent environment and uses other available libraries and applications installed. GDL language enables to process keyboard input, data files, images and can visualize data tables, charts, pictures. GDL is effective in the analysis of numerical data, vector representation, the use of mathematical functions and procedures. This tool is suitable for wide use in science, research, and as an alternative to known mathematical and visualization tools.

Key words and phrases. GDL, IDL, scientific data, programming, visualization.

1 Úvod

GNU Data Language (GDL)¹ je jazyk na spracovanie vedeckých dát a zároveň prostredie (interpreter a inkrementálny prekladač) na spúšťanie programov v tomto jazyku. GDL je

¹GDL 0.9.1 (29.3.2011) <http://gnudatalanguage.sourceforge.net>

slobodný softvér kompatibilný s komerčne licencovaným Interactive Data Language (IDL)². Historický vývoj jazyka začal v roku 1977³.

GDL je platformovo nezávislé prostredie, programované pre systémy spĺňajúce štandard POSIX. Využíva dostupné nainštalované knižnice a aplikácie pre rozšírenie svojich funkcií a možností, alebo ako lepšie optimalizované implementácie štandardných, ale menej efektívnych funkcií. Inštalácia GDL je vďaka tomu modulárna, prispôsobiteľná použitiu. GDL sa stále vyvíja a prispôbuje sa jazyku IDL v najnovšej verzii. GDL je programované v C++ s použitím viacerých ďalších programovacích a matematických jazykov [1], [2].

Jazyk GDL umožňuje spracovávať vstupy z klávesnice, dátové súbory, obrázkové súbory, a to jednotlivo, aj hromadne. Výstup môže byť textový v terminálovom okne, grafický, alebo do súboru. Je možné vizualizovať dáta tabuľkami, grafmi, obrázkami. GDL má široké možnosti matematického spracovania dát – numerická analýza, vektorová reprezentácia, výpočet funkcií a procedúr.

GDL, ako jazyk a zároveň prostredie, je vhodný pre široké použitie vo vede, výskume, aj ako alternatíva k známym matematickým a vizualizačným nástrojom.

2 Inštalácia

Prostredie GDL je programované multiplatformovo, pre systémy spĺňajúce štandard POSIX (napr. UNIX, BSD, GNU/Linux, Mac OS X), po doplnení potrebných súčastí môže fungovať aj v MS Windows.

Sú k dispozícii spustiteľné balíky⁴ pre viaceré distribúcie GNU/Linux, BSD, Mac OS X. Základná inštalácia hotového balíka je jednoduchá a rýchla. Prostredie GDL hneď môžeme vyskúšať v interaktívnom príkazovom riadku.

Ak nevyhovuje ani jeden spustiteľný inštalačný balík, musíme nainštalovať GDL zo zdrojového kódu⁵ po preklade. Pri inštalácii musíme vyhovieť závislostiam. GDL nutne potrebuje knižnice *plplot*, *gsl*, *readline*. Podľa účelu použitia GDL si vyberieme voliteľné knižnice *Xlib*, *Magick++*, *FFTW*, *netCDF*, *HDF*, *GRIB API*, *UDUNITS*, *wxWidgets*, *libproject*, *GSHHS*, *Python*, *CMSVLIB*, *MPFIT*, *Astron lib*, *TeXtoIDL* a iné. Na preklad zdrojového kódu GDL a ručnej inštalácii je potrebné mať dobrú znalosť operačného systému a programovacích jazykov, lebo je potrebné zvoliť správne poradie prekladu a inštalácie knižníc a tie v správnych verziách. Hlavnou nutnou knižnicou je *gsl* (*GNU scientific library*), do ktorej sa pripájajú voliteľné knižnice a spolu s nimi poskytujú funkcie a procedúry volané prekladačom/interpreterom prostredia GDL. Často je potrebné ručne nastavovať viaceré parametre prekladu vzhľadom na použitý operačný systém. Medzi podporné knižnice patria knižnice pre algebru (*blas*, *cblas*, *lapack*) prekladané ako súčasť *gsl*, alebo pripojené zvlášť.

²ITT Visual Information Solutions IDL 8.0 (30.9.2010) <http://www.itvis.com/language/en-us/products/services/idl.aspx>

³Wikipedia: IDL: [http://en.wikipedia.org/wiki/IDL_\(programming_language\)](http://en.wikipedia.org/wiki/IDL_(programming_language))

⁴<http://gnudatalanguage.sourceforge.net> (download)

⁵<http://sourceforge.net/projects/gnudatalanguage/>

Pri väčšom počte knižníc trvá preklad a inštalácia bežne niekoľko hodín a spolu s nastuďovaním dokumentácie ubehne aj niekoľko dní. Inštalácia obsahuje viaceré pripravené testy funkčnosti a optimalizácie prostredia, knižníc, funkcií.

3 Spustenie prostredia GDL

Prostredie GDL sa môže spustiť v terminálovom okne (príkazovom riadku):

```
$ gdl
```

Nainštalované môže byť aj grafické prostredie ako samostatná nadstavba *gdl*, alebo ako modifikovaná aplikácia *gdl*. Dokumentácia ku GDL nie je dostatočne podrobná alebo zrozumiteľná, preto ďalej uvádzam postup pri prvom spustení.

Prostredie GDL potrebuje mať nastavené systémové premenné. Existujú však odlišné inštalácie GDL/IDL, preto sa odporúča nastaviť duplicitné premenné s rôznymi názvami, aby sa vyhovelo rôznym situáciám, napr.:

```
~/ .bashrc (~/.profile):
```

```
export GDL_DIR=/opt/gdl
```

```
export IDL_DIR=/opt/gdl
```

```
export GDL_PATH=/opt/gdl/pro
```

```
export IDL_PATH=/opt/gdl/pro
```

```
export GDL_STARTUP=~/ .gdl/gdlrc
```

```
export IDL_STARTUP=~/ .gdl/idl_startup/pro
```

GDL/IDL hľadá súbor s nastaveniami, ktorý takisto môže mať rôzne názvy:

```
~/ .gdl/gdlrc
```

```
~/ .gdl/idl_startup.pro -> gdlrc
```

V tomto súbore môžu byť direktívy pre beh prostredia GDL, príkazy operačného systému, príkazy jazyka GDL – vykonajú sa pri spustení prostredia. Dôležité je vložiť do premennej *PATH* cestu k funkciám, procedúram, programom, ktoré má prostredie GDL vedieť nájsť podľa mena:

```
!PATH='/opt/gdl/' ; príklad cesty ku knižnici GDL
```

```
!PATH=expand_path('+'+!PATH) ; rozvinúť cestu do stromu
```

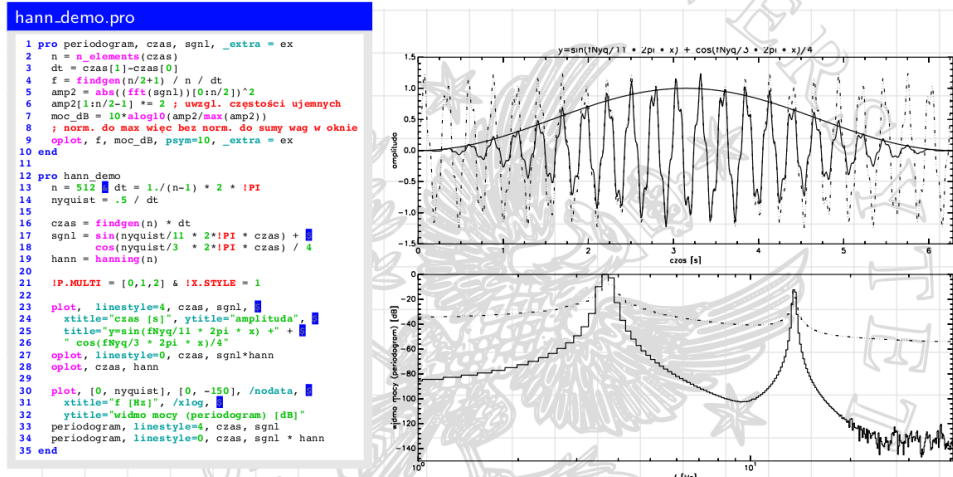
Pri spustení GDL môžeme zmeniť aktuálny priečinok príkazom *cd*, alebo spustiť GDL na takom mieste, kde máme svoje súbory.

Nainštalované prostredie GDL obsahuje zhruba 100 súborov *.pro* so základnými funkciami/procedúrami jazyka. Na toto miesto môžeme nakopírovať ďalšie súbory, alebo pripojiť svoju cestu do premennej *PATH*.

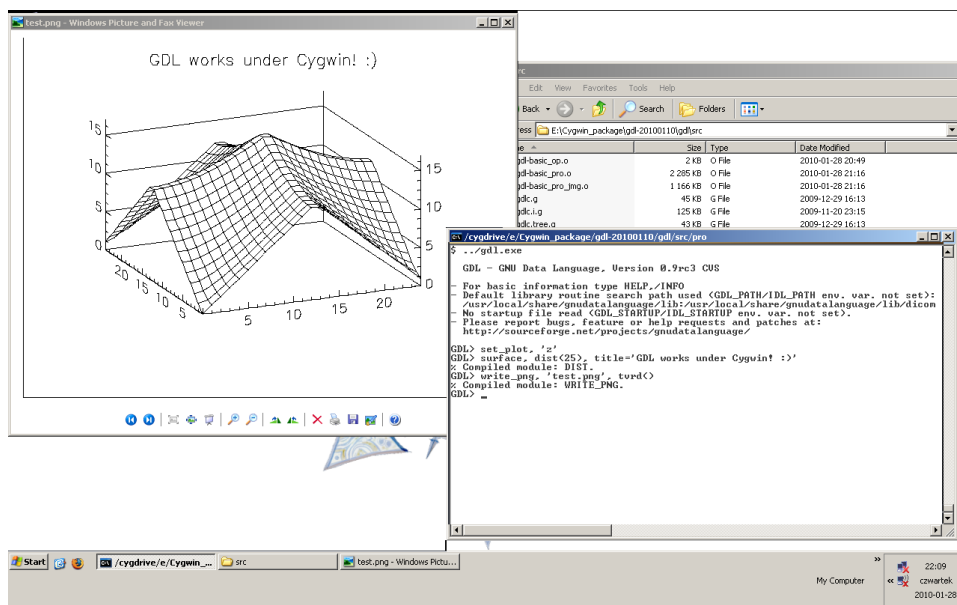
Príkladom je knižnica *Astron lib*⁶ pre astronómov, fyzikov. Podobných knižníc je mnoho. Jazyk GDL ešte nie je tak známy ako starší jazyk IDL, preto sa ľahšie hľadajú knižnice s kľúčovým slovom *IDL*. Autor programu v jazyku GDL/IDL sa môže rozhodnúť, aké

⁶Goddard Space Flight Center IDL Astronomy User's Library (Astron lib) <http://idlastro.gsfc.nasa.gov>

licenčné podmienky požaduje pre svoje zverejnené programy a je výhodné, ak taký program funguje v ľubovoľnom prostredí, teda aj v GDL.



Obr. 1: Ukážka programu v jazyku GDL s grafickým výstupom, upravený z <http://gnudatalanguage.sourceforge.net/screenshots/screenshot06.png>



Obr. 2: GDL funguje v MS Windows + Cygwin, upravený z <http://gnudatalanguage.sourceforge.net/screenshots/screenshot09.png>

4 Jazyk GDL

Program v GDL sa píše ľahko. Jazyk GDL sa podobá čiastočne jazyku FORTRAN, čiastočne jazyku C, využíva sa množstvo výhod podporných knižníc prostredia, hlavne matematické výrazy. V príkazovom riadku je dostupná pomoc, v grafickom prostredí aj základná príručka, príklady, zoznamy príkazov. Stručný návod pre začiatočníkov nájdeme na internete⁷, ako aj obsiahly zoznam zdrojov pre široké použitie jazykov GDL/IDL⁸. Funkcie, procedúry, programy a dokumentácia časom pribúdajú, lebo ich tvorí odborná verejnosť pri svojej práci a obvykle sú sprístupnené všetkým. Podrobná dokumentácia IDL pomôže aj pri práci s GDL [3–5].

Nasleduje stručne ukázaná syntax a zopár základných príkazov z jazyka GDL: Nerozlišujú sa malé a veľké písmená. Príkaz sa ukončuje koncom riadka (nie bodkočiarkou). Príkaz operačného systému začína znakom \$. Systémová premená sa použije s uvádzajúcim znakom !. Komentár začína na riadku znakom ;.

<príkaz>, <parameter> ; komentár

Dátové typy sú automatické, stačí premennú prvý raz použiť, nie je nutné ju deklarovať. Celé číslo (int) má štandardne 2 B, alebo veľkosť podľa operačného systému. Niektoré parametre funkcií alebo procedúr začínajú znakom /.

A=5 ; priradenie hodnoty do premennej A

print, A ; výpis premennej

help, A ; výpis informácie o premennej

B=[1,2,3] ; vektor/pole/matica

print, B[0:1] ; vypíš prvky poľa B s indexom 0 až 1

C=A+5 ; matematická operácia

S='prvy'+'druhy' ; operácia s textovým reťazcom

openr,subor,'in.txt',/get_lun ; otvorenie súboru na čítanie

readf,subor,S ; načítanie súboru do premennej S

print,subor ; výpis súboru

free_lun,subor ; zavrieť súbor

Vektorové/maticové operácie sú vykonávané automaticky, ak je to potrebné, upravia sa rozmery (dimenzie aj počet prvkov) poľa a jeho dátový typ.

Programový súbor má príponu .pro, končí sa príkazom end. Program sa musí najprv preložiť a potom sa môže vykonať:

.compile <program> ; preklad

.run <program> ;(preklad a) vykonanie programu

<program> ; vykonanie programu (procedúry)

Blok je ohraničný kľúčovými slovami begin, end. Podprogram môže byť vo forme procedúry alebo funkcie:

⁷Sven Geier, GDL intro, 2006: <http://www.sgeier.net/tools/GDL-intro.pdf>

⁸IDL resources: <http://www.astro.virginia.edu/class/oconnell/astr511/IDLresources/astro-dept-idlresources.html>

```

    pro program, x ; procedúra
x=x+1
print,x
return

    end

    function funkcia, x ; funkcia
x=x+1
return, x

    end

```

Jazyk umožňuje používať podmienku, vetvenie, skok, cyklus, napr.:

```

    if A eq B then print,S
    if A gt B then begin
print,A
print,B
endif else begin
print,S
endelse

    for I=1,10,1 do print,I
    while A eq B do A=A+1
    repeat A=A+1 until A eq B

```

Spracovanie parametrov programu:

```

    if(n_params(0) eq 0) then print,'Neboli zadane parametre'

```

GDL je veľmi vhodný na spracovanie obrázkov, čo sa využíva hlavne v astronómii, ale napr. aj v medicíne:

```

read_jpeg,'1.jpg',image,true=1 ; načítanie obrázka
ysize=size(image,/dimensions) ; zistenie rozmerov
window,0,xsize=xysize[1],ysize=xysize[2],title=Obrázok ; okno
tvsc1,image,true=1 ; zobrazenie obrázka
obrazok=fltarr(3,300,300) ; deklarácia obrázka v reálnom
poli (RGB, x, y)

```

Knižnica obsahuje veľa funkcií/procedúr pre transformáciu obrázkov, podobne ako kvalitné grafické aplikácie. Tieto operácie sa dajú algoritmizovať a sú vykonávané efektívne.

Ďalej existujú knižnice pre prácu s rôznymi formátmi súborov, s databázou, na paralelné vykonávanie programov, simulácie. Dáta možno vypísať do terminálu na obrazovke, zapísať do súboru, vyrobiť z nich tabuľku, alebo zobraziť ako obrázok pomocou rôznych vizualizačných nástrojov, napr. Plplot.

5 Aplikácie jazyka GDL

Veľká skupina používateľov jazyka GDL/IDL pracuje vo fyzike, špeciálne v astronómii. Okrem knižnice *Astron lib*⁹ je možné použiť GDL ako API k špecializovaným lokálnym alebo webovým aplikáciám, napr. na ovládanie astronomického programu Aladin¹⁰, prístup k virtuálnym observatóriám¹¹. Jazyk GDL/IDL sa používa na spracovanie a analýzu obrázkov z astronomických pozorovaní, napr. snímky Slnka, vzdialených vesmírnych objektov, nočnej oblohy.

GRID je distribuovaný výpočtový systém, pozostávajúci zo samostatných počítačov, alebo klastrov a navonok sa tvári ako virtuálny superpočítač, ktorý môže poskytnúť používateľovi ohromné výpočtové zdroje. GDL je univerzálne prostredie, dá sa nainštalovať aj na slabší jedno-processorový počítač, aj na virtuálny počítač. GDL v GRIDE vie obslužiť silne paralelizovateľné úlohy a vykonať veľmi efektívne vektorové/maticové operácie definované v matematických knižniciach. Tieto možnosti robia z GDL výkonný simulačný nástroj [6], [7].

Jazyk GDL sa používa vo viacerých vedných oblastiach aj napr. na spektrálnu analýzu, fotometriu, redukovanie obrázkov, fitovanie.

6 Záver

GNU Data Language (GDL) je jazyk na spracovanie vedeckých dát. Je vhodným slobodným softvérovým nástrojom pre vedu a výskum, kompatibilný s komerčne licencovaným Interactive Data Language (IDL).

GDL možno porovnať s matematickými nástrojmi¹², napr. *Octave*, *Scilab*, *Pylab*, alebo komerčne licencovanými *Matlab*, *Mathematica*, ďalej so simulačnými nástrojmi, napr. *GENESIS*. Veľký prínos má GDL s knižnicou *Astron lib* pre fyzikov, astronómov, kde môžu analyzovať a spracovávať obrázky vesmírnych objektov, čo sa dá porovnať napr. s *AIPS*, *IRAF*.¹³

Pri použití GDL v GRIDE máme v rukách nástroj so silným výpočtovým výkonom a poradíme si aj veľmi veľkým objemom vstupných dát.

⁹Goddard Space Flight Center IDL Astronomy User's Library (*Astron lib*) <http://idlastro.gsfc.nasa.gov>

¹⁰Aladin <http://aladin.u-strasbg.fr/aladin.gml>

¹¹Virtual Observatory lib <http://www.ctio.noao.edu/~chrism/V0lib>

¹²Matematický softvér <http://www.personal.kent.edu/~rmuhamma/Mathematics/mathsoftware.html>

¹³Astronomický softvér <http://tdc-www.harvard.edu/astro.software.html>

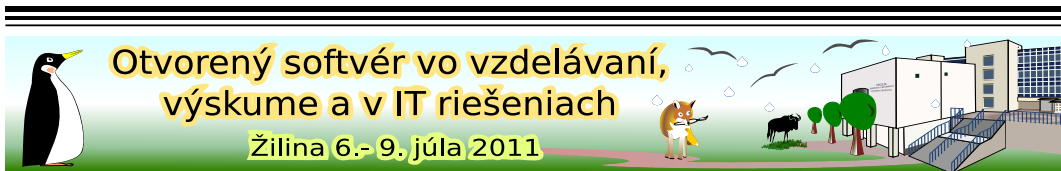
Literatúra

- [1] ARABAS, S: GDL – GNU Data Language, Free and Open Source Developers’ European Meeting 2011, <http://www.igf.fuw.edu.pl/~slayoo/gdl-fosdem-2011-talk.pdf>
- [2] COULAIS, A. – MERMET, T.: Status of GDL – GNU Data Language, Astronomical Data Analysis Software and Systems XIX, ASP Conference Series, Vol. XXX, 2009, arXiv:1101.0679v1 [astro-ph.IM] 4 Jan 2011, <http://arxiv.org/abs/1101.0679>
- [3] ITT Visual Information Solutions: Getting Started with IDL, 2007, 1107IDL70GS, <http://www.astro.virginia.edu/class/oconnell/astr511/IDLresources/getting-started-IDL-v7.0.pdf>
- [4] ITT Visual Information Solutions: IDL Quick Reference, 2009, 0509IDL71QR, http://www2.fz-juelich.de/vislab/software/idl_docs/quickref.pdf
- [5] Research Systems: IDL HandyGuide, 2000, <http://www.astro.virginia.edu/class/oconnell/astr511/IDLresources/IDL-handiguide-v5.4.pdf>
- [6] WANG, M., FOX, G., PERCE, M.: Grid-based Collaboration in Interactive Data Language Applications, In Proceedings of ITCC (1), 2005, 335–341 2005, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.68.7212&rep=rep1&type=pdf>
- [7] SHASHARINA, S. – VEITZER, S.: GRIDL: High-Performance and Distributed Interactive Data Language, <http://www.txcorp.com/pdf/publications/GRIDL.pdf>

Kontaktná adresa

Martin ŠECHNÝ, (Mgr.),

Stredná priemyselná škola elektrotechnická, Plzenská 1,
080 47 Prešov, martin.sechny@shenk.sk



DOI: 10.5300/2011-OSSConf/111

VYUŽITÍ SCILABU PŘI KOORDINACI SVĚTELNÝCH KŘÍŽOVATEK

TUREK, Michal, (CZ)

Abstrakt. Příspěvek se zabývá navrhováním koordinace světelných křižovatek v Max-plus algebře. V první části příspěvku je definován problém a uveden teoretický rozbor Max-plus algebry. V další části příspěvku je navržena koordinace světelných křižovatek na základě rovnic Max-plus algebry. V závěrečné části je provedeno vyhodnocení.

Klíčová slova. Koordinace, světelné řízení, křižovatka, Max-plus algebra, Scilab.

SCILAB USE IN COORDINATION OF SIGNALIZED CROSSROADS

Abstract. The paper deals with the coordination of the design traffic lights at the Max-plus algebra. In the first part the problem is defined and given a theoretical analysis of max-plus algebra. In another part of the paper is designed to coordinate traffic lights based on the equation, Max-plus algebra. In the final part is an evaluation.

Key words and phrases. Coordination, Signalized control, Crossroads, Max-plus algebra, programming, Scilab, \LaTeX .

1 Úvod

Princip světelného řízení křižovatek umožňuje současné jízdy pouze nekolizním, resp. podmíněně kolizním dopravním proudům. Zajišťuje srozumitelné zobrazování návěstí včetně srozumitelného zobrazování změny návěstí, aby byla zajištěna bezpečnost provozu, a efektivně přiřazuje doby zelené jednotlivým dopravním proudům, aby byla zajištěna plynulost provozu. Uvedený princip lze využít u křižovatek, které se nacházejí ve velkých vzdálenostech. V případě, že se křižovatky nacházejí v malých vzdálenostech, je nutné základní princip světelného řízení rozšířit o koordinaci křižovatek, protože vstupy dopravních proudů do křižovatek jsou navzájem ovlivňovány.

Rozšířením principu světelného řízení o koordinaci dojde k tomu, že koordinované dopravní proudy získají na sousedních křižovatkách shodné doby zelené v odpovídajících

časových úsecích. Tím bude zajištěna plynulost provozu na sousedních křižovatkách a pozitivní psychologický vliv na řidiče, kteří budou moci vstoupit do sousedních křižovatek bez omezení rychlosti. Nekoordinovaným dopravním proudům zůstane přiřazována doba zelené obdobně jako u izolovaných křižovatek. Některé začátky a konce zelených pro nekoordinované dopravní proudy budou ovlivňovány dobami zelené koordinovaných dopravních proudů, ale nesmí dojít k tomu, že nabízená doba zelené pro nekoordinovaný dopravní proud bude nižší než požadovaná doba zelené.

2 Motivace

K návrhu koordinace křižovatek lze v současné době použít především technické podmínky TP 81, v nichž je uveden grafický a numerický způsob koordinace křižovatek [1]. Dále je možné použít matematický model pro tvorbu signálních plánů soustavy světelně řízených křižovatek, který vychází z matematického modelu pro řízení dopravy na křižovatce vytvořeného řešitelským kolektivem ve Výzkumném ústavu dopravním v Žilině, Černý a kolektiv [2].

Kromě přístupů, které se v současné době pro návrh světelného řízení se zajištěním koordinace křižovatek používají, je vhodné z hlediska efektivity hledat perspektivnější způsoby řešení koordinace křižovatek. Perspektivní řešení koordinace křižovatek představují obsáhlé matematické modely, které ve spojení s výpočtovou silou software řešícího úlohy matematického programování a matematického aparátu Max-plus algebry umožňují efektivně řídit koordinované a nekoordinované dopravní proudy na světelně řízených křižovatkách.

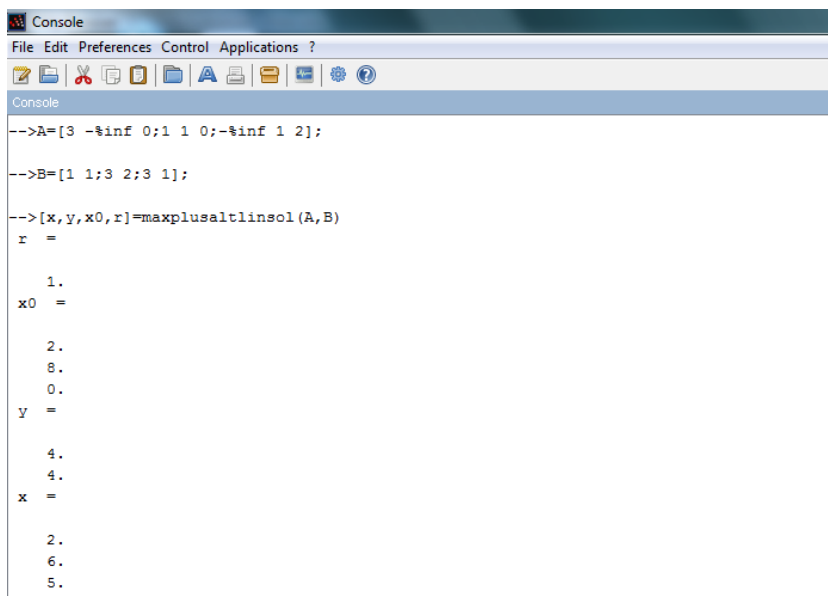
3 Max-plus algebra

Max-plus algebra se objevila v oblasti operačního výzkumu jako matematický přístup k rozhodování procesů v roce 1950. Většina problémů v operačním výzkumu zahrnuje hledání optima. Max-plus algebra využívá k hledání optima specifické operace vhodné pro modelování a analýzu synchronizovaných systémů. Představuje matematický nástroj, v němž se aritmetická operace sčítání nahrazuje určováním maxima a aritmetická operace násobení se nahrazuje sčítáním. Jedná se tedy o originální aplikace nelineární algebry při řešení synchronizačních problémů a nový přístup k řešení optimalizačních problémů, který může být použit pro návrh světelného řízení křižovatek a jejich koordinaci. Podrobnější informace o Max-plus algebře čtenář nalezne v literatuře [3], [4].

4 Software pro Max-plus algebru

Pro zpracování rovnic Max-plus algebry byl francouzskými výzkumnými institucemi INRIA (The French National Institute for Research in Computer Science and Control) a ENPC (École Nationale des Ponts et Chaussées) vytvořen vědecký program pro numerické výpočty Scilab [6].

Při spuštění programu Scilab se zobrazí hlavní pracovní okno (*Obrázek 1*) obsahující menu, ikony a pracovní plochu. Menu a ikony slouží k obsluze vytvářených, resp. již vytvořených souborů v programu Scilab, nebo ke spuštění nápovědy, ve které jsou vysvětlovány jednotlivé příkazy na demonstračních příkladech. Pracovní plocha umožňuje řešit konkrétní příklady prostřednictvím příkazů, které se zadávají do dialogových řádků.



```

Console
File Edit Preferences Control Applications ?
[Icons]
Console
-->A=[3 -%inf 0;1 1 0;-%inf 1 2];
-->B=[1 1;3 2;3 1];
-->[x,y,x0,r]=maxplusaltlinsol(A,B)
r =
    1.
x0 =
    2.
    8.
    0.
y =
    4.
    4.
x =
    2.
    6.
    5.

```

Obrázek 1: Hlavní pracovní okno software Scilab

5 Koordinace křižovatek v Max-plus algebře

Návrh koordinace křižovatek byl zpracován pro křižovatku Vápenice-Olomoucká-Svatoplukova a křižovatku Svatooplukova-Újezd v Prostějově (*Obrázek 2*) s ohledem na zásady, že délky cyklu na sousedních křižovatkách mají být shodné a koordinované dopravní proudy mají do křižovatek vstupovat ve fázích, které následují bezprostředně po sobě [1].

5.1 Sestavení matematického modelu koordinace dvou křižovatek

V rámci modelování koordinace křižovatek byly sestaveny rovnice Max-plus algebry. Před modelováním koordinace křižovatek bude uvedeno schéma koordinovaných křižovatek s vyznačenými vozidlovými proudy (*Obrázek 3*) a popsány veličiny, které v rovnicích Max-plus algebry vystupují. Následně bude provedeno modelování koordinace křižovatek.

Použité označení:

V_i – vozidlový proud i , $i \in \{A, B, \dots, L\}$,

$z_i(k)$ – začátek zelené pro i -tý vozidlový proud v k -té fázi,



Obrázek 2: Letecký pohled na koordinované křižovatky

$t_i(k)$ – doba zelené pro i -tý vozidlový proud vstupující v k -té fázi,

t – doba průjezdu mezi křižovatkami,

m_{ij} – čas mezi koncem vstupu i -tého proudu a začátkem vstupu j -tého proudu do křižovatky.

Systém následujících lineárních rovnic zajišťuje koordinaci dvou křižovatek:

$$z_A(k+1) = \max(z_C(k) + t_C(k) + m_{CA}, z_I(k) + t_I(k) + m_{IJ}) \quad (1)$$

$$z_B(k+1) = \max(z_C(k) + t_C(k) + m_{CB}, -\infty) \quad (2)$$

$$z_J(k+1) = \max(z_C(k) + t_C(k) + m_{CA}, z_I(k) + t_I(k) + m_{IJ}) \quad (3)$$

$$z_D(k+1) = \max(z_B(k) + t_B(k) + m_{BD}, -\infty) \quad (4)$$

$$z_E(k+1) = \max(z_B(k) + t_B(k) + m_{BE}, z_J(k) + t) \quad (5)$$

$$z_F(k+1) = \max(z_A(k) + t, -\infty) \quad (6)$$

$$z_G(k+1) = \max(z_J(k) + t_J(k) + m_{JG}, -\infty) \quad (7)$$

$$z_H(k+1) = \max(z_J(k) + t_J(k) + m_{JH}, -\infty) \quad (8)$$

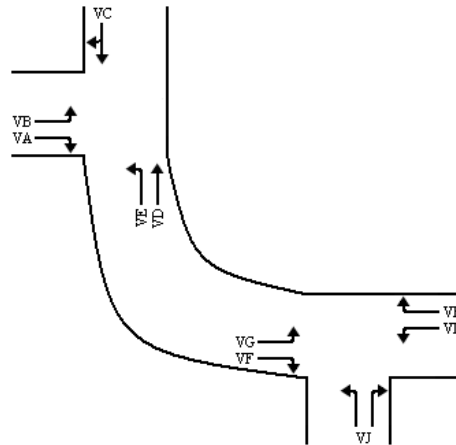
$$z_C(k+1) = \max(z_E(k) + t_E(k) + m_{EC}, -\infty) \quad (9)$$

$$z_I(k+1) = \max(z_F(k) + t_F(k) + m_{FI}, z_G(k) + t_G(k) + m_{GI}). \quad (10)$$

Otevřena zůstává volba počátečních začátků zelených pro vozidlové proudy. Ke stanovení počátečních začátků zelených na koordinovaných křižovatkách byla využita hodnota vlastního vektoru.

5.2 Analýza vstupních podkladů

Před zajištěním koordinace křižovatek prostřednictvím Max-plus algebry byla provedena analýza vstupních podkladů. Do každé křižovatky vstupuje 5 vozidlových proudů, hodnoty



Obrázek 3: Schéma koordinovaných křižovatek

mezičasů pro dvojice kolizních proudů na křižovatkách jsou uvedeny v *Tabulce 1* a fázová schémata, stanovená s ohledem na princip koordinace, jsou uvedena na *Obrázku 4*.

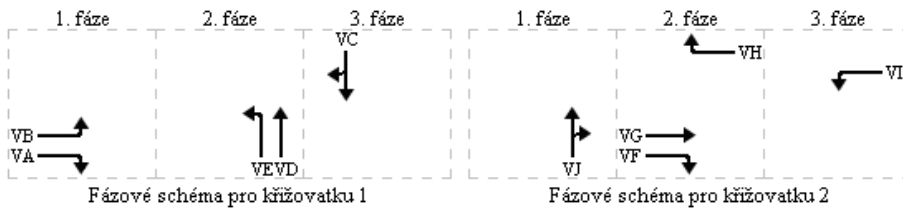
Tabulka 1: Stanovené mezičasy

Vyklizuje	Najíždí					Vyklizuje	Najíždí				
	VA	VB	VC	VD	VE		VF	VG	VH	VI	VJ
VA	-	-	3	-	-	VF	-	-	-	2	-
VB	-	-	3	2	2	VG	-	-	-	3	2
VC	3	3	-	-	2	VH	-	-	-	-	2
VD	-	2	-	-	-	VI	2	3	-	-	3
VE	-	2	2	-	-	VJ	-	2	2	3	-

Doby zelených v sekundách (*Tabulka 2*) odpovídají intenzitám účastníků silničního provozu vstupujících do křižovatek při zohlednění požadavků na minimální dobu zelené pro vozidlové proudy a minimální dobu, kterou potřebují vozidla pro vjezd do křižovatky. Doba průjezdu mezi křižovatkami činí 9 sekund.

5.3 Stanovení začátků zelených

V matematickém software Scilab byla pro matici A v (11) sestavenou na základě rovnic Max-plus algebry (1)-(10) vypočtena hodnota vlastního čísla a vlastního vektoru. Hodnota vlastního čísla umožňuje vymezení začátků zelených, které se po určité době opakují. Hodnota vlastního vektoru umožňuje definovat začátky zelených, po kterých nastane období,



Obrázek 4: Fázová schémata

Vozidlové proudy	VA	VB	VC	VD	VE	VF	VG	VH	VI	VJ
Požadované doby zelené pro vozidlové proudy	22	7	20	12	23	22	9	12	11	23

Tabulka 2: Doby zelených

kteří se opakuje. Následně byly zjišťovány pomocí vlastního čísla matice A začátky zelených pro vozidlové proudy.

$$A = \begin{array}{c|cccccccccc}
 & VA & VB & VJ & VD & VE & VF & VG & VH & VC & VI \\
 \hline
 VA & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 23 \\
 VB & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 23 \\
 VJ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 23 \\
 VD & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 VE & 0 & 9 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 VF & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 VG & 0 & 0 & 25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 VH & 0 & 0 & 25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 VC & 0 & 0 & 0 & 0 & 25 & 0 & 0 & 0 & 0 & 0 \\
 VI & 0 & 0 & 0 & 0 & 0 & 0 & 24 & 12 & 0 & 0
 \end{array} \quad (11)$$

5.3.1 Zjišťování hodnot vlastního čísla a vlastního vektoru

V software Scilab se nachází pro stanovení hodnoty vlastního čísla a vlastního vektoru příkaz $[1, v, d] = \text{maxplusmaxalgol}(A)$, přičemž prvek 1 reprezentuje vlastní číslo matice A , v reprezentuje vlastní vektor matice A a d je pozitivní celé číslo, které představuje délku cyklu vektorů matice A .

Nyní bude uveden zápis, kterým byly v software Scilab zjišťovány hodnoty vlastního čísla a vlastního vektoru. -->s=maxplusscg(A)

-->[l,v,d] = maxplusmaxalgol(A)

d = 3.

v = 86, 86, 86, 76, 76, 76, 92, 92, 82, 85.

l = 19.

5.3.2 Zjišťování začátků zelených

Po vymezení počtu zelených, které se po určité době opakují, byly vymezenému počtu zelených přiřazeny konkrétní hodnoty, přičemž za počáteční hodnotu opakovacího cyklu byly uvažována hodnota vlastního vektoru.

V software Scilab se nachází pro stanovení následných začátků zelených pomocí vlastního čísla příkaz $[X] = \text{maxplussys}(A, x0, p)$, přičemž prvek A reprezentuje matici A , $x0$ reprezentuje počáteční vektor a p reprezentuje vlastní číslo.

Nyní bude uveden zápis, kterým byly v software Scilab definovány začátky zelených pomocí vlastního čísla matice A .

-->x0=[86;86;86;76;76;76;92;92;82;85];

-->p=19;

-->[X]=maxplussys(A,x0,p)

Zjištěné začátky zelených pro vozidlové proudy (Tabulka 3) nabízejí řešiteli pouze výchozí řešení, při kterém jsou do křižovatek umožněny současné vjezdy všem vozidlovým proudům. Proto byly zjištěné začátky zelených přiřazeny jednotlivým skupinám tak, aby odpovídaly fázovým schémátům.

6 Závěr

V příspěvku bylo prezentováno zajištění koordinace křižovatek prostřednictvím Max-plus algebry, která představuje perspektivní přístup k řešení optimalizačních problémů, protože umožňuje prostřednictvím poměrně jednoduchých lineárních rovnic řešit složité optimalizační úlohy. Uvedený přístup byl aplikován v podmínkách soustavy křižovatek Vápenice-Olomoucká-Svatoplukova a Svatooplukova-Újezd v Prostějově, na které byly s ohledem na vlastní číslo a vlastní vektor stanoveny začátky zelených pro vozidlové proudy, které se po určité době opakují. Ke stanovení začátku zelených pro vozidlové proudy byl použit matematický program Scilab určený pro numerické výpočty [6].

Na ověření výstupů získaných z obsáhlých matematických modelů se ukazují perspektivními časově závislé Petriho sítě, které prostřednictvím vizualizace nabízejí dopravním inženýrům poměrně pohodlnou možnost detailní analýzy kritických událostí na křižovatkách [5].

Tabulka 3: Začátky zelených

Začátky zelených v jednotlivých fázích [s]									
VA	05:00:16	05:00:35	05:00:54	05:01:13	05:01:32	05:01:51	05:02:10	05:02:29	05:02:48
VB	05:00:16	05:00:35	05:00:54	05:01:13	05:01:32	05:01:51	05:02:10	05:02:29	05:02:48
VJ	05:00:16	05:00:35	05:00:54	05:01:13	05:01:32	05:01:51	05:02:10	05:02:29	05:02:48
VD	05:00:06	05:00:25	05:00:44	05:01:03	05:01:22	05:01:41	05:02:00	05:02:19	05:02:38
VE	05:00:06	05:00:25	05:00:44	05:01:03	05:01:22	05:01:41	05:02:00	05:02:19	05:02:38
VF	05:00:06	05:00:25	05:00:44	05:01:03	05:01:22	05:01:41	05:02:00	05:02:19	05:02:38
VG	05:00:22	05:00:41	05:01:00	05:01:19	05:01:38	05:01:57	05:02:16	05:02:35	05:02:54
VH	05:00:22	05:00:41	05:01:00	05:01:19	05:01:38	05:01:57	05:02:16	05:02:35	05:02:54
VC	05:00:12	05:00:31	05:00:50	05:01:09	05:01:28	05:01:47	05:02:06	05:02:25	05:02:44
VI	05:00:15	05:00:34	05:00:53	05:01:12	05:01:31	05:01:50	05:02:09	05:02:28	05:02:47
VA	05:03:07	05:03:26	05:03:45	05:04:04	05:04:23	05:04:42	05:05:01	05:05:20	05:05:39
VB	05:03:07	05:03:26	05:03:45	05:04:04	05:04:23	05:04:42	05:05:01	05:05:20	05:05:39
VJ	05:03:07	05:03:26	05:03:45	05:04:04	05:04:23	05:04:42	05:05:01	05:05:20	05:05:39
VD	05:02:57	05:03:16	05:03:35	05:03:54	05:04:13	05:04:32	05:04:51	05:05:10	05:05:29
VE	05:02:57	05:03:16	05:03:35	05:03:54	05:04:13	05:04:32	05:04:51	05:05:10	05:05:29
VF	05:02:57	05:03:16	05:03:35	05:03:54	05:04:13	05:04:32	05:04:51	05:05:10	05:05:29
VG	05:03:13	05:03:32	05:03:51	05:04:10	05:04:29	05:04:48	05:05:07	05:05:26	05:05:45
VH	05:03:13	05:03:32	05:03:51	05:04:10	05:04:29	05:04:48	05:05:07	05:05:26	05:05:45
VC	05:03:03	05:03:22	05:03:41	05:04:00	05:04:19	05:04:38	05:04:57	05:05:16	05:05:35
VI	05:03:06	05:03:25	05:03:44	05:04:03	05:04:22	05:04:41	05:05:00	05:05:19	05:05:38

Poděkování

Článek byl zpracován s podporou grantu Fakulty strojní VŠB-TU Ostrava č. SP2011/129 Výzkum v oblasti modelování pro podporu řízení dopravy ve městech.

Literatura

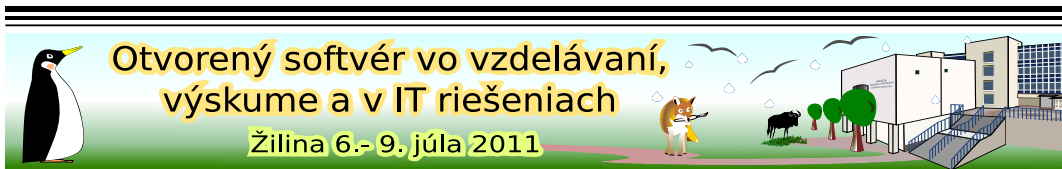
- [1] Technické podmínky TP 81.: *Navrhování světelných signalizačních zařízení pro řízení provozu na pozemních komunikacích*. Brno: CDV, Ministerstvo dopravy České republiky, 2006
- [2] ČERNÝ, J. – KLUVÁNEK, P.: *Základy matematickej teórie dopravy*. Bratislava: VEDA, 1991. 279 s. ISBN 80-224-0099-8
- [3] ANDERSEN, M. H.: *Max-plus algebra: properties and applications*. 2002

- [4] BACELLI, F. – COHEN, G. – OLSDER, G. J., – QUADRAT, J. P.: *Synchronization and Linearity*. 2001
- [5] TUREK, M.: *Navrhování řízení světelných křížovatek Petriho sítěmi*, Sborník z konference Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach, Žilina, 2010. s.193-203. ISBN 978-80-970457-0-8
- [6] *Scilab*. Dostupné z <<http://www.scilab.org>>.

Kontaktní adresa

Michal TUREK (Ing.),

Institut dopravy FS VŠB – TU Ostrava, 17. listopadu 15,
708 33 Ostrava – Poruba, michal.turek@atlas.cz



DOI: 10.5300/2011-OSSConf/121

MOŽNOSTI SYNCHRONIZACE LINEK MHD VE SCILABU

TUREK, Richard (CZ)

Abstrakt. Příspěvek se zabývá modelováním synchronizace odjezdů spojů z přestupních zastávek pomocí Max-plus algebry ve Scilabu. Problém přestupu cestujících je formulován pomocí matematického aparátu Max-plus algebry. Následuje seznámení s operacemi Max-plus algebry a popis softwarového nástroje Scilab který je využit při modelování synchronizace linek MHD Prostějov.

Klíčová slova. Synchronizace, přestupní zastávka, modelování, Max-plus algebra, Scilab.

SYNCHRONIZATION OF PUBLIC TRANSPORT LINES IN SCILAB

Abstract. The paper deals with the modeling of synchronization departure from the transfer stations using Max-plus algebra in Scilab. The problem of transfer passengers is formulated using mathematical Max-plus algebra. The following introduction to the operations of the max-plus algebra and a description of the Scilab software tool that is used in modeling the synchronization of public transport lines Prostějov.

Key words and phrases. Synchronization, Interchange bus stops, modeling, Max-plus algebra, Scilab.

1 Úvod

Řešení hromadné osobní dopravy představuje komplex dílčích problémů, které musí být řešeny společně. Klíčovým problémem každého systému MHD je snaha o maximální snížení ekonomické ztrátovosti. S uvedeným problémem velice úzce souvisí ekonomická efektivita rozsahu dopravní sítě MHD především ve vztahu k počtu nasazených vozidel a provozní délce jednotlivých linek, který by korespondoval s požadavky cestující veřejnosti. Jedním z nežádoucích důsledků takovýchto do značné míry protichůdných požadavků může být ztráta možnosti vhodných přestupů mezi spoji některých linek na významných zastávkách sítě.

Existence přímého spojení znamená výhodu pro cestujícího spočívající v tom, že nemusí při svých cestách přestupovat. V důsledku racionalizace však často dochází ke snížení

počtu linek, což pro cestující veřejnost představuje zvýšení potřeby přestupovat v některých relacích. Relace bez možnosti přímého spojení jsou charakteristické nižšími intenzitami přepravního proudu v daných relacích a vyšší mírou souběhu linek.

2 Motivace

K řešení problému časové koordinace spojů v přestupních uzlech je možno přistoupit různými způsoby. Nejčastěji je využíván tzv. zkušenostní přístup, kdy pověření zaměstnanci dopravců zajišťují možnosti přestupu na vybraných přestupních zastávkách zpravidla na základě svých logických úvah vyplývajících z historicky vzniklých přestupních vazeb.

Pokročilejší přístup v dopravní praxi v tuzemsku i v zahraničí představují heuristické metody, mezi které patří metoda pravidelných n -segmentů na kružnici známá též jako tzv. úloha o žilinských n -úhelnících, a metody založené na lineárním programování, v rámci kterých byl vytvořen matematický model časové koordinace spojů sestavený řešitelským kolektivem Výzkumného ústavu dopravního v Žilině, Černý a kolektiv [1].

S ohledem na výpočtovou sílu solverů a matematického aparátu Max-plus algebry se nabízí možnost rozšíření pokročilejších přístupů prostřednictvím tvorby robustních matematických modelů pro synchronizaci technologických omezení dopravců i požadavků cestující veřejnosti.

3 Max-plus algebra

Většina problémů v operačním výzkumu zahrnuje hledání optima. Max-plus algebra využívá při formulaci rovnic specifické operace a proto je zajímavým kandidátem pro matematický popis problémů v oblasti operačního výzkumu.

Max-plus algebra představuje matematický aparát, ve kterém jsou klasické aritmetické operace sčítání a násobení nahrazeny následujícími operacemi

$$a \oplus b = \max(a, b), \quad a \otimes b = a + b.$$

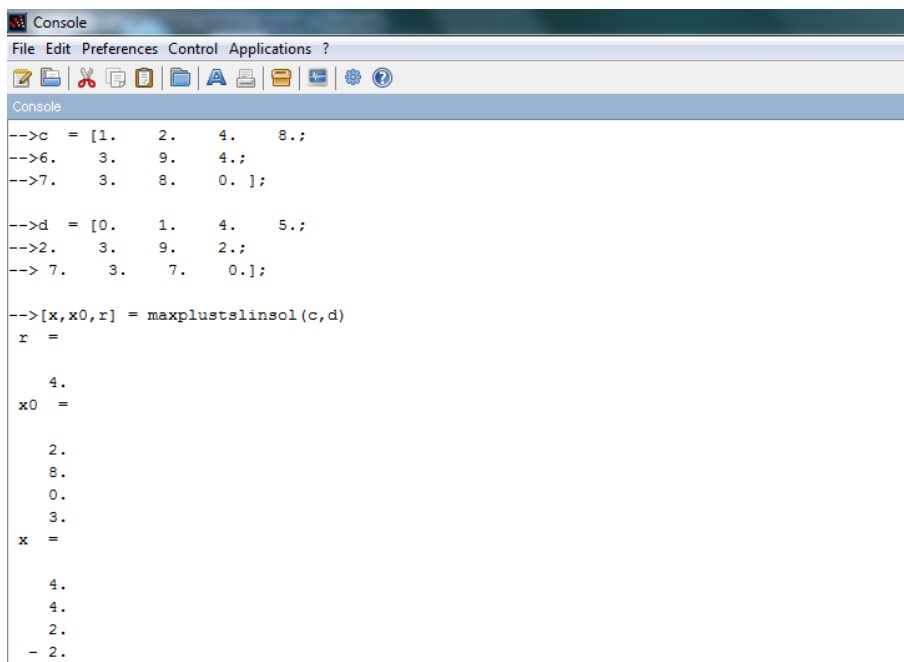
Uvedený matematický přístup nabízí netradiční způsob vhodný pro modelování systémů diskrétních událostí DES (Discrete Event Systems) a optimalizaci problémů ve výrobě a dopravě. Navíc se ukazuje silná podobnost s klasickou lineární algebrou, což umožňuje například analogické řešení soustav lineárních rovnic a efektivní výpočet vlastního čísla a vlastních vektorů [2], [3] a [4].

4 Softwarový nástroj pro Max-plus algebru

K řešení rozsáhlejších modelů nejen v oblasti Max-plus algebry je vždy nutné využít dostupný software, prostřednictvím kterého je možné získat vyhodnocení řešené problematiky. Pro

zpracování rovnice Max-plus algebry byl francouzskými výzkumnými institucemi INRIA (The French National Institute for Research in Computer Science and Control) a ENPC (École Nationale des Ponts et Chaussées) vytvořen v roce 1990 vědecký program pro numerické výpočty s názvem Scilab [6].

Při spuštění programu Scilab se zobrazí hlavní pracovní okno (*Obrázek 1*) obsahující menu, ikony a pracovní plochu. Menu a ikony slouží k obsluze vytvářených, resp. již vytvořených souborů v programu Scilab, např. otevírání a ukládání souborů prostřednictvím záložky *File*. Pracovní plocha slouží k řešení konkrétních příkladů prostřednictvím příkazů, které se zadávají do dialogových řádků. Např. příkazem *maxplusmaxalgol* se zjišťuje vlastní číslo a vlastní vektor matice. K vysvětlení jednotlivých příkazů lze využít nápovědy, která se po spuštění zobrazí v okně Scilab Browse Help obsahující syntax každého příkazu včetně demonstračního příkladu.



```

Console
File Edit Preferences Control Applications ?
[Icons]
Console
-->c = [1.  2.  4.  8.;
-->6.  3.  9.  4.;
-->7.  3.  8.  0.];

-->d = [0.  1.  4.  5.;
-->2.  3.  9.  2.;
-->7.  3.  7.  0.];

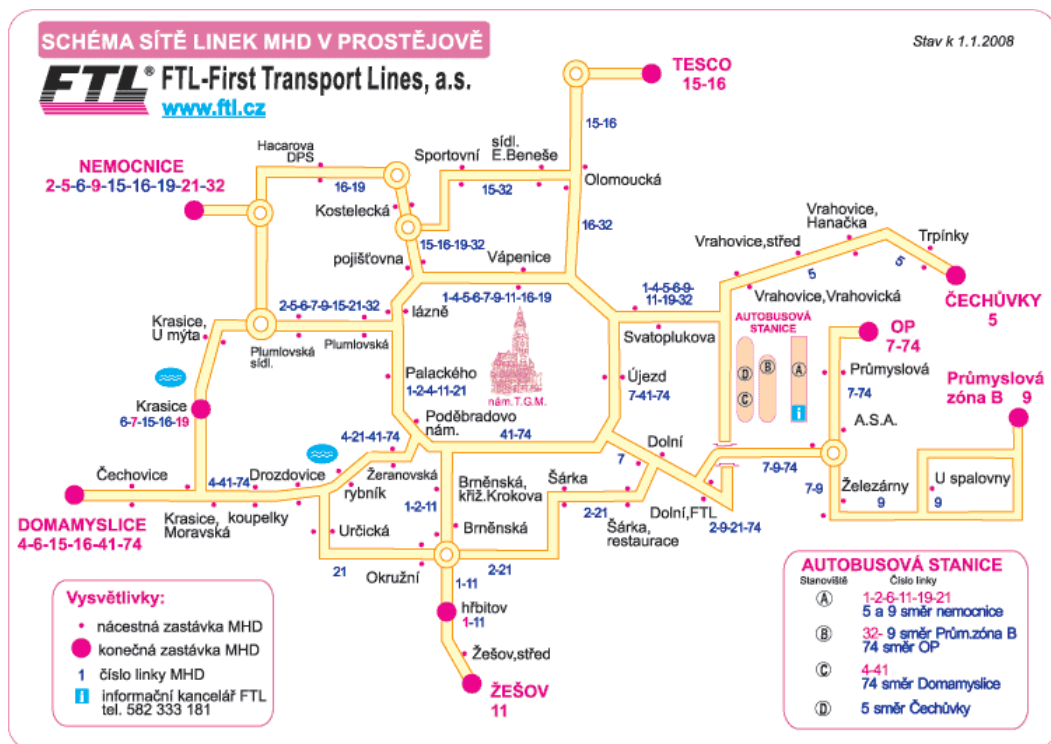
-->[x,x0,z] = maxpluslinsol(c,d)
z =
    4.
x0 =
    2.
    8.
    0.
    3.
x =
    4.
    4.
    2.
    - 2.

```

Obrázek 1: Pracovní okno software Scilab

5 Synchronizace tří linek v podmínkách MHD Prostějov

Území města je obsluhováno prostřednictvím 15 linek a jedné linky komerční, na které neplatí tarif MHD. Schéma sítě linek MHD Prostějov je znázorněno na *Obrázku 2*. Městskou hromadnou dopravu ve městě Prostějov zajišťuje 22 autobusů. Vozidlový park tvoří standardní autobusy Karosa a Sor s pohonem na zemní plyn. Průměrné stáří vozidlového parku



Obrázek 2: Schéma sítě linek MHD v Prostějově

činí cca 8 let. Autobusy obsluhují spoje linek MHD podle předem stanovených turnusů. V jednotlivých turnusech jsou pro každé vozidlo definovány činnosti (přistavení, spoje, bezpečnostní přestávky, odstavení), zastávky, časové údaje, rychlost, ujetá vzdálenost atd. Autobusy v současné době nejsou přiřazeny určité lince, ale v průběhu dne obsluhují spoje různých linek, které na sebe navazují v konečných zastávkách. Některá vozidla navíc obsluhují dle rozpisu také spoje příměstských linek. Počet nasazených autobusů ve špičce činí 19, množství autobusů v sedle je 7. Intervaly na jednotlivých linkách jsou kromě komerční linky nepravidelné, ostatně jak je ve většině našich okresních měst obvyklé.

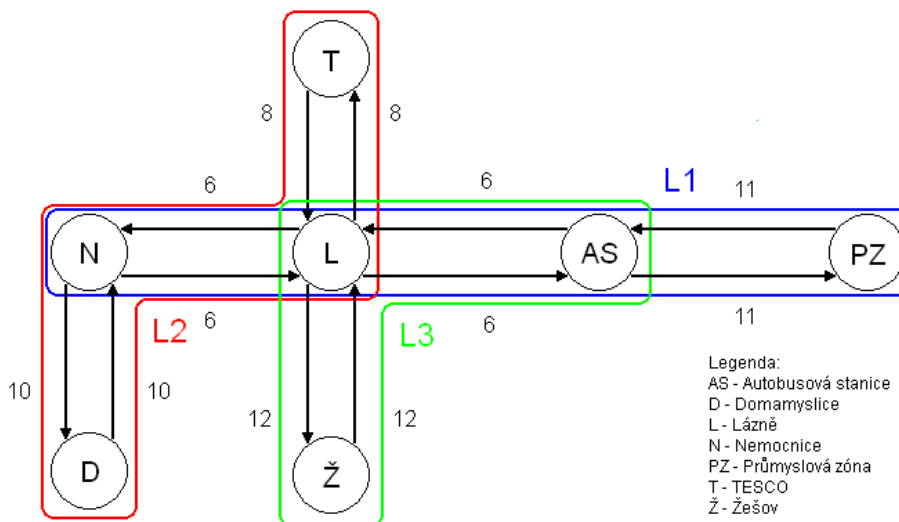
Současný stav MHD v Prostějově je charakteristický nepřiměřeně velkým počtem linek na větší okresní město, což přispívá k nepřehlednosti, zejména pro cestujícího, který MHD v Prostějově nevyužívá pravidelně.

V důsledku racionalizace by došlo ke snížení počtu autobusů a také linek. Výsledek studie činí 15 autobusů obsluhujících 10 linek. Uvedené řešení představuje efektivní využití autobusů. Snížení počtu linek přispívá ke zpřehlednění sítě MHD Prostějov a eliminaci souběhů linek. V navrženém řešení neobsluhují autobusy během dne spoje různých linek, ale každý autobus je přidělen jedné lince. Navržené řešení umožňuje synchronizaci provozu

na třech linkách, včetně navazujících výhod – vytvoření lepších podmínek pro koordinaci odjezdu spojů jednotlivých linek.

Pro obyvatele Krasic, severní části města a části průmyslové zóny představuje předkládané řešení zhoršení dostupnosti autobusové stanice a železniční stanice Prostějov hlavní nádraží v podobě absence přímého spojení. Ve výše uvedených relacích budou cestující nuceni více přestupovat, ve všech případech však půjde pouze o 1 přestup. V současné době je největší přestup realizován obousměrně mezi MHD, PAD a železniční dopravou a nejméně přestupů je realizováno v relaci mezi spoji MHD. Realizací navržených změn však vznikne vyšší potřeba přestupovat. V takovém případě bude nutné zabývat se časovou koordinací spojů na jednotlivých linkách na vytípaných zastávkách.

V další části příspěvku bude pozornost věnována modelování jízdních řádů autobusových linek v dopravní síti MHD Prostějov, resp. stanovení časů odjezdů spojů synchronizovaných linek uvedených na *Obrázku 3*. Jedná se o zajištění přestupu mezi linkami L1 a L2 na zastávce nemocnice a zajištění přestupu mezi linkami L2 a L3 na zastávce lázně.



Obrázek 3: Fragment dopravní sítě MHD Prostějov

5.1 Sestavení matematického modelu synchronizace linek MHD

Modelování synchronizace linek MHD bylo vytvořeno rovnicemi Max-plus algebry pro každou zastávku tak, aby při každém odjezdu ze zastávek byly zohledněny příjezdy vozidel MHD ze sousedních zastávek.

Před modelováním synchronizace linek MHD budou z důvodu srozumitelnosti popsány symboly, které budou dále používány:

$x_{ij}(k)$ – k -tý synchronizovaný odjezd z i -té zastávky směrem k j -té zastávce (proměnná

veličina), $i, j \in \{A, B, \dots, E\}$,

t_{ij} – jízdní doba mezi i -tou a j -tou zastávkou (konstantní veličina), $i, j \in \{A, B, \dots, E\}$,

i_l – interval na lince l (konstantní veličina).

Systém následujících lineárních rovnic zajišťuje synchronizaci linek MHD:

$$x_{AS-L}(k+1) = \max(x_{AS-L}(k) + i_3, x_{Z-L}(k) + t_{Z-AS}) \quad (1)$$

$$x_{D-N}(k+1) = \max(x_{D-N}(k) + i_2, x_{T-L}(k) + t_{T-D}) \quad (2)$$

$$x_{N-D}(k+1) = \max(x_{N-D}(k) + i_2, x_{PZ-AS}(k) + t_{PZ-N}, x_{T-L}(k) + t_{T-N}) \quad (3)$$

$$x_{N-L}(k+1) = \max(x_{D-N}(k) + t_{D-N}, x_{N-L}(k) + i_1, x_{PZ-AS}(k) + t_{PZ-N}) \quad (4)$$

$$x_{L-AS}(k+1) = \max(x_{L-AS}(k) + i_3, x_{T-L}(k) + t_{T-L}, x_{Z-L}(k) + t_{Z-L}) \quad (5)$$

$$x_{L-T}(k+1) = \max(x_{AS-L}(k) + t_{AS-L}, x_{D-N}(k) + t_{D-L}, x_{L-T}(k) + i_2) \quad (6)$$

$$x_{PZ-AS}(k+1) = \max(x_{N-L}(k) + t_{N-PZ}, x_{PZ-AS}(k) + i_1) \quad (7)$$

$$x_{T-L}(k+1) = \max(x_{D-N}(k) + t_{D-T}, x_{T-L}(k) + i_2) \quad (8)$$

$$x_{Z-L}(k+1) = \max(x_{AS-L}(k) + t_{AS-Z}, x_{Z-L}(k) + i_3) \quad (9)$$

Problematickou je otázka stanovení počátečních odjezdů. Při modelování synchronizace tří linek MHD Prostějov byla ke stanovení počátečních odjezdů využita hodnota vlastního vektoru.

5.2 Charakteristika vstupních údajů

Vybraná část dopravní sítě MHD Prostějov je tvořena třemi linkami, které jsou vždy obsluhovány jedním vozidlem. Na každé lince jsou definovány linkové intervaly (*Tabulka 1*) a jízdní doby mezi zastávkami (*Tabulka 2*).

Tabulka 1: Linkové intervaly

Číslo linky	Linkový interval na lince l (min)
Linka 1	56
Linka 2	58
Linka 3	46

5.3 Stanovení odjezdů spojů ze zastávek MHD

Na základě hodnot vstupních údajů dosazených do rovnic Max-plus algebry (1) až (9), kterými se modelovala synchronizace linek MHD, byla sestavena matice A v (10).

Tabulka 2: Jízdní doby mezi zastávkami

Úsek mezi i -tou a j -tou zastávkou	Jízdní doba mezi i -tou a j -tou zastávkou (min)	Úsek mezi i -tou a j -tou zastávkou	Jízdní doba mezi i -tou a j -tou zastávkou (min)
AS-L	6	L-Z	12
AS-PZ	11	N-D	10
D-N	10	N-L	6
L-AS	6	PZ-AS	11
L-N	6	T-L	8
L-T	8	Z-L	12

$$A = \begin{array}{c|cccccccccc} & NL & PZAS & DN & LT & ND & TL & ASL & LAS & ZL \\ \hline ASL & 0 & 0 & 0 & 0 & 0 & 0 & 46 & 0 & 18 \\ DN & 0 & 0 & 58 & 0 & 0 & 24 & 0 & 0 & 0 \\ ND & 0 & 23 & 0 & 0 & 58 & 14 & 0 & 0 & 0 \\ NL & 56 & 23 & 10 & 0 & 0 & 0 & 0 & 0 & 0 \\ LAS & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 46 & 12 \\ LT & 0 & 0 & 16 & 58 & 0 & 0 & 6 & 0 & 0 \\ PZAS & 23 & 56 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ TL & 0 & 0 & 24 & 0 & 0 & 58 & 0 & 0 & 0 \\ ZL & 0 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 46 \end{array} \quad (10)$$

Následně byl pro matici A zjišťován vlastní vektor a vlastní číslo. Vlastní vektor byl zjišťován za účelem stanovení počátečních odjezdů ze zastávek MHD a vlastní číslo bylo zjišťováno za účelem stanovení odjezdů ze zastávek MHD, při kterých bude opět dosaženo počátečních odjezdů.

Pomocí vlastního čísla byly dále zjišťovány každé následné odjezdy ze zastávek MHD, čímž byly získány hodnoty odjezdů mezi počátečními odjezdy a opakovanými počátečními odjezdy.

5.3.1 Stanovení vlastního čísla a vlastního vektoru

Hodnota vlastního čísla a vlastního vektoru byla stanovena v software Scilab prostřednictvím příkazu $[1, v, d] = \text{maxplusmaxalgol}(A)$, přičemž jednotlivé prvky uvedeného příkazu mají následující význam:

l – vlastní číslo matice A ,

v – vlastní vektor matice A ,

d – pozitivní celé číslo, které představuje délku cyklu vektorů matice A .

Stanovení vlastního čísla a vlastního vektoru matice A v software Scilab:

```
-->s=maxplusscg(A)
```

```
-->[l,v,d] = maxplusmaxalgol(A)
```

```
d = 8.
```

```
v = 599.5, 606.5, 603, 601, 599.5, 604.5, 608, 608, 571.5
```

```
l = 54.5.
```

5.3.2 Stanovení následných odjezdů

Časy následných odjezdů ze zastávek MHD v časovém úseku, který odpovídá hodnotě vlastního čísla, byly stanoveny v software Scilab prostřednictvím příkazu:

```
[X] = maxplussys (A, x0, p),
```

přičemž jednotlivé prvky uvedeného příkazu mají následující význam: A – matice A , x_0 – počáteční vektor, p – vlastní číslo.

Stanovení následných odjezdů v rámci vlastního čísla matice A v software Scilab:

```
-->x0=[600;607;603;601;600;605;608; 608;572];
```

```
-->p=55;
```

```
-->[X]=maxplussys(A,x0,p)
```

Získané hodnoty synchronizovaných odjezdů ze zastávky lázně jsou uvedené v *Tabulce 3*.

6 Závěr

V článku bylo prezentováno modelování synchronizace odjezdů spojů vybraných linek MHD Prostějov prostřednictvím Max-plus algebry, která představuje perspektivní přístup k řešení optimalizačních problémů, protože umožňuje prostřednictvím poměrně jednoduchých rovnic řešit složité optimalizační úlohy. Ke stanovení odjezdů spojů z vybraných zastávek MHD Prostějov, které se po určité době opakují, byl použit matematický program Scilab určený pro numerické výpočty [6].

Budoucí výzkum se zaměří na ověření prezentovaných rovnic (1) až (8) a jejich zobecnění na celou síť MHD prostřednictvím vizualizace, jež umožňují Petriho síť [5].

Poděkování

Článek byl zpracován s podporou grantu Fakulty strojní VŠB-TU Ostrava č. SP2011/129 Výzkum v oblasti modelování pro podporu řízení dopravy ve městech.

Tabulka 3: Synchronizované odjezdy ze zastávky lázně

Relace	Příjezd	Relace	Odjezd
TESCO – lázně – aut. st.			
TESCO - lázně	5:16	lázně - aut.st.	5:38
	6:11		6:33
	7:05		7:27
	8:00		8:22
aut.st - lázně	5:06	lázně - TESCO	5:23
	6:00		6:17
	6:55		7:12
	7:49		8:06
Domamyslice – lázně – aut. st.			
Domamyslice - lázně	5:23	lázně - aut.st.	5:38
	6:17		6:33
	7:12		7:27
	8:06		8:22
aut.st - lázně	5:06	lázně - Domamyslice	5:16
	6:00		6:11
	6:55		7:05
	7:49		8:00

Literatura

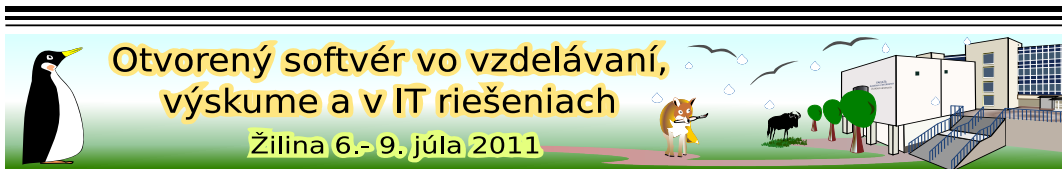
- [1] ČERNÝ, J. – KLUVÁNEK, P.: *Základy matematickej teórie dopravy*. Bratislava: VEDA, 1991. 279 s. ISBN 80-224-0099-8
- [2] BURKARD, R. E. – BUTKOVIČ, P.: *Max algebra and the linear assignment problem*. 2003
- [3] ANDERSEN, M. H.: *Max-plus algebra: properties and applications*. 2002
- [4] BACELLI, F. – COHEN, G. – OLSDER, G. J. – QUADRAT, J. P.: *Synchronization and Linearity*. 2001
- [5] TUREK, R.: *Koordinace linek MHD s využitím Petriho sítí*, Sborník z konference Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach, Žilina, 2010. s. 205–214. ISBN 978-80-970457-0-8
- [6] *Scilab*. Dostupné z <http://www.scilab.org>

Kontaktní adresa

Richard TUREK (Ing.),

Institut dopravy FS VŠB – TU Ostrava, 17. listopadu 15,
708 33 Ostrava – Poruba, richardtunek@seznam.cz

OPEN GIS
OTVORENY SOFTVÉR V GIS



DOI: 10.5300/2011-OSSConf/133

ZAŘAZENÍ GEOGRAFICKÝCH INFORMAČNÍCH SYSTÉMŮ DO VÝUKY PŘEDMĚTU INFORMATIKA VE VEŘEJNÉ SPRÁVĚ

DOLEJŠOVÁ, Miroslava (CZ)

Abstrakt. Článek představuje ideu výuky informatiky se zaměřením na geografické informační systémy (GIS) pro obor Veřejná správa a regionální rozvoj na Fakultě managementu a ekonomiky Univerzity Tomáše Bati ve Zlíně. Je upozorněno především na užití praktických ukázek, aby to bylo srozumitelné běžnému občanovi i lidem z veřejnosprávních institucí. V článku je několik ukázek zmíněno.

Klíčová slova. Informatika, veřejná správa, výuka, GIS.

INCLUSION OF THE GEOGRAPHICAL INFORMATION SYSTEMS IN INFORMATICS IN THE PUBLIC ADMINISTRATION

Abstract. This paper deals with a new conception of the subject "Informatics in the public administration" taught in the second year of the field of study "Public administration and the regional development" and guaranteed by the Department of Statistics and Quantitative Methods at the Tomas Bata University in Zlín. The main purpose is to point out the significance of the geographical information systems (GIS) in the public administration, to describe and clearly demonstrate the ways of their application both from the eye of citizens and the public authorities and to indicate the further development in this area.

Key words and phrases. Informatics, Public Administration, Teaching, GIS.

1 Úvod

V současné době informační a znalostní společnosti se bez kvalitních informací prakticky neobejdeme. Na jedné straně existuje dostatek informací, ale stačí nám to k rozhodování? Víme, které informace máme použít, abychom učinili správné rozhodnutí? Máme k dispozici nástroje, které by nám byly schopny pomoci rozhodování usnadnit? Odpověď je poměrně jednoduchá. Tyto nástroje k dispozici jsou a nazývají se geografické informační systémy (GIS).

Geografické informační systémy mají uplatnění v řadě oblastí. Jednou z nejvyužívanějších oblastí je oblast veřejné správy. Cílem tohoto příspěvku je popsat možnosti zavedení geografických informačních systémů do předmětu Informatika ve veřejné správě vyučovaném na Fakultě ekonomiky a managementu Univerzity Tomáše Bati ve Zlíně.

2 Stručné představení předmětu IVS

Ústav statistiky a kvantitativních metod zajišťuje výuku povinného předmětu Informatika ve veřejné správě ve 2. ročníku zimního semestru ve studijním oboru Veřejná správa a regionální rozvoj ve všech formách studia. Výuka v prezenční formě studia probíhá v rozsahu dvou cvičení po dobu 13 týdnů na počítačových učebnách, pro kombinovanou a celoživotní formu studia je vymezeno 9 konzultací.

Předmět je v první řadě zaměřen na práci s informačními zdroji ve veřejné správě. Dalším cílem předmětu je rozšíření počítačové gramotnosti studentů o kreslení diagramů v programu Microsoft Word a rozšíření znalostí programu Excel. Předmět je ukončen klasifikovaným zápočtem. Podmínkou pro jeho získání je vypracování seminární práce a úspěšné absolvování znalostního testu.

3 Proč právě GIS v předmětu IVS?

Zásadním nedostatkem oboru Veřejná správa a regionální rozvoj jsou chybějící předměty, které jsou vyučovány Ústavem statistiky a kvantitativních metod ve studijním oboru Management a ekonomika (předměty Počítačové zpracování dat a Základy kvantitativních metod). Studenti oboru Veřejná správa a regionální rozvoj mají jen předmět Informatika pro ekonomy (společný předmět pro oba obory) a ve druhém ročníku předmět Informatika ve veřejné správě.

Pouhé vyhledávání informačních zdrojů, rozšíření počítačové gramotnosti o možnosti Wordu a Excelu podle názoru autorky nestačí. Návaznost na další předměty oboru Veřejná správa a regionální rozvoj (především předmět Regionální analýza) vyvolala potřebu úpravy obsahu tohoto předmětu. Předmět Informatika ve veřejné správě tak může doplňovat předmět Regionální analýza či další předměty garantované Ústavem regionálního rozvoje, veřejné správy a práva. Kromě obohacení obsahu obou předmětů a rozšíření znalostí studentů by došlo také k navázání úzké pedagogické i vědeckovýzkumné spolupráce mezi oběma ústavy.

4 Proč využívat GIS?

K nejčastějším typům informací, které potřebuje každý člověk i organizace, patří geografické informace. Pravidelně hledáme cestu do konkrétního místa, rozhodujeme se, kde umístíme další provozovnu, chceme vědět, jak dlouho nám přeprava do zvoleného místa bude trvat,

a také chceme znát vzdálenost mezi oběma místy. Geografické informační systémy nám tyto (a nejenom tyto) typy informací poskytují.

Na první pohled se může zdát, že geografické informační systémy jsou jen software. GIS však možnosti klasického programu převyšují. Usnadňují nám rozhodování a také se jedná o kvalitní grafický nástroj pro analýzu a modelování dat. Geografické informační systémy jsou obvykle spojovány s mapami a navigačním systémem. Mapy jsou však výstupem geografických informačních systémů a navigační systém (myšleno systém globální navigace GPS) nebývá zahrnut do těchto systémů.

5 Představení geografických informačních systémů

5.1 Tři pohledy na GIS

Různí autoři pohlíží na geografické informační systémy různě. Maguire [3] v příspěvku „An Overview and Definition of GIS“ rozlišuje tři pohledy na geografické informační systémy: mapový, databázový a modelový (ve smyslu provádění prostorových analýz).

Mapový pohled je nejjednodušší. Pomocí geografických informačních systémů můžeme mapy vytvářet a upravovat. Výsledná mapa je pak souborem námi vybraných vrstev, které chceme v mapě zobrazit.

Mapa jako obraz nám nebude stačit. Potřebujeme ji doplnit o konkrétní údaje. Tato data jsou uložena v různých databázích, které se často označují jako geodatabáze. S databází je možné provádět jednoduché analýzy. Neméně důležitou funkcí databází je i vizualizace dat.

Pokud máme k dispozici obraz i data, můžeme prostřednictvím různých analytických nástrojů získávat další informace potřebné pro rozhodování. Výběrem dat, různými dotazy, aplikací statistických a analytických funkcí a dalších vhodných nástrojů získáme nová data, která nám umožní získat komplexnější pohled na studovaný problém. Oproti databázovému pohledu je provádění prostorových analýz daleko složitější.

Voženílek [5] vychází z myšlenek Maguirea [3]. Podle něj mají geografické informační systémy také tři aspekty: databázový, analytický a kartografický. Databázový aspekt je založen na správně navržené databázi, pomocí které lze data shromažďovat, třídít, vybírat a prezentovat. Analytický aspekt je spojen s prostorovou analýzou a modelováním. Kartografický aspekt se týká tvorby, zpracování a prezentace map.

Oproti tomu Rapant [4] chápe geografické informační systémy jako software, konkrétní aplikace nebo jako informační technologie. Chápání geografických informačních systémů jen jako software je příliš zúženo, spíše se lze přiklonit k chápání geografických informačních systémů jako konkrétní aplikace, v nejširším slova smyslu i jako informační technologie.

Geografické informační systémy lze proto definovat jako prostředek zobrazení, manipulace a analýzy prostorových dat.

5.2 Typy dat, se kterými GIS pracují

Geografické informační systémy pracují s různými typy dat. K nejpoužívanějším z nich patří vektorová a rastrová data. Vektorová data mohou být vyjádřena jako polygony (typické pro lesy, území krajů a velkých měst, vodní plochy), čáry pro zobrazení silnic, železnic, řek, ulic nebo jako body (stromy, obchody, malá města a obce). Rastrová data jsou vyjádřena jako matice bodů. K rastrovým datům patří družicové a letecké snímky, zobrazení terénů a sítí.

5.3 Co můžeme zjišťovat prostřednictvím GIS?

- Zjišťovat počty (Kolik?): počet nemocnic, počet úřadů.
- Zjišťovat hustotu (Jak mnoho?): počet lékařů připadajících na 1000 pacientů.
- Zjišťovat, co je uvnitř určité oblasti: počet obchodů v určitém regionu.
- Zjišťovat, co je poblíž: nejbližší nemocnice, nejbližší rekreační středisko.
- Modelovat změny (obvykle předpovědi počasí).

6 Komerční a otevřený GIS software

K nejčastěji využívaným komerčním aplikacím patří ArGIS společnosti ESRI. Tyto komerční produkty jsou však poměrně nákladné. Z tohoto důvodu je vhodné využít otevřené geografické informační systémy k prostorové analýze dat.

K nejstarším GIS software patří GRASS (Geographical Resources Analysis Support System). Tento systém je orientován na rastrovou i vektorovou analýzu dat. Dále obsahuje nástroje pro prostorové modelování, vizualizaci a řízení a zpracování satelitního obrazu. Poslední verze 6.4.1 je ze dne 12. dubna 2011. Bližší informace o tomto GIS software jsou k dispozici na <http://grass.osgeo.org/>.

Dalším otevřeným geografickým informačním systémem je Quantum GIS (QGIS). Tento GIS umožňuje práci s vektorovými i rastrovými daty a pracuje s databázovými formáty. Při práci se systémem není potřeba konvertovat formáty dat. Pomocí QGIS je možné vytvářet mapy a interaktivně provádět průzkum prostorových dat, tato prostorová data editovat, publikovat a exportovat do různých formátů. QGIS spolupracuje i s geografickým informačním systémem GRASS ve formě digitalizace dat a provádění prostorových analýz. Poslední verze 1.6.0 byla vydána dne 27. prosince 2010. Bližší informace o tomto GIS software jsou k dispozici na <http://www.qgis.org/>.

K analýze biologické rozmanitosti slouží geografický informační systém DIVA-GIS. Pomocí tohoto geografického informačního systému lze vytvářet nejen mapy světa, ale i mapy velmi malých oblastí. Do mapy lze přidávat hranice států, řeky, satelitní snímky a naleziště míst jednotlivých druhů rostlin a živočichů. K dispozici jsou i prostorová data z celého světa. Poslední verzí je verze 7.4. Bližší informace o DIVA-GIS software jsou na <http://www.diva-gis.org/>.

Kromě uvedených geografických informačních systémů lze k analýze prostorových dat použít systém Kristýna-GIS. Kristýna-GIS je freeware aplikace, pomocí níž můžeme vytvářet mapy. Hlavním nástrojem analýzy dat je práce s dotazy. K dispozici je Kristýna-GIS prohlížečka (freeware) a Kristýna-GIS 3.1 (shareware). Bližší informace o systému Kristýna-GIS jsou k dispozici na <http://www.christine-gis.com/cz/>.

7 Praktické ukázky

Další část příspěvku bude popisovat možné aplikace geografických informačních systémů (nebo lépe řečeno ne přímo jejich podoby) v předmětu Informatika ve veřejné správě. Vzhledem k tomu, že se studenti seznamují i s nejpoužívanějším informačním zdrojem ve veřejné správě, Portálem veřejné správy České republiky, bude potřeba jejich znalosti rozšířit i o mapové služby. Příspěvek bude charakterizovat ilustrativní příklady, které byly vytvořeny pomocí Portálu veřejné správy, <http://geoportal.gov.cz/web/guest/welcome> [1] a prostřednictvím internetové stránky <http://www.mapy.cz/> [2].

Vezměme si Rusavu. Při pokusu najít přesnou polohu rekreačního zařízení Rusava (chata Jestřabí) budeme úspěšní přes internetové stránky <http://www.mapy.cz/>. Pokud se o totéž pokusíme přes Portál veřejné správy, zobrazí se jen turistická mapa. Práce s mapovými službami není vůbec složitá. Mapové služby spustíme kliknutím na odkaz Mapy v pravé horní části Portálu veřejné správy. Největší část této aplikace tvoří prostor pro zobrazení mapy. Postupným kreslením obdélníků pomocí myši specifikujeme region, který chceme prozkoumat. Jakmile jsme s výběrem zcela spokojeni, vybereme z části Funkce aplikace odkaz na obrázek tří map, který představuje seznam tematických úloh. K dispozici je jich celkem padesát.

Velmi důležitou součástí aplikace jsou karty Vrstvy a Legenda. Na kartě Vrstvy lze vybrat, co konkrétního si přejeme zobrazit, na kartě Legenda vidíme vysvětlení popisků mapy a vybrané statistické údaje. Zrušením zaškrtnutí před příslušnou vrstvou odstraníme z mapy všechno, co v ní vidět nechceme. Změna se však provede pouze překreslením mapy (zakulacená šipka nacházející se vlevo od karty Vrstvy). Podle barevného rozlišení jsme schopni zjistit potřebné informace. Rozsah využívaných dat je však omezen pouze na údaje Českého statistického úřadu a na informace týkající se životního prostředí. Přesto můžeme zjišťovat velmi zajímavé informace.

Budeme-li se zajímat o hustotu zalidnění v okolí Rusavy, vybereme tematickou úlohu Hustota zalidnění. Z karty Vrstvy necháme zobrazeny hranice územních jednotek, obce a komunikace a hustotu zalidnění (obr. 1). Z obrázku je patrné, že v okolí Rusavy je hustota obyvatel asi 2 až 14 obyvatel na km².

Autorku příspěvku také zajímalo, zda má obec Rusava veřejnou knihovnu. Přímo v Rusavě i v jejím okolí se nachází jedna knihovna (obr. 2). Další zajímavou otázkou je vybavení obce Rusava kanalizací. K velkému překvapení bylo zjištěno, že v okolí Rusavy není kanalizace vybudována vůbec (obr. 3).

K dalším součástí mapových služeb patří i vektorová mapa pozemních komunikací. Tato mapa umožňuje zkoumat, jaké objekty se v dané lokalitě vyskytují. Z obrázku 4 je patrné, že v Rusavě je pouze 7 mostů (malá kolečka na silnici).

Jinou, velmi zajímavou možností, je měření vzdálenosti mezi dvěma náhodně zvolenými místy. Protože máme zájem vidět dvě konkrétní místa v okolí Rusavy, musíme si vybrat jiný, způsob zobrazení mapy. Zobrazení podrobnějších informací nejlépe nabízí topografická mapa Armády České republiky. Hledáme možnosti, jak se dostat z místa A (zelený text RUSAVA pod kótou 536 vlevo nad obcí Rusava) do místa B (část Ráztoka, kóta 422). První možností je chůze po silnici (obr. 5). Délka trasy je delší než 4 km (přesněji 4406 m). Druhou možností je vydat se přímo bez ohledu na překážky v terénu, což bude činit přes 2,5 km (přesněji 2566 m). To znázorňuje obrázek 6.

To, co mapové služby Portálu veřejné správy nedovedou, je plánování tras. Zde byly použity údaje internetové stránky <http://www.mapy.cz/>. Můžeme si naplánovat trasu ze Zlína do Rusavy a zjišťovat jednak nejrychlejší trasu, která trvá 40 minut a její délka je 30,88 km a jednak nejkratší trasu, která trvá 45 minut a její délka je téměř 30 km (přesněji 29,48 km). Můžeme si detailně prohlédnout celou trasu.

Internetová stránka <http://www.mapy.cz/> umožňuje také vyhledávat nejbližší objekty v okolí. Pokud se budeme zajímat o nejbližší ubytování v Rusavě, najdeme celkem tři možnosti, které jsou označeny číselně. V pravé části obrazovky si můžeme prohlížet i detailní informace o daném místě. Chata Jestřabí však nebyla v seznamu ubytování na Rusavě nalezena.

Po zadání klíčového slova „rekreační zařízení“ se chata Jestřabí již objeví. Nyní si z tohoto místa naplánujeme cestu do Zlína. Výsledkem bude popis trasy s uvedením počtu kilometrů na jednotlivých úsecích trasy a odhadovaný čas přepravy. Celková délka trasy bude cca 31 km (přesněji 30,88 km) a cesta bude trvat 40 minut, což je nejrychlejší cesta.

Jak již bylo řečeno, je rozsah zjišťovaných informací, které jsou k dispozici na Portálu veřejné správy, dost omezený. Můžeme čerpat podklady o životním prostředí, údaje z integrovaného registru znečišťování, hranice územních jednotek, volebních obvodů a příslušných úřadů, lze využívat vojenské i staré mapy, údaje Českého statistického úřadu týkající se obyvatelstva a vybavenosti obcí, údaje České pošty, zjišťovat informace o kvalitě koupacích vod a omezené informace o dopravě. I když není tématických úloh tak velký počet, umožňují i tyto informace pomoci nalézt odpovědi na otázky, které nás zajímají. Uživatelé, kteří nedovedou ještě pracovat s geografickými informačními systémy, získají alespoň základní přehled možností mapových služeb Portálu veřejné správy i informační stránky <http://www.mapy.cz/>. Dalším krokem pro ně již bude naučit se pracovat s konkrétním geografickým informačním systémem.

8 Další možnosti výuky předmětu Informatika ve veřejné správě

Z praktických ukázek je patrné, že si geografické informační systémy svou pozornost určitě zaslouhují. Studenti prezenční formy studia si již vyzkoušeli práci s produktem ArcExplorer.

Studenti kombinované formy studia navrhovali zařazení geografických informačních systémů do výuky. Studenty oboru Veřejná správa a regionální rozvoj bude vhodné naučit pracovat prakticky s konkrétními geografickými informačními systémy minimálně v rozsahu dvou až tří cvičení.

Z komerčních aplikací se uvažuje o geografickém informačním systému ArcGIS. Z otevřených GIS se uvažuje o systémech GRASS a QGIS. Systémy Kristýna-GIS a DIVA-GIS budou určeny pro pokročilejší uživatele. Studenti se naučí samostatně vytvářet a upravovat mapy, získávat data z různých zdrojů, data editovat, pracovat s dotazy a v případě zájmu se seznámí s podrobnějšími metodami analýzy prostorových dat.

Obsah předmětu lze rozšířit o analýzu rizik ve veřejné správě a aplikaci kvantitativních metod ve veřejné správě. Nabízí se celá řada možností, které obohacují obsah tohoto předmětu ve všech jeho stránkách. Předmět může být i inspirací pro zpracování bakalářských i diplomových prací a současně jednou z oblastí vědeckovýzkumné činnosti ústavu.

Autorka příspěvku přivítá jakékoliv podněty i kritické připomínky týkající se geografických informačních systémů a problematiky veřejné správy.

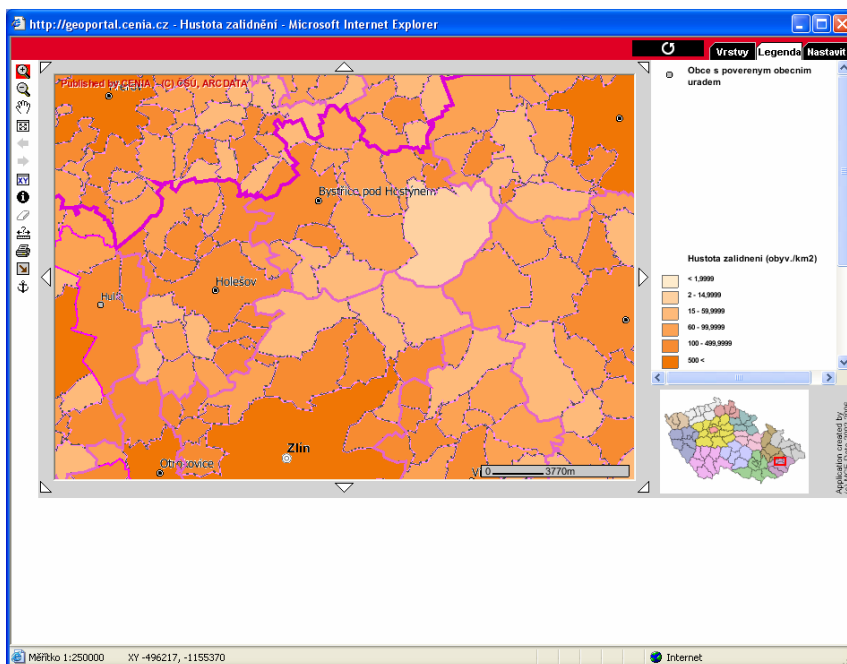
Literatura

- [1] Ministerstvo vnitra. *Portál veřejné správy České republiky* [online]. 2003–2011 [cit. 2011-06-01]. Dostupný z WWW: <http://geoportal.cenia.cz/mapmaker/cenia/portal/>
- [2] Seznam.cz. *Mapy.cz – mapa Evropy, České republiky, plány měst a obcí v ČR* [online]. 1996–2011 [cit. 2011-06-01]. Dostupný z WWW: <http://www.mapy.cz/>
- [3] MAGUIRE, D. J. An Overview and Definition of GIS. In D. J. Maguire, M. F. Goodchild, & D. W. Rhind (Eds.). *Geographical Information Systems, Volume 1 : Principles*. Volume 1 : Principles (Vol. 1, pp. 9-20). Harlow, Essex, England: Longman Scientific & Technical. Dostupný z WWW: http://www.wiley.com/legacy/wileychi/gis/Volume1/BB1v1_ch1.pdf
- [4] RAPANT, P. *Úvod do geografických informačních systémů*. Ostrava: VŠB-TU Ostrava, 2002. Dostupný z WWW: <http://gis.vsb.cz/publikace/ugis>
- [5] VOŽENÍLEK, Vít. *Geografické informační systémy I : pojetí, historie, základní komponenty*. 1. vyd. Olomouc : Univerzita Palackého, 2000. 173 s.

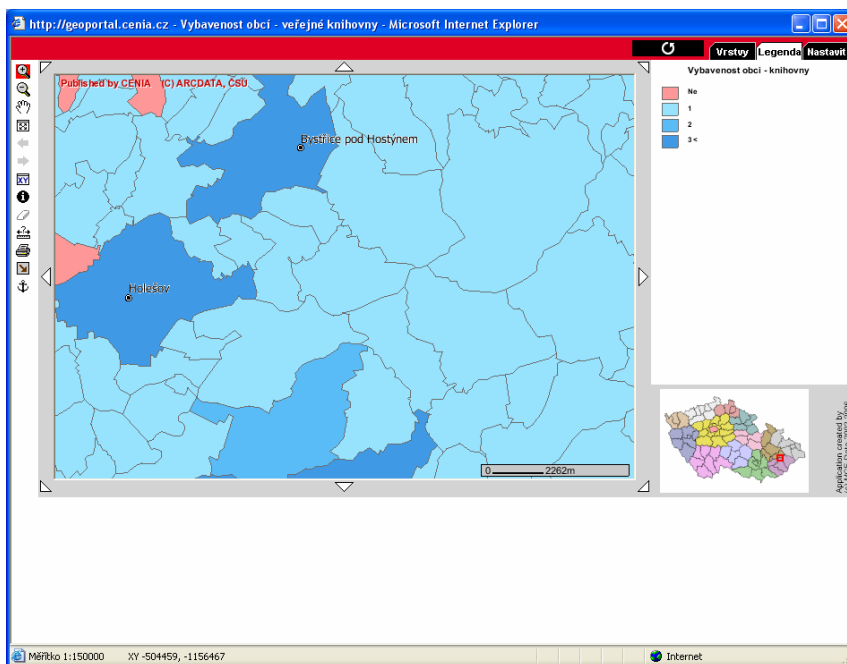
Kontaktní adresa

Miroslava DOLEJŠOVÁ (Ing., Ph.D.),

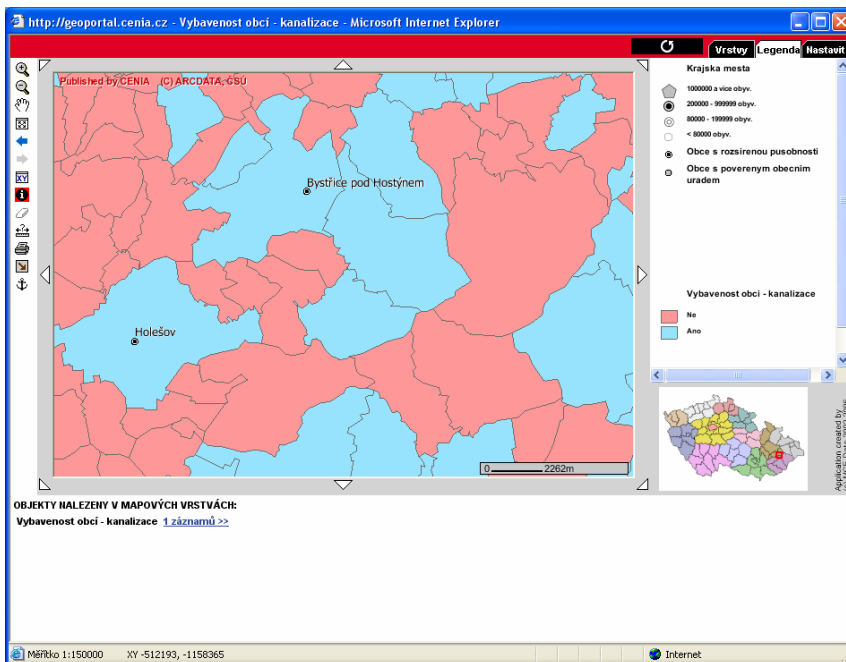
Tomas Bata University in Zlín, nám. T. G. Masaryka 5555,
76001 Zlín, Czech Republic, dolejsova@fame.utb.cz



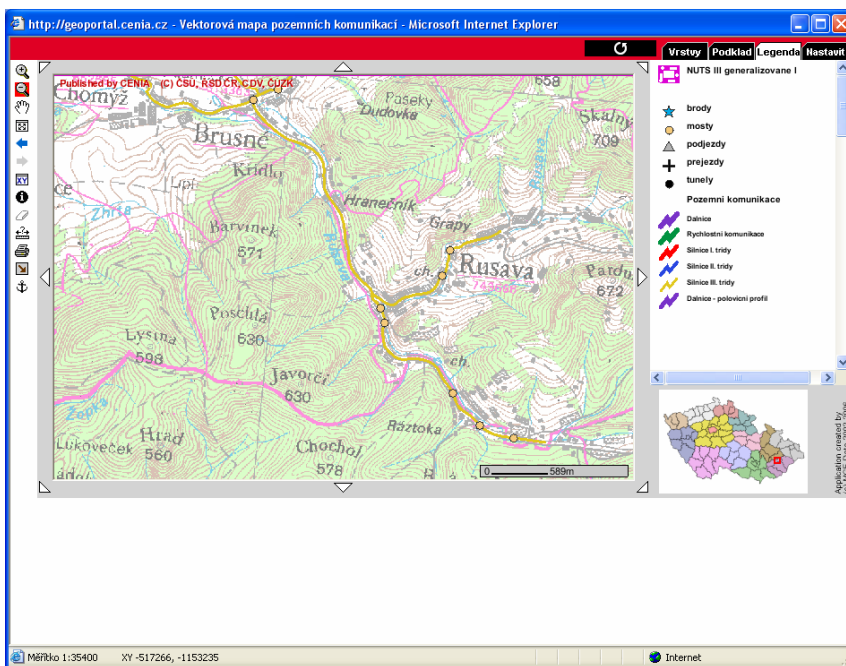
Obrázek 1: Hustota zalidnění v obci Rusava [1]



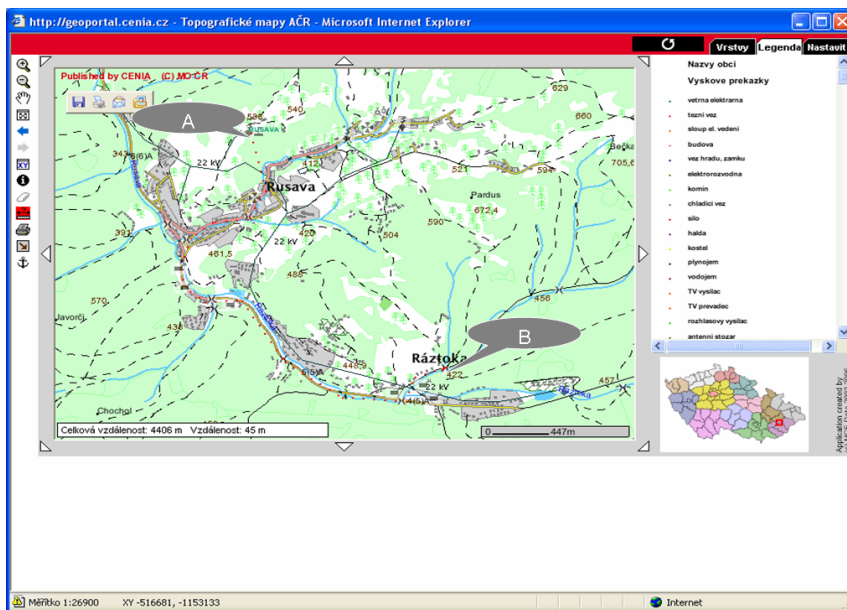
Obrázek 2: Vybavenost obce Rusava veřejnou knihovnou [1]



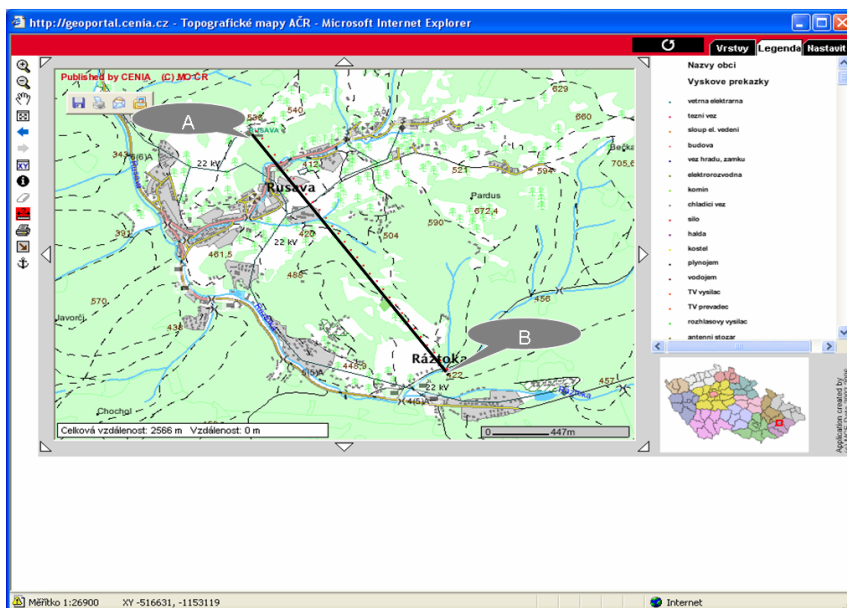
Obrázek 3: Vybavenost obce Rusava kanalizací [1]



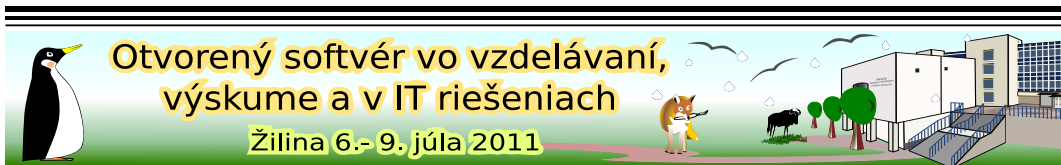
Obrázek 4: Vektorová mapa pozemních komunikací [1]



Obrázek 5: Měření vzdálenosti mezi dvěma zvolenými místy: delší trasa [1]



Obrázek 6: Měření vzdálenosti mezi dvěma zvolenými místy: přímá trasa [1]



DOI: 10.5300/2011-OSSConf/143

VYUŽITIE OPEN SOURCE TECHNOLOGIÍ PRI VÝVOJI A APLIKÁCIÍ WEBOVÝCH GIS RIEŠENÍ NA PRÍKLADE SYSTÉMU GISPLAN 2011

HOFIERKA, Jaroslav (SK); MIČANÍK, Rastislav (SK)

Abstrakt. Prudký rozvoj v oblasti open-source technológií priniesol nové možnosti pri vývoji softvérových riešení pre spracovanie a publikáciu geografických informácií prostredníctvom webu. Príkladom praktického a efektívneho využitia takýchto technológií na Slovensku je webový systém Gisplan. Tento systém umožňuje rýchle a flexibilné zobrazovanie rastrových a vektorových údajov a zároveň poskytuje štandardné funkcie a nástroje pre interakciu s geografickými údajmi. Systém je navrhnutý tak, aby plnil svoj účel v rámci intranetu prevádzkovateľa, ale aj pre externých používateľov, v prostredí internetu. Prevádzkovateľ systému môže flexibilne využívať svoje geografické údaje a zároveň poskytovať prístup k týmto údajom aj verejnosti jednoducho prostredníctvom webového prehliadača. Systém bol spracovaný aj pre mobilné technológie na báze operačného systému Android, čo vytvára nové možnosti využívania technológie a geografických údajov v spojení s technológiou GPS priamo v teréne.

Kľúčové slová. GIS, web, Gisplan.

THE USE OF OPEN-SOURCE TECHNOLOGIES IN THE DEVELOPMENT AND APPLICATION OF WEB-BASED GIS SOLUTIONS ON THE EXAMPLE OF THE GISPLAN SYSTEM 2011

Abstract. Rapid development in the open-source technology has brought new possibilities in the development of software solutions for processing and publication of geographic information on the web. The Gisplan web system is such an example for useful and effective use of these technologies. The system provides a fast and flexible visualization of raster and vector data and at the same time provides standard functions and tools for interaction with geographic data. The system has been designed to operate within the user's intranet or also to benefit external users on the Internet. So the operator of the system has the flexibility to provide an access to the data also for public users via the standard web browsers. The system has been also developed for a mobile technology based on the Android operating system providing new possibilities for the use of this technology and geographic data along with the GPS technology during various field works.

Key words and phrases. GIS, web, Gisplan.

1 Úvod

Open-source technológie nachádzajú široké uplatnenie aj v oblasti geopriestorových technológií [1], [2], [3]. Používatelia majú možnosť priamo využívať rôzne aplikácie alebo nástroje, či už v oblasti operačných systémov, databázových riešení alebo konkrétnych špecializovaných aplikácií (napr. GRASS, SAGA, QGIS a podobne). Špecifickosť využívania geopriestorových technológií spočíva v nutnosti kombinácie viacerých technologických prvkov, ktorá bude vytvárať softvérovú zložku geografického informačného systému (GIS). I keď open-source poskytuje nesporné výhody pre určitú skupinu používateľov (otvorenosť, flexibilita, nízke náklady na obstaranie), konkrétne prevádzkové podmienky mnohých používateľov neumožňujú vytvárať zložitejšie riešenia vo vlastnej réžii a len na báze jedného produktu. Títo používatelia potrebujú na prácu s geografickými údajmi rozhranie, ktoré im bude dostatočne známe a príjemné a bude od nich vyžadovať len základné znalosti technologického pozadia a metód spracovania geografických údajov.

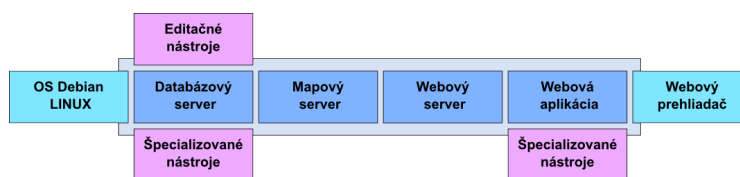
Takýmto riešením je webová technológia so štandardným webovým prehliadačom. Používateľom poskytuje možnosti prehľadného zobrazenia veľkého množstva mapových a priestorových informácií vo vrstvách, rýchle vyhľadávanie a identifikovanie objektov spolu s ich atribútovými vlastnosťami. Zároveň ponúka sadu základných nástrojov pre prácu s mapou (napr. meranie dĺžok a plôch, tlač mapových výstupov a podobne). Táto funkcionality je používateľom dostupná v predpripravenej, veľmi jednoduchej a intuitívnej forme. Tento trend je podporený aj existenciou populárnych mapových služieb ako sú Google Maps alebo OpenStreetMap, ktoré sú výbornou ukážkou prieniku webových GIS technológií do každodenného života širokého spektra používateľov [4]. Celosvetový trend nástupu webových GIS technológií je tiež v značnej miere vyvolaný existenciou vyzretých open-source technológií ako je PostGIS, UMN MapServer, GeoServer či OpenLayers a moderných serverových technológií Cloud Computingu, ktoré umožňujú pokročilé implementácie mapových služieb vo veľkom rozsahu za cenovo veľmi priaznivých podmienok. Webové GIS systémy tak postupne nahrádzajú klasické desktopové softvérové riešenia a systémy a prispievajú tak k rapidnému nárastu používateľov GIS aj medzi bežnou verejnosťou, bez nutnosti veľkých odborných znalostí z oblasti GIS-ov a geopriestorových technológií všeobecne.

Tento trend sa prejavil aj v oblasti geopriestorového softvéru pre informačné systémy o území pre samosprávy a obchodné spoločnosti. Je evidentný posun od úzko špecializovaných pracovísk geodetov, architektov či GIS špecialistov k celopodnikovým GIS systémom, ktorý využívajú desiatky aktívnych používateľov podľa aktuálnych potrieb. Celkovo možno konštatovať, že webové GIS systémy sa v posledných rokoch zaradili medzi žiadané a využívané informačné systémy, čo v konečnom dôsledku otvorilo aj nové možnosti aplikácie open-source riešení v tomto softvérovom segmente.

Do tejto kategórie open-source GIS riešení môžeme zaradiť aj systém Gisplan od spoločnosti GISTA s.r.o. zameraný na prácu s geografickou informáciou vo webovom prostredí. Cieľom tohto príspevku je podrobnejšie charakterizovať toto GIS riešenie, prezentovať oblasť jeho aplikácie a načrtnúť jeho ďalšie smerovanie do budúcnosti.

2 Webový systém Gisplan

Analýza potrieb významnej skupiny používateľov ukázala, že webový systém predstavuje v súčasnosti optimálny spôsob komunikácie geografickej informácie. Spoločnosť GISTA, s.r.o. preto od roku 2008 vyvíja vlastný softvérový produkt na báze open-source technológií s názvom Gisplan (www.gisplan.sk). Tento produkt je po systémovej stránke tvorený operačným systémom Debian Linux, databázovým serverom PostGIS, mapovým serverom UMN Mapserver a webovým serverom Apache. Webové rozhranie aplikácie je postavené na technológii OpenLayers. Schematizované riešenie systému Gisplan je na obr. 1.



Obr. 1: Základné systémove zložky softvéru Gisplan

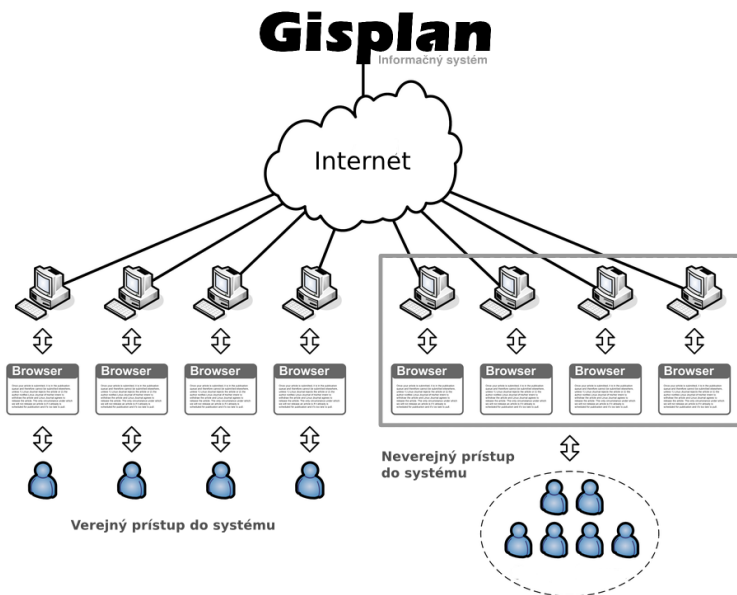
Celý systém Gisplan beží na operačnom systéme Debian Linux (www.debian.org). Jedná sa o všestranný, výkonný, bezpečný a maximálne flexibilný operačný systém využívaný vo svete predovšetkým v serverových riešeniach.

Databázovým riešením systému Gisplan je open-source objektovo-relačný databázový systém PostgreSQL (www.postgresql.org) a jeho open-source nadstavba PostGIS (www.postgis.org). Aplikácia PostGIS slúži na zabezpečenie podpory pre špecifické geografické údaje v databáze PostgreSQL. Rozširuje teda možnosti PostgreSQL serveru tak, aby ho bolo možné využívať ako priestorovú databázu pre GIS a pridáva množstvo pokročilých funkcií pre analýzu priestorových objektov. Geografické údaje uložené v systéme PostGIS je možné jednoducho editovať, ukladať alebo exportovať do viacerých formátov. Môžu byť využívané množstvom desktopových komerčných i open-source GIS systémov a tiež mapovými webaplikáciami prostredníctvom UMN Mapservera alebo GeoServera.

O samotné publikovanie geografických údajov na internete sa stará ďalšia open-source aplikácia, mapový server UMN MapServer (www.mapserver.org). Prostredníctvom webového servera prijíma požiadavku používateľa a vytvára výstup v podobe obrázku pre samotný obsah mapy, legendy, mierku a podobne. Samotný mapový server je vo svojom základe programom CGI, no môže byť rozšírený aj o aplikáciu MapScript, ktorá umožňuje programovať aplikačné programové rozhranie MapServeru v rôznych skriptovacích jazykoch (napr. Python, PP, Perl a iných), čo rozširuje možnosti dynamických výstupov mapového servera.

Vizualizačnú zložku systému Gisplan má na starosti webová aplikácia. Jej základom je technológia OpenLayers (www.openlayers.org), čo je vlastne JavaScriptová knižnica, ktorá zabezpečuje zobrazenie mapových údajov vo väčšine súčasných webových prehliadačov (napr. Internet Explorer, Mozilla Firefox, Google Chrome a iné). Podobne ako v Google

Maps, aj OpenLayers poskytuje aplikačné programové rozhranie (API), ktoré umožňuje integrovať ovládacie prvky a funkcie OpenLayers do vlastnej webovej aplikácie.



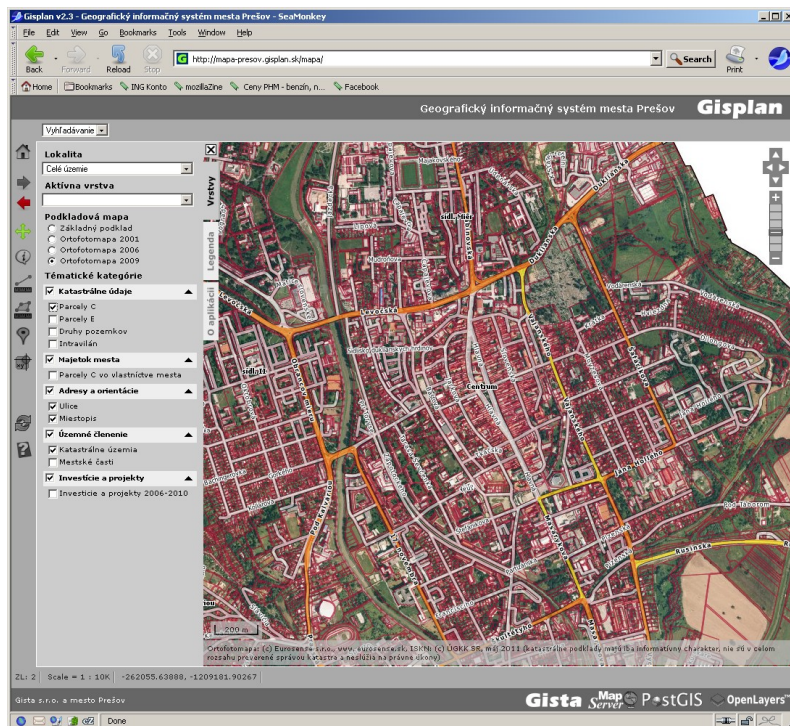
Obr. 2: Gisplan umožňuje jednoducho kontrolovať verejný a neverejný prístup k údajom

Webová aplikácia systému Gisplan môže byť pre používateľov prístupná vo neverejnom aj verejnom rozhraní. Ľahko a intuitívne sa ovláda, nevyžaduje žiadne odborné znalosti a nemá špeciálne nároky na výkon počítača na strane klienta. Svojím otvoreným a flexibilným riešením odstraňuje bariéry využívania GIS technológií plynúcich z používateľskej náročnosti desktopových GIS, umožňuje zobrazovanie rastrových a aj vektorových údajov, poskytuje štandardné funkcie a nástroje pre ich využívanie (napr. výber vrstiev, zmeny mierky zobrazenia, vyhľadávanie, identifikácia a zobrazenie atribútových údajov, meranie vzdialeností a plôch, vkladanie bodov, tvorba tlačových výstupov). Na strane správcu poskytuje jednoduché a prehľadné webové správčovské rozhranie. Celkové riešenie je preto veľmi flexibilné, škálovateľné pre rôzne typy a množstvo údajov.

System Gisplan je spoločnosťou GISTA s.r.o. poskytovaný ako komplexná softvérová služba na báze Cloud Computingu dostupná prostredníctvom internetu a bežného webového prehliadača.

3 Aplikácie

Webový systém Gisplan je primárne určený na rýchlu komunikáciu geografických údajov v rastrovom alebo vektorovom formáte bez obmedzenia cieľa alebo účelu informačného systému. Môže tvoriť základnú softvérovú zložku GIS-u ktoréhokoľvek informačného systému

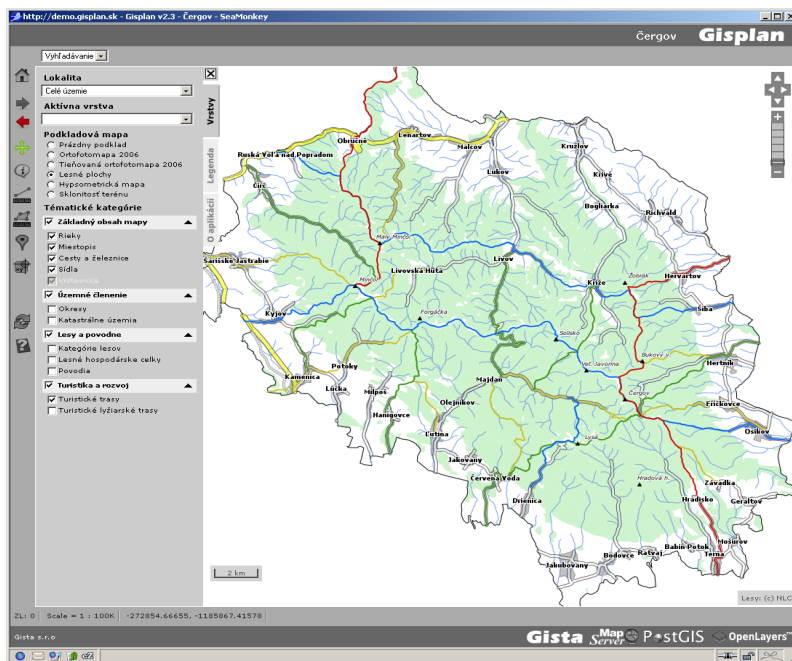


Obr. 3: Základné používateľské rozhranie systému Gisplan
(<http://mapa-presov.gisplan.sk/mapa/>)

o území a teda je určený všetkým používateľom pracujúcim s geografickým (priestorovo lokalizovanými údajmi). V súčasnosti prevažuje nasadenie tohto systému v samosprávach miest. Systém bol úspešne nasadený v mestách Prešov, Bardejov, Banská Bystrica, Snina, Stará Ľubovňa a ďalších. Aktívne sa využíva aj v rámci správy infraštruktúry a záujmového územia v prostredí mestských polícii a mestských podnikov.

Z obsahového hľadiska Gisplan poskytuje základné údaje o území v podobe ortofotomáp z rôznych časových období, čo umožňuje lepšie sledovať dynamiku zmien v zastavanom území. Jeho dôležitou súčasťou sú tiež katastrálne mapy a ich výstupy (napr. parcely vo vlastníctve mesta), adresy, budovy, uličné systémy, územné členenia ako aj špecializované mapové podklady napr. pre potreby mestskej polície, správy mestskej zelene, komunikácií a podobne. Obsah je vždy individuálne prispôbovaný aktuálnym potrebám a požiadavkám.

Gisplan je možné využiť aj pre špeciálne potreby, napr. v rámci podpory regionálneho rozvoja alebo ochrany prírody. Napríklad lesoochránárske združenie Vlk využíva Gisplan na evidenciu základných údajov o pohorí Čergov. Okrem základných topografických mapových prvkov obsahuje aj podrobnejšie údaje o lesoch a turistických trasách (obr. 4).



Obr. 4: Gisplan pre pohorie Čergov (<http://demo.gisplan.sk/>)

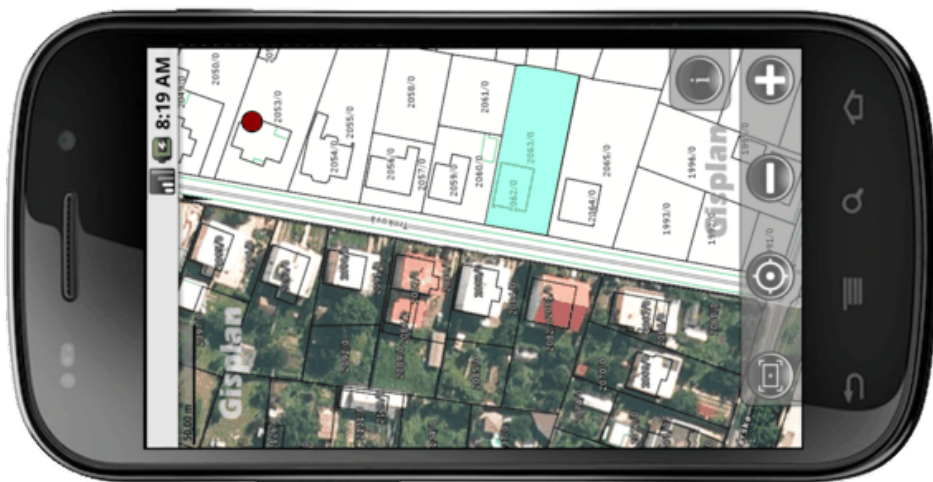
Jedným z významných faktorov úspešného nasadenia GIS-u v praxi je využívanie geografických údajov nielen pre potreby danej organizácie, ale aj pre iných používateľov mimo danej organizácie prostredníctvom verejného rozhrania. Napríklad mestá potrebujú komunikovať so svojimi občanmi, poskytovať informácie o územnom pláne, volebných obvodoch a podobne a najlepším spôsobom, ako takéto informácie zverejniť je prostredníctvom webu. Navyše, schválením európskej smernice INSPIRE o infraštruktúre pre priestorové informácie v Európskom spoločenstve č. 2007/2/EC a zákona NR SR č. 3/2010 Z.z. o národnej infraštruktúre pre priestorové informácie sa pre mnohé verejné inštitúcie stalo povinnosťou poskytovať základné informácie o geografických údajoch, ktoré vytvárajú alebo spravujú. Predpokladá sa, že prostredie webu bude základným infraštruktúrnym prostredím pre poskytovanie tohto typu údajov. To svedčí o tom, že záujem o tento typ riešení bude stúpať a poskytovanie geografických informácií prostredníctvom webu sa stane štandardom pre rôzne organizácie.

Explozívny rozvoj mobilnej komunikácie za posledné desaťročie priniesol nové možnosti aj pre geopriestorové technológie. Cenová dostupnosť technológie GPS na určovanie polohy a navigáciu spôsobila, že zariadenia GPS sa stávajú bežnou súčasťou mobilných telefónov. Mobilný telefón s GPS a digitálnou mapou sa stáva aj navigačným zariadením, ktoré je možné využiť na vyhľadávanie lokalít podľa adresy, stanoviť optimálne trasy na dosiahnutie cieľovej lokality a pod.. Túto mobilnú technológiu podporujú napríklad spoločnosti Nokia

(OviMaps) alebo Google (Google Maps). Spoločnosť GISTA, s.r.o. vyvinula vlastné riešenie na podporu systému Gisplan určené pre mobilné technológie na báze operačného systému Android (obr. 5).

Toto riešenie nazvané Gisplan do vrecka je určené na praktické využívanie špecializovaných digitálnych máp a databázových údajov v teréne prostredníctvom internetu a malých prenosných zariadení (inteligentných telefónov alebo tabletov). Používateľom poskytuje mobilný prístup do systému Gisplan a umožňuje tak flexibilnú a efektívnu prácu s údajmi, ktoré sú v ňom uložené. Ponúka pritom výber rozličných mapových vrstiev, identifikáciu objektov v mape a zobrazenie ich databázových údajov a určenie presnej polohy používateľa pomocou GPS a digitálneho kompasu. Jeho ovládanie je jednoducho riešené pomocou moderného dotykového displeja. Gisplan do vrecka komunikuje s centrálnym serverom systému Gisplan prostredníctvom GSM alebo WiFi sietí.

Predpokladá sa, že táto mobilná aplikácia nájde uplatnenie najmä medzi pracovníkmi inštitúcií pracujúcich priamo v teréne (sledovanie polohy pracovníkov, obhliadky lokalít pri vydávaní povolení, mestská polícia a pod.).



Obr. 5: Gisplan do vrecka umožňuje využívať mobilné GIS technológie priamo na displeji mobilného telefónu vybaveného prijímačom GPS

4 Záver

Súčasný rozvoj geopriestorových technológií, nová legislatíva v oblasti geografických informácií, ako aj široké využívanie a dostupnosť internetu a webových služieb vytvorilo vhodné podmienky pre rozvoj webových GIS riešení. Jedným z úspešných príkladov je webový systém Gisplan, v súčasnosti využívaný na prácu s geografickými údajmi v niektorých mestách na Slovensku. Open-source technológie v systéme Gisplan tvoria základnú

systémovú kostru, ktorá umožňuje veľmi flexibilne naplňovať údajovú zložku systému podľa potrieb zákazníka a zároveň elegantným spôsobom saturovať potrebu komunikácie vybraných údajov aj pre verejnosť. Bezprecedentný rozvoj mobilnej technológie našiel svoju odozvu aj v riešeníach pre systém Gisplan, a to na báze operačného systému Android. Ďalší rozvoj tejto technológie sa zameria najmä využívanie pokročilejších priestorových funkcií GIS-u, čo významným spôsobom rozšíri analytické funkcie Gisplan-u a aplikácie v oblasti priestorového rozhodovania. Zvyšuje sa aj záujem o publikovanie 3D údajov (napr. vo podobe virtuálnych 3D modelov miest), čo otvára ďalšie možnosti v aplikácii tejto technológie v segmente virtuálneho 3D modelovania.

PodĎakovanie

Príspevok je súčasťou riešenia projektu APVV VMSP-II-0019-09 a VEGA 1/0355/09.

Literatúra

- [1] NETELER, M., MITASOVA, H.: *Open Source GIS: A GRASS GIS Approach*, Kluwer Academic Publishers, Boston, 2004.
- [2] HOFIERKA, J.: *Geografické informačné systémy a diaľkový prieskum Zeme*. Vysokoškolské učebné texty. Fakulta humanitných a prírodných vied Prešovskej univerzity, Prešov, 2003.
- [3] HOFIERKA, J.: *Geopriestorové a kartografické technológie v regionálnom rozvoji*. In: Michaeli, E., Matlovič, R., Ištók, R. (eds.): *Regionálny rozvoj a regionálne politika pre geografov*. Vysokoškolská učebnica. Prešovská univerzita v Prešove, Prešov, 2010, s. 225–292, ISBN: 978-80-555-0065-2.
- [4] HOFIERKA, J.: *Nové možnosti komunikácie geografickej informácie pomocou geopriestorovej internetovej technológie Google Maps a Google Earth*. Acta Facultatis Studiorum Humanitatis et Naturae Universitatis Prešoviensis. Prírodné vedy, Folia Geographica 17, 2011, s. 100–108.

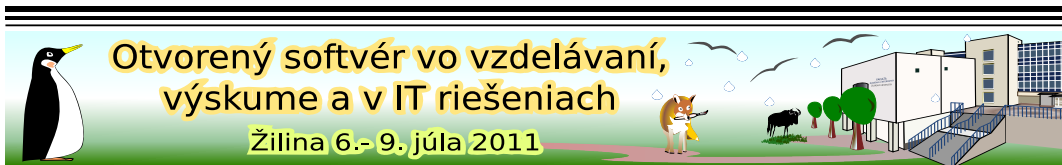
Kontaktná adresa

Jaroslav HOFIERKA (doc., Mgr., PhD.),

GISTA, s. r. o., Hlavná 133, Prešovská univerzita v Prešove, Ul. 17. novembra 1
08001 Prešov, jhofierka@gmail.com

Rastislav MIČANÍK,

GISTA, s. r. o., Hlavná 133, Prešovská univerzita v Prešove, Ul. 17. novembra 1
08001 Prešov, rastislav.micanik@gista.sk



DOI: 10.5300/2011-OSSConf/151

VYUŽITIE SLOBODNÝCH SOFTVÉROV V NÁVRHU KONCEPCIE ROZVOJA FAKULTY V OBLASTI GIS

JAKAB, Imrich, (SK), BUGYA, Titusz, (HU)

Abstrakt. *Katedra ekológie a environmentalistiky na Fakulte prírodných vied Univerzity Konštantína Filozofa v Nitre vypracovala začiatkom roku 2011 novú koncepciu rozvoja v oblasti Geografických informačných systémov (GIS). Návrh koncepcie v sebe zahŕňa inováciu v oblasti výučby (zmeny v oblasti obsahu a formy vzdelávania), softvérovej podpory (orientácia na otvorený GIS softvér vo vyučovaní), organizácie GIS laboratória (prispôbenie súčasného GIS laboratória plánovaným zmenám), vznik fakultnej GIS databanky (konceptia tvorby, editácie, prístupu a zdieľania dát) a spoluprácu v oblasti GIS na fakultnej, univerzítnej a medzi-univerzítnej úrovni. Základnou pohnútkou pre zmenu súčasného systému a pre vytvorenie novej koncepcie je okrem zlepšenia súčasného stavu aj spolupráca s Inštitútom geografie na Univerzite v Pécsi v Maďarsku v rámci spoločných vedeckých a edukačných projektoch KEGA 030UKF-4/2011 and TÁMOP-4.2.2/08/1/2008-0011 IKT – Science, Please! Innovative Resarch Team.*

Kľúčové slová. *Koncepcia, GIS, otvorený softvér, laboratórium, databanka, spolupráca.*

FACULTY DEVELOPMENT CONCEPTION IN THE FIELD OF GIS USING OPEN SOURCE GIS SOFTWARES

Abstract. *Department of Ecology and Environmental Sciences at the Faculty of Natural Sciences, Constantine the Philosopher University in Nitra has developed a new concept of development in the field of geographic information systems (GIS). The concept contains the innovation in learning (changes in content and form of education), software support (focus on an open GIS software), the organization of GIS laboratory (adaptation of the current GIS laboratory for the planned changes) the creation of GIS databank (creation, editing, access and data sharing) and cooperation in the field of GIS at the level of faculty, university and on the inter-university level. The main motivation for changing the current system and developing the new concept is to improve the current state based on cooperation with the Institute of Geography at the University of Pécs in Hungary within the joint educational and scientific projects KEGA 030UKF-4/2011 and TÁMOP-4.2.2/08/1/2008-0011 IKT – Science, Please! Innovative Resarch Team.*

Key words and phrases. *GIS, open source, laboratory, databank, cooperation.*

1 Úvod

Spôsoby získavania, spracovania a vizualizácie informácií sa neustále zefektívňujú a modernizujú, na čo by mala reagovať každá univerzita inováciou nielen obsahu, formy vzdelávania ale aj podmienok v ktorých sa samotná výučba realizuje. Pre univerzity je modernizácia vyučovania neľahkou úlohou prispôsobovania sa potrebám odbornej praxe. Od absolventov fakúlt prírodných vied, sa očakáva, že okrem nadobudnutých vedomostí budú disponovať aj určitou úrovňou zručností v oblasti informačno-komunikačných technológií (IKT). Jedným z prioritných zameraní prírodovedného vzdelávania na FPV UKF v Nitre v oblasti IKT je využívanie nástrojov Geografických informačných systémov (GIS). Modernizácia v oblasti GIS v podmienkach univerzity musí tvoriť dobre premyslený systém v ktorom je proces a obsah výučby realizovaný v kvalitne pripravenom prostredí informačných technológií a vhodne vybraných softvérových nástrojov.

2 Geografický informačný systém

Jednoznačná a všeobecne prijateľná definícia pojmu geografický informačný systém v súčasnej dobe stále neexistuje. Väčšina definícií, ktoré boli doposiaľ zostavené, sú silne poznamenané prostredím, z ktorého ich autori pochádzajú.

Jednou z príčin tejto situácie môže byť aj skutočnosť, že rôzni autori definícií vnímajú tento pojem na odlišných úrovniach. Rapant [1] rozlišuje tri úrovne chápania pojmu GIS:

- GIS ako softvér,
- GIS ako konkrétne aplikácie,
- GIS ako informačné technológie.

2.1 Softvérové chápanie pojmu geografický informačný systém

Na tejto najnižšej úrovni je pojem GIS postavený na úroveň programových produktov. Typickým prejavom tohto stavu je preferovanie určitých softvérových nástrojov a predstava, že konkrétny produkt od konkrétnej firmy je to pravé GIS. Súhlasíme s Rapantom [1], s Tučekom [2] a s množstvom ďalších autorov, že používanie pojmu GIS len v tejto súvislosti je nesprávne.

2.2 Geografický informačný systém ako konkrétna aplikácia

Používanie pojmu GIS v tejto rovine uvažovania je celkom oprávnené, pojem GIS je tu možné pomerne ľahko definovať. Odborná literatúra ponúka niekoľko desiatok rôznych definícií. Väčšina z nich je však silne ovplyvnená subjektívnym prístupom autorov. Osobne považujeme za najvhodnejšie definovať pojem GIS podľa Rapanta [1]:

GIS je funkčný celok vytvorený integráciou technických a programových prostriedkov, dát, pracovných postupov, obsluhy, užívateľov a organizačného kontextu, zameraných na zber, ukladanie, správu, analýzu, syntézu a prezentáciu priestorových dát pre potrebu popisu, analýzy, modelovanie a simulácie okolitého sveta s cieľom získať nové informácie potrebné pre racionálnu správu a využívanie tohto sveta.

V oblasti rozvoja GISov na vysokých školách a univerzitách by pravdepodobne táto definícia pri snahe o jej konkretizáciu nadobudla rôzne smery najmä v oblastiach:

- použitých technických a programových prostriedkov,
- využívaných dát a pracovných postupov,
- definovaní práv pre obsluhu, užívateľov a organizačného kontextu,
- v obsahovej a procesuálnej stránke pri správe, analýze, syntéze a prezentácii priestorových dát.

2.3 Geografický informačný systém ako informačné technológie

Jedná sa o najcelistvejšiu rovinu chápania pojmu GIS, v ktorej je však najťažšie tento pojem akokoľvek vymedziť. V podstate by sa malo jednať o celkové prostredie, v ktorom vznikajú aplikácie GIS. Navrhovaná koncepcia vychádza z tohto komplexného prístupu k problematike GIS na FPV UKF v Nitre.

3 Súčasný stav v oblasti GIS na FPV UKF v Nitre

Výučbe v oblasti GIS sa na Fakulte prírodných vied (FPV) venujú dve katedry:

- Katedra ekológie a environmentalistiky (KEE)
- Katedra geografie a regionálneho rozvoja (KGRR)

Výučba GIS prebieha v počítačovej miestnosti, ktorú poznáme pod názvom GIS Laboratórium. K dispozícii je 16 počítačov pracujúcich pod OS Windows. Študenti sa prihlasujú pod kontom „studenti“ ktoré je bez hesla. Účet je síce obmedzený a študenti dokážu zmazať len to, čo bolo vytvorené pod týmto účtom, ale to stačí na to, aby si jednotliví užívatelia navzájom vymazávali zložky a prepisovali dáta.

Pedagóg má na jednotlivých počítačoch rovnaké práva ako študent a ním nahrané dáta určené pre študijné účely môžu byť priebehu niekoľkých vyučovacích hodín zničené a nepoužiteľné. Na počítačoch sa miešajú dáta všetkých študentov a viacerých katedier. Pracovať so študentmi na dlhodobějších projektoch je v takýchto podmienkach sťažené a komplikované.

Miestnosť kvôli absentujúcej autentifikácii študentov môže byť rizikom pre bezpečnosť a integritu siete a môže byť zneužitá na šírenie nelegálneho obsahu, spamu, prípadne inú nežiaducu činnosť.

V súčasnosti je v miestnosti nainštalovaných niekoľko desktop GIS aplikácií. Okrem platených softvérov od firmy ESRI (ArcView 3.2, ArcGIS 9.2), využívaných hlavne Katedrou

geografie a regionálneho rozvoja sa tu objavujú aj otvorené softvéry, v ktorých zabezpečuje výučbu Katedra ekológie a environmentalistiky.

Jedná sa o nasledovný softvér:

- QuantumGIS – Softvér využíva KEE pre výučbu GISov v 2. ročníku bakalárskeho štúdia, ako náhradu za ArcView3.2. QuantumGIS je užívateľsky prívetivá desktop GIS aplikácia s otvoreným kódom, podporujúca vektorové, rastrové a databázové formáty. Umožňuje prezerat' a vytvárat' mapy a podporuje veľké množstvo dátových formátov pre spracovanie geopriestorových informácií (napr. ESRI ShapeFile, GeoTIFF). QGIS umožňuje rozšírenie základných funkcií pomocou modulov (napr. pre prácu s GPS zariadeniami, GRASS GIS funkcionality) [3]. Softvér je bezplatný a umožňuje nastavenie svojho prostredia aj v slovenskom jazyku. Výučba pod týmto softvérom sa na Katedre ekológie a environmentalistiky realizuje od akademického roka 2010–2011.
- GrassGIS – GRASS GIS (Geographic Resources Analysis Support System) je slobodný softvér – geografický informačný systém (GIS) umožňujúci správu a analýzu geopriestorových dát, spracovanie obrazových dát, produkciu mapových výstupov, geopriestorové modelovanie a vizualizáciu dát. GRASS je bežne používaný ako v akademickej tak v komerčnej sfére po celom svete. Je oficiálnym projektom Open Source Geospatial Foundation, multiplatformnou aplikáciou šírenou pod licenciou GNU GPL [4]. Náročnejší užívatelia mnohokrát oceňujú dostupnosť zdrojových programov celého systému, čo im umožňuje vývoj vlastných „grassovských“ programov riešiacich ich špecifické problémy. Preto sa GRASS stáva svojím spôsobom predstaviteľom koncepcie otvoreného GISu (Open GIS) [4]. Výučbu pod GRASS GIS plánujeme začať akademickým rokom 2011–2012. Moduly tohto softvéru určené pre priestorové analýzy plánuje využívať KEE v prvom ročníku magisterského stupňa štúdia.

4 Návrh koncepcie rozvoja GIS

Iniciatíva i samotný návrh koncepcie rozvoja GIS na FPV UKF v Nitre vyšli z dielne Katedry ekológie a environmentalistiky. Základnou pohnútkou pre zmenu súčasného systému a pre vytvorenie novej koncepcie je okrem snahy o zlepšenie súčasného stavu aj spolupráca s Inštitútom geografie na Univerzite v Pécsi v Maďarsku v rámci spoločných vedeckých a edukačných projektov so zameraním na modelovanie a simulácie procesov v krajine s využitím nástrojov GIS. Inštitút má bohaté skúsenosti s výučbou GIS a s využívaním otvorených softvérov nielen vo vyučovacom procese, ale aj vo výskumnej činnosti. V súčasnosti má vytvorené dve GIS laboratóriá pre spolu 48 užívateľov. Počas piatich rokov štúdia majú študenti možnosť používať viaceré slobodné softvéry. Okrem softvérov so zameraním na GIS, kde prioritnú úlohu zohráva softvér GRASS GIS, je pozornosť venovaná aj softvérom QuantumGIS, Merkaartor a OpenJUMP. Študenti sa stretávajú aj s ďalším slobodným softvérom ako OpenOffice.org, Inkscape, GIMP, ImageJ, gedit a s operačným systémom Linux.

Návrh koncepcie rozvoja GIS v sebe zahŕňa inováciu v oblasti:

1. Výučby GIS – zmeny v oblasti obsahu a formy vzdelávania,
2. Softvérovej podpory - používaných GIS softvérov vo vyučovaní,
3. Organizácie GIS laboratória - prispôbenie GIS laboratória plánovaným zmenám,
4. Vzniku katedrovej/fakultnej GIS databanky – koncepcia tvorby, editácie a zdieľania dát využitím spoločnej GIS databanky,
5. Spolupráca v oblasti GIS – spolupráca na fakultnej, univerzitnej a medzi univerzitnej úrovni.

4.1 Koncepcia výučby v oblasti GIS

Potreba inovácie obsahu a formy vzdelávania vychádza zo snahy o modernizáciu vyučovania s dôrazom na požiadavky odbornej praxe. Základnou snahou je vytvoriť viac možností a priestoru pre študentské projekty, podporiť využívanie softvérových nástrojov pre tvorbu mapových výstupov, analytických výpočtov, modelovanie a simulácie, aby boli absolventi schopní využívať nástroje GIS v budúcom zamestnaní. Pritom našim zámerom je učiť študentov porozumieť jednotlivým nástrojom geografických informačných systémov a nie, aby vedeli pracovať s jedným konkrétnym softvérom. Ak študenti rozumejú, čo do systému vstupuje, aké premenné ho ovplyvňujú a prečo, a čo znamenajú hodnoty na výstupe, sú schopní používať viaceré softvéry pre GIS.

Dôležitým faktorom, ktorý ovplyvňuje obsah vzdelávania v oblasti GIS, je problém zabezpečenia kontinuity výučby kvôli prechodu na dvojstupňové štúdium. Na magisterské štúdium sa prijímajú aj študenti, ktorí neštudovali na našej univerzite, čiže nemusia mať dostatočné vedomosti a zručnosti v oblasti GISov. Preto nie je možné plynule nadviazať na učivo z bakalárskeho štúdia. Časť našich študentov zase skončí na úrovni bakalárov a je potrebné, aby si so sebou odniesli nejaký ucelený balík vedomostí a zručností v oblasti GIS a vedeli ho v odbornej praxi využívať. Preto je nevyhnutné rozmýšľať o dvoch dimenziách: o GISoch pre bakalárske štúdium a o GISoch pre magisterské štúdium.

4.1.1 GIS pre bakalárske štúdium

Výučba GIS pre študentov Katedry ekológie a environmentalistiky pozostáva z dvoch úrovní: z teoretickej prípravy pre pochopenie problematiky GISov a z praktickej práce pod GIS softvérom. Cieľom je, aby boli študenti schopní využívať prístroje GPS pri priamom zbere priestorových údajov, ich transformácie a ďalšej editácie, vytvárať vlastné mapové vrstvy použitím digitalizácie mapových podkladov, ich klasifikácie a kartograficky správnej mapovej vizualizácie. Úroveň a rozsah obsahu vzdelávania v oblasti GIS by mala vyplývať z tém bakalárskych prác konkrétnej katedry. Pre potrebu výučby je pre našu katedru (KEE) najideálnejším riešením použitie softvéru Quantum GIS.

Obsah prednášok GIS v bakalárskom štúdiu environmentalistiky tvoria témy:

1. Základná charakteristika GIS
2. Stručný vývoj GIS
3. Aplikácie GIS
4. Geografické údaje, komponenty geografických údajov
5. Modelovanie geo-priestoru
 - Geoid
 - Elipsoid
 - Kartografické zobrazenia
6. Súradnicové systémy
 - Systémy globálne
 - Systémy lokálne
7. Z reálneho sveta do sveta GIS
 - Mapové vrstvy
8. Údajové typy v GISoch
 - Vektorové údaje
 - Rastrové údaje
9. Zber priestorových údajov
 - Priame metódy zberu údajov
 - Nepriame metódy zberu údajov
10. Globálny polohový systém
 - Definície GPS
 - Štruktúra systému GPS
 - Určenie polohy
 - Chyby a nepresnosti v meraní
 - Transformácie medzi systémami
 - Výhody a nevýhody merania s GPS
 - Oblasti využitia GPS
11. Priestorové databázy
12. Vizualizácia a kartografická prezentácia
13. Desktop GIS aplikácie

Obsahom cvičení pre bakalárske štúdium environmentalistiky je:

1. Inštalácia Quantum GIS
2. Pracovné prostredie GIS softvérov na príklade Quantum GIS
 - hlavné menu
 - panel nástrojov
 - stavovo-pomocný riadok
 - prekrývanie vrstiev
 - možnosti zobrazovania
 - nastavenia vlastností prostredia GIS softvérov
 - mapové nastavenia
3. Práca s vektorovými dátami
 - pridávanie vektorových vrstiev
 - symbolika
 - popisy
 - atribúty
 - kartodiagramy
 - vytváranie nových vrstiev – body, línie, polygóny (vo formáte ESRI shapefile)
 - špecifikácia súradnicového systému
 - editácia bodových líniových a polygónových vrstiev
4. Práca s atribútovou tabuľkou
 - selekcia mapových prvkov
 - pridanie nových atribútov k mapovej vrstve
 - dátové typy
 - mapový kalkulátor
5. Práca s rastrovými dátami
 - pridávanie rastrových vrstiev
 - symbolika a paleta farieb
 - možnosti nastavenia priehľadnosti
6. Vizualizácia a kartografická prezentácia
 - správca tvorby dát
7. Spracovanie dát z GPS
 - transformácia dát medzi súradnicovými systémami
 - možnosti pridávania, editácie a vizualizácie vrstvy vo formáte: delimited text layer

4.1.2 GIS pre magisterské štúdium

V magisterskom štúdiu je prioritou nadobudnúť vedomosti a skúsenosti v oblasti priestorových analýz (rastrové a vektorové analýzy), transformácie údajov, 3D analýz, modelovaní a simulácii procesov. Pre potreby výučby vidíme ako najideálnejším riešením použitie softvéru GRASS GIS. Tu je dôležité vysvetliť možnosti editácie dát vytvorených v GRASS GIS v prostredí Quantum GIS a opačne. Súčasný rozsah výučby neposkytuje dostatočný priestor pre realizáciu týchto zámerov. Preto bolo nevyhnutné riešiť možnosti posilnenia oblasti GIS a ponúknuť viac možností pre študentov, ktorí majú o danú problematiku hlbší záujem. Riešením je zavedenie voliteľného predmetu Priestorové analýzy v GIS v magisterskom štúdiu, ktorého výučba sa začne akademickým rokom 2011/2012.

Obsah prednášok a cvičení pre magisterské štúdium environmentalistiky tvoria nasledovné témy (témy, ktoré sú súčasťou voliteľného predmetu sú označené *):

1. Databanka GRASS
2. Mapset a PERMANENT
3. Štruktúra príkazov v GRASS
4. Premenné prostredia GRASS
5. Inštalácia GRASS
6. Založenie location v GRASS
7. Import rastrových, vektorových a bodových dát
8. Práca s rastrovými dátami:
 - Vizualizácia rastrových máp
 - Vybrané aplikácie rastrových máp - výpočet profilov, analýza viditeľnosti, prekryv jednotlivých rastrových máp, rastrové buffery
 - Zmena tabuľky farieb rastrovej mapy
 - Štatistické charakteristiky máp
 - Metódy pre manipuláciu dát s rastrovými mapami – reklasifikácia, maskovanie*
 - Vektorizácia rastrových dát*
9. Správa atribútov vektorových máp
 - Výpis atribútov
 - Priradenie atribútov
 - Manipulácie s atribútovými dátami
10. Práca s vektorovými dátami
 - Funkcia prieniku, prekryvu, spojenia a vystrihnutia
 - Extrakcia dát
 - Správa topológie
 - Vektorizácia v GRASS

- Sieťové analýzy - analýza najkratšej cesty, úloha minimálneho Steinerovho stromu, úloha obchodného cestujúceho, analýza nákladov*

11. Konverzia dát*

- Vektorizácia rastrových dát
- Rasterizácia vektorových dát

12. Interpolácie*

- Interpolácia rastrových dát - Metódy IDW, RST (spline)
- Interpolácie dát vo vektorovom formáte

13. Mapová algebra s modulom r.mapcalc*

14. 3D vizualizácia a animácia v GIS*

15. Vizualizácia dát a mapový výstup

4.2 GIS softvér

V súčasnosti vo výučbe uprednostňovaný softvér ArcGIS vyvoláva niekoľko sporných otázok. Jedná sa o platený softvér a zakúpené práva na multilicenciu programu neobsahujú nástroje pre priestorové analýzy a 3D analýzy, nevyhnutné pre výučbu a vedeckovýskumnú činnosť v oblasti GIS na univerzite. Aj keby sa nakúpila rozšírená verzia, stále sa jedná o platený a nie lacný softvér. Fakulta teda nedokáže zabezpečiť, aby sa študenti mohli doma pripravovať na predmety, aby si mohli na vlastných počítačoch riešiť svoje projekty a plnohodnotne využívať tento systém pre vedeckovýskumnú činnosť spojenú so záverečnými prácami. Možným riešením je sprístupnenie počítačovej miestnosti študentom v čase mimo výučby. Vzhľadom na vysoký počet študentov študujúcich na KGRR a KEE je táto možnosť málo efektívna. Len na KEE sa v zimnom semestri v roku 2010 vystriedalo počas výučby v GIS laboratóriu okolo 150 ľudí. Tento počet predstavujú len študenti absolvujúci predmety GIS. Je však potrebné nechať priestor aj pre študentov, ktorí riešia záverečné práce a cvičenia (semináre) pre iné predmety, v ktorých sa očakávajú mapové výstupy a GIS analýzy.

Ako vhodná alternatíva, ktorá by spomínané problémy do širokej miery eliminovala, je používanie už vyššie spomenutých OpenSource softvérov (QGIS, GrassGIS). Ich používanie vo výučbe by prinieslo niekoľko výhod:

- Jedná sa o profesionálne softvéry, na ktoré univerzita nemusí vyčleniť peniaze na nákup softvéru, nakoľko ich zaobstaranie a používanie je pre univerzity bezplatné,
- Aktualizácia softvéru je takisto bezplatná, takže OpenSource softvér nám po čase nezostarne,
- Otvorenosť softvéru umožňuje bezproblémové exporty a transformácie dát z jedného softvéru do druhého, je možné kombinovať viaceré GIS softvéry pre riešenie jednej úlohy,

- Ich hlavnou výhodou je, že študenti si ich môžu slobodne inštalovať na svojich počítačoch a ich práca pod profesionálnym GIS softvérom nie je limitovaná možnosťami počítačovej miestnosti.
- Ponúknutím určitých práv a možnou dostupnosťou k zdrojovému kódu softvérov sa vývoj GIS nástrojov môže odohrávať aj mimo veľkých firiem. Táto skutočnosť môže byť silným motivujúcim prvkom pre študentov, ktorí môžu vyvíjať nové nástroje a pretvárať existujúce.
- Študenti, ktorí po absolvovaní štúdia prechádzajú do odbornej praxe, sú nielen odborne spôsobilí, ale poznajú možnosti a výhody Open Source softvérov ako alternatívu pre používanie profesionálnych GIS softvérov a sú schopní ich priamo využívať.
- Využitie OpenSource GIS softvérov je výhodnou alternatívou pre celoživotné vzdelávanie učiteľov v oblasti GIS. Pre učiteľov prírodovedných predmetov je používanie GIS softvéru vo vyučovacom procese nižšieho a vyššieho sekundárneho vzdelávania moderným a efektívnym vyučovacím prostriedkom použiteľným v pedagogickej praxi súčasného školstva. Preto je o túto oblasť zvýšený záujem a možnosť vzdelávania v Open Source ponúka okrem vyškolených pedagógov aj bezplatný nástroj pre školy.
- K dispozícii je množstvo dostupných materiálov a manuálov pre užívateľov slobodných softvérov dostupných na webových stránkach, napríklad:
 - Markus Neteler: GIS GRASS – Praktická rukovieť ke geografickému informačnému systému GRASS, dostupné na: http://grass.fsv.cvut.cz/wiki/index.php/GIS_GRASS__Praktická_rukovět
 - Markus Neteler, Oto Dassau, Stephan Holl: GISS GRASS 6.0 Kursskript GDF Hannover, dostupné na: http://grass.fsv.cvut.cz/wiki/index.php/GIS_GRASS_6.0__Praktická_rukovět_záčínajících_uživatelů
 - Sherman, Sutton et. al: Quantum GIS User's Guide, dostupné na: <http://qgis.org/index.php>

4.3 GIS laboratórium

Možným riešením pre efektívne fungujúce GIS laboratórium by bolo použitie modelu ktorý pre svoje počítačové miestnosti vytvoril Inštitút geografie na Univerzite v Pécsi. Miestnosti majú má 25 počítačov a je pre študentov otvorená nepretržite v čase výučby i mimo nej. Študent si po zapnutí počítača vyberie pod akým operačným systémom chce pracovať (k dispozícii sú dva: Windows XP a Linux Ubuntu 10.04) a do vybraného systému sa prihlási. Každý študent má svoje prihlasovacie meno a heslo. Všetky dáta, s ktorými študent pracuje vo svojom vytvorenom účte, sú uložené na serveri. Tým sú dostupné v každej počítačovej miestnosti na univerzite a aj prostredníctvom bezdrôtovej siete v priestoroch univerzity.

Inštitútom zakúpené a pedagógom pripravené GIS vrstvy sú umiestnené na serveri, odkiaľ si ich študenti inštitútu môžu pre účely štúdia skopírovať. Tento systém je ošetrený vyššie uvedeným spôsobom individuálneho prihlasovania a definovanými právami študentov, ktoré neumožňujú zmenu zdrojových vrstiev. Zároveň sa k týmto vrstvám študenti dostávajú prostredníctvom GrassGIS softvéru, kde časť permanent, obsahuje zdrojové nemenné vrstvy, a študentmi vytvorené vrstvy sa ukladajú pod študentmi vytvorené mapsety. Práve táto možnosť zdieľania dát v GrassGIS s možnosťou rozdelenia dát na zdrojové (nemenné) a študentmi pretvorené a vytvorené vrstvy robia s GrassGIS softvérom výhodný nástroj pre správu dát z GIS servera. Zdrojové dáta sa v takto vytvorenom systéme neničia, neduplikujú, nemiešajú so študentskými dátami, a každá katedra či inštitút má svoje dáta pod svojou správou.

Prihlasovanie na pracovnú stanicu má niekoľko ďalších výhod:

- práca študentov je spätne skontrolovateľná - vieme overiť na čom študent pracoval počas výučby, aké dáta sťahoval, aké webstránky navštívil
- nahlasovanie funguje ako preventívny nástroj pred vandalizmom v miestnosti – neanonymný prístup bráni študentom ničeniu pracovných staníc a okolia počítačov
- znižuje riziko vzniku softvérovej kriminality – zabráňuje sťahovaniu materiálov nevyplývajúcich z výučby a návštevu neželaných web stránok.

Vychádzajúc z vyššie spomínaných problémov navrhujeme niekoľko zmien v GIS laboratóriu na FPV UKF v Nitre:

- Základným centrálnym spravovaným úložiskom dát by mal byť server, kde by každý užívateľ mal vyčlenený svoj vlastný priestor a definované práva. Pedagóg by mal mať právo pristupovať k cvičným dátam, meniť ich, dopĺňať prípadne mazať. Študenti by nemali zasahovať do učiteľom pripravených dát. Môžu ich otvoriť, editovať, kopírovať, zmeniť, ale zmenené ich môžu ukladať len na vyčlenenom mieste. Takto sa cvičné dáta zachovajú nezmenené počas celého semestra.
- Počítače by mali nainštalované 2 operačné systémy. Užívatelia by sa tým istým heslom dostali pod oba operačné systémy a zároveň aj k svojim dátam na servery. Pravdepodobne KGRR by naďalej vyučovala GISy v ARC softvéry pod OS Windows. KEE by využívala prednostne OS Linux.
- Zmeny pre študentov počas výučby by boli nasledovné:
- výber OS a prihlásenie sa cez login do systému pri štarte počítača,
- študent by mal k dispozícii len svoje osobné dáta s možnosťou prístupu k študijným materiálom,

- jeho práca by nebola obmedzená na používanie jediného počítača, nakoľko dáta by boli na servery dostupné z ľubovoľného počítača,
- každý študent by mal väčšiu možnosť na prístup ku GIS softvéru (na oboch operačných systémoch ich bude nainštalovaných niekoľko) a väčšiu možnosť profesionálneho rastu,
- študent bude mať naďalej možnosť využívať okrem softvéru, ktorý je nainštalovaný na počítačoch aj služby ktoré ponúka internet, jeho prácu je však možné kontrolovať, prípadne monitorovať.

4.4 Tvorba katedrovej a fakultnej GIS databanky

Jedným zo základných pilierov vedeckovýskumnej činnosti ale aj výučby v oblasti GISov je práca s geodátami – ich zber, editovanie, transformácia, vytváranie, zhromažďovanie a následné využívanie.

Práve skutočnosť, že na katedrách vznikajú hodnotné dáta, ktoré sú použiteľné aj ďalšími odborníkmi v iných oblastiach výskumu (pri zachovaní autorských práv) otvára otázku, prečo neexistuje priestor, kde by sa k vytvoreným dátam konkrétni užívatelia mohli dostať. Vytvorená GIS databanka:

- umožní efektívny prístup ku GIS dátam z ľubovoľného miesta planéty, čo je výhodné pri prezentácii výsledkov našich projektov a pri výskumných stážach v zahraničí,
- zabezpečí výmenu dát so spolupracujúcimi inštitúciami na Slovensku i v zahraničí napr. pri medzinárodných, vedeckovýskumných projektoch,
- zjednoduší zdieľanie dát pri výučbe GISov na FPV – pedagógovia, vyučujúci GISy, môžu na tomto mieste zhromažďovať dáta potrebné k výučbe,
- sprehľadní v súčasnosti existujúce GIS dáta – vyhneme sa tak duplikovanej tvorbe podaktorých GIS vrstiev (napríklad ak jeden pedagóg rektifikuje konkrétnu mapu pre svojho diplomanta, môže ju ponúknuť aj ďalším s presne určenými autorskými právami),
- zvýši autorom vytvorených máp a mapových vrstiev potenciál citovateľnosti – každé dostupné geodáta budú mať pri sebe aj informáciu, kto je ich autorom, na základe akých podkladov dáta vznikli a podobne, čím sa umožní citácia dát,
- zhromaždí okrem nami vytvorených a zakúpených dát aj súbor aktuálnych, voľne dostupných a slobodne šíriteľných dát,
- prinesie do budúcnosti veľký súbor geodát ponúkateľných vedecko-výskumným inštitúciám i súkromnému sektoru.

Informačno-technologickým základom pre GIS databanku je kvalitne pripravený GIS server s presne definovanými právami pre užívateľov.

Tvorcami a užívateľmi dát v GIS databanke budú:

- študenti – nielen ako užívatelia, ale aj nimi vytvorené, digitalizované mapy počas cvičení na hodinách GIS sa stanú súčasťou geodatabanky,
- odborní asistenti, vyučujúci v oblasti GISov,
- vedeckí pracovníci – jednotlivci ale aj skupiny pracujúce na spoločných projektoch.

4.5 Spolupráca v oblasti GIS

V ďalšom období plánuje Katedra ekológie a environmentalistiky využiť rozvíjajúce sa GISy na FPV pre novú vedeckú spoluprácu, využiť jednotné softvéry (Grass GIS) pre výmenu skúseností v oblasti priestorových analýz, a vytvorenú GEO databanku využiť i v budúcnosti pre porovnávacie štúdie.

Zároveň je dôležité pokračovať v existujúcej spolupráci v oblasti geografických informačných systémov a prehlbovať vzťahy so zahraničnými partnermi v oblasti výskumu krajiny s využitím GIS:

- Inštitút geografie, Fakulta prírodných vied, Univerzita v Pécsi, Maďarská republika
- Inštitút environmentalistiky, Fakulta prírodných vied, Univerzita v Pécsi
- Institute of Geography and Regional Science, University of Graz
- Institute of Ecology, University of Innsbruck
- Department of Physical Geography, Faculty of Biology and Earth Sciences, Jagiellonian University Kraków
- Department of Geographical Information Systems, Cartography and Remote Sensing, Faculty of Biology and Earth Sciences, Jagiellonian, Kraków
- Ústav systémové biologie a ekologie, Akademie věd v ČR, Brno
- Katedra sociální geografie a regionálního rozvoje, Karlova Univerzita, Praha

5 Záver

Navrhnutá koncepcia bola skonzultovaná s Katedrou geografie a regionálneho rozvoja a spoločne ponúknutá a odprezentovaná vedeniu Fakulty prírodných vied začiatkom roka 2011. Koncepcia bola následne schválená a v súčasnosti sa vykonávajú všetky potrebné zmeny z nej vyplývajúce. Výučba Geografických informačných systémov v nových podmienkach začne už na jeseň tohto roku.

Pod'akovanie

Koncepcia vznikla ako výsledok doterajšej spolupráce Katedry ekológie a environmentalistiky na FPV UKF v Nitre s Inštitútom geografie na Univerzite v Pécsi v rámci projektov KEGA 030UKF-4/2011 a projektu TÁMOP-4.2.2/08/1/2008-0011 IKT - Science, Please! Innovative Resarch Team.

Literatúra

- [1] RAPANT, P.: Úvod do geografických informačných systémů. Ostrava: VŠB – TU Ostrava, 2002. online: <http://gis.vsb.cz/publikacie/ugis>
- [2] TUČEK, J.: Geografické informační systémy – Principy a praxe. Praha, Computer Press, 1998, ISBN 807226091X
- [3] <http://geo.enviroportal.sk/infrastruktura/desktop-gis>, 3. 4. 2011
- [4] http://gama.fsv.cvut.cz/gwiki/GRASS_GIS, 3. 4. 2011

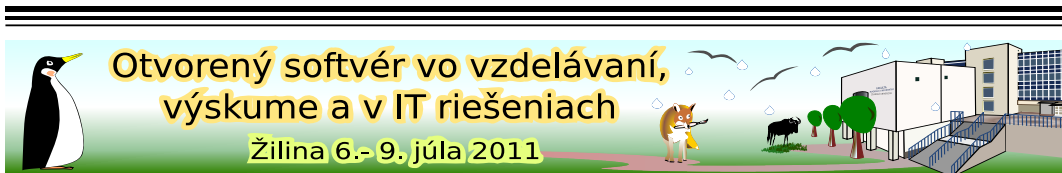
Kontaktná adresa

JAKAB, Imrich, (Mgr., PhD.),

Katedra ekológie a environmentalistiky FPV UKF v Nitre,
Tr. A. Hlinku 1, 949 74 Nitra,
Research Group for High Performance Computing, SP! Project Office,
University of Pécs, H-7633 Pécs,
Szántó Kovács János u. 1/B,
ijakab@ukf.sk

BUGYA, Titusz, (MSc., PhD.),

Institute of Geography,
University of Pécs, H-7633 Pécs,
Szántó Kovács János u. 1/B.,
titusz@gamma.ptt.ttk.hu



DOI: 10.5300/2011-OSSConf/165

SPRACOVANIE DÁT SLOVENSKEHO KATASTRA POMOCOU OPEN SOURCE APLIKÁCIÍ PRE WEBGIS TECHNOLOGIE A ICH PUBLIKOVANIE 2011

KISZTNER, Juraj, (SK)

Abstrakt. Prvá časť sa zaoberá spracovaním dát slovenského katastra. Údaje katastra nehnuteľností sú jednou zo základných priestorových vrstiev na území SR. V projekte boli vytvorené programy, ktoré riešia konverziu z proprietárneho formátu vgi používaného v rezorte geodézie, kartografie a katastra do formátu GML s otvorenou špecifikáciou. Vďaka správnej konverzii s pripojením atribútov program nahrádza čiarovú kresbu typickú pre katastrálne mapy, mapou s farebným odlíšením spôsobu využívania pozemku. Druhá časť projektu prináša ukážky, ako sa tieto dáta dajú využiť vo WMS. Tieto open source riešenia môžu slúžiť na evidenciu pozemkov na území obce alebo ako základnú vrstvu v iných geoinformačných systémoch.

Kľúčové slová. Kataster, konverzia, wms.

SLOVAK CADASTRAL DATA PROCESSING BY USING OPEN SOURCE APPLICATIONS FOR WEBGIS TECHNOLOGIES AND THEIR PUBLICATION 2011

Abstract. This project deals with the Slovak Cadastral data processing. The cadastral data are one of the basic spatial layers in Slovakia. In the project, we created programs which design conversion from proprietary format .vgi used in the section of Geodesy, Cartography and by Slovak Cadastral to format GML with an open specification. Due to the correct conversion with attached attributes the program replaces line drawing typical for Cadastral maps with a map with colored categories based on landuse. The second part of the project offers the actual illustrations of how the data can be utilized in WMS with orientation on smaller municipalities, which can use open source systems to register the grounds in the municipal territory or as a basic layer in another geoinformational systems.

Key words and phrases. Cadastre, conversion, WMS.

Úvod

Kataster nehnuteľností je základná evidencia nehnuteľností na Slovensku. Jeho súčasťou je katastrálna mapa (obr. 1), ktorá by mala slúžiť spolu s popisnými údajmi ako základná mapa veľkej mierky (1:1000, 1:2000, 1:2880) pre širokú škálu IS a GIS. Pretože existuje len slabá podpora ako tieto dáta využívať, bolo cieľom tohto projektu vytvoriť podporné aplikácie na konverziu formátu „vgi“ do formátu „GML“. Rovnako bolo cieľom aj sprehľadnenie katastrálnej mapy a popisných údajov katastra nehnuteľností. Na koniec projekt prináša ukážku, ako vhodne implementovať tieto výsledky v prostredí internetu. Celý tento projekt je zameraný na spracovanie a publikovanie dát v otvorených softvéroch. Jeho cieľom je pripraviť dáta katastra najlepším a zároveň najmenším nákladným spôsobom.



Obr. 1: Ukážka katastrálnej mapy s typickou líniovou kresbou

Kataster ako zdroj dát

V katastri nehnuteľností sa spoločne evidujú všetky nehnuteľnosti nachádzajúce sa v jednom katastrálnom území. Z tohto pohľadu je kataster zdrojom veľmi aktuálnych informácií o nehnuteľnostiach. Priestorovou zložkou katastra nehnuteľností je katastrálna mapa – ide o mapu veľkej mierky, ktorá sa v digitálnej podobe aktualizuje spoločne s popisnou zložkou. Z tohto pohľadu nám kataster má slúžiť ako základná priestorová vrstva na evidenciu nehnuteľností, a tieto dáta sa dajú využiť pre: adresný IS, modelovanie povodňového rizika.

Spracovanie dát katastra

V katastri sa na obnovu mapy a prenos údajov vektorových máp, preberaní a poskytovaní údajov ako sú geometrické plány, pozemkové úpravy, ale aj pre poskytovanie údajov pre iné informačné systémy používa výmenný formát vgi. Tento formát nemá otvorenú špecifikáciu a má malú podporu v GIS softvéroch.

Popisná zložka katastra nehnuteľností je evidovaná pomocou štruktúry tabuliek vo formáte dbf, ktoré ukladajú všetky popisné informácie o jednom katastrálnom území. Spojenie grafickej a popisnej časti katastra nehnuteľností je parcelné číslo, ktoré slúži ako jednoznačný identifikátor parcely v jednom katastrálnom území.

Spracovanie grafickej časti

Spracovanie formátu vgi bolo jedným z hlavných cieľov pri vytváraní tohto projektu. Na automatizované spracovanie boli použité sady python skriptov, konvertujúce formát vgi do formátu shp, ktorý ma otvorenú špecifikáciu a je dnes akýmsi štandardom na ukladanie priestorových vrstiev v GIS. Pomocou týchto python skriptov sa líniová kresba uložená vo formáte vgi konvertuje na polygóny, kde jeden objekt tvorí práve jednu parcelu. Pomocou automatizovaného spracovania popisnej zložky sa pripoja čísla listov vlastníctva a kódy spôsobu využívania pozemku (tab. 1). Vďaka týmto kódom je následne možné vykresliť farebnú mapu (obr. 2).

1- obilniny, okopaniny
11- vodný tok
15- bytová budova
16- nebytová budova
17- budova bez s.c.
18- dvor
19- spoločný dvor
20- železnica
21- diaľnica
22- cesta
23- letisko
24- plavebný kanál
25- ostatné inžinierske stavby
26- rozostavaná stavba
29- okrasná zahrada
3- vinic
30- ihrisko, stadion
32- cintorin
34- skladová plocha
37- skaly, medze
38- les
4- zastavané územie obce
5- zahrady
6- ovocné sady
7- luky
8- skleník
99- ine



Obr. 2: Ukážka spracovania údajov katastra

OpenGIS® Geography Markup Language Encoding Standard (GML): Je z rodiny XML jazykov. Slúži na opis geografických prvkov ako modelovací jazyk v geografických systémoch. Jeho hlavnou výhodou je schopnosť integrovať všetky formy geografických informácií.

Formát vgi: Výmenný formát vektorových máp katastra – VGI je textový súbor, ktorý vychádza z textového formátu výkresov grafického systému Kokeš – VTX. Rozdiely sú len

v niektorých voliteľných vetách (napr. diakritika je len Latin2) a povinnosti uvádzať niektoré, vo VTX implicitné, údaje (napr. K=1). Výmenný formát sa používa pri všetkých prenosoch údajov vektorových máp, preberaní i poskytovaní údajov napr. spracovanie registrov (ROEP), pozemkové úpravy, geometrické plány, poskytovaní údajov pre iné IS, prenosi údajov okres → centrum. (Podľa: http://www.zememeric.cz/default.php?/clanek_tisk.php?zaznam=1770)

Tab. 1: Tabuľka spôsobu využívania pozemkov, ktorá slúžila ako podklad na farebnú klasifikáciu spracovanej katastrálnej mapy

Spôsob využívania pozemku	
Kód	Popis
1	Pozemok využívaný pre rastlinnú výrobu, na ktorom sa pestujú obilniny, okopaniny...
2	Pozemok vysadený chmeľom alebo pozemok vhodný na pestovanie chmeľu...
3	Pozemok, na ktorom sa pestuje vinič alebo pozemok vhodný na pestovanie viniča...
4	Pozemok prevažne v zastavanom území obce alebo v záhradkárskej osade...
5	Pozemok v rámci záhradného centra, na ktorom sa pestuje okrasná nízka a vysoká zeleň...
6	Pozemok súvisle vysadený ovocnými stromami, ovocnými krami a ovocnými sadenicami...
7	Pozemok lúky a pasienku trvalo porastený trávami...
8	Na pozemku je postavený skleníkový, japonský, parenisko a iné
9	Na pozemku je škôlka pre chmeľové sadivo, viničová škôlka...
10	Na pozemku je účelová ochranná poľnohospodárska a ekologická zeleň...
11	Vodný tok (prirodzený – rieka, potok; umelý – kanál, náhon a iné)
12	Vodná plocha (jazero, umelá vodná nádrž, odkryté podzemné vody – štrkovisko, bagrovisko a iné)
13	Rybník – umelá vodná nádrž určená na chov rýb vrátane stavieb
14	Močiar
15	Pozemok, na ktorom je postavená bytová budova označená súpisným číslom
16	Pozemok, na ktorom je postavená nebytová budova označená súpisným číslom
17	Pozemok, na ktorom je postavená budova bez označenia súpisným číslom
18	Pozemok, na ktorom je dvor
19	Pozemok, na ktorom je spoločný dvor
20	Pozemok, na ktorom je postavená inžinierska stavba – železničná, lanová a iná dráha...
21	Pozemok, na ktorom je postavená inžinierska stavba – diaľnica a rýchlostná komunikácia...
22	Pozemok, na ktorom je postavená inžinierska stavba – cestná, miestna a účelová komunikácia...
23	Pozemok, na ktorom je postavená inžinierska stavba – vzletová, pristávací a rolovacia dráha...
24	Pozemok, na ktorom je postavená inžinierska stavba – prístav, plavebný kanál a komora...
25	Pozemok, na ktorom je postavená ostatná inžinierska stavba a jej súčasť
26	Pozemok, na ktorom je rozostavaná stavba
27	Pozemok, na ktorom je zrúcanina
28	Pozemok, na ktorom je postavený vstupný portál do podzemnej stavby alebo pivnice
29	Pozemok, na ktorom je okrasná záhrada, uličná a sídlisková zeleň, park a iná funkčná zeleň...
30	Pozemok, na ktorom je ihrisko, štadión, kúpalisko, športová dráha, autokemp, táborisko a iné
31	Pozemok, na ktorom je botanická a zoológická záhrada, skanzen, amfiteáter, pamätník a iné
32	Pozemok, na ktorom je cintorín alebo umový háj
33	Pozemok, ktorý slúži na ťažbu nerastov a surovín
34	Pozemok, na ktorom je manipulačná a skladová plocha, objekt a stavba slúžiaca lesnému hosp.
35	Pozemok, na ktorom je skládka odpadu
36	Pozemok, ktorý nie je využívaný žiadnym z uvedených spôsobov
37	Pozemok, na ktorom sú skaly, svahy, rokliny, výmole, vysoké medze s krovím alebo kamením...
38	Pozemok s lesným porastom, dočasne bez lesného porastu na účely obnovy lesa...
99	Pozemok využívaný podľa druhu pozemku

Spracovanie popisnej časti

Automatizované spracovanie tabuliek s popisnými informáciami rovnako prebehlo pomocou python skriptu, ktorý vygeneroval listy vlastníctva vo formáte pdf (obr. 3) a na pripojenie listov vlastníctva k jednotlivým parcelám slúžilo číslo listu vlastníctva, ktorý je jednoznačným identifikátorom listu vlastníctva v jednom katastrálnom území.

* údaje nemôžu slúžiť na právne úkony, majú informatívny charakter

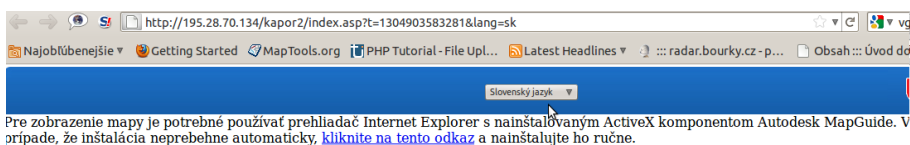
List vlastníctva c.516

LV,PODIEL,VLASTNIK
516, 1/1, ZEMAN MIROSLAV, B - PODZAHRADNA 19

Obr. 3: Ukážka spracovania popisných údajov do pdf listov vlastníctva

Publikovanie dát katastra v prostredí internetu

V prípade riešenia na stránkach katastrálneho portálu Úradu geodézie kartografie a katastra SR <https://www.katasterportal.sk/kapor/> pri zobrazení mapovej časti sa používateľ stretne s riešením postavenom na bázy Active-X komponentu. Táto mapová aplikácia je spustiteľná len v prehliadači Internet Explorer po nainštalovaní komponentu. Riešenie nepriamo núti k používaniu jedného prehliadača a jedného operačného systému (obr. 4). Rovnako je náročné pripojenie týchto údajov do GIS softvéru.



Obr. 4: Zobrazenie mapovej aplikácie katastrálneho portálu v prehliadači Firefox 4.0.1

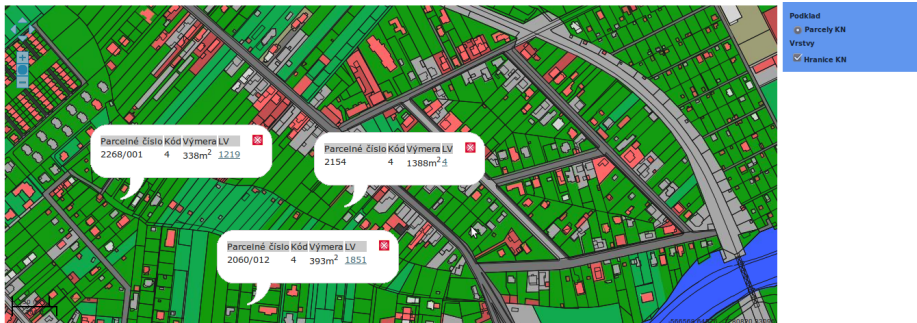
Preto jednou zo základných myšlienok pri vytváraní tohto projektu bolo publikovanie katastrálnych dát v prostredí internetu s pomocou štandardu wms, pretože ide o najpraktickejší a dnes čoraz častejší spôsob publikovania priestorových dát.

Po konverzii do formátu GML môžeme využiť celú radu produktov na publikovanie takto pripravených dát. V našom prípade bol zvolený produkt UMN Mapserver s knižnicou OpenLayers na vytvorenie užívateľského rozhrania v prostredí internetu (obr. 5). Ide o open source riešenie ktoré je nezávislé na operačných systémoch a internetových prehliadačoch. Medzi hlavné výhody takéhoto riešenia patrí:

1. Nezávislosť od operačného systému
2. Nezávislosť od web prehliadača

3. Populárne riešenie
4. Dostupnosť
5. Prepojitelné do DESKTOP aplikácii
6. Prepojitelné do MOBILE aplikácii

Takto publikované údaje katastra môžu slúžiť ako podklad pre obce alebo ich základný informačný systém s variabilným využitím.



Obr. 5: Publikovanie dát katastra v prostredí www

Záver

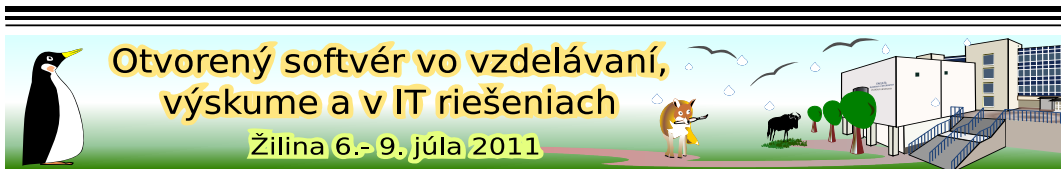
Spracovaním dát katastra do formátu GML sa údaje katastra stávajú prístupnejšie pre širokú škálu softvérov z oblasti GIS napr. QGIS, JUMP, GRASS GIS. Rovnako vďaka konverzii s použitím spracovaných skriptov sa katastrálna mapa stáva prehľadnejšia pre laickú verejnosť a zároveň vďaka prevodníku ogr môže priniesť ľahšie využitie v širokom spektre riešení a produktov. Takto spracované riešenie je nezávislé na operačnom systéme, webovom prehliadači. Rovnako je možnosť tieto dáta pripojiť do vlastného, či už desktopového alebo mobilného softvéru. A zároveň takto spracovaná vrstva vo formáte GML môže slúžiť ako podklad v iných informačných systémoch ako napríklad modelovanie protipovodňového rizika, lavínové záchranársko, územné plánovanie.

Literatúra

- [1] UMN Maperver. <http://mapserver.org/>
- [2] Python. <http://python.org/>
- [3] Geometrické plány a výmenný formát katastra nehnuteľností na Slovensku.
http://www.zememeric.cz/default.php?clanek_tisk.php?zaznam=1770
- [4] Kataster nehnuteľností. <http://katasternehnutelnosti.sk/>

Kontaktná adresa

Juraj Kisztner,
juro.kisztner@gmail.com



DOI: 10.5300/2011-OSSConf/171

PUBLIKÁCIA KOMPLEXNÉHO DIGITÁLNEHO MODELU RELIÉFU PROSTREDNÍCTVOM OPEN SOURCE TECHNOLOGIÍ 2011

PETŘÍČEK, Gabriel, (SK); MAHRÍK, Martin, (SK)

Abstrakt. *Predložený príspevok sa zaoberá operačnými možnosťami geografickej bázy údajov a publikovaním komplexného digitálneho modelu reliéfu (ďalej KDMR) prostredníctvom open source technológií, pričom popisuje aj proces tvorby KDMR a jeho morfometrických parametrov. Celý postup je rozčlenený do troch okruhov. Prvým krokom je spracovanie vstupného bodového súboru a KDMR a jeho morfometrických parametrov. V ďalšom kroku sa vykoná import do databázového prostredia, kde budú vykonané atribútové a priestorové dopyty za účelom generovania nových priestorových údajov. Posledným krokom je publikácia kartografických modelov za použitia webových mapových služieb WMS a WFS a ich zobrazenie prostredníctvom tenkého a stredného klienta.*

Kľúčové slová. *KDMR, geografická база údajov, WMS, PostgreSQL, PostGIS, mapový server, GIS, dopyty, OpenLayers.*

PUBLICATION OF COMPLEX DIGITAL ELEVATION MODEL THROUGH OPEN SOURCE TECHNOLOGIES 2011

Abstract. *The presented paper deals with operational capabilities of spatial database and publication of the digital elevation model (aka DEM) by utilizing open-source tools, while describing the entire creation process of the DEM and its morphometric parameters at the same time. The whole procedure is divided into three ambits. The first step is processing of input point file and modeling the DEM and its morphometric parameters. The next step describes import DEM and morphometric parameters into spatial database, where attribute and spatial queries to generate new spatial data are performed. The final step is the publication of cartographic models using web map services WMS and WFS and displaying them by the thin and middle client.*

Key words and phrases. *DEM, geographic database, WMS, WFS, PostgreSQL, PostGIS, map server, GIS, queries, OpenLayers.*

1 Úvod

Modelovanie reliéfu patrí medzi najvýznamnejšie metódy výskumu v geografických disciplínach, ktoré sa za posledné obdobie prenieslo do roviny širokej praxe. Metódy modelovania sú imanentnou súčasťou špecifických analytických nástrojov na modelovanie priestorovej diferenciácie geografických objektov a fenoménov [1]. V súčasnej dobe existuje množstvo GIS programov, ktoré obsahujú vo svojej štandardnej výbave moduly na modelovanie KDMR a jeho morfometrických parametrov. Podrobne sa problematike modelovania KDMR a jeho morfometrických parametrov venuje práca [2].

Každá organizácia si v súčasnej dobe buduje spoľahlivý informačný systém, ktorého súčasťou je aj databáza (nazývaná tiež báza údajov) a softvér, ktorý s databázou pracuje. Preto dnešné informačné systémy sú úzko spojené s databázami a databázovými systémami. Geografická (priestorová) báza údajov predstavuje model reality, ktorý vyjadruje základné vlastnosti a stav krajiny uvažovanej ako objekt riadenia na základe zvolených hľadísk výskumu, využitia a ochrany prírody [3]. Poskytuje dopyty na tvorbu nových kartografických modelov na základe vstupných priestorových údajov. Dopyty sú žiadosti o geografickú informáciu z geografickej bázy údajov. Delia sa na priestorové a atribútové. Pri priestorových dopytoch sa využívajú priestorové operátory, pri atribútových dopytoch zas logické [4].

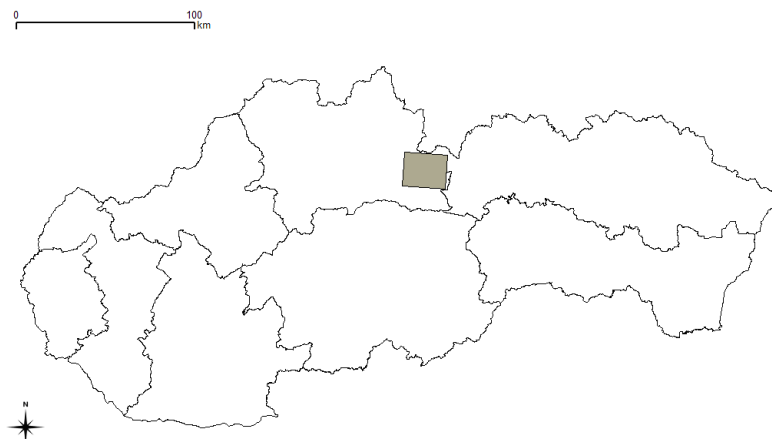
Neodlučiteľnou súčasťou posledného obdobia v oblasti geografických vied je publikácia vytvorených kartografických modelov prostredníctvom celosvetovej siete internet. Publikácia je umožnená za použitia webových mapových služieb alebo webových grafických aplikácií. Webové mapové služby poskytujú užívateľom grafické výstupy kartografických modelov vo formáte rastrovom (WMS), vektorovom (WFS), poskytujú katalógové služby (CSW) a iné. Ich zobrazenie je možné pomocou naskriptovaných webových aplikácií (tzv. tenkých klientov), prostredníctvom stredných klientov, ktorí poskytujú minimálnu funkcionálnu prácu s geografickými informáciami alebo pomocou rôznych GIS softvérov (inak nazývaných aj hrubých klientov), ktorí podporujú štandardy webových mapových služieb OGC¹ a zabezpečujú komunikáciu s webovými mapovými službami. Podrobná špecifikácia štandardov webových mapových služieb je popísaná na URL adrese <http://www.opengeospatial.org/>.

Pri tvorbe KDMR a jeho morfometrických parametrov sa použil mapový list 26-44 v mierke 1:50 000. Údaje boli poskytnuté Výskumným ústavom geodézie a kartografie a používajú súradnicový systém S-JTSK so základným poludníkom Ferro (EPSG:2065). Mapový list pokrýva oblasť Západných Tatier a leží na rozhraní Žilinského a Prešovského kraja, resp. medzi okresmi Liptovský Mikuláš a Poprad (obr. 1).

Metodika predloženého príspevku pozostáva z troch fáz:

- modelovanie KDMR a jeho morfometrických parametrov,
- generovanie nových priestorových informácií v databázovom prostredí,
- publikácia kartografických modelov pomocou webových mapových služieb a webovej aplikácie.

¹OGC je nezisková organizácia, ktorá vyvíja štandardy pre geopriestorové a lokalizačné služby.



Obr. 1: Poloha pracovnej lokality v rámci SR

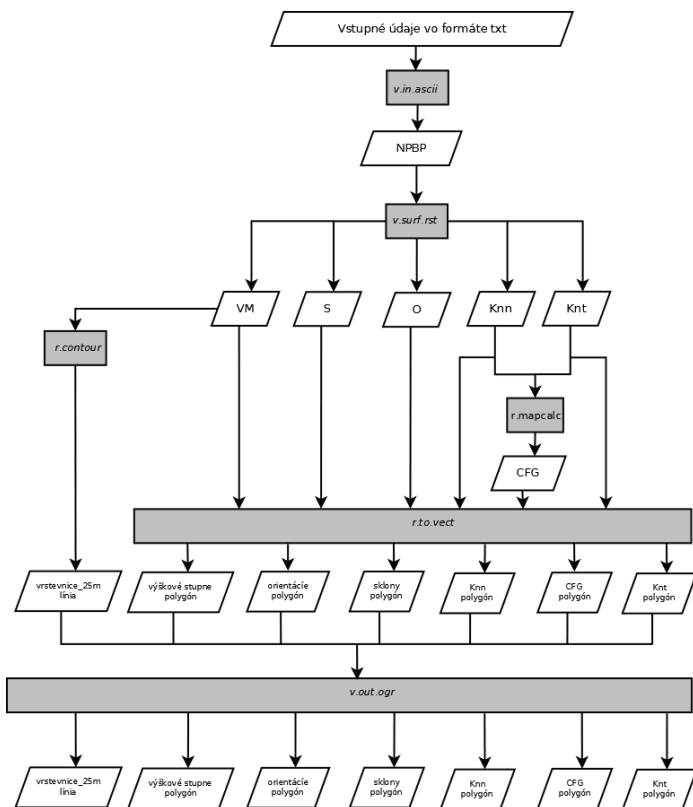
V prvej fáze bolo potrebné namodelovať KDMR a jeho morfometrické parametre za použitia interpolačnej metódy regularizovaného splajnu s tenziou. Hodnota v interpolačnom bode sa vypočíta na základe matematickej funkcie, ktorá simuluje správanie sa tenkej, elastickej platne a prechádza cez zadané body [5]. Na overenie vhodnosti a správnosti interpolačnej metódy regularizovaného splajnu s tenziou sa použila metóda cross validation. Metóda cross validation odoberie jeden bod zo vstupného bodového poľa a vypočíta výškovú odchýlku medzi vstupným bodovým poľom a namodelovaným DMR. Potom postup prechádza na ďalší bod v poradí a celá procedúra sa zopakuje až pokiaľ sa neotestuje posledný bod vstupného bodového poľa. Zdrojové údaje boli vo formáte textového dokumentu, ktorý obsahoval súradnice x, y a z (výšky). Procesný diagram na obr. 2 popisuje pracovný postup od importovania vstupných údajov vo forme textového súboru do prostredia GRASS po import vytvorených kartografických modelov do databázového prostredia PostgreSQL s nadstavbou PostGIS.

Podrobná dokumentácia jednotlivých modulov prostredia GRASS GIS je dostupná na URL adrese <http://grass.fbk.eu/> alebo v publikácii [6].

Po uskutočnení prvej fázy boli vytvorené kartografické modely, ktoré boli importované do open source databázového prostredia PostgreSQL s využitím priestorovej knižnice PostGIS. Bol navrhnutý katalóg objektov KDMR, ktorý predstavuje tab. 1. V danom databázovom prostredí boli generované nové kartografické modely na základe atribútových a priestorových dopytov (obr. 3). Pri atribútových dopytoch sa použili logické operátory ako $>$, $<$ a $=$. Pomocou nasledovného príkazu sa zvolili konvexné (vypuklé) formy reliéfu normálovej krivosti $Kn_{(n)}$ ²:

```
select * from knn where hodnota=2
```

² $Kn_{(n)}$ predstavuje normálovú krivosť georeliéfu v smere spádnice.



Obr. 2: Procesný diagram tvorby KDMR a jeho morfometrických parametrov

Ďalší výsledok atribútového dopytu je zobrazený na obr. 5, ktorý znázorňuje sklony georeliéfu od 3 do 25 stupňov. Pri priestorových dopytoch sa využili príkazy priestorovej knižnice PostGIS. Priestorové dopyty umožňujú na základe dvoch a viacerých vstupných údajov vygenerovať nový. Na obr. 6 je demonštrovaný výsledok priestorového dopytu, ktorý zobrazuje územia so sklonom do 7° a do nadmorskej výšky 1000 metrov. Syntax dopytu má nasledujúci tvar:

```

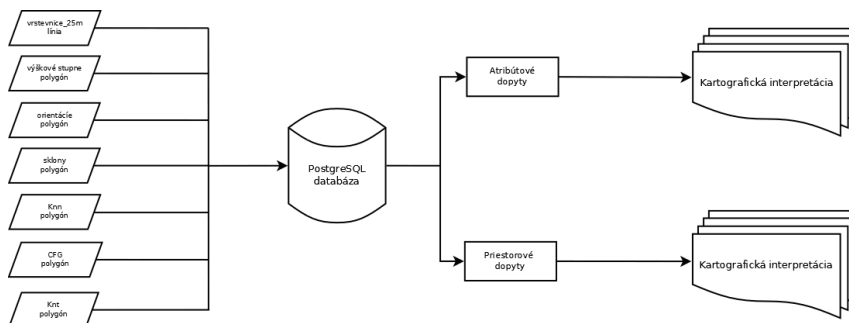
select s.* from sklony as s, vyskove_stupne as v~where s.hodnota<=3
and v.hodnota <=2 and ST_Contains (s.the_geom, v.the_geom)
  
```

Poslednou fázou bola publikácia dosiahnutých kartografických modelov pomocou mapovej služby WMS a WFS. Na publikáciu novo vytvorených priestorových informácií v databázovom prostredí bola použitá open source technológia, mapový server Geoserver, ktorá podporuje komunikáciu s geografickou bázou údajov PostgreSQL s nadstavbou PostGIS. Mapové služby sa zobrazili prostredníctvom desktop GIS Quantum GIS 1.7.0. Na obr. 4 až 6 je demonštrovaná publikácia údajov prostredníctvom webovej mapovej služby WFS.

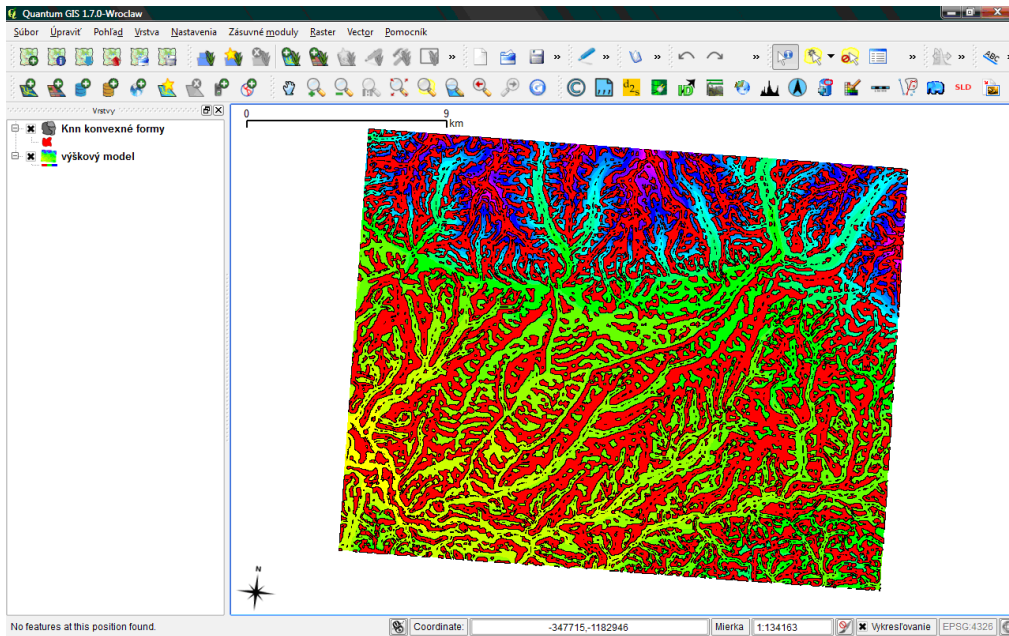
Tab. 1: Ukážka katalógu objektov komplexného digitálneho modelu georeliéfu (KDMR)

Kategória	Subkategória	Objekt	Typ geometrie	Atribút	Hodnota atribútu	Popis atribútu	
Komplexný digitálny model reliéfu	Výšky	Vrstevnice	línia	ID	integer	Číselný identifikátor objektu (1, 2, 3, ...)	
				hodnota	integer	Hodnota nadmorskej výšky (650,85)	
	Sklony	Výškové stupne	polygón	ID	integer	Číselný identifikátor objektu (1, 2, 3, ...)	
				hodnota	integer	Identifikátor výškových stupňov (800-900)	
	Orientácie	Areály sklonov	polygón	ID	integer	Číselný identifikátor objektu (1, 2, 3, ...)	
				hodnota	integer	Identifikátor areálu sklonov (1, 2, 3, ...7)	
	Geometrické formy	Orientácie	Areály orientácií	polygón	ID	integer	Číselný identifikátor objektu (1, 2, 3, ...)
					hodnota	integer	Identifikátor areálu orientácií (1, 2, 3, ...8)
		Normálové formy $Kn_{(n)}$		polygón	ID	integer	Číselný identifikátor objektu (1, 2, 3, ...)
					hodnota	integer	Identifikátor areálu $Kn_{(n)}$ (1, 2)
	Normálové formy $Kn_{(t)}^3$		polygón	ID	integer	Číselný identifikátor objektu (1, 2, 3, ...)	
				hodnota	integer	Identifikátor areálu $Kn_{(t)}$ (1, 2)	
Celkové formy georeliéfu		polygón	ID	integer	Číselný identifikátor objektu (1, 2, 3, ...)		
			hodnota	integer	Identifikátor areálu celkových foriem (1...4)		

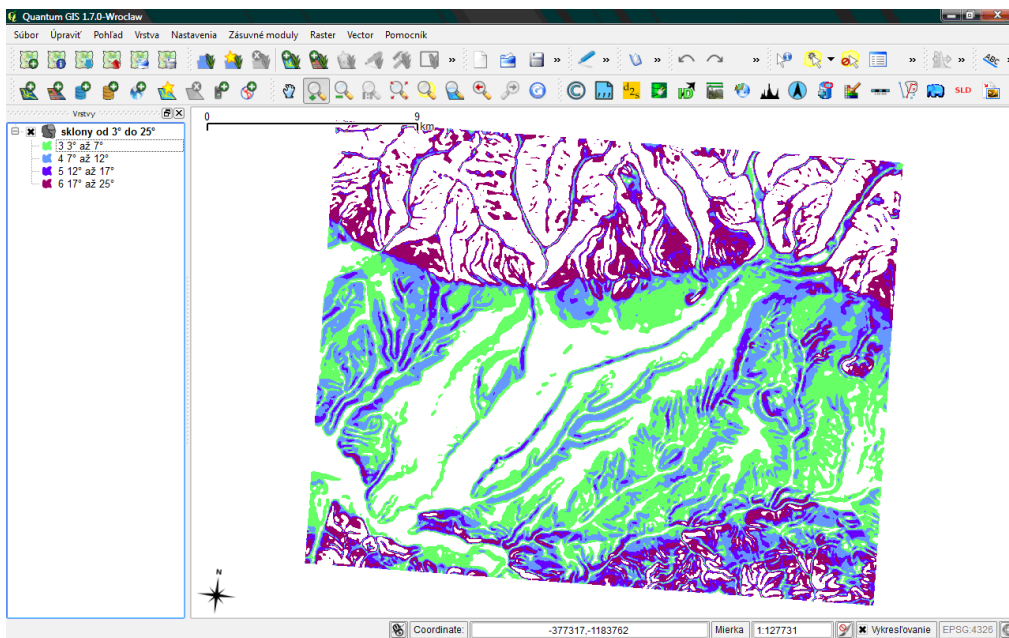
³ $Kn_{(t)}$ predstavuje normálovú krivost' georeliéfu v smere dotýčnice k vrstevnici.



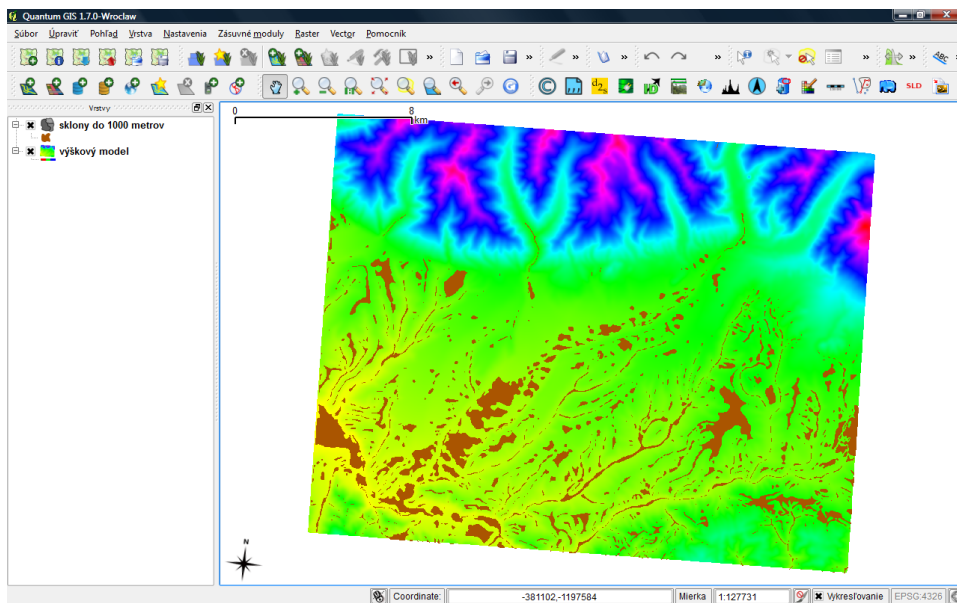
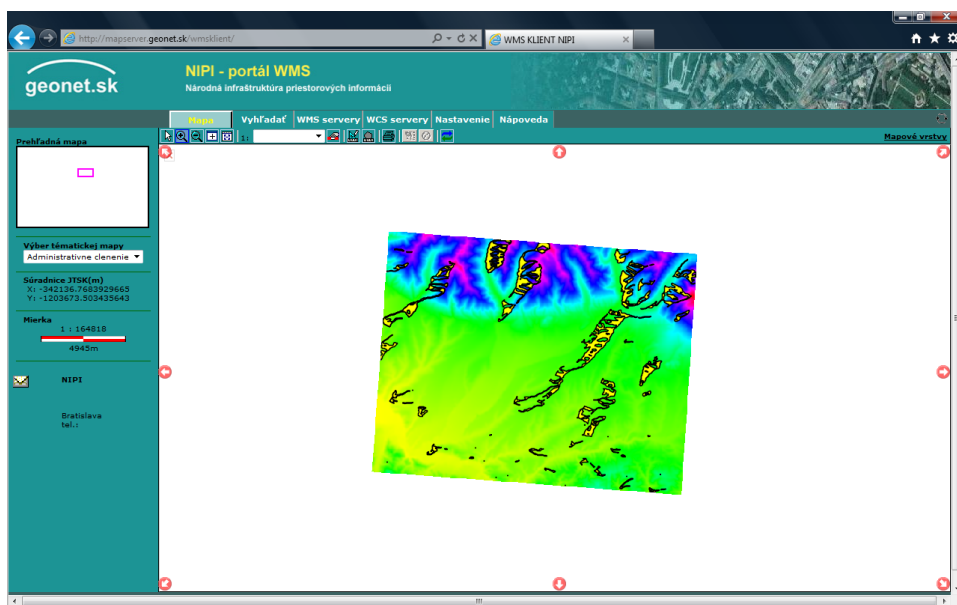
Obr. 3: Vývojový diagram GBÚ KDMR



Obr. 4: Konvexné formy georeliéfu normálovej krivosti $Kn_{(n)}$



Obr. 5: Sklony georeliéfu od 3° do 25°

Obr. 6: Sklony georeliéfu do 7° v nadmorskej výške do 1000 metrovObr. 7: Zobrazenie konkávných foriem georeliéfu $Kn_{(n)}$ orientovaných západným smerom prostredníctvom tenkého klienta dostupného na <http://mapserver.geonet.sk/wmsklient>



Obr. 8: Zobrazenie stredných vysočín prostredníctvom OpenLayers

Druhou možnosťou publikácie je publikácia prostredníctvom tzv. tenkých klientov. Tenký klient poskytuje iba používateľské rozhranie, ktoré komunikuje so serverom a zobrazuje výsledky. Procesy prebiehajú na serverovom počítači, preto sú minimalizované hardvérové nároky klientskeho počítača, ktorý iba zobrazuje dáta poskytované a spracované serverom [7]. Webová stránka <http://mapserver.geonet.sk/wmsklient> poskytuje tenkého klienta, ktorý podporuje mapovú službu WMS (obr. 7). Pomocou priestorovej knižnice OpenLayers skriptovacieho jazyka JavaScript je možné si ľahko a rýchlo vytvoriť webovú mapovú aplikáciu, ktorá podporuje OGC štandardy [8]. Webová aplikácia vytvorená prostredníctvom priestorovej knižnice OpenLayers je znázornená na obr. 8.

2 Záver

Použitie voľne dostupné technológie v príspevku sa preukázali ako rýchly, efektívny a jednoduchý spôsob modelovania a publikácie KDMR a jeho morfolometrických parametrov. V súčasnej dobe existuje široké spektrum GIS programov, ktoré disponujú podporou webových mapových služieb. Publikácia geografických informácií sa už stáva rutinou kvôli jej jednoduchosti spracovania a hlavne kvôli rýchlemu poskytovaniu geografických informácií prostredníctvom internetu. Operačné možnosti GBÚ poskytujú efektívnu tvorbu nových geografických informácií z KDMR a jeho morfolometrických parametrov, ktoré môžu nájsť uplatnenie v geografických vedných disciplínach a prostredníctvom mapových služieb sa takto publikované geografické informácie môžu stať aj voľne dostupnými pre širokú verejnosť.

Literatúra

- [1] Mičietová, E., Kožuch, M.: *Špecializované informačné technológie v prírodovednom výskume: Geoinformačné technológie*. ELITA, Bratislava, s. 77, 2008.
- [2] Krcho, J.: *Morfometrická analýza a digitálne modely georeliéfu*. Veda, Bratislava, 1990.
- [3] Mičietová, E.: 14. kartografická konferencia, *Geografický informačný systém: štruktúra, integrita, interoperabilita, implementácia*. Plzeň, s. 40, 2008.
- [4] Mičietová, E., Čulen, O.: *Geografická databáza a modelovanie komplexných priestorových štruktúr georeliéfu a topografických prvkov*. Kartografické listy 14, s. 57, 2006. ISBN 80-89060-08-0.
- [5] Hofierka, J.: *Geografické informačné systémy a diaľkový prieskum Zeme*. Prešovská Univerzita v Prešove, str. 50, 2003.
- [6] Neteler, M., Mitasova, H.: *Open Source GIS: A GRASS GIS Approach*. Boston, Kluwer Academic Publisher, 2004. ISBN 1-4020-8064-6.
- [7] Zhong-Ren, P., Ming-Hsiang, T.: *Internet GIS: Distributed geographic information services for the Internet and wireless network*. New Jersey, John Wiley and Sons, 2003.
- [8] Hazzard, E.: *OpenLayers 2.10: Beginner's Guide*. Birmingham, UK, 2011. ISBN 978-1-849514-12-5.

Kontaktná adresa

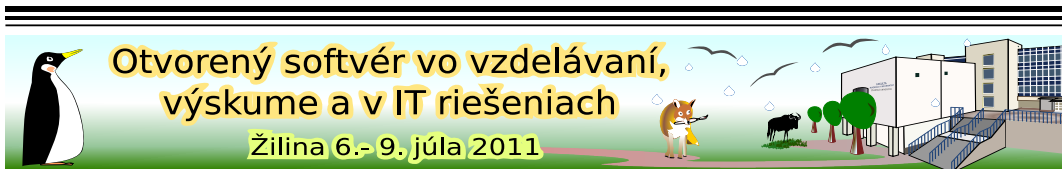
Gabriel PETŘÍČEK (Mgr.),

Katedra kartografie, geoinformatiky a DPZ
Prírodovedeckej fakulty UK v Bratislave, Mlynská dolina,
842 15 Bratislava 4, petricek@fns.uniba.sk

Martin MAHRÍK,

Katedra kartografie, geoinformatiky a DPZ
Prírodovedeckej fakulty UK v Bratislave, Mlynská dolina,
842 15 Bratislava 4, matkom369@gmail.com

T_EX A JEHO PRIATELIA



DOI: 10.5300/2011-OSSConf/183

$\text{T}_{\text{E}}\text{X}$ a otevřený software v malé firmě aneb Efektivní kancelář

ČAČKOVÁ, Petra (CZ)

Abstrakt. *Typografický systém $\text{T}_{\text{E}}\text{X}$ se dlouho a úspěšně používá v akademickém prostředí i pro profesionální publikování, především pro rozsáhlá díla. V tomto příspěvku je představen odlišný pohled, kdy se $\text{T}_{\text{E}}\text{X}$ uplatňuje při automatizaci zpracování dokumentů v malé firmě. Součástí řešení přitom je i netradiční přístup ke zdrojům dat. Kromě samotného $\text{T}_{\text{E}}\text{X}$ ového řešení se příspěvek věnuje i ekonomickým přínosům pro firmu.*

Klíčová slova. *$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, firma, kancelář, efektivita, zpracování dat.*

$\text{T}_{\text{E}}\text{X}$ and open-source software in a small business or Efficient office

Abstract. *The $\text{T}_{\text{E}}\text{X}$ typesetting system has been successfully used in academic and professional publishing for a long time, particularly for large works. This paper presents a different perspective, when $\text{T}_{\text{E}}\text{X}$ is applied in the document processing automation in a small business. The solution takes the advantage of an unconventional approach to data sources. In addition to the $\text{T}_{\text{E}}\text{X}$ solution itself it concerns the economic benefits for the company.*

Key words and phrases. *$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, business, office, effectiveness, data processing.*

1 Úvod

Typografický systém $\text{T}_{\text{E}}\text{X}$ se dlouho a úspěšně používá pro publikování. Jeho doménou je tradičně akademické prostředí napříč všemi obory, od matematiky a fyziky (sazba vzorců) k lingvistice, filosofii a teologii (sazba více jazyků a abeced) [18]. Podpora kvalitní typografie a možnosti automatizace procesu otevřely $\text{T}_{\text{E}}\text{X}$ u cestu do nakladatelství. I po téměř třiceti letech od svého vzniku se stále používá, dokonce lze říci, že vzhledem ke zdokonalování editorů i k dostupnosti online [1, 10] se možnosti využití $\text{T}_{\text{E}}\text{X}$ u a jeho nadstaveb zvyšují.

O tomto přístupu svědčí opakovaná vydání $\text{T}_{\text{E}}\text{X}$ ových a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ových příruček a učebnic, kurzy sazby na vysokých školách, články o výhodách sazby v $\text{T}_{\text{E}}\text{X}$ u a nespočetné diskuze o tom, zda Microsoft Word je nebo není lepší. [5, 14]

Je-li T_EX takto propagován a představován laikům, obvykle se uvádějí jednoduché a neutrální příklady využití, popř. se počítá s používáním především v akademickém prostředí.

Cílem tohoto příspěvku je prozkoumat přístup k T_EXu (a především k L^AT_EXu) v netypické oblasti, kterou je podnikání – zda se T_EX využívá ve firmách, k jakým činnostem, s jakým úspěchem. Na podporu tohoto přístupu je uveden příklad firmy, která vhodně zapojila T_EX.

S tím pak souvisí i širší problematika přístupu firem k softwaru obecně – zda používat komerční software nebo jeho otevřené alternativy a proč.

2 T_EX a otevřený software ve firemním prostředí

Jednou z klíčových otázek, kterou firmy potřebují řešit, je softwarová politika a výběr používaných aplikací. Přehled kritérií pro výběr softwaru uvádí např. Macák [13]. I v této oblasti se hojně publikují články a vedou diskuze o tom, zda je lepší komerční software nebo zda je otevřený software dostatečně vhodný i pro firemní použití.

Problematika licencí a orientace v oblasti softwaru zřejmě není pro uživatele snadná a situace je mnohdy opředená řadou mýtů (např. [3], snímek 72). Vysvětlení podávají například články o komerci, opensource, licencích a obchodním modelu [7, 8] a přicházejí s osvětou, že nejde jen o cenu softwaru.

Častým požadavkem na software je právě nízká/nulová cena, pokud možno legálně. Petr Hájek v článku „Koupím software do firmy. Zn.: Hlavně levně“ [4] uvádí, že firmy na software vydávají asi 2 % svého IT rozpočtu. Nezdá se to mnoho, při bližším pohledu však může být finanční úspora významná (konkrétní výpočet uvádí např. [16]). Dopady však jsou nejen finanční, volba ovlivňuje procesy a způsob komunikace v celé firmě. V prezentaci „Free Software: An Introduction“ [3] jsou uvedeny přínosy pro firmu. Mezi ně můžeme počítat kompatibilitu s ostatními aplikacemi a systémy i do budoucna, nezávislost na konkrétním podniku, možnost najmout si vlastní vývojáře, internetovou podporu a v neposlední řadě i úsporu nákladů na licence. Kromě toho jsou zde i společenské přínosy, jako omezování monopolů, zamezení sběru dat o uživatelích a jejich sledování, sdílení znalostí nebo spolupráce na vývoji.

Má tedy firma používat svobodný software? Postřehy ze setkání Business Tuesday o tom, zda a kdy se to firmám vyplatí, uvádí článek P. Krčmáře [12].

I zde, podobně jako v mnoha dalších příspěvcích a diskuzních fórech, se velmi často projevuje přístup, že opensource software \equiv Linux. Není to však jediná alternativa a zde se zaměříme na případy, kdy není celý počítač/kancelář/firma 100% opensource, ale používá (např. pro začátek) alespoň některé produkty z této kategorie.

Ve firmě přicházejí v úvahu především kancelářské aplikace. Rozdíly mezi Microsoft Office a OpenOffice.org (a případně dalšími variantami) jsou opět vděčným námětem článků (např. [2]) a diskuzí. Často řešeným problémem je kompatibilita, funkčnost a rozdílnost konkurenčních aplikací. S vývojem nových a zdokonalených verzí se zdá, že otevřený kancelářský balík si nachází stále častěji cestu do firem, škol a úřadů (více viz [15]).

Samotný T_EX je ve spojení s nasazením ve firmách vnímán spíše jako okrajová záležitost. Příležitostné návody a dotazy (například jak udělat vizitku [20], zda se T_EX používá pro firemní hlášení [6]) se týkají již konkrétního nasazení a předpokládají uživatele, který se v T_EXu sám vyzná natolik, že je schopen jej samostatně používat. Jak uvádí L. Maciak [14], „Users should at least have basic understanding or programming, markup languages, compiling and debugging software.“ Častými doporučeními tedy bývá používat grafickou nástavbu LyX nebo pověřit zkušeného uživatele T_EXu (nebo, kvůli maximálnímu zjednodušení, L^AT_EXu) přípravou šablon, do nichž se bude pouze doplňovat text.

Další cestou k T_EXu mohou být komerční implementace [9], které nabízejí například placené nástavby, písma, konvertory a další bonusy.

Vhodným řešením v případě používání dokumentů v T_EXu i v textovém procesoru Microsoft Word mohou být konvertory Word2T_EXa T_EX2Word [11]. Tento software je placený, může ale (paradoxně?) napomoci rozšíření kvalitního otevřeného softwaru.

3 Nasazení T_EXu – případová studie

Zkušenosti z praxe ukazují, že ne pro všechny případy je vhodná stále tatáž softwarová aplikace. Použití textového procesoru je vhodné například pro dopis, ale už ne pro rozsáhlou knihu nebo grafický plakát, který by se zrovna tak nedobře tvořil v T_EXu. Pro orientaci v této džungli softwaru může posloužit nástroj pro optimalizaci výběru programového vybavení, který je navržen a popsán v samostatném článku [17]. V dalším textu se budeme zabývat případy, kdy je vhodný právě typografický systém T_EX.

Výhody, které bývají uváděny v souvislosti s T_EXem (nebo L^AT_EXem) [19], obvykle zahrnují možnost precizní sazby se všemi typografickými požadavky, matematickou sazbu, jednoznačnou strukturu textu (strukturní značkování a srozumitelnost značkování obecně), rozšiřitelnost, programovatelnost a automatizované prostředky, nezávislost na platformě, stálost . . . a cena.

Jako nevýhody se obvykle uvádějí složitost, náročné učení a neinteraktivní zpracování, které mohou odradit uživatele od běžné práce. Pokud se ale v seznamu výhod zaměříme na typografii, možnost přizpůsobení sazby těmito prostředky a na automatizaci procesu sazby, může se T_EX použít jako sazební stroj pro automatické generování dokumentů na základě připravené předlohy.

Toto využití je poměrně běžné, proto se v dalším textu zaměříme na případ, kdy je T_EX zapojen do procesu automatizovaného generování formulářů v malé firmě.

3.1 Situace

Modelovým případem je soukromá firma, která provozuje malé call centrum. Majitelé firmy jsou i manažery, zaměstnávají asi 20 telefonistů, asistentku a vedoucího call centra. Jeden z manažerů má na starosti i správu IT ve firmě a firemní procesy v této oblasti, správu serverů už zabezpečuje firma, která poskytuje hosting.

Hlavním obchodním partnerem je jeden z mobilních operátorů a úkolem telefonistů je komunikace se stávajícími firemními zákazníky tohoto operátora a poté administrativa spojená s uzavíráním smluv. Mobilní operátor trvá na množství formulářů, které je vždy třeba vyplnit, a část pracovní doby by tak byla strávena touto administrativou, tj. neproduktivně.

Mobilní operátor používá platformu Microsoft Office a dokumenty, včetně všech typů formulářů, distribuuje partnerským firmám ve formátu DOC, popř. XLS a automaticky očekává též používání Microsoft Office a uzavřených formátů. (Další variantou je ruční vyplnění vytištěného formuláře.) Vyplněný formulář se odevzdává v papírové podobě, podepsaný.

Malá firma neplýtvá finančními prostředky na softwarové licence. Pro kancelářské aplikace používá OpenOffice.org; ke spolupráci, sdílení a komunikaci slouží i aplikace Dokumenty Google. Specializovaným potřebám souvisejícím s telefonováním slouží vlastní software vyvinutý přímo ve firmě. V rámci firmy funguje komunikace, sdílení dokumentů apod. bez problémů, používání formulářů v DOC a XLS ale nedávalo dobré výsledky, navíc bylo třeba je stále vyplňovat napůl „ručně“, vyhledáním údajů o firmě a zkopírováním do formuláře.

Malé firmě, která ohledně používaného softwaru a souborových formátů nemá žádné možnosti vyjednávat, tak nezbyvá, než na to tvůrčím způsobem reagovat. . .

3.2 Způsob řešení

Vedení call centra řešilo úkol, jak vyplnit „wordové“ formuláře bez Wordu, pokud možno automatizovaně. Vzhledem k tomu, že ve výsledku na souborovém formátu nezáleží (mobilní operátor potřebuje podepsaný list papíru), vzniklo řešení, které nepotřebuje žádný kancelářský systém a spoléhá na L^AT_EX.

Řešení využívá toho, že různé typy smluv jsou standardizované, jejich obsah se mění pouze sporadicky a doplňují se do nich pouze údaje o firmách, jejich zástupcích a požadovaných službách. Proto byly vytvořeny L^AT_EXové šablony pro každý typ smlouvy. Aplikace „Generátor formulářů“ pak vytvoří dokumenty, které jsou – díky možnostem tohoto systému – k nerozeznání podobné od originálů. Zbývá do nich doplnit požadované údaje.

Automatizované zpracování v tomto případě vypadá následovně:

1. Call centrum komunikuje s firemními zákazníky, kteří jsou všichni evidováni buď v Obchodním rejstříku, nebo alespoň v Živnostenském rejstříku, a to včetně požadovaných identifikačních údajů (název firmy, sídlo, identifikační číslo. . .). Téměř všechny údaje do smluv jsou dostupné.
2. Na serveru s OS Linux, který malá firma využívá, je k dispozici *textový webový prohlížeč*. Skrze něj se přistupuje ke stránkám Obchodního nebo Živnostenského rejstříku. Výstupem je textový dokument, který je za vydatného použití regulárních výrazů v jazyce Perl prohledán a údaje se dosadí na patřičná místa ve formulářích.
3. Díky tomuto způsobu zpracování je obsluha pro telefonisty velmi jednoduchá – vstupem je tlačítko pro generování formuláře pro určitou firmu (podle identifikačního

čísla) a výstupem je formulář s doplněnými údaji, kam lze v případě potřeby něco doplnit. Další tlačítko vytvoří soubor ve formátu PDF.

4 Závěr

System L^AT_EX ve spojení s Linuxem a textovým prohlížečem (tj. běžnými uživateli často zatracované nástroje) jsou využity pro maximální automatizaci zpracování formulářů. Produkují tak výsledek, kterého komerční systémy za současného stavu nejsou schopny dosáhnout. Tento přístup vyvrací tvrzení, že co je zdarma, nemůže být dost kvalitní.

Nezanedbatelná je i ekonomická úspora – pokusným měřením bylo ve firmě zjištěno, že vyplňování formulářů „ručním“ kopírováním dat trvá (u vícestránkových smluv, kde se údaje i opakují) – podle typu smlouvy – 8 až 14 minut, generátor formulářů poskytuje výsledky v řádu sekund.

Tento stručný příklad ilustruje skutečnost, že otevřený software, včetně T_EXu, má nebo může mít své místo nejen v akademickém prostředí, ale i v běžných firmách. Asi ne jako nástroj pro přímé použití všemi pracovníky a pro sekretářky na psaní dopisů, ale může se při vhodné přípravě uplatnit při procesu automatizace.

Z hlediska firmy může být T_EX nejen nejen levná a svobodná alternativa, ale přináší i efektivnost, která už alternativy v komerčním světě nemá.

Literatura

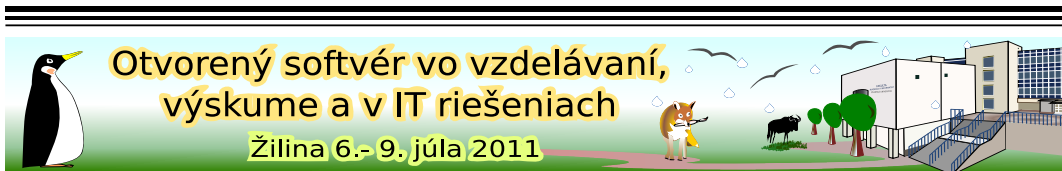
- [1] ČERNÝ, M. *Online L^AT_EX editory – plnohodnotná náhrada?* [online]. 2010 [cit. 2011-05-24]. <http://www.lupa.cz/clanky/online-latex-editory-plnohodnotna-nahrada/>
- [2] ČERNÝ, M. *Osm důvodů proč používat OpenOffice.org 3.1* [online]. 2009 [cit. 2011-05-26]. <http://www.dsl.cz/clanek/1459-osm-duvodu-proc-pouzivat-openoffice-org-3-1>
- [3] *Free Software: An Introduction* [online]. 2009 [cit. 2011-05-26]. <http://www.slideshare.net/PlusOrMinusZero/an-introduction-to-free-software>
- [4] HÁJEK, P. *Koupím software do firmy. Zn.: Hlavně levně* [online]. 2010 [cit. 2011-05-26]. <http://www.root.cz/clanky/koupim-software-do-firmy-zn-hlavne-levne/>
- [5] HANKE, H. *Svobodný software 17 – T_EX vs. MS Word?* [online]. 2001 [cit. 2011-05-24]. <http://www.zive.cz/clanky/svobodny-software-17--tex-vs-ms-word/sc-3-a-103611/default.aspx>
- [6] *Is anybody using T_EX for business reporting?* [online]. 2010 [cit. 2011-05-25]. <http://tex.stackexchange.com/questions/3506/is-anybody-using-tex-for-business-reporting>
- [7] *Komerce a svobodný software* [online]. 2010 [cit. 2011-05-26]. <http://linuxzblizka.blog.zive.cz/2010/08/komerce-a-svobodny-software/>

- [8] *Komerce postavená na open source* [online]. 2010 [cit. 2011-05-26].
<http://linuxzblizka.blog.zive.cz/2010/03/komerce-postavena-na-open-source/>
- [9] KREJČÍ, R. *Komerční implementace T_EXu* [online]. 2003 [cit. 2011-05-26].
<http://www.grafika.cz/art/sazba/komtex.html>
- [10] KREJČÍ, R. *Sazba v L^AT_EXu jako webová služba: Možnosti a řešení* [online]. 2009 [cit. 2011-05-26]. <http://www.grafika.cz/art/sazba/latex-online.html>
- [11] KREJČÍ, R. *Word2T_EXa T_EX2Word aneb Word a T_EX pospolu* [online]. 2002 [cit. 2011-05-24].
<http://www.grafika.cz/art/sazba/word2tex.html>
- [12] KRČMÁŘ, P. *Kdy se firmě vyplatí přejít na open source?* [online]. 2010 [cit. 2011-05-26].
<http://www.root.cz/clanky/kdy-se-firme-vyplati-prejit-na-open-source/>
- [13] MACÁK, P. *Kritéria výběru software pro malé a středně velké společnosti* [online]. 2011 [cit. 2011-05-26].
http://www.cssi.cz/cssi/system/files/all/si_2011_01_10_Macak.pdf
- [14] MACIAK, L. *Why L^AT_EX is Superior to Office* [online]. 2007 [cit. 2011-05-25].
<http://www.terminally-incoherent.com/blog/2007/07/16/why-latex-is-superior-to-office/>
- [15] OpenOffice.org *Používají OpenOffice.org* [online]. 2011 [cit. 2011-05-24].
<http://www.openoffice.cz/pouzivaji-openoffice-org>
- [16] *Případová studie využití open source v obchodní kanceláři* [online]. 2006 [cit. 2011-05-24].
http://www.linuxservices.cz/Open_source_pripadova_studie.pdf
- [17] Rybička, J. – Talandová, P. – Přichystal, J. *Optimalizace výběru programového vybavení*. In ŽUFAN, P. *Firma a konkurenční prostředí 2010*. 1. vyd. Brno: Provozně ekonomická fakulta, Mendelova univerzita v Brně, 2010, s. 132–133. ISBN 978-80-7375-385-6.
- [18] *What professions use T_EX/L^AT_EX besides CS?* [online]. 2011 [cit. 2011-05-25].
<http://tex.stackexchange.com/questions/940/what-professions-use-tex-latex-besides-cs>
- [19] *Why T_EX?* [online]. 2011 [cit. 2011-05-25].
<http://lpar.ath0.com/2011/04/05/why-tex/>
- [20] WIDMANN, T. *How to make business cards in L^AT_EX* [online]. 2009 [cit. 2011-05-26].
<http://blog.widmann.org.uk/2009/05/27/1297/>

Kontaktní adresa

Petra ČAČKOVÁ (Ing., Ph. D.),

Ústav informatiky, Provozně ekonomická fakulta, Mendelova univerzita v Brně,
Zemědělská 1 613 00 Brno, petra.cackova@mendelu.cz



DOI: 10.5300/2011-OSSConf/189

NECH $\text{T}_{\text{E}}\text{X}$ NÁS BÝT: POHLED NA KRUŠNÉ ZAČÁTKY SAZEČSKÉ Z POHLEDU NESAZEČŮ

SARGA, Libor, (CZ)

Abstrakt. Článek popisuje tvorbu sborníku příspěvků v systému $\text{T}_{\text{E}}\text{X}$, formátu $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, z pohledu amatérských sazečů spolu s problémy a těžkostmi, které byly při této práci zaznamenány a vyřešeny. Zvláštní pozornost je věnována peripetiím při tvorbě obsahu a popiskům grafických objektů, jsou však nastíněny také drobné problémy a triky, které je vyřešily. Rovněž je popsán systém správy a synchronizace verzí souborů, použitý při decentralizované manipulaci s hlavní šablonou sborníku, jenž je využitelný a vyzkoušený v praxi.

Klíčové slová. sazba, $\text{T}_{\text{E}}\text{X}$, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, *tocloft*, *caption*, tabulky, obrázky.

GUIDE $\text{T}_{\text{E}}\text{X}$ IT: UNEASY BEGINNINGS OF TYPESETTERS FROM THE PERSPECTIVE OF NON-TYPESETTERS

Abstract. The article describes the process of typesetting a proceedings in $\text{T}_{\text{E}}\text{X}$, the $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ format, from the perspective of prospective typesetters along with challenges and obstacles encountered and solved during the work. Focused on the problems of generating a desired Table of Contents and captions of graphic objects, it further lists minor annoyances and tricks used to solve them. Also described is a field-proven electronic content management and synchronization system for different file versions utilized while working on the project in a decentralized fashion.

Key words and phrases. typesetting, $\text{T}_{\text{E}}\text{X}$, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, *tocloft*, *caption*, tables, figures.

1 Úvod

Dříve, než se pustíme do rozboru, jak jsme se dostali k sázení monografie v systému $\text{T}_{\text{E}}\text{X}$, formátu $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, a jaké peripetie při tom zažili, bych rád zmínil, jaké byly mé zkušenosti dříve, než jsem tuto nabídku dostal a přijal. Bude to sice pojednání velmi krátké, ale pro pochopení pointy článku velmi důležité.

V září 2010 jsem nastoupil do prvního ročníku doktorského studia na Ústav statistiky a kvantitativních metod Fakulty managementu a ekonomiky Univerzity Tomáše Bati ve Zlíně. Protože jsem si už tehdy byl vědom, že pro profesionální úpravu svých článků a dalších písemných výstupů bude nutné zvolit sofistikovanější systém, než je na univerzitách běžně dostupný komerční software, začal jsem hledat. Poměrně brzy jsem narazil na systém \TeX . Stáhl jsem si nepříliš krátký úvod¹ a slíbil si, že se na něj během krátké doby určitě podívám.

To jsem také udělal, zjistil, co je to preambule, co do ní napsat, jak se načítají balíčky a další podstatné náležitosti, jež jsem však neměl z důvodu chybějícího vybavení možnost prakticky vyzkoušet. Situace se změnila začátkem října 2010, kdy jsem potřebné zázemí a čerstvou instalaci systému \TeX Live 2010 získal. Měl jsem to štěstí, že jak Lubor Homolka, další doktorand na našem ústavu, tak Pavel Stríž se o tento nástroj živě zajímají a dodnes mi slouží radou i ukázkou. Stále jsem však zůstával a zůstávám pouhým začátečníkem, který se rozhodl pozvolna proniknout do tajů „texu“ (fonetický přepis mé výslovnosti v počátcích) s pomocí editoru \TeX Maker a touhou se učit. Když jsem poprvé zkompiloval svůj vlastní PDF dokument, bylo mé nadšení nepopsatelné:

```
\documentclass[a4paper]{article}
\begin{document}
Hello world!
\end{document}
```

Byl jsem se sebou spokojen, experimentoval jsem podle návodu s balíčky, zkoušel vkládat obrázky, naučil jsem se přepínat jazyky. Lubor Homolka v té době již objevil krásu sázení matematických vzorců. Závistivě jsem přihlížel, jak své dokumenty, čítající na desítky řádků, kompiloval bez chyb, zatímco já zkoumal esteticky pěkně vyvedená červená hlášení, kterými mě editor častoval. Dodnes s těžko skrývaným očekáváním doufám, že se po spuštění kompilace a hláše

```
Process started
```

```
dočkám úspěchu ve formě
```

```
Process exited normally.
```

```
nebo se místo tohoto požehnání objeví obávané
```

```
Process exited with error(s).
```

Poslední zmíněnou zprávu jsme později při sázení vídávali více než cokoliv jiného, čítající v to naše přátele, rodiny či dna pivních sklenic.

¹www.ctan.org/tex-archive/info/lshort/english/lshort.pdf

2 Kniha

Relativně klidná situace se rychle změnila po e-mailu Pavla Stříže, tázajícího se, zda bychom neměli zájem o vysázení konferenčního sborníku příspěvků pro nakladatelství jeho bratra². Již toto gesto důvěry ve mě zvedlo vlnu nadšení: konečně budu moct své vytříbené znalosti, sice podtržené občasným neúspěchem ve formě chyb, ale o to triumfálněji završené ukázkovými PDF dokumenty, ukázat světu. Již tehdy, před polovinou října, jsem však v textu nemohl přehlédnout jistý tón spěchu. Indicie typu „hořící zakázka“ či „termín dva, možná až tři týdny“ prozrazovaly, že se bude jednat o nárazovou akci. Také jsme se dozvěděli, že sborník bude obsahovat zhruba 250 stran, 60 obrázků a 50 tabulek. Svůj varovný hlas jsem rychle zaplašil a zakázku jsme po vzájemné dohodě přijali. Nermalou měrou se na tom podílel i Pavel Stříž, slibivší nám pomoc v případě jakýchkoliv problémů. Toho jsme se ale již na začátku rozhodli kontaktovat až po vyčerpání všech dalších možností. Z ukázky, kterou jsme získali, se nám jako hlavní problém jevily tabulky, které zřejmě bude nutné převést do jednotného formátu, a obrázky, vyžadující konverzi do PDF (jak, to jsme zatím nevěděli).

Dne 15. října 2010 jsme získali dokument se všemi příspěvky ve formátu .doc. Tento 3,822 MB velký soubor obsahoval všechny příspěvky v anglickém jazyce spolu s obrázky a tabulkami. Pustili jsme se do práce. Od tohoto dne také vznikl hojně využívaný komunikační kanál s bratrem Pavla Stříže, Martinem. I ten poskytl cenné rady, především co se týče základního nastavení, typu sazby i dalších náležitostí, o nichž jsme neměli tušení (chlácholili jsme se tím, že to ani vědět nemůžeme, protože nejsme tiskaři). Kniha měla být ve formátu A5 (148 × 210 mm), čemuž bylo přizpůsobeno i nastavení tiskového zrcadla v \TeX ové šabloně. Zároveň jsme museli uznat svou slabost a odmítnout vytvoření obálky přes PSTricks, což skutečně bylo nad naše síly.

Hned na začátku jsme byli donuceni vyhledat jiný prohlížeč PDF, než byl univerzitní Acrobat Reader 9. Ten si totiž každý PDF soubor při otevření zamyká a není možné jej příkazem z editoru překreslit. Až později jsem zjistil, že je to způsobeno tím, že stejně jako mnoho dalších (Foxit PDF Reader) přímo nepodporuje DDE (Dynamic Data Exchange). Řešením může být místo odkazu na absolutní umístění vytvořit jednoduchý dávkový soubor, který po každém spuštění kompilace program zavře a znovu otevře, tedy formu nepřímé výměny dat. To jsme však netušili a zvolili program Sumatra PDF, s nímž jsme byli velmi spokojeni.

Dále jsme ve svých editorech synchronizovali nastavení, především kódování UTF-8, neboť Lubor Homolka se rázně distancoval od komerčních operačních systémů ve prospěch Linuxu, a tato nesourodá prostředí by mohla působit problémy. Navíc se bez UTF-8 komentáře s českou diakritikou měnily ve směs otazníků a dalších roztodivných znaků, jež se v nich původně vůbec nevyskytovaly, a to i mezi stejnými verzemi Windows. Balíček `inputenc` s předvolbou `[utf8]` poskytl řešení. Pro vyříznutí obrázků jsme se rozhodli využít licencovaný Adobe Acrobat 8 ve verzi Professional, který kromě dalších obsahuje funkci oříznutí (`crop`). Vhodnou alternativou je využití interního parametru `viewport` příkazu

²<http://www.striz.cz/>

`\includegraphics`, kde se jako parametr vyskytují souřadnice. Tak je možné celý proces pohodlně automatizovat. Toto jsme však sami nevyzkoušeli, protože se nám při předtuše souřadnicových deklarácí udělaly mžitky před očima.

Náš postup extrakce obrázků tedy vypadal tak, že jsme si celý dokument s příspěvky převedli do PDF, pomocí open source nástroje PDF Split and Merge (<http://www.pdfsam.org/>) extrahovali pouze stránky s obrázky a ty pak v Adobe Acrobat Professional vyřízli a exportovali do PDF, přičemž vhodnou alternativou pro první krok je také `pdftk` (<http://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/>), což jsme však neměli možnost vyzkoušet. Tak byla zaručena škálovatelnost, jíž „statickými“ formáty typu JPEG či PNG (o něž jsme se také pokoušeli) nebylo možné docílit. První jmenovaný formát je navíc v základní verzi ztrátový – existuje ale i novější JPEG2000, který už podporuje bezztrátovou kompresi – a trpí syndromem tzv. generation loss, kdy se při opakovaném ukládání snižuje kvalita (přesněji zvyšuje se poměr šumu k obrazové informaci), což je i při malém přiblížení okem snadno pozorovatelné, což řešil právě bezztrátový PNG. Výhodou PDF oproti oběma zmíněným formátům je právě možnost zoomovat, bohužel použitý program byl, jak vtipně poznamenal Martin, „dražší jak průměrný plat v ČR“. Byl to však také jediný komerční software, použitý během tvorby sborníku.

Objevili jsme i makro „`Vlna`“³ autora Petra Olšáka. V dokumentech se totiž často stává, že se na konci řádku vyskytne některá ze spojek „a, i, o, k“, což i my sazeči-laici považujeme za nedostatek. Příkazem

```
vlna -l -m -n dokument.tex
```

bude zajištěno, že namísto manuálního vkládání „`vlnek`“ (správně česky tild, slovensky povedané vlnoviek) se toto provede automaticky. Přepínač `-l` zapíná \LaTeX ový mód, `-m` makru nakáže ignorovat stav matematických prostředí a `-n` to stejné udělá s prostředím verbatim.

Bohužel toto makro lze aplikovat pouze na český text, což nám ve sborníku, vydávaném v anglickém jazyce, příliš nepomohlo. Dávali jsme si proto zvláštní pozor na čísla a jednotky, dělící se přes dva řádky, a několik z nich jsme již v raném stádiu opravili. Pak jsme však zjistili, že jakékoliv manipulace s objekty či tabulkami nám text deformují, takže velké množství tild bylo v textu zcela zbytečných. Postřeh, jenž jsem si z této události odnesl, byl, že není nutné řešit všechny problémy najednou, ale vytvořit si určitou hierarchii.

3 Drobné radosti a starosti

Od října tedy začal souboj z našeho pohledu s nevypočítatelným systémem. Velmi záhy jsme zjistili, že schůdnou cestou (takovou, která bude generovat nejmenší množství chyb) bude vytvořit hlavní šablonu, do níž budou příkazem `\input{*}` vkládány jednotlivé kapitoly, což značným způsobem usnadňovalo orientaci v kódu. Co se týče balíčků, zabývali jsme se především dvěma: `tocloft` a `caption`.

³<http://ftp.linux.cz/pub/tex/local/cstug/olsak/vlna/>

3.1 Seznamy: tocloft

`tocloft`⁴ je známý soubor maker pro parametrizaci při tvorbě obsahů. My jsme potřebovali vytvořit seznamy tři: kapitol (ToC – Table of Contents), obrázků (LoF – List of Figures) a tabulek (LoT – List of Tables). Právě u posledních dvou zmíněných se objevil první velký problém, protože když byly v rámci `caption` změněny názvy „Obrázek“ na „Figure“ a „Tabulka“ na „Table“, vysázelo se do seznamu pouze jejich číslo, a nikoliv toto slovo. Zkoušeli jsme situaci obejít tak, že bychom nechali základní názvy a pomocí příkazů

```
\renewcommand*\figurename}{Figure}
\renewcommand*\tablename}{Table}
```

je změnili mimo balíček, ale výsledek byl stejný. Nejsem si jist, zda by byl problém vyřešen pouze tím, že bychom před `babel` napsali `[english]`, ale doufám, že jsme toto v průběhu našich experimentů vyzkoušeli. Nakonec jsme se museli uchýlit k nepřilíh čistému řešení. To spočívalo nejprve v odsazení všech položek v seznamu tabulek a obrázků příkazy

```
\addtolength{\cftfignumwidth}{25pt}
\addtolength{\cfttabnumwidth}{20pt}
```

a následně přidáním slov „Figure“ a „Table“ takříkajíc natvrdo do uvolněného místa, čehož jsme docílili příkazy

```
\renewcommand{\cftfigpresnum}{Figure }
\renewcommand{\cfttabpresnum}{Table }
```

Jemným doladěním hodnot jsme docílili výsledku na obrázku 1 na další straně.

Považovali jsme to za první viditelný úspěch našeho snažení, navíc pro dřívějšího uživatele WYSIWYG⁵ nástrojů to byl rovněž milník v pochopení toho, že za zdánlivě jednoduchými výstupy se skrývá hodně práce.

Mým soukromým poznatkem bylo rovněž to, že dokumentace k balíčkům jsou neocenitelným zdrojem informací, přestože některé z nich jsou hodně technické, například právě `tocloft`. Proto jsme se domluvili na systému, kdy já měl na starosti oba balíčky (`tocloft` a `caption`) a vždy, když se vyskytl nějaký problém, jsem automaticky otevíral příslušný soubor s nápovědou. Lubor pak na sebe vzal taktéž nelehký úkol manipulace a ořezávání obrázků, sice nevyžadující žádné manuály, ale vykoupené nemalými požadavky na trpělivost a přesnost.

⁴<http://www.ctan.org/tex-archive/macros/latex/contrib/tocloft/>

⁵What You See Is What You Get editor umožňuje ihned vidět všechny provedené změny bez nutnosti znovu kompilovat celý dokument.

Table of Contents	List of Figures	List of Tables
List of Figures 5	Figure 1 Three Possible Scenarios for the Development of Potential Product in the Eurozone 44	Table 1 The depth of the two crises: Ten industrialized countries 21
List of Tables 8	Figure 2 Average yearly inflation in Eurozone countries, 2002-2008 48	Table 2 GDP development in main regions 24
Introduction 10	Figure 3 GDP development in Eurozone countries before and after the introduction of the single currency Euro 51	Table 3 Comparison of the two crises: industrialized vs. non industrialized countries: Growth of real GDP 29
I. Global Economic Crisis Explored		
1 Core versus Periphery in the Recent Recession as Compared to the Great Depression 16	Figure 4 Development in long-term interest rates in selected Eurozone countries (1999-2007) 52	Table 4 Comparison of the two crises: New Member Countries and Neighbors: Growth of real GDP 30
2 Eurozone Before and During the Financial and Economic Crisis 39	Figure 5 Growth in Government Default Risk (the interest rate on CDS contracts for government bonds in selected Eurozone countries) 56	Table 5 Macroeconomic performance of the EU-10 31
3 Growth Crisis in the EU 60	Figure 6 Potential growth in the country-groups of the EU 69	Table 6 External and fiscal balances in the EU-10 32
4 Asset Price Fluctuations and the Financial Crisis 83	Figure 7 Contribution of the capital accumulation to the potential growth 73	Table 7 Main Macroeconomic Indicators: Comparison 42
II. Global Economic Crisis Outside of Eurozone		
5 The Effects of the Global Crisis on Turkish Economy and Existing Fiscal Policies for this Crisis 118	Figure 8 Contribution of the TFP to the potential growth 74	Table 8 Development of inflation and interest rates: comparison 46
6 Determinants and Absorption of Exchange Market Pressure in Selected New EU Members 149	Figure 9 Potential growth in the new MSs (annual change in %) 75	Table 9 Public indebtedness and debt servicing costs as a % of GDP 53
7 Impact of the World Economic Crisis upon Measures of Convergence and Preparedness of the Candidate Countries to Join the Eurozone: Are We Better Prepared for the Euro? 168	Figure 10 Selected indicators on the size and impact of the economic crisis in EU-27 Member States 55	Table 10 Selected indicators on the size and impact of the economic crisis in EU-27 Member States 55
III. Impact of Crisis across the Economic Landscape		
8 Incentives to Irresponsible Behavior and Present Crisis 184	Figure 11 Potential GDP growth under different shocks (annual growth rate) 79	Table 11 Potential GDP growth rate (annual average as percentage) 61
9 The Influence of Official Development Assistance on Economic Development of the Selected Group of Developing Countries around the World 210	Figure 12 Three stylized paths of asset prices 85	Table 12 Labour productivity (annual average growth rate as percentage) 67
10 Evaluation of the Development of Unemployment Rates with respect to the Real GDP Growth Rate 245	Figure 13 The movements of the Dollar/Euro exchange rate and technical trading signals, 1999-2008 89	Table 13 Potential growth in the European Union 70
IV. In Lieu of Conclusion		
11 The Information in the Economic Discourse and Analysis (Some thoughts about the role and uses of information) 254	Figure 14 Technical trading signals based on intraday Dollar/Euro exchange rates, June, 6-13, 2003, 5-minute data 89	Table 14 Potential growth and its factors in the country groups 72
	Figure 15 Technical trading signals for WTI crude oil futures contract, July and August, 2000 96	Table 15 Potential growth in the EU, USA and Japan 77
	Figure 16 Technical trading signals for 1092 technical models and the dynamics of oil futures prices, January 2007 to June 2008 99	Table 16 Potential growth in the European Union 77
	Figure 17 Aggregate trading signals of 1092 technical models and the dynamics of oil futures prices, January 2007 to June 2008 99	Table 17 Features of three hypothetical "worlds" of financial markets 87
	Figure 18 Dollar/Euro exchange rate and purchasing power parity 101	Table 18 Runs of the S&P 500 exchange rate 1999/2005, daily data 92
	Figure 19 Dollar exchange rate and oil price fluctuations 104	Table 19 Non-random components in the duration of exchange rate runs, daily data 93
	Figure 20 World market for crude oil, oil futures trading and oil price movements 102	Table 20 Non-random components in the duration and slope of exchange rate runs, 30-minute data 94
	Figure 21 Dynamics of commodity futures prices and derivatives trading activities, 2007-2008 104	Table 21 Hypothetical transaction tax receipts in the global economy 2007 113
	Figure 22 Stock market value and net worth of non-financial corporations 105	Table 22 Environment of the Crisis 126
		Table 23 Results of F-tests 150
		Table 24 Results of Davidson-MacKinnan and Miss-Robust tests 156
		Table 25 Estimation of Model 3 157
		Table 26 Estimation of Model 4 163
		Table 27 Descriptive statistics of GDP growth in Eurozone members, CEE and Baltic countries 172

Obrázek 1: Obsah a první strany Seznamu obrázků a tabulek.

3.2 Obrázky: caption

K balíčku `caption`⁶ jsme po prvních zkušenostech s anglickými názvy obrázků a tabulek přistupovali s nedůvěrou, velmi mu ale v našich očích pomohla srozumitelná a snadno přístupná dokumentace autora Axela Sommerfeldta. S tímto balíčkem nakonec byla velmi snadná práce. Mohli jsme například zcela eliminovat příkazy `\renewcommand{*}{*}` a změnit názvy „Obrázek“ a „Tabulka“ na „Figure“ a „Table“ nastavením těchto parametrů přímo v preambuli

```
\usepackage{caption}
\captionsetup{figurename=Figure,tablename=Table},
```

což je velice elegantní řešení. Snažili jsme se totiž maximálně využít možností nastavení, jež nám jednotlivé balíčky poskytovaly, namísto využívání globálních změn, které by mohly mít neočekávané důsledky.

Obavy jsme měli z odkazů na internetové stránky v popisech, čehož jsme si všimli ihned. Dříve, než jsme přišli na balíček `url`, jsme všechna lomítka v odkazech museli předsadit lomítkem zpětným, což se projevilo jako kontraproduktivní, neboť po využití `url` jsme museli všechny tyto nadbytečné znaky vymazávat. Navíc bez použití `url` byly odkazy brány jako text a nebylo možné je rozkliknout. Další z poznatků sazeče-začátečníka do budoucna, seznámit se se všemi možnostmi `TeX`u dříve, než začnu vymýšlet náhradní, provizorní a pracnější řešení.

Výsledný popis obrázků a tabulek je znázorněn na obrázku 2 na další straně.

Všechny grafické objekty byly navíc při vkládání označeny řetězcem `[!hpb]`, který jsme získali jako radu od Pavla Striže poté, co se nám všechny větší obrázky sázely na

⁶<http://www.ctan.org/tex-archive/macros/latex/contrib/caption/>

Table 7 Main Macroeconomic Indicators: Comparison

Source: EC (2008)

Figure 41 ISE National 100 Index (Based on the Closing Prices)Source: The Undersecretariat Turkish Treasury, Ekonomi Sunumu, http://www.hazine.gov.tr/irj/go/km/docs/documents/TreasuryWeb/Statistics/EconomicIndicators/egosterge/Sunumlar/Ekonomi_Sunumu_ENG.pdf, p. 129

Obrázek 2: Ukázka popisu tabulky a obrázku.

konec příspěvků, což poněkud kazilo dojem ze čtení, zejména v situaci, kdy na ně autor odkazoval v textu. Až později jsem se dozvěděl, že jednotlivé znaky symbolizují, kam všude povolujeme \TeX u objekty umisťovat.

Posledním poznatkem je výhoda při využívání `\textwidth` a `width` namísto `scale`, což jsme rovněž zjistili až v praxi. Zatímco první příkaz škáluje objekt v poměru k šířce textového zrcadla, druhý toto nerespektuje a škáluje absolutně. Často se nám pak stávalo, že jsme byli nuceni manuálně měnit hodnoty `scale`, neboť objekt přesahoval šířku tisknutelné plochy A5. Použitím jakékoliv hodnoty menší nebo rovnu jedné v první deklaraci (například `width=0.5\textwidth`) tato situace nastat nemůže.

3.3 Tabulky

Tabulky byly po seznamech dalším kámenem úrazu. Měli jsme v tomto směru v zásadě dvě možnosti: všechny překlomit do prostředí \TeX u, nebo se spolehnout na osvědčený způsob vyříznutí z PDF. Protože mým největším úspěchem v tomto ohledu bylo úspěšné zvládnutí deklarace `{c|c|c}`, kterým se obsah třech sloupců tabulky zarovnával na střed, a protože některé zdrojové tabulky by se lépe vyjímaly na listu formátu A3, než A5, rozhodli jsme se pro druhý způsob.

Problémy nastaly takřka okamžitě, především co se formátu výstupu týkalo. Několik extrémně velkých tabulek, rozdělených na dvě strany, bylo nutné určitým způsobem dostat na jedinou, aby bylo možné je vyříznout. Toto jsem na svém neširokoúhlém monitoru vyřešil jednoduše tak, že jsem v Microsoft Word nastavil formát papíru na velikou hodnotu (spolehlivě fungovala B0, tedy 1000×1414 mm), poté jsem pomocí virtuální tiskárny PDFCreator⁷ nastavil shodný formát a vytiskl do PDF.

Než jsem však přišel na správnou velikost papíru, uplynul nějaký čas. Stejně tak bylo i výsledný dokument nutné v Adobe Acrobat škálovat, neboť 1,4 metrů široký monitor jsme k dispozici neměli. Byli jsme ale spokojení. . . dokud jsme nezjistili, že jsme opomněli převést všechny tabulky do jednotné podoby. To byl závažný nedostatek, který znamenal pro

⁷<http://sourceforge.net/projects/pdfcreator/>

část hotových tabulek nutnost proces znovu opakovat. Naštěstí nám v tomto ohledu pomohl Martin Stříž, který se tohoto repetitivního úkolu zhostil a pomohl nám tím ve chvíli, kdy nám došlo, že jsme několik dní pracovali na něčem, co teď můžeme smazat.

3.4 Různé

Jedním z poznatků, který jsem si z této práce odnesl, je rovněž nutnost pojmenovávání a správy souborů. V průběhu času se nám v e-mailech, složkách i vyměnitelných discích ocitaly šablony, pokusy funkční i nefunkční, tabulky, obrázky atd. Udržet pořádek bylo imperativní především proto, že jsme pracovali decentralizovaně, každý na své části. Každý z nás však měl jiný systém „údržby“ souborů, proto se občas fungující pokusy musely opakovat jenom proto, že šablona, v níž byly umístěny, byla nahrazena starší verzí, jenž však zase měla fungující jinou část kódu. Řešením bylo v mém případě dělat si každý den před začátkem „ \TeX ování“ zálohu včerejší šablony, pracovat, a na konci dne nové dodatky kopírovat do této starší verze. Tak vždy existovala aktuální šablona jako jádro, jenž byla dle potřeby doplňována. Navíc jsme se mohli spolehnout, že existuje vždy něco (maximální ofset byl 24 hodin) funkčního pro případ kontroly.

Co se týče pojmenování souborů, spontánně jsme zavedli systém, k dokonalosti vypilován až v průběhu prací. Obrázek jsme zkrátili na `figXX`, tabulku na `tabXX`, v případě, že byla tabulka umístěná na dvou stranách, značili jsme soubory `tabXXa` a `tabXXb`, celkovou tabulku pak opět `tabXX`. Pro názvy jednotlivých kapitol jsme využívali notace `kapXX`, hlavní šablonu jsme označovali jako `sablona_hlavni`. Tak bylo i z rychlého pohledu zřejmé, jak je sborník členěn a rychle šlo vyhledat žádaný soubor.

Vyskytlo se také mnoho dalších drobných překážek, které jsme vyřešili svépomocí, přesněji pomocí vyhledávání na fórech uživatelů \LaTeX u. Dodnes mám například uloženu tak základní akci, jako je umístit za číslo sekce tečku a mnoho dalších tipů, jež začátečníci potřebují, a které jsou pro zkušené uživatele otázkou invokace správného příkazu (`\renewcommand{*}{*}`), potažmo balíčku, obsahujícího potřebnou funkcionalitu (`titlesec`). Spoustu věcí jsme již také určitě zapoměli, neboť jsme je už znovu \TeX em nepoužili, spoustu z nich bychom také dnes již řešili jinak, například místo balíčku `fancyhdr`, který jsme používali pouze proto, abychom mohli nadeklarovat styl kapitol bez číslování stránek, lze jednoduše použít příkaz `\thispagestyle{empty}`. Také vdovám a sirotkům⁸, jež jsme nejprve provizorně řešili mírným zvětšením obrázku tak, aby tento jev nenastal, se dá jednoduše předejít nastavením vyšší hodnoty ve `\widowpenalty=XX` a `\clubpenalty=XX`.

Věc, kterou jsme na začátku opomenuli, byly uvozovky. V této oblasti jsme nebyli příliš kovaní, ani co se týče českého jazyka, a proto jsme se neváhali zeptat, nebo přesněji řečeno, byli jsme upozorněni, že se v tomto dělají chyby a abychom se jich vyvarovali. Slíbili jsme, že ano, ale stejně jsme je udělali. Situace totiž byla ztížena tím, že příspěvky psali zahraniční autoři (uvozovky byly „nečeské“), ale sborník byl vydán v České republice

⁸Vdovou je v typografii označován první řádek odstavce, který se vyskytne na posledním řádku stránky, sirotkem pak poslední řádek odstavce, vyskytující se na prvním řádku nové stránky.

(tedy by zřejmě uvozovky „české“ být měly), čemuž jsme nakonec dali přednost. To nás stálo nové procházení textu a vyhledávání. Abychom se vyhnuli dalším nedorozuměním s vyhledáváním uvozovek přes Alt+ kódy, zkopírovali jsme si dvojici uvozovek z článků na české a anglické verzi Wikipedie. České uvozovky vypadají v textu „takto“, anglické pak “takto”, neboli české uvozovky se podobají dvojici 99 dole a 66 nahoře, zatímco anglické 66 a 99 nahoře. To jsme pochopili a řídili se tím.

Dokud jsme neobjevili jednu záludnost, a to jsou uvozovky v uvozovkách, které se rovněž liší. České vložené uvozovky „vypadají ,takto““, anglické “ale ‘takto’” (čili 6 dole, 9 nahoře a 6 a 9 nahoře). Další prohledání celého textu a ujištění se, že i všechny vložené uvozovky jsou skutečně anglické. Byly a my to považovali za konec našich útrap s tímto typografickým gulášem. Těžko se nám pak chápalo, že někteří autoři pro vložené uvozovky nevyužívali běžně respektovaných znaků (uvedených výše), ale znaků apostrof (’), ve zdrojovém textu pouze těžko rozlišitelných. Další kontrola celého textu, převedení apostrofů na vložené uvozovky a prohledání anglické klávesnice za účelem zjištění, které znaky by se ještě daly využít místo vnořených uvozovek, nám přinesly kýžený klid: žádné. Až později jsme zjistili, že se v textu vyskytl ještě jeden „typ“ těchto uvozovek, vytvořený za pomoci čárky, běžně používané nad písmeny. Kontrola textu nám pak po předchozích zkušenostech již nezabrala moc času.

4 Závěr

Tvorba sborníku byla pro nás oba velmi dobrou zkušeností. Dílo jsme s vydatnou pomocí Pavla Stříže a jeho bratra Martina odevzdali ještě dříve, než byl smluvený termín. Po drobných úpravách, korektuře a externí tvorbě obálky jsme přibližně v polovině listopadu obdrželi zprávu, že je kniha hotová. Když jsme dostali do rukou fyzickou kopii sbírky, byli jsme na sebe velmi pyšní, protože každý obrázek, tabulka, vzorec, uvozovka, čárka a spojka námi byla alespoň jednou (spíše několikrát) zkontrolována. I když se v textu před tiskem objevily další chyby, nedávali jsme si za cíl stoprocentní úspěšnost, ale spíše zjištění, zda takový úkol zvládneme.

Zvládli jsme a naučili se přitom spoustu nových věcí, které nyní využíváme při tvorbě materiálů na výuku, kde samozřejmě neopomíjíme zmínit, že se jedná o výstupy zpracovávané v systému \TeX . Studenti by totiž měli mít možnost vidět, že kvalitní typografie lze docílit právě bez použití komerčního software, pouze s open source řešeními. Vezmeme-li pak v úvahu doplňkový software pro oblast statistiky, například jazyk R^9 , nástroj gretl¹⁰ či grafickou utilitu gnuplot¹¹, je možné konstatovat, že kvalitní vědecká díla lze tvořit s minimálními (nulovými) náklady.

⁹<http://www.r-project.org/>

¹⁰<http://gretl.sourceforge.net/>

¹¹<http://www.gnuplot.info/>

Post Mortem

Nebojte se a začněte!

Poděkování

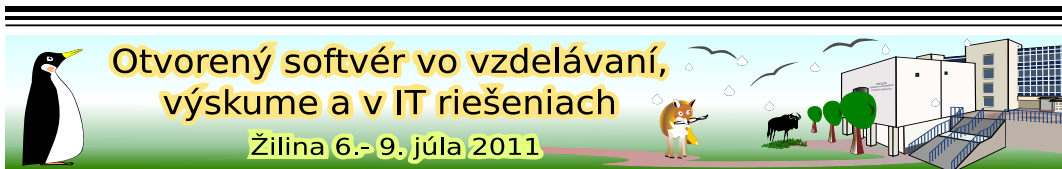
Tento příspěvek vznikl za částečné podpory ESF projektu č. CZ.1.07/2.2.00/07.0361.

Literatura

- [1] LACINA L. – ROZMAHEL P. – RUSEK A. (editors): *Financial Crisis: Institutions and Policies*. Bučovice, Czech Republic: Martin Stříž Publishing, 2010. 268 pp. Více na <http://www.striz.cz/36crisis.php>. ISBN 978-80-87106-37-2.

Kontaktní adresa

Libor SARGA (Ing.),
Tomas Bata University in Zlín,
nám. T. G. Masaryka 5555,
760 01 Zlín, Czech Republic,
sarga@fame.utb.cz



DOI: 10.5300/2011-OSSConf/199

TYPOGRAFIE A ČÁROVÉ KÓDY

STRÍŽ, Pavel (CZ)

Abstrakt. *Letos organizátoři OSSConf zkusili propagaci akce i pomocí QR kódu, autora napadlo, že by nemuselo být marné se podívat čárovým kódům na zoubek hlouběji. Tento příspěvek je o sazbě čárových kódů v typografickém systému \TeX , především o projektu PSTricks (<http://tug.org/PSTricks/>). V ukázkách se podíváme na vývoj sazby čárových kódů, od ean13.tex [1, 2] přes ean13isbn [6] a vyzkoušíme si konkrétní příklady u projektu PSTricks [5], speciálně pomocí balíčku pst-barcode (dokumentace).*

Klíčová slova. Čárový kód, ISBN, EAN, \XeTeX , PSTricks, QR Code, PDF 417, balíčky ocr-b, ocr-b-outline, pst-barcode a auto-pst-pdf.

TYPOGRAPHY AND BARCODES

Abstract. *This article introduces typesetting possibilities of barcodes in \TeX . It presents several examples using ean13.tex macro and packages such as ean13isbn, ocr-b-outline, ocr-b, auto-pst-pdf as well as the pst-barcode package from the PSTricks project.*

Key words and phrases. Barcode, ISBN, EAN, \XeTeX , PSTricks, QR Code, PDF 417, packages ocr-b-outline, ocr-b, pst-barcode and auto-pst-pdf.

1 Makro ean13.tex

Nejstarší autorovi známá sada maker pro čárové kódy EAN (European Article Number) a ISBN (International Standard Book Number) v \TeX u je od Petra Olšáka z roku 1995. Dokumentaci lze vyvolat pomocí `texdoc ean13`, případně lze přímo nahlédnout do souboru `ean13.tex`, který naleznete ve složce `texmf-dist/tex/generic/ean/` (přímý odkaz). Více o tomto zdrojovém kódu ve Zpravodaji [1] či v časopisu TUGboat [2].

Vytvoříme si soubor `test1.tex`, který bude vypadat takto:

```
\documentclass[a4paper]{article}
\begin{document}
```

```

\pagestyle{empty}
\input ean13.tex
\font\ocrbsmall=ocrb7 scaled 890
\def\ISBNb #1 {\def\ISBNnum{
  \makebox[10pt][r]{ISBN }#1}
  \barheight=45.151515\X\relax}
\ISBNb 978-80-87106-36-5
\EAN 9788087106365
\end{document}

```

Tento kód byl poprvé použit na skutečné knize u nakladatelství Martin Stříž, použil se L^AT_EX formát a zasáhlo se do formátování tak, aby textové výstupy lícovaly. Výstup ve formátu PDF (na straně A4 nebo na americkém letter, dle instalace a nastavení T_EX kolekce) dostaneme spuštěním:

```
pdflatex test1.tex
```

Takto díky dvěma příkazům (`\ISBNb` a `\EAN`) můžeme s čárovým kódem pracovat kdekoliv v dokumentu. Někdy je však výhodné mít takový grafický element ve zvláštním souboru a ořezaný o volné bílé oblasti, případně nastavenou ochrannou zónu. Toho dosáhneme nejrychleji např. takto:

```
pdfcrop --hires --margins 2 test1.pdf test1out.pdf
```

Nově vzniklé PDF je ořezané s dvěma body (typografické jednotky bp) jako ochrannou zónou. Pomocí balíčku `graphicx` a příkazu `\includegraphics` můžeme rovnou načíst do libovolného dokumentu.



Obrázek 1: Horní i spodní část je rastrovaná.

Např. pro účely náhledu příspěvku pro openMagazin se vytváří i rastrovaná verze, to můžeme zrealizovat pomocí nástrojů ImageMagick (<http://www.imagemagick.org/>) či GraphicsMagick (<http://www.graphicsmagick.org/>) takto:

```
convert -density 300 test1out.pdf test1a.png
```

Výstupní PNG je rastrovaný obrázek v rozlišení 300 dpi. Konverzi lze zrealizovat přímo pomocí GhostScriptu (<http://pages.cs.wisc.edu/~ghost/>), pod Microsoft Windows:

```
gswin32c -r300 -sOutputFile=test1b.png -dNOPAUSE
-dBATCH -sDEVICE=pngalpha test1out.pdf
```

2 Balíček ean13isbn

S příchodem ISBN-13 (1. ledna 2007) bylo potřeba drobných zásahů. V roce 2008 vznikl balíček ean13isbn od Zdeňka Wagnera. Dokumentaci vyvoláte v distribuci T_EX Live zapsáním texdoc ean13isbn. Na délku čárového kódu formátuje ISBN (horní část) a přichází s možností nastavit si přepínačem v podobě textového řetězce rozměr EAN13. Pro zájemce zmiňujeme, že Zdeněk Wagner vytvořil i balíček makebarcode pro prokládaný čárový kód 2/5 a Code 39 [6, str. 172].

Konkrétní ukázka necht' je uložena v souboru test2.tex:

```
\documentclass{article}
\usepackage[SC7]{ean13isbn}
\pagestyle{empty}
\begin{document}
\EANisbn[ISBN=978-80-87106-36-5]
\end{document}
```

Výstup opět můžeme ořezat a rastrovat, standardně v PDF nalezneme po spuštění pdflatex test2.tex následující:



Obrázek 2: Jen spodní část je rastrovaná.

3 Používáme X_YL^AT_EX na soubory PFB

Jak si mnozí jistě všimli, výstupní PDF obsahuje rastrované písmo (horní i spodní popisek v první ukázce a spodní v druhé ukázce). Je to tím, že písmo ocrb7 (celá starší písmová rodina ocr-b; autor Norbert Schwarz; <http://ctan.org/tex-archive/fonts/ocr-b/>) je z dob aktivního používání METAFONTu, kdy se rastrovalo závisle na výstupním zařízení.

Zdeněk Wagner v roce 2011 zvektoroval rodinu ocr-b z MF do podoby PFB a OTF souborů. My si s takovým písmovým formátem (PFB) dokážeme poradit i bez dalších konverzí (např. programem FontForge) pomocí X_YL^AT_EXu, který je stručně představen ve Zpravodaji [3], a je o něm zvaná přednáška od doc. Rybičky na letošním ročníku OSSConf.

Naše první ukázka test1.tex v novém hávu by vypadala jako test3.tex, ten spustíme pomocí xelatex test3.tex.

```

\documentclass[a4paper]{article}
\usepackage{fontspec}
\pagestyle{empty}
\begin{document}
\input ean13.tex
\font\ocrb=ocrb9.pfb
\font\ocrbsmall=ocrb7.pfb scaled 890
\def\ISBNb #1 {\def\ISBNnum{
  \makebox[10pt][r]{ISBN }#1}
  \barheight=45.151515\X\relax}
\ISBNb 978-80-87106-36-5
\EAN 9788087106365
\end{document}

```

Výstup je stejný jako v první ukázce, jen PDF je krásně celé vektorové.



Obrázek 3: Obě textové části jsou vektorové.

U naší druhé ukázky pomocí balíčku `ean13isbn` si stačí do adresáře s \TeX ovým souborem nahrát soubor `ocrb9.pfb` a spustit si `xelatex test2.tex`.

Zdeněk Wagner doporučuje cestu, rozbalit si `ocr-b-outline.zip` z CTAN.ORG do adresáře `texmf-local/fonts/type1/local/` a aktualizovat si \TeX u dostupné soubory pomocí příkazu `mktexlsr` (\TeX Live). Tím odpadá povinnost mít soubor vždy v adresáři s příslušným \TeX ovým souborem.

Dobrá zpráva je, že balíček `ocr-b-outline` se aktualizuje automaticky v distribuci \TeX Live. Pokud bychom tedy vymazali `pfb` v naší poslední ukázce, tak při `pdflatex test3.tex` se automaticky načtou PFB+TFM soubory a při `xelatex test3.tex` se načtou OTF soubory. Při případné absenci OTF souborů, si \XeTeX vyhledá PFB soubory.

Závěr tedy je, že pokud spustíme (doinstalujeme či aktualizujeme si balíček Zdeňka Wagnera `ocr-b-outline`):

```

tlmgr update --self
tlmgr update --all

```

a některou z doposud zmíněných ukázek, tak výstupy budou vždy vektorové. Tím by se aktualizovaly starší ukázky používající bitmapová písma rodiny `ocr-b`.

4 Projekt PStricks: balíček pst-barcode

Typografický vývoj nelze zastavit a ve světě čárových kódů nelze vystačit s kódy ISBN a EAN, viz ukázky na další straně.

Jistou typografickou evolucí je balíček pst-barcode autorů Terryho Burtona a Herberta Voše. Dokumentaci lze v \TeX Live vyvolat `texdoc pst-barcode`. Původní projekt Terryho Burtona lze nalézt na webové stránce <http://www.terryburton.co.uk/barcodewriter/generator/>, s možností si výstup stáhnout jako EPS, BMP, JPEG, PNG či TIFF soubor. Nástavbu pro KDE lze nalézt na stránkách <http://www.kbarcode.net/>.

4.1 Tvorba samostatného PDF

Zkusíme si prvně naši ISBN ukázkou dle rady z článku Herberta Voše [5, str. 66], tedy tak, že výstup bude tvořit samostatné a na hlavním dokumentu nezávislé PDF. O této problematice píše podrobněji článek [4].

Po projití technické dokumentace balíčku pst-barcode (`texdoc pst-barcode`, přímý odkaz) získáváme ukázkou v podobě souboru `test4.tex`:

```
\documentclass{article}
\pagestyle{empty}
\usepackage{pst-barcode}
\begin{document}
\begin{pspicture}(3,1in)
\psbarcode{978-80-87106-36-5}{includetext guardwhitespace}{isbn}
\end{pspicture}
\end{document}
```

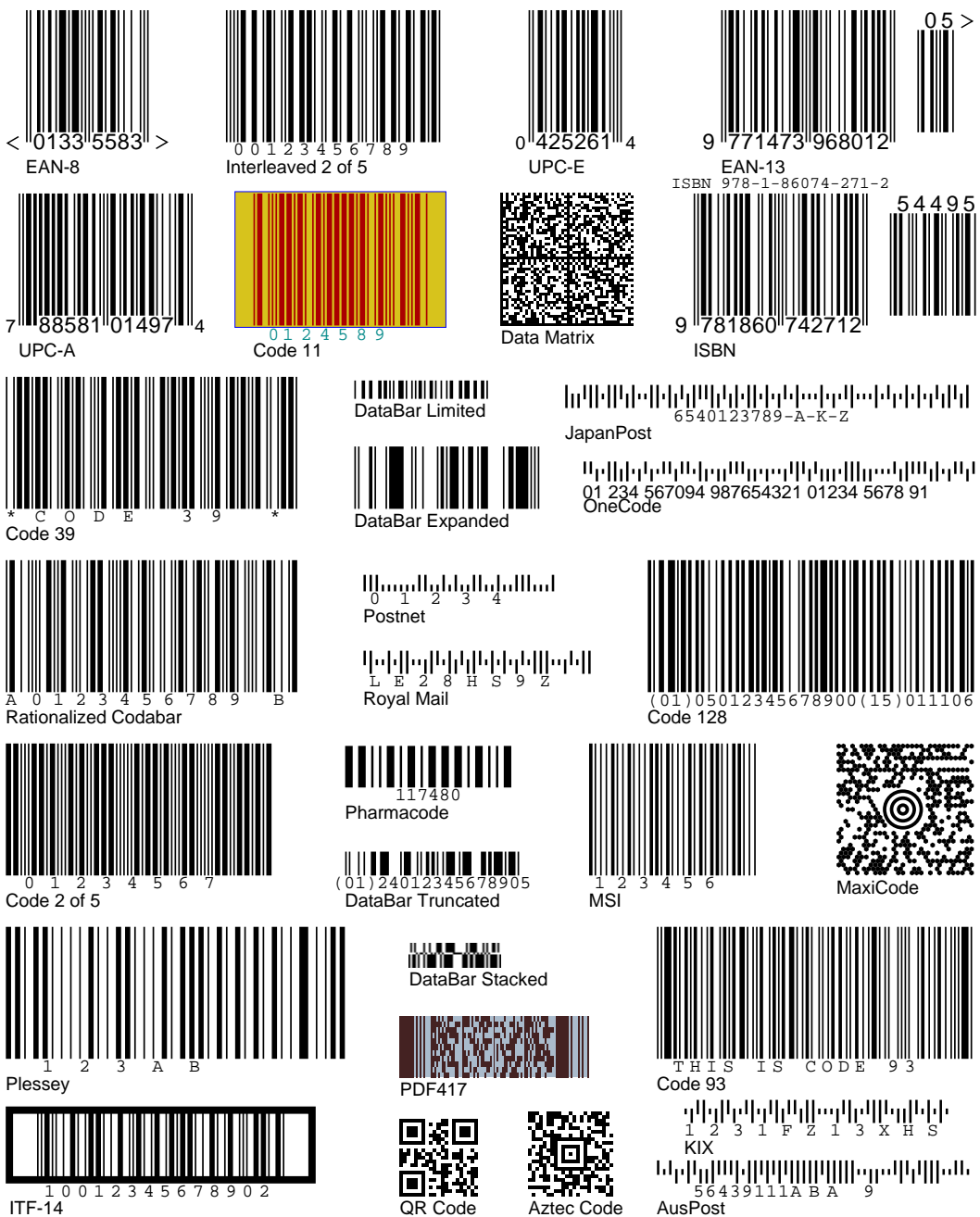
Spustíme hned několik kroků za sebou (pro usnadnění práce na to samozřejmě užijte pomocný MAKEFILE, SH či BAT soubor):

```
latex test4.tex
dvips test4.dvi
ps2pdf test4.ps test4in.pdf
pdfcrop --hires --margins 2 test4in.pdf test4out.pdf
```

Popisky jsou vektorové, centrované na délku čárového kódu a s ochranným symbolem (>) v části spodní.

4.2 Kód PStricks v \TeX souboru určeném pro zpracování pdf \LaTeX em

Nebyl by to Will Robertson – autor významného balíčku fontspec a správce balíčků Petera Wilsona – aby nám život ještě trochu neusnadnil. Přeskočíme možnost pomocí balíčku pst-pdf a nahlédneme rovnou na balíček auto-pst-pdf.



Obrázek 4: Ukázky čárových kódů dle původního projektu Terryho Burtona (přímý odkaz) či dle balíčku pst-barcode autorů Terryho Burtona a Herberta Voße při projektu PSTricks [5].



Obrázek 5: Výstup z balíčku pst-barcode.

Tento balíček nám zajistí kompletní automatizaci PostScriptových operací v pozadí vznikajícího PDF souboru. U další ukázky `test5.tex` si můžeme navíc všimnout nastavování parametrů. Více o možnosti volby parametrů v dokumentaci balíčku (`texdoc pst-barcode`), u webové verze je popis na <http://groups.google.com/group/postscriptbarcode/web/Options>. Volba parametrů je místy úplnější na webové stránce než v dokumentaci balíčku `pst-barcode`; je to dané vytížeností Herberta Voše, kdy jeho ToDo seznam obsahuje důležitější a naléhavější položky.

Pro zkušenější \TeX istky a \TeX isty poznámka, že první sada parametrů se odděluje čárkou, další pak mezerou. Přístup k barvám zajišťuje balíček `xcolor` (`texdoc xcolor`, přímý odkaz), který je balíčkem `pstricks` načten automaticky, pokud soubor `xcolor.sty` v \TeX ové kolekci, např. na vašem pevném či USB flash disku, existuje.

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[czech]{babel}
\usepackage[IL2]{fontenc}
\usepackage{auto-pst-pdf}
\usepackage{pst-barcode}
\pagestyle{empty}
\begin{document}
Text před čárovým kódem\ldots\par
\begin{pspicture}(2,1in)
\psbarcode[scalex=1.1,linecolor={[cmyk]{0,0.85,0.87,0.35}}]
{978-80-87106-36-5}
{guardwhitespace textsize=10 inkspread=0.5 includetext}
{isbn}
\end{pspicture}\par
Text za čárovým kódem\ldots
\end{document}
```

Potřebujeme spustit jen jeden řádek (první je pro instalaci \TeX Live, druhý je pro $\text{MiK}\TeX$) s možností návratu do systémového prostředí za běhu \TeX u:



Obrázek 6: Výstup z projektu PSTricks.

```
pdflatex -shell-escape test5.tex
#pdflatex -enable-write18 test5.tex
```

V dokumentaci tohoto balíčku Will Robertson zmiňuje, že vyvíjí další balíček nazvaný `pstool` (`texdoc pstool`, přímý odkaz), který by měl v budoucnosti balíček `auto-pst-pdf` (`texdoc auto-pst-pdf`, přímý odkaz) překonat.

5 Ukázka z Japonska

Z celé škály čárových kódů ze strany 204 si ještě ukážeme relativně nový typ dvojrozměrného čárového kódu QR Code, který je již dávno zaběhlým standardem v Japonsku, čtený např. mobilním telefonem za pochodu na ulici. Čárový kód odkazuje např. na webovou stránku nějaké firmy, objednávkový systém koncertu, divadelního kousku nebo tedy konferenci OSSConf2011. Jedna ukázka za všechny uložená pod `test6.tex`:

```
\documentclass{article}
\usepackage{auto-pst-pdf}
\usepackage{pst-barcode}
\pagestyle{empty}
\begin{document}
\begin{pspicture}(1,1in)
\psbarcode[linecolor=blue]{Viva QR Code!}
  {eclevel=H version=8}{qrcode}
\end{pspicture}
\end{document}
```

Po spuštění `pdflatex -shell-escape test6.tex` získáváme ukázkou na další straně.

Dekódování lze zrealizovat např. programem `zbarimg` (manuál). Lze též ohodnotit URL či soubor z vašeho počítače na <http://zxing.org/w/decode.jspx> (projekt ZXing alias zebra crossing). Však formuláři zkuste podsunout http://ossconf2011.soit.sk/static/images/qrcode_oss11.png. Sami uvidíte!



Obrázek 7: Ukázka čárového kódu QR Code.

6 Rozloučení se z Česka

Za pozornost stojí, že kvazidvourozměrný čárový kód PDF 417 nalezneme na České daňové správě pod Interaktivními formuláři¹. Jeden konkrétní formulář za všechny, viz Daň z příjmů fyzických osob ze závislé činnosti². V Adobe Readeru 9.4 přepočítává čárový kód (v pravém dolním rohu na každé straně) při vyplňování formulářových políček za běhu. Můžete vyzkoušet funkčnost ve svém oblíbeném prohlížeči PDF souborů.

Na úplný závěr ukázka statického čárového kódu PDF 417 s čárově kódovaným textem „Na shledanou!“ připravené v souboru `test7.tex`.

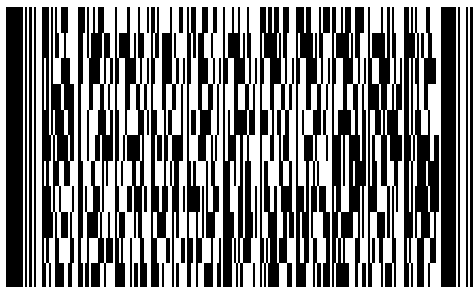
```
% pdflatex -shell-escape test7.tex
\documentclass{article}
\pagestyle{empty}
\usepackage[svgnames]{xcolor}
  % nebo lze užít x11names
  % dvipsnames není takto použitelný
\usepackage{pst-barcode}
\usepackage{auto-pst-pdf}
\begin{document}
\begin{pspicture}(1,1in)
\psbarcode[linecolor=Black]{Na shledanou!}
  {columns=9 rows=11 elevel=5 rowmult=12}{pdf417}
\end{pspicture}
\end{document}
```

Poděkování

Tento příspěvek vznikl za částečné podpory ESF projektu č. CZ.1.07/2.2.00/07.0361.

¹<http://cds.mfcr.cz/cps/rde/xchg/cds/xsl/10276.html?year=>

²http://cds.mfcr.cz/cps/rde/xbcr/cds/InterAktiv5241_4.pdf



Obrázek 8: Ukázka čárového kódu PDF 417.

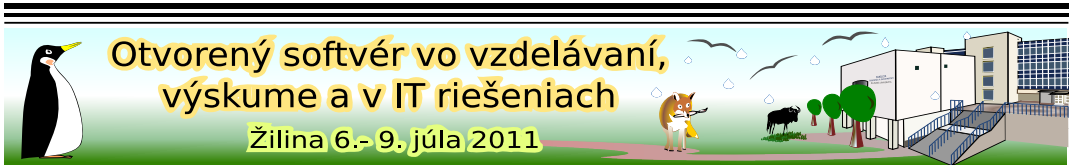
Literatura

- [1] OLŠÁK, P.: Čárové kódy EAN v $\text{T}_{\text{E}}\text{X}$ u. [The EAN barcodes by $\text{T}_{\text{E}}\text{X}$.] In *Zpravodaj Československého sdružení uživatelů $\text{T}_{\text{E}}\text{X}$ u*, Volume 6, Number 2, pp. 86–96, 1996. ISSN 1211-6661. doi: 10.5300/1996-2/86
- [2] OLŠÁK, P.: The EAN barcodes by $\text{T}_{\text{E}}\text{X}$. [Čárové kódy EAN v $\text{T}_{\text{E}}\text{X}$ u.] In *TUGboat*, Volume 15, Number 4, pp. 459–464, 1994. URL: <http://www.tug.org/TUGboat/Articles/tb15-4/tb45olsa.pdf>
- [3] STRÍŽ, P.: Úvod do $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u. [Introduction to $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.] In *Zpravodaj Československého sdružení uživatelů $\text{T}_{\text{E}}\text{X}$ u*, Volume 20, Number 4, pp. 292–296, 2010. ISSN 1211-6661. doi: 10.5300/2010-4/292
- [4] VOSS, H.: From PostScript to PDF with epstopdf, pdftricks, pst-pdf, auto-pst-pdf, pst2pdf, and more. In *TUGboat*, Volume 31, Number 2, pp. 145–147, 2010. ISSN 0896-3207. Na rok od vydání jen pro členy TUGu, poté bude článek veřejně dostupný. URL: <https://www.tug.org/members/TUGboat/tb31-2/tb98voss.pdf>
- [5] VOSS, H.: The current state of the PSTricks project. [Současný vývoj a novinky v balíčcích PSTricks.] In *Zpravodaj Československého sdružení uživatelů $\text{T}_{\text{E}}\text{X}$ u*, Volume 20, Number 1–2, pp. 39–67, 2010. ISSN 1211-6661. doi: 10.5300/2010-1-2/39
- [6] WAGNER, Z.: $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ v sazečské praxi. [$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ in the Typographer’s Profession.] In *Zpravodaj Československého sdružení uživatelů $\text{T}_{\text{E}}\text{X}$ u*, Volume 18, Number 3, pp. 161–174, 2008. ISSN 1211-6661. doi: 10.5300/2008-3/161

Kontaktní adresa

Pavel STRÍŽ (Ing., Ph.D.),

Tomas Bata University in Zlín, nám. T. G. Masaryka 5555,
760 01 Zlín, Czech Republic, striz@fame.utb.cz



DOI: 10.5300/2011-OSSConf/209

BALÍČEK PGFPLOTS: O TVORBĚ JEDNOHO GRAFU NA ZAKÁZKU

STRÍŽ, Pavel (CZ)

Abstrakt. Článek představuje vybrané partie balíčku `pgfplots` z \TeX ového světa. Zdrojové kódy jsou pod 1 KiB. Autor ukazuje dílčí situace, které vyústí v netradiční, programovatelný graf na konci našich snah. Bude se jednat o graf (metrovou grafovou housenku), která je autorem představena v samotném závěru tohoto konferenčního příspěvku. A jako taková bude mít premiéru v podobě dárku příchozím právě na OSSConf2011 v Žilině.

Mezi netradiční řešené partie můžeme zařadit: stupňovité měřítko vodorovné osy, několik hlavních a vedlejších svislých os v jednom grafu, mřížka umístěná do pozadí, umístění objektů absolutně či relativně vůči souřadnicovému systému, úprava dat za běhu \TeX u, změna vzhledu značek, nastavení stylu a sazba matematiky za účasti \TeX u, zvětšení grafu se zachováním os, příprava podkladů pro balíček `animate`, jako vhodné dělení grafu (s i bez překryvu) pro vložení dílčích částí pod sebe atp.

Klíčová slova. \TeX , \LaTeX , balíčky: `TikZ`, `pgfplots`, `pgfplotstable` a `animate`.

THE PGFPLOTS PACKAGE: ABOUT PREPARATION OF ONE HIGHLY CUSTOMIZED GRAPH

Abstract. The article presents several features from the `pgfplots` package from the \TeX world. The author introduces these selected parts using a minimal example (snippet, chunk). All presented codes have file size below 1 KiB. The source code is always briefly commented and followed by its preview. All our efforts shall help to create a special, highly programmable graph at the end. The one-meter caterpillar graph is introduced at the very end of this paper. Such as, the graph will have its first performance at the OSSConf2011 conference in Žilina, Slovakia.

Among the nontrivial tasks we may find: cascade ticks, independent tick labeling, grid to be in the background layer, absolute and relative coordinate system object placement, input data modification on-the-fly, markers adjustment, direct interaction with \TeX typography system, graph zooming with added axes, slicing graph for use with the `animate` package as well as cutting graph into several pieces included below each other etc.

Key words and phrases. \TeX , \LaTeX , `TikZ`, `pgfplots`, `pgfplotstable`, and `animate`.

Poděkování: Tento příspěvek vznikl za částečné podpory
ESF projektu č. CZ.1.07/2.2.00/07.0361.

1 O grafu samotném

Oslovila mě jedna studentka medicíny z cizí Alma Mater, že řeší SVOČ (Studentská vědecká a odborná činnost), a že se ji nedaří nastavit v grafu si vše dle jejích přání. Z obdržených vzorků lékařský dat jsem postupně skládal mozaiku až vzniklo dílo víceméně dle jejích představ. Dílo, které na soutěži nakonec nepoužila, ale sloužilo ji inspirací.

Graf, pracovně nazývaný „grafová housenka“, nám běží v článku na str. 218. Na první pohled vypadá komplikovaně a sofistikovaně, ale jak by měl tento článek čtenáři dokázat, je to vsutku jen mozaika, skládačka složená z dílčích vyřešených situací.

2 O pgfplots z $\text{T}_\text{E}\text{X}$ ového světa

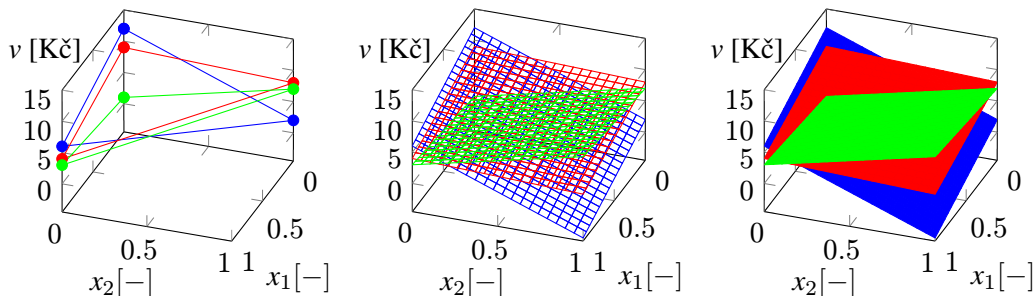
Na své první setkání s PGFPLOTS (dokumentace verze 1.4.1) si vzpomínám dobře. Tehdy jsem měl za sebou již několik experimentů v TikZ (autor Till Tantau; ukázky; s dilematem, jestli METAPOST stále ještě používat či ne, zvláště, když jsem se s vývojářem Tacem Hoekwaterem setkal osobně), a když mě bratr, nakladatelství Martin Stříž, požádal o jednoduchý sloupcový 2D graf, tušil jsem problém. V METAPOSTu je to dřina, TikZ se do toho také moc nehrnul, tak jsem hledal. Brzy jsem hledání završil úspěchem, s vědomím, že takto používaná oblast bude určitě řešena. Byl to objev hoden PGFPLOTS! Podporou mu je balíček PGFPLOTSTABLE.

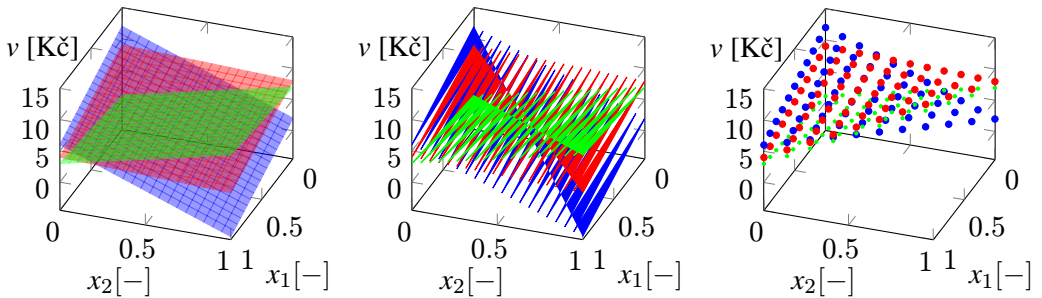
Neuplynulo ani pár týdnů a v teorii her jsem potřeboval nakreslit 3D grafy, grafické řešení her rozměru $3 \times n$ a $m \times 3$. Nu, a byl jsem doma! Grafy vznikaly jedna radost. A když jsem dokázal vykreslit graf výpočtem jednotlivých bodů (na úrovni $\text{T}_\text{E}\text{X}$ ového programování; na další straně, v pořadí šestý 3D graf), zahrnout si i podmínky, tak si mě PGFPLOTS získal.

Nemluvě o tom, že mně autor PGFPLOTS, Christian Feuersänger z Německa, přes noc udělal nadstavbu (nyní součástí experimentální verze; unstable version, str. 265 a 266) na vrstvy v PDF pro balíček OCG. Balíček PGFPLOTS je založen na TikZu, kdo o TikZ ještě neslyšel, jistě uslyší, neb $\text{T}_\text{E}\text{X}$ ový svět (část grafika) se pohnul tímto směrem.

0 Ukázky 3D grafů a zároveň geometrických klamů

Šest ukázek různých grafů ze stejných vstupních dat (zadány jsou tři roviny). Žádný z těchto grafů nelze v reálném světě fyzicky sestrojít. Věřte, autor to se studenty zkoušel! Nebo správně, skepticky autorovi nevěřte! . . . ☺ Na konferenci autor prozradí víc. . .





```

\begingroup \parindent=0pt
\pgfplotsset{width=0.33\textwidth, height=0.33\textwidth, compat=1.3}
\def\podminky{samples=20,domain=0:1,y domain=0:1,xlabel=$x_1[-]$,ylabel=$x_2[-]$,
zlabel=$v$ [Kč], view={110}{35}, zmin=-4.5, zmax=15, ztick={0,5,10,15},
every axis x label/.style={at={(rel axis cs:1,1.4,-0.2)}},
every axis y label/.style={at={(rel axis cs:1.4,0.5,-0.2)}},
every axis z label/.style={at={(rel axis cs:1,-0.1,1.3)}}, }
% První graf... Popisek v je jako výhra či p jako payoff,
\begin{tikzpicture} % další osy jsou pravděpodobnosti voleb hráčů.
\begin{axis}[\podminky]
\addplot3[blue, mark=*] coordinates {(0,0,12) (0,1,2) (1,0,6) (0,0,12)};
\addplot3[red, mark=*] coordinates {(0,0,9) (0,1,8) (1,0,4) (0,0,9)};
\addplot3[green,mark=*] coordinates {(0,0,1) (0,1,7) (1,0,3) (0,0,1)};
\end{axis}
\end{tikzpicture}
% Definování základu pro druhý až pátý graf.
\def\graf#1#2{%
\begin{tikzpicture}
\begin{axis}[\podminky, #1]
\addplot3[fill=blue, draw=blue, #2]{-6*x-10*y+12};
\addplot3[fill=red, draw=red, #2]{-5*x-1*y+9};
\addplot3[fill=green, draw=green, #2]{2*x+6*y+1};
\end{axis}
\end{tikzpicture}
}% Konec příkazu \graf.
% Vygenerování druhého až pátého grafu.
\graf{mesh}{}% 2. graf
\graf{shader=flat, surf}{}\par% 3. graf
\graf{shader=flat, surf}{opacity=0.4}% 4. graf
\graf{}{}% 5. graf
% Šestý graf...
\begin{tikzpicture}
\begin{axis}[\podminky, enlargelimits=0.001]
\foreach \myx in {0,0.1,...,1.05} {
\foreach \myy in {0,0.1,...,1.05} {
\pgfmathparse{\myx+\myy<1.1} % Sestavení podmínky.
\ifnum\pgfmathresult=1 % Test pravdivosti podmínky.
\pgfmathparse{-6*\myx-10*\myy+12} \let\mya=\pgfmathresult
\pgfmathparse{-5*\myx-1*\myy+9} \let\myb=\pgfmathresult
\pgfmathparse{2*\myx+6*\myy+1} \let\myc=\pgfmathresult

```

```

\typeout{x=\myx, y=\myy, a=\mya, b=\myb, c=\myc} % Výpis do log souboru.
\addplot3[blue, mark=*,mark options={scale=0.6}]coordinates{(\myx,\myy,\mya)};
\addplot3[red, mark=*,mark options={scale=0.6}]coordinates{(\myx,\myy,\myb)};
\addplot3[green,mark=*,mark options={scale=0.3}]coordinates{(\myx,\myy,\myc)};
\fi% Konec \ifnum podmínky.
}% Konec \myy.
}% Konec \myx.
\end{axis}
\end{tikzpicture}
\endgroup%
% Velikost nad 1 KiB: 2370 bajtů, tohle je však 6 snippetů. :-)

```

3 Skládačky mozaiky aneb dílky hrají svůj part

Postupně nahlédneme na jednotlivé partie grafu a uděláme si jejich minimální ukázky (anglicky a snippet nebo a chunk, slovensky zlomok kódu). V kódech nebudu dále uvádět řádky nutné zavést do preambule dokumentu:

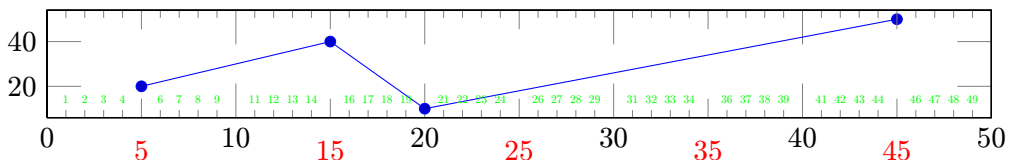
```

\usepackage{pgfplots,pgfplotstable,animate}
\pgfplotsset{width=1\textwidth,height=3cm}
% Velikost: 112 bajtů.

```

1 Stupňovité měřítko vodorovných os

Vyrobil jsem je tak, že jsem přes sebe překresloval axis o identickém xmin a xmax, dále se nulovaly popisky na svislé ose: `only marks, no markers`. Ačkoliv můžeme použít `major` a `minor tick` u každé axis, v ukázce vidíme třikrát typ hlavního měřítka (major).



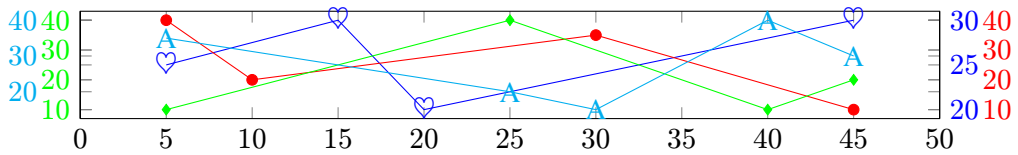
```

\def\nemenne{xmin=0, xmax=50}
\def\nemenneb{only marks, no markers}
\def\data{\addplot coordinates {(5,20) (15,40) (20,10) (45,50)};}
\begin{tikzpicture}
\begin{axis}[\nemenne, xtick={0,10,...,50}, major tick length=4.5mm] \data
\end{axis}
\begin{axis}[\nemenne, xtick={5,15,...,45}, \nemenneb, ytick=\empty,
tick label style={red, yshift=-1ex}, major tick length=2.5mm] \data
\end{axis}
% Jistá nezvyklá možnost, jak přeskočit čísla dělitelná pěti.
\foreach \osy/\osyb in {1/6,2/7,3/8,4/9} {
\begin{axis} [\nemenne, xtick={\osy,\osyb,...,50}, \nemenneb, ytick=\empty,
tick label style={green, scale=0.4, yshift=5ex}, major tick length=1mm] \data
\end{axis}
} % Konec \foreach cyklu.
\end{tikzpicture}% Velikost: 754 bajtů.

```


2 Násobné hlavní a vedlejší svislé osy

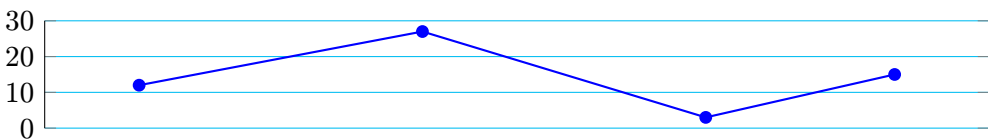
V grafu můžeme orientovat (left, right) popisky svislé osy (hlavní a vedlejší osa).



```
\def\nemenne{xmin=0, xmax=50}
\begin{tikzpicture}[ every axis/.append style={width=0.92\textwidth}]
\begin{axis}[\nemenne, yticklabel pos=right, yticklabel style={blue}]
\addplot [blue, mark=text, text mark=$\heartsuit$]
coordinates {(5,25) (15,30) (20,20) (45,30)};
\end{axis}
\begin{axis}[\nemenne, yticklabel pos=right, yticklabel style={xshift=2.5ex,
red}, axis x line=none]
\addplot [red, mark=*] coordinates {(5,40) (10,20) (30,35) (45,10)};
\end{axis}
\begin{axis}[\nemenne, yticklabel pos=left, axis x line=none, yticklabel style={green}]
\addplot [green, mark=diamond*] coordinates {(5,10) (25,40) (40,10) (45,20)};
\end{axis}
\begin{axis}[\nemenne, yticklabel pos=left, axis x line=none,
yticklabel style={xshift=-2.5ex, cyan}]
\addplot [cyan, mark=text, text mark=A] coordinates {(5,35) (25,20) (30,15) (40,40) (45,30)};
\end{axis}
\end{tikzpicture}% Velikost: 896 bajtů.
```

3 Nastavitelná mřížka

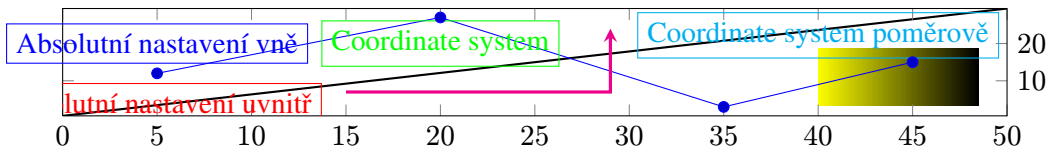
Z mřížek (bez zobrazení vstupních dat, nejčastěji znovunačtených z textového souboru) jsem si připravil speciální osy. Tudíž mohly být mřížky v pozadí jako nejspodnější vrstva. Na ukázce modrá vystupuje nad azurovou mřížkou a jsou při sazbě nezávisle nastavitelné.



```
\def\nemenne{xmin=0, xmax=50, axis x line=none, ymin=0, ymax=30}
\begin{tikzpicture}
\begin{axis}[\nemenne, grid=major, grid style=cyan!75, only marks, no markers]
\addplot coordinates {(5,12) (20,27) (35,3) (45,15)};
\end{axis}
\begin{axis}[\nemenne, axis y line=none]
\addplot [blue, thick, mark=*] coordinates {(5,12) (20,27) (35,3) (45,15)};
\end{axis}
\end{tikzpicture}% Velikost: 405 bajtů.
```

4 Objekty vkládané do grafu

V grafové housence na str. 218 si můžeme všimnout šipek, žlutých kruhů s číslem s i bez šipečky i obdélníkových přechodů. Po bližším prozkoumání zjistíme, že jsou ve spodní, středních i nejvrchnější vrstvě. U některých šipek lze nalézt mřížky nad i pod.



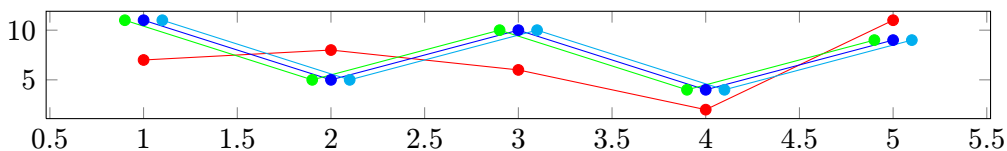
```
\begin{tikzpicture}
\begin{axis}[axis x line=none, axis y line=none, only marks, no markers]
\addplot coordinates {(0,0)}; % Stejný graf, linka bude v pozadí pod vším ostatním.
\draw[black,thick] (rel axis cs:0,0)--(rel axis cs:1,1); % Linka přes celý graf.
\shade[left color=yellow, right color=black] (rel axis cs:0.8,0.1)
rectangle (rel axis cs:0.97,0.63); % Přechod poměrově v prázdném grafu.
\end{axis}
\begin{axis}[xmin=0, xmax=50, yticklabel pos=right, every axis/.append style={
after end axis/.code={\node[blue,draw] at (50,170) {Absolutní nastavení vně};}}]
\addplot coordinates {(5,12) (20,27) (35,3) (45,15)};
\node[red, draw] at (50,10) {Absolutní nastavení uvnitř};
\node[green,draw] at (axis cs:20,20) {Coordinate system};
\node[cyan, draw] at (rel axis cs:0.8,0.75) {Coordinate system poměrově};
\draw[magenta,->,>=stealth,very thick] (axis cs:15,7)-|(axis cs:29,24);
\end{axis}
\end{tikzpicture}% Velikost: 976 bajtů.
```

5 Výpočty ze vstupních dat

Uvažme soubor data.txt (středníky jsou přechody na nový řádek, v článku záměrně takto použito z důvodu úspory místa; velikost souboru je 59 bajtů):

Time Mu1 Mu2; 1 7 11; 2 8 5; 3 6 10; 4 2 4; 5 11 9;

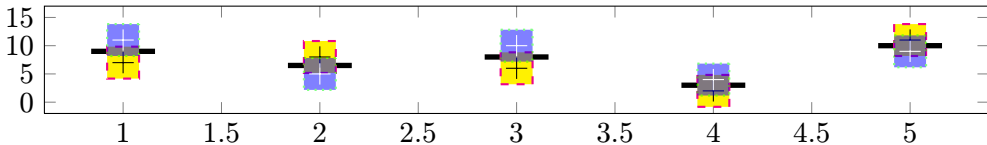
Překreslení mezi červenými a modrými daty nepostřehneme, ale pokud by data byla identická, jeden graf začne překreslovat druhý. Proto jemně posuneme zelenou (doleva) i azurovou křivku (doprava). Ačkoliv máme identická data, všechny naměřené údaje vidíme.



```
\pgfplotstableread{figures/data.txt}\loadedtable
\begin{tikzpicture} \begin{axis}
\addplot[color=red, mark=*] table [x=Time, y=Mu1] {\loadedtable};
\addplot[color=blue, mark=*] table [x=Time, y=Mu2] {\loadedtable};
\addplot[color=green,mark=*] table [x expr=\thisrow{Time}-0.1,y=Mu2]{\loadedtable};
\addplot[color=cyan, mark=*] table [x expr=\thisrow{Time}+0.1,y=Mu2]{\loadedtable};
\end{axis} \end{tikzpicture}% Velikost: 441 bajtů.
```

6 Data vykreslená jako uzly

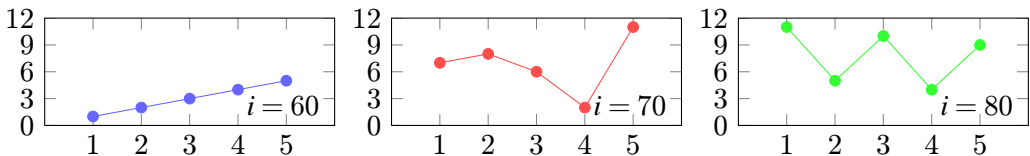
Nepřekvapí nás, že si lze z TikZ volat téměř cokoliv, například překreslení značek se děje přes parametr `mark options`, což je de facto volání `\node` v TikZ.



```
\pgfplotstableread{figures/data.txt}\loadedtable % Průměr jde do pozadí,
\begin{tikzpicture} \begin{axis} [ymin=-2, ymax=17] % poté následují značky...
\addplot[black, only marks, mark=-, mark options={scale=6, line width=2pt}]
table [x=Time, y expr=0.5*(\thisrow{Mu1}+\thisrow{Mu2})] {\loadedtable};
\addplot[magenta, only marks, mark=square*, mark options={scale=3, fill=yellow,
thick, dashed}] table [x=Time, y=Mu1] {\loadedtable};
\addplot[black, only marks, mark=+, mark options={scale=2, thin}]
table [x=Time, y=Mu1] {\loadedtable};
\addplot[green, only marks, mark=square*, mark options={scale=3, fill=blue,
opacity=0.5, thick, dotted}] table [x=Time, y=Mu2] {\loadedtable};
\addplot[white, only marks, mark=+, mark options={scale=2, thin}]
table [x=Time, y=Mu2] {\loadedtable};
\end{axis} \end{tikzpicture}% Velikost: 876 bajtů.
```

7 Přímá interakce s TeXem

Jak bylo doposud vidět, nepracujeme v ničem jiném (vyjma záměrného načtení dat) než v TeXovém souboru. To oceníme především u přípravy stylů, užití barev, volby písma, vkládání poznámek (znak procenta), zařazení matematiky a prvků automatizace.



```
\pgfplotstableread{figures/data.txt}\loadedtable
\def\nemenne{xmin=0,xmax=6,xtick=data,ytick={0,3,...,12},ymin=0,ymax=12}
\def\graf #1; #2, #3 {%
\begin{tikzpicture} [every axis/.append style={width=0.385\textwidth},
mujstyl/.style={#1!#2, mark=*}]
\begin{axis}[\nemenne]
\addplot[mujstyl] table [x=Time, y=#3] {\loadedtable};
\node [anchor=south east] at (axis cs:5.9,0) {$i=#2$};
\end{axis} % i je jako intenzita užitá barvy
\end{tikzpicture}
}% Konec definice příkazu \graf.
\graf blue; 60, Time
\graf red; 70, Mu1
\graf green; 80, Mu2
% Velikost: 587 bajtů.
```

4 Bonusy

Zmíníme dva plus jeden bonusy, které se na papír už nedaly zahrnout.

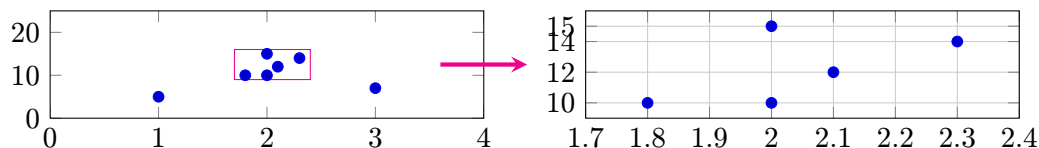
8 Zvětšování grafu

Balíček PGFPLOTS přinesl několik osvěžujících myšlenek (řada z nich ještě není ve verzi v \TeX distribucích, ale je testována pod experimentální verzí).

Zvětšit si výřez zkušenější \TeX ista už zvládne parametry `viewport` a `clip` pod příkazem `\includegraphics` z balíčku `graphicx`, viz též poslední ukázka tohoto článku, str. 218 a 219. Ovšem okamžitě přijdeme o informaci o osách a užitém měřítku. To je u grafů poměrně zásadní záležitost.

V dokumentaci PGFPLOTS na str. 230 a 231 najdeme toto zvětšování grafu, viz prostředí `groupplot`. Ukažme si jednodušší variantu této úvahy.

Ukázka se tváří nebezpečně, ale způsob je jednoduchý. Nakreslit graf znovu se stejnými daty, jen nastavit nové `xmin`, `xmax`, `ymin` a `ymax`. U 3D grafů navíc `zmin` a `zmax`.



```

\def\mabarva{magenta}%
\def\mujxmin{1.7}%
\def\mujxmax{2.4}%
\def\mujymin{9}%
\def\mujymax{16}%
\def\data{\addplot coordinates {(1,5)(2,15)(1.8,10)(2.1,12)(2.3,14)(2,10)(3,7)};}%
\begin{group}
\pgfplotsset{width=0.52\textwidth}%
\begin{tikzpicture}
\begin{axis} [only marks, xmin=0, xmax=4, ymin=0, ymax=25, xtick={0,1,...,4},
every axis/.append style={after end axis/.code={
\draw[magenta,ultra thick,->,>stealth] (rel axis cs:0.9,0.5)
--(rel axis cs:1.1,0.5); }} ]
\draw[mabarva] (axis cs:\mujxmin,\mujymin) rectangle (axis cs:\mujxmax,\mujymax);
\data
\end{axis}
\end{tikzpicture}
% Nový graf, na levém grafu nezávislý...
\begin{tikzpicture}
\begin{axis} [only marks, xmin={\mujxmin}, xmax={\mujxmax}, ymin={\mujymin},
ymax={\mujymax},xtick={1.7,1.8,...,2.5}, ytick=data,
grid=major, grid style=black!20]
\data
\end{axis}
\end{tikzpicture}
\end{group}
% Velikost: 925 bajtů.

```

9 Běžící grafová housenka

Rozporcováním dlouhého PDF či rastrového obrázku, např. PNG či JPG, lze získat běžící banner. Vytvoříme si ve speciálním souboru graf.tex dlouhý graf:

```
\documentclass{article}
\pagestyle{empty}
\usepackage{pgfplots}
\paperwidth=42in
\pdfpagewidth=\paperwidth
\textwidth=0.9\paperwidth
\pgfplotsset{width=1\textwidth,height=3cm}
\begin{document}
\begin{tikzpicture}
\begin{axis} [xmin=0, xmax=360, xtick={0,5,...,360}]
\addplot coordinates {(0,10) (180,30) (360,20)};
\end{axis}
\end{tikzpicture}
\end{document}% Velikost: 394 bajtů.
```

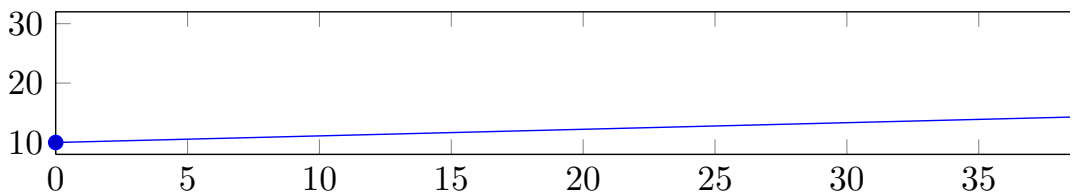
Ten si ořežeme s např. 1 bp (cca třetina milimetru) ochrannou zónou (velikost 70 bajtů):

```
pdflatex graf.tex
pdfcrop --hires --margins 1 graf.pdf graf-final.pdf
```

Například v programu PS_View zjistíme rozměry pravého horního rohu v bp jednotkách (dolní levý roh je obvykle nula-nula). Pokud použijeme `pdfinfo graf-final.pdf` zjistíme pt jednotky, tedy náš výsledek bude drobně jiný. Převést délkové jednotky lze operativně, např. užitím serveru `www.jednotky.cz`, z menu v levé části zvolíme Délka.

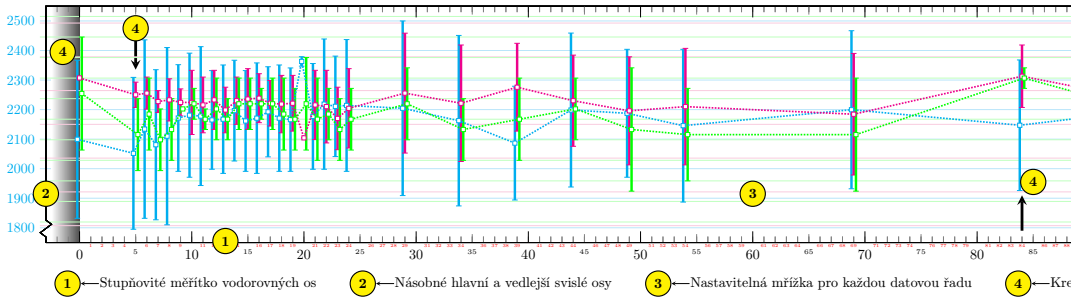
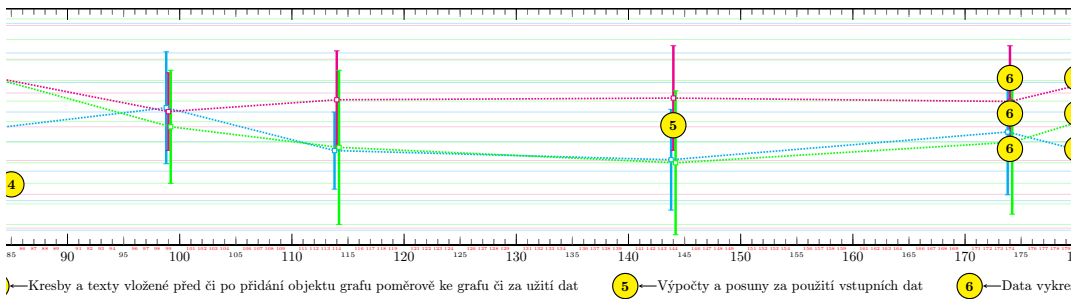
Po chvíli zkoumání balíčku ANIMATE (byla o něm řeč na ročníku OSSConf2010) si nastavíme počty snímků, levý dolní a pravý horní roh zobrazované oblasti, případný přesah a rychlost zobrazování snímků (počet snímků za vteřinu, očekává se kladná hodnota).

Nu a pak si animaci můžeme vložit do dokumentu, připravované prezentace, např. ve třídě BEAMER (o tom více příspěvek Aleše Kozubíka na letošním ročníku OSSConf), atp. Adobe Reader umí v PDF vložený banner zobrazit.

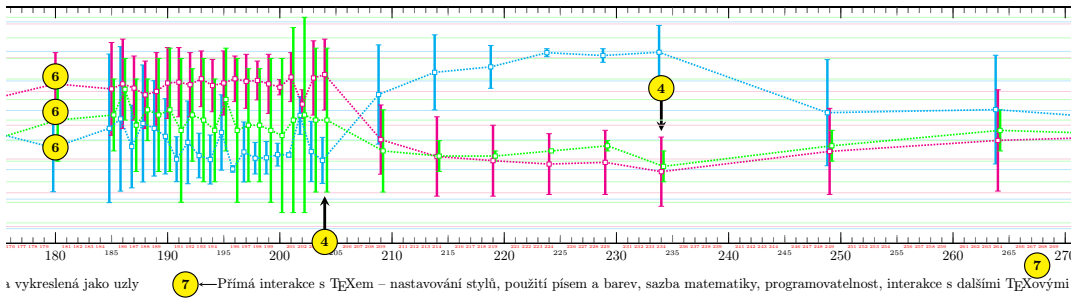


```
\noindent % Běž, grafe, běž!
\begin{animateinline}[autoplay,loop]{1} % Jeden snímek za vteřinu.
\multiframe{13}{ioldmuj=0+240, imuj=300+240}{%
\includegraphics[viewport={\ioldmuj} 0 {\imuj} 52.5, clip,
width=1\textwidth]{graf-final.pdf}%
}% Velikost souboru: 309 bajtů.
\end{animateinline}
```

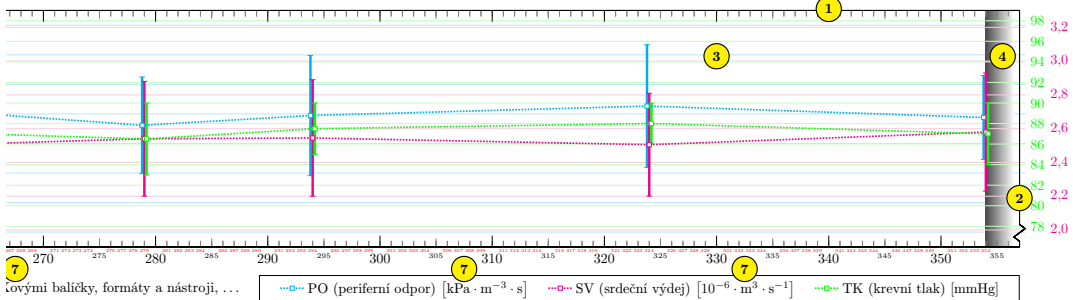
Třistapadesátčtyřminutové odměřené cykly... (maximum, průh

a, průměr, minimum)... měřený čas [min]... ukázka práce s \TeX ovými balíčky TikZ (v2.1), pgfplots (v1.4.1) a pgfplotstable (v1.4.1) z distribuce \TeX Live (v2010), tech

b), technická dokumentace balíčků se volá z příkazové řádky: texdoc mezera název_balíčku enter... reálná data byla použita z lékařského experimentu šesti dobrovolných pac



řech pacientů s laskavým svolením Adély Krajčové a jejího vedoucího...



X Stříhání grafu místo Závěru

Pokud v \TeX u použijeme cyklus, můžeme docílit podobného efektu, jako v předchozím dílu skládačky. Na závěr článku se pokusme i o to!

Máme připravený a ořezaný soubor `housenka-orez.pdf`, přípravou i vzhledem podobný souboru `graf-final.pdf` ze strany 217. Vložíme si následující kód, který nám obrázek připraví na n částí. Výsledek pro $n = 4$ s přesahem 30 bp (přibližně jeden centimetr původního rozměru před zmenšením) můžeme shlédnout na předchozí straně, str. 218.

```
\begin{figure}[p]
\label{housenka}%Na křížové odkazy
\newcount\casti \casti=4% Počet dílků
\newcount\citac \citac=0% Čítač dílků
\newdimen\delka \delka=2930bp% Délka obrázku
\newdimen\delkaold \delkaold=0bp% x-rozměr levý
\newdimen\delkanew \delkanew=30bp% x-rozměr pravý
\advance\delka by -\delkanew% Zrušení závěrečného přesahu
\divide\delka by \casti% Délka zobrazované části
\loop% y-rozměry jsou fixní
\advance\citac by 1%
\advance\delkanew by \delka%
%Kontrolní výpis:
%\noindent\the\citac. díl: \the\delkaold\ \the\delkanew\par
%1. díl: 0.0pt 757.83124pt
%2. díl: 727.71875pt 1485.54999pt
%3. díl: 1455.4375pt 2213.26874pt
%4. díl: 2183.15625pt 2940.98749pt
\noindent%Části jsou s přesahem hodnoty \delkanew
\includegraphics[viewport={\the\delkaold} 0 {\the\delkanew} 228,
clip, width=1\textwidth]{housenka-orez.pdf}\par\vspace{8mm}%
\advance\delkaold by \delka%
\ifnum\citac<\casti\repeat%
\end{figure}% Velikost: 999 bajtů.
```

5 Post Mortem

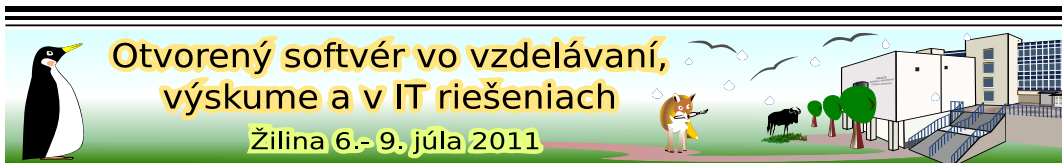
Každá část zdrojového kódu byla pod jeden KiB, včetně části X. Výborně, zadařilo se nám! Pro vášnivého čtenáře tip, aby si zkusil obdobnou housenku složit sám jako domácí úkol. Toho se nebojme, jde o to mít ze sazby radost a zaexperimentovat si přitom ve fázi učení se!

Kontaktní adresa

Pavel STRÍŽ (Ing., Ph.D.),

Tomas Bata University in Zlín, nám. T. G. Masaryka 5555,
760 01 Zlín, Czech Republic, `striz@fame.utb.cz`

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



DOI: 10.5300/2011-OSSConf/223

L^AT_EX NIE JE FARBA NA MAĽOVANIE, ALE NA PÍSANIE

BLAŠKO, Rudolf, (SK)

Abstrakt. Príspevok nadväzuje na článok „L^AT_EX nie je farba na maľovanie“, ktorý bol súčasťou tejto konferencie v minulom roku. Jeho cieľom je, aby každý čitateľ dokázal napísať svoj „vlastný dokument“ v L^AT_EXu.

Kľúčové slová. T_EX, L^AT_EX, programovanie.

L^AT_EX IS NOT THE COLOR FOR PAINTING BUT FOR WRITING

Abstract. This contribution is the continuation of the paper „L^AT_EX is not the color for painting“, which was published in the previous year conference proceedings. The aim for the readers is to make them able to create the own documents in L^AT_EX.

Key words and phrases. T_EX, L^AT_EX, programming.

1 Úvod

Na úvod stručne zopakujeme hlavné zásady práce v programe L^AT_EX (L^AT_EX je typografický systém, ale z dôvodu prehľadnosti ho budeme označovať program). L^AT_EX má dve zdanlivo veľké nevýhody, ktoré odrádzajú väčšinu potenciálnych užívateľov, ale pre normálneho užívateľa nie sú problém. Prvú z nich predstavujú minimálne, ale predsa nejaké znalosti tohto programu. Druhou nevýhodou je, že sa dokument tvorí analogickým spôsobom ako program v klasickom programovacom jazyku. Napíše sa zdrojový súbor *.tex (je to obyčajný textový súbor), ktorý sa preloží prekladačom do *.pdf alebo *.dvi súboru a ten sa následne vhodným prehliadačom zobrazí. V praxi na to postačia dve systémové okná (so zdrojovým textom a s výsledným grafickým súborom). Jedným príkazom alebo kliknutím myši sa zdrojový súbor automaticky preloží, aktualizuje sa výsledný súbor a hneď vidíme výsledok.

Táto zdanlivá nevýhoda je v skutočnosti výhoda. Nezáleží na tom, v akom operačnom prostredí pôsobíme, aké kódovanie používame, koľko a akých jazykov naraz používame, akú tlačiareň používame a „výsledok je vždy rovnaký“ (samozrejme až na možnosti zobrazovačov a tlačiarní). To znamená, že zdrojový súbor môžeme bez problémov modifikovať a beztrešne prekladať v ľubovoľnom inom operačnom systéme a „výsledok je vždy rovnaký“. Takýto stav absolútnej väčšine programov určených pre bežného užívateľa nehrozí. Jednoduchý text bez zložitých vzorcov si dokáže predstaviť každý. Čo si nedokáže predstaviť je jeho usporiadanie na stránke, ale to uvidí ihneď po preklade v druhom okne. S trochou praxe dokáže bez problémov každý priemerný počítačový užívateľ tvoriť aj zložitejšie vzorce.

L^AT_EX je určený predovšetkým na písanie vedeckých a technických dokumentov (z matematiky, fyziky, informatiky, chémie, biológie, ...). Do textu sa dajú jednoducho vkladať súbory v rôznych formátoch (*.eps, *.ps, *.jpg, *.png, ...). Dajú sa v ňom písať zložité notové zápisy (Mus_IX_TE_X), ale samozrejme aj jednoduché texty.

Nesmieme zabúdať na najväčšiu výhodu L^AT_EXu, ktorou sú nulové náklady na jeho zaobstaranie. To znamená, že to nie je komerčný produkt, ale produkt voľne šíriteľný, ktorý veľké množstvo nadšencov rozvíja a neustále zdokonaľuje.

Základnou činnosťou pri vytváraní publikácie v programe L^AT_EX je zápis zdrojových textov, t. j. súborov *.tex (majú pevnú štruktúru, zloženú z **preambuly** a **vlastného zdrojového textu** – pozri [1]). Pre naše účely budeme používať pracovný názov zdroj.tex. Súbor zdroj.tex môžeme písať v ľubovoľnom editore. Uložiť ho musíme ako obyčajný textový ASCII súbor bez formátovacích znakov editora, resp. operačného systému. Z dôvodu prehľadnosti a hlavne pre jednoduché dodatočné opravy je výhodne vlastný zdrojový text štrukturovať do súborov podľa ucelených celkov (napr. kapitoly, podkapitoly). Tiež je vhodné každý dokument písať do nového zdrojového súboru. Myslím, že v L^AT_EXu užívateľa ani nenapadne „zlepšovák“, ktorý sa často praktizuje: Písať všetky svoje dokumenty do jedného súboru a potom tlačíť iba potrebné strany.

Najmenšiu skupinu textu, ktorú L^AT_EX spracováva je odstavec. Pomocou dobre premyslených algoritmov upraví delenie slov, nastaví medzislovné medzery. Text ako celok spracováva na úrovni kapitoly, t. j. sa snaží sformátovať celý text tak, aby bola celá kapitola vyvážená (aby neboli medzi odstavcami veľké medzery, aby neboli strany prázdne, aby boli obrázky a tabuľky vhodne usporiadané, ...). Inými slovami to znamená, že sa staráme iba o obsah textu a nie o jeho vizuálnu stránku.

Odskok textu na nový riadok zabezpečí príkaz `\newline` alebo príkaz `\\` (používa sa hlavne v tabuľkách). Odskok textu na novú stranu zabezpečí príkaz `\newpage`. Zvyšok strany ostane prázdny a text pokračuje na novej strane ako nový odstavec.

2 Začínáme písať dokument

Na napísanie svojho prvého L^AT_EX–dokumentu už máme dosť informácií. Keďže sa všetko unifikuje a do popredia sa dostáva kódovanie Unicode¹ UTF-8, predpokladám, že v budúcnosti bude väčšina populácie používať toto kódovanie (nastavenia iných kódovaní pozri napr. [1], resp. [3, 5]). Súbor zdroj.tex pre jednostranný slovenský text, písaný v kódovaní Unicode UTF-8 na stránku formátu A4 s veľkosťou písma 12pt má tvar (ostatné nastavenia vrátane iných kódovaní pozri [1, 3, 5]):

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[slovak]{babel}
\include{mojeMakra.tex}
\begin{document}
```

¹Vo všetkých distribúciách Linuxu a aj vo Windows 7 je implicitne nastavené toto kódovanie.

```
\include{listy/list01}
\end{document}
```

Náš text budeme písať do súboru `list01.tex`, ktorý je uložený v podadresári `listy` (aktuálneho adresára, t. j. adresára so súborom `zdroj.tex`). Vlastné makrá, môžeme ukladať do súboru `mojeMakra.tex` v aktuálnom adresári. Tento súbor (aj prázdny) musí existovať, v opačnom prípade hlási L^AT_EX chybu.

3 Kapitoly a obsah

V L^AT_EXu sa obsah vytvára automaticky. Stačí v mieste textu, kde má obsah byť uložený, použiť príkaz `\tableofcontents`. V tomto prípade sa na nový riadok vysádza nadpis „Obsah“ (v anglickej mutácii „Contents“) a pod nim názvy jednotlivých častí s číslami strán.

Na dosiahnutie kompletného obsahu je nevyhnutné preložiť dokument `zdroj.tex` aspoň dvakrát. Pri prvom preklade sa vytvorí súbor `zdroj.toc`, do ktorého sa uložia potrebné informácie. Pri druhom preklade sa tieto informácie vložia do súboru `zdroj.dvi` (preklad pomocou `cslatex zdroj.tex` alebo `cslatex zdroj-koncovka.tex` nemusíme písať), resp. `zdroj.pdf` (preklad `pdfcslatex zdroj.tex`, resp. `pdfcslatex zdroj`).

Obsah zvykne byť v texte umiestnený na samostatnej strane, preto je rozumnejšie ho vložiť do súboru `obsah.tex` a na miesto výskytu napísať `\include{obsah}`. Súbor `obsah.tex` by mal okrem `\tableofcontents` obsahovať aj príkaz `\clearpage`, resp. `\cleardoublepage`. Príkaz² `\clearpage` zabezpečí prechod na novú stranu (zvyšok strany za posledným riadkom obsahu ostane prázdny). Pri obojstrannom dokumente (voľba `twoside` v príkaze `\documentclass`) sa použije analogický príkaz `\cleardoublepage`, ktorý zabezpečí prechod na nepárnu stranu.

Do obsahu sa automaticky pridávajú názvy jednotlivých častí (vrátane číslovania) definovaných príkazmi `\chapter` (trieda `book`), `\section`, `\subsection` `\subsubsection`. Tieto príkazy môžu mať nepovinný parameter [názov_do_obsahu], ktorý sa zobrazí v obsahu. Ak obsahujú hviezdičku *, potom sa nečíslujú a nezobrazujú v obsahu (pozri [1]). Na ešte jemnejšie členenie textu existujú príkazy `\paragraph{názov}`, `\subparagraph{názov}`, ale tie sa nezobrazujú v obsahu.

Príkazom `\addcontentsline{toc}{úroveň}{text_do_obsahu}` môžeme do obsahu pridať ľubovoľný text (`toc` je koncovka súboru `zdroj.toc`, `úroveň` je jeden z parametrov `chapter`, `section`, `subsection`, resp. `subsubsection`). Na druhej strane príkazom `\addtocontents{toc}{text_do_obsahu}` tiež vložíme text do obsahu (môže byť formátovaný), ale bez čísla stránky. Uvedieme príklad použitia týchto príkazov a ich interpretáciu v obsahu:

```
\section{Číslovaná sekcia}\label{sek01}
Text číslovanej sekcie \dots
```

²Ak použijeme tento príkaz v texte, potom sa okrem prechodu na novú stranu tiež vysádzajú všetky tzv. plávajúce objekty, t. j. obrázky a tabuľky.

```

\section[D-sekcia]{Druhá číslovaná sekcia}\label{sek02}
  Text druhej číslovanej sekcie \dots
\section*{Nečíslovaná sekcia}\label{sek03}
  Text nečíslovanej sekcie \dots
\subsection{Číslovaná subsekcia}\label{ssek01}
  Text číslovanej subsekcie \dots
\subsubsection[C-subsubsekcia]{Číslovaná subsubsekcia}\label{sssek01}
  Text číslovanej subsubsekcie \dots
\subsubsection*{Nečíslovaná subsubsekcia}\label{sssek02}
  \addcontentsline{toc}{subsubsection}{N-subsubsekcia}
  Text nečíslovanej subsubsekcie \dots
  \addtocontents{toc}{\dotfill toto je náš text \dotfill}
\paragraph{Náš paragraf}\label{p01}
  \addcontentsline{toc}{subsection}{Náš paragraf}
  Text paragrafu \dots
\subparagraph{Subparagraf}\label{sp01}
  Text subparagrafu \dots

```

Preložený výsledný text je zobrazený v tabuľke 1 a obsah celého článku na obrázku 1. V texte sú použité návestia `\label{návestie}`, na ktoré sa môžeme odkazovať (pozri [1]).

Každá z úrovní textu má svoje číslovanie, reprezentované vlastným počítadlom, ktoré môžeme podľa potreby rôzne nastavovať a meniť. Počítať môžeme ľubovoľný objekt v texte, nemusí to byť iba kapitola alebo sekcia (pozri [5]). \LaTeX má preddefinovaných niekoľko počítadiel, ktoré väčšinou korešpondujú s názvom počítaného príkazu alebo prostredia: `part` (príkaz `\part` má zvláštny význam a neovplyvňuje číslovanie ostatných úrovní), `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`, `subparagraph` (evidujú čísla kapitol, sekcií, ...), `page` (číslo strany), `equation`, `figure`, `table` (čísla rovníc, obrázkov a tabuliek), `footnote` (číslo poznámok pod čiarou), `mpfootnote` (číslo poznámok pod čiarou v prostredí `minipage`), `enumi`, `enumii`, `enumiii`, `enumiv` (čísla rôznych úrovní v prostredí `enumerate`). Po každom použití sa hodnota daného počítadla zväčší o hodnotu 1, pri príkazoch s hviezdičkou `*` sa hodnota počítadla nemení.

Pre výpis aktuálnej hodnoty počítadla s názvom `meno_počítadla` sa používa príkaz `\themenom_počítadla`. Príkazom `\addtocounter{meno_počítadla}{n}` sa zväčší jeho hodnota o celé číslo n (resp. zmenší pre záporné n). Na hodnotu n nastavíme počítadlo `meno_počítadla` príkazom `\setcounter{meno_počítadla}{n}`.

Počítať objekty môžeme nielen arabskými číslami. Používajú sa na to nasledujúce príkazy `\arabic{meno_počítadla}` (arabské čísla – hodnoty môžu byť celé čísla), `\roman{meno_počítadla}`, `\Roman{meno_počítadla}` (malé a veľké rímske čísla – iba kladné celé čísla), `\alph{meno_počítadla}`, `\Alph{meno_počítadla}` (malé a veľké latinské písmená – iba prirodzené čísla, maximálna hodnota počítadla je 26).

Hodnotu počítadla môžeme nastaviť pre celý dokument v preambule alebo podľa potreby meniť na ľubovoľnom mieste dokumentu. Postupnosť príkazov

```

\setcounter{section}{-3}\thesection
\addtocounter{section}{36}\thesection
  \renewcommand{\thesection}{\roman{section}}\thesection
\addtocounter{section}{-14}
  \renewcommand{\thesection}{\Roman{section}}\thesection
  \renewcommand{\thesection}{\alph{section}}\thesection
  \renewcommand{\thesection}{\Alph{section}}\thesection

```

postupne nastaví hodnotu počítadla `section` na hodnoty: -3, 33, xxxiii, XIX, s, S.

Aby sme uzavreli túto časť nastavíme správne číslovanie sekcie na hodnotu 3 príkazmi `\setcounter{section}{3}` a `\renewcommand{\thesection}{\arabic{section}}`.

2 Číslovaná sekcia

Text číslovanej sekcie ...

3 Druhá číslovaná sekcia

Text druhej číslovanej sekcie ...

Nečíslovaná sekcia

Text nečíslovanej sekcie ...

3.1 Číslovaná subsekcia

Text číslovanej subsekcie ...

3.1.1 Číslovaná subsubsekcia

Text číslovanej subsubsekcie ...

Nečíslovaná subsubsekcia

Text nečíslovanej subsubsekcie ...

Paragraf Text paragrafu ...

Subparagraf Text subparagrafu ...

Tabuľka 1: Ukážka štrukturovaného textu

4 Úprava odstavca a výpočtové prostredia

L^AT_EX implicitne zarovnáva text do odstavcov a delí slová (pozri [1, 3, 5]). Text môžeme na ľubovoľnom mieste rozdeliť na viacej stĺpcov a potom sa vrátiť k jednotlípovej sadzbe. Služi na to balíček `multicol`,

t. j. do preambuly musíme napísať príkaz `\usepackage{multicol}`.

Zápis textu do n stĺpcov dosiahneme pomocou prostredia `multicols` príkazmi (v našom prípade sú to 2 stĺpce, t. j. $n=2$):

Obsah		
1	Úvod	1
2	Začíname písať dokument	2
3	Kapitoly a obsah	3
4	Číslovaná sekcia	5
5	D-sekcia	5
	5.1 Číslovaná subsekcia	5
	5.1.1 C-subsubsekcia	5
	N-subsubsekcia	5
 toto je náš text	
	Náš paragraf	5
4	Úprava odstavca a výpočtové prostredia	5
5	Tabuľky	9
6	Záver	12

Obrázok 1: Ukážka obsahu dokumentu

```
\begin{multicols}{n}
```

Text ľubovoľného množstva textu sádzaný do n odstavcov \dots

```
\end{multicols}
```

Text (s viacerými odstavcami) môžeme bez problémov zarovnať na ľavý, resp. pravý okraj alebo centrovat' na vertikálnu os (v tomto prípade sa nedelia slová). Takéto formátovanie odstavcov sa nazýva sadzba na zástavku (po česky sazba na praporek). Sadsbu na zástavku môžeme bez problémov kombinovať s viacstĺpcovou sadzbou.

Sadsbu na ľavý okraj zabezpečíme pomocou príkazu `\raggedright` alebo prostredia

`flushleft`. Príkaz:

```
\raggedright{ľubovoľný text \dots}
```

je ekvivalentný príkazom:

```
\begin{flushleft}
```

 ľubovoľný text \dots

```
\end{flushleft}
```

Sadsbu na pravý okraj zabezpečíme pomocou príkazu `\raggedleft` alebo

prostredia `flushright`. Príkaz:

```
\raggedleft{ľubovoľný text \dots}
```

je ekvivalentný príkazom:

```
\begin{flushright}
```

 ľubovoľný text \dots

```
\end{flushright}
```

Centrovanú sadzbu zabezpečíme pomocou príkazu `\centering` alebo prostredia

`center`. Príkaz:

```
\centering{ľubovoľný text \dots}
```

je ekvivalentný príkazom:

```
\begin{center}
```

 ľubovoľný text \dots

```
\end{center}
```

Poznámky pod čiarou môžeme chápať ako špeciálny prípad odstavcov. Sú sádzané menším stupňom písma v dolnej časti danej strany „pod čiarou“ a zadávajú sa príka-

zom `\footnote[vlastné_číslo]{text_pod_čiarou}`, ktorý sa napíše za slovo, ktoré chceme označiť. Ak nie je zvolený nepovinný parameter `[vlastné_číslo]`, potom sa použije aktuálna hodnota počítadla `footnote`. V niektorých prostrediach (napr. `tabular`) príkaz `\footnote` nefunguje. Príkaz `\footnotemark[vlastné_číslo]` vytvorí v mieste uvedenia iba odkaz (nie poznámku pod čiarou) podľa rovnakých pravidiel ako `\footnote`. Príkaz `\footnotetext[vlastné_číslo]{text_pod_čiarou}` vytvorí poznámku pod čiarou s číslom identickým s posledným použitím príkazov `\footnote`, resp. `\footnotemark`, prípadne s hodnotou `[vlastné_číslo]`. Ak chceme dané miesto odkázať na už použitú poznámku pod čiarou (ľubovoľne v texte), môžeme použiť dvojicu príkazov `\label`, `\ref` (pozri [1, 3, 5]). Uvedieme príklad použitia poznámok pod čiarou:

```
Náš text\footnote{Prvá poznámka
  pod čiarou.\label{p01}}
pokračuje A\footnotemark
\footnotetext{%
  Druhá poznámka pod čiarou.}
\footnotetext{%
  Tretia poznámka pod čiarou
  s rovnakým číslom ako druhá.}
a pokračuje B\footnote[77]{%
  Štvrtá poznámka pod čiarou
  s číslom 77.}
a pokračuje C\footnote[\ref{p01}]{%
  Piata poznámka pod čiarou
  s rovnakým číslom ako prvá.}
a pokračuje D\footnotetext{%
  Šiesta poznámka pod čiarou
  s rovnakým číslom ako druhá,
  ale bez značky.}
a pokračuje E\footnotemark{
  bez poznámky pod čiarou \dots
```

Vysádaná strana s poznámkami pod čiarou bude mať tvar (pre prehľadnosť je počítadlo `footnote` nastavené na začiatok):

Náš text¹ pokračuje A² a pokračuje B⁷⁷ a pokračuje C¹ a pokračuje D a pokračuje E³ bez poznámky pod čiarou . . .

¹Prvá poznámka pod čiarou.
²Druhá poznámka pod čiarou.
²Tretia poznámka pod čiarou s rovnakým číslom ako druhá.
⁷⁷Štvrtá poznámka pod čiarou s číslom 77.
¹Piata poznámka pod čiarou s rovnakým číslom ako prvá.
²Šiesta poznámka pod čiarou s rovnakým číslom ako druhá, ale bez značky.

Na zložitejšie členenie odstavcov formou číslovaných alebo nečíslovaných zarážok môžeme použiť niekoľko prostredí. Môžeme ich do seba ľubovoľne vnárať do štvrtej úrovne. Pre označenie zarážok sa vo všetkých z nich používa príkaz `\item[vlastná_značka]`.

Odsadenie zarážok od ľavého okraja definujú registre `\leftmargini`, `\leftmarginii`, `\leftmarginiii`, `\leftmarginiv`. Príkazy `\renewcommand{\leftmargini}{.2cm}`, resp. `\leftmargini=.2cm`, resp. `\leftmargini 0.2cm` nastaví odsadenie prvej úrovne na 0,2 cm. Podobne `\leftmarginii 2.5\parindent` nastaví odsadenie druhej úrovne na 2,5-násobok `\parindent`. Zvislú medzeru medzi zarážkami meníme príkazom `\itemsep`, napr. `\itemsep 0pt` nastaví túto medzeru iba na medziriadkovú medzeru.

Prostredie `itemize` označuje zarážky postupne na jednotlivých úrovňach symbolmi

- (bullet), – (stredná pomlčka), * (hviezdička) a . (centrovaná bodka).

Prostredie `enumerate` čísluje zarážky postupne na jednotlivých úrovňach arabskými číslami s bod-

kou (`1.` počítadlo `enumi`), malými latinskými písmenami v zátvorkách (`(a)` počítadlo `enumii`), malými rímskymi čísly s bodkou (`i.` počítadlo `enumiii`) a veľkými latin-
skými písmenami s bodkou (`A.` počítadlo `enumiv`). Prostredie `description` predstavuje
popisný výpis hesiel, ktoré sú definujú ako nepovinný parameter [*vlastná_značka*] prí-
kazu `\item`. Tieto heslá sú automaticky na všetkých úrovniach sádzané tučným písmom.

Na označenie zarážok môžeme použiť ľubovoľné symboly. Väčšina z nich je považovaná
za matematické symboly a preto sa musia písať v matematickom móde, t. j. medzi dolármi
`$` a `$`. Je ich veľmi veľké množstvo a čitateľ ich nájde napríklad v [2–5].

Náš text rozdelíme na štyri úrovne,
aby sme videli odsadenia:

```
\begin{itemize}\itemsep 0.5em
\item Prvá úroveň \dots
\item[{$\heartsuit$}] Prvá úroveň \dots

\begin{itemize}\itemsep 0.25em
\item Druhá úroveň \dots
\item[{$\clubsuit$}] Druhá úroveň \dots

\begin{itemize}\itemsep 0em
\item Tretia úroveň a na
ukážku je tu dlhší text \dots
\item[{$\star$}] Tretia úroveň \dots

\begin{itemize}\itemsep 0em
\item Štvrtá úroveň \dots
\item[{$\pi$}] Štvrtá úroveň \dots
\end{itemize}
\end{itemize}
\item[{$\clubsuit$}] Druhá úroveň \dots
\item[{$\clubsuit$}] Druhá úroveň \dots
\end{itemize}
\item[{$\heartsuit$}] Prvá úroveň \dots
\end{itemize}
A text pokračuje \dots
```

```
\begin{description}
\item[Heslo 1] Text pre~heslo~1 \dots
\begin{enumerate}
\item Prvý podtext \dots
\item Druhý podtext \dots
\item[7a] Tretí podtext \dots
\end{enumerate}
\item[Heslo 2] Text pre~heslo~2 \dots
\renewcommand{\theenumi}{\alph{enumi}}
\begin{enumerate}
\item Prvý podtext \dots
\item[{$\ast$}] Druhý podtext \dots
\end{enumerate}
\item Text bez~hesla \dots
\end{description}
```

Predchádzajúci text ilustruje použitie
prostredia `itemize` do štvrtej úrovne. Vy-
sádzaný text vyzerá nasledovne:

Náš text rozdelíme na štyri úrovne, aby
sme videli odsadenia:

- Prvá úroveň ...
 - ♡ Prvá úroveň ...
 - Druhá úroveň ...
 - ♣ Druhá úroveň ...
 - * Tretia úroveň a na ukážku
je tu dlhší text ...
 - * Tretia úroveň ...
 - Štvrtá úroveň ...
 - π Štvrtá úroveň ...
 - ♣ Druhá úroveň ...
 - ♣ Druhá úroveň ...
 - ♡ Prvá úroveň ...
- A text pokračuje ...

Po preložení dostaneme výsledný text
v nasledujúcom tvare:

- Heslo 1** Text pre heslo 1 ...
1. Prvý podtext ...
 2. Druhý podtext ...
 - 7a Tretí podtext ...
- Heslo 2** Text pre heslo 2 ...
- a. Prvý podtext ...
 - * Druhý podtext ...
- Text bez hesla ...

5 Tabuľky

Tabuľky sa na prvý pohľad robia zložito, ale nie je to pravda. Ak si užívateľ píše zdrojový text prehľadne (t. j. aby sa v ňom aj neskôr vyznal) a predvída potenciálne zmeny, potom sa tabuľky tvoria elegantne a pár príkazmi dokážeme meniť ich obsah a vzhľad.³ Pre sadzbu tabuliek sa používa prostredie `tabbing` (text môže presahovať na novú stranu), resp. `tabular` (text sa musí zmestiť na jednu stranu) a jeho matematická analógia `array`.

Prostredie `tabbing` umožňuje rozmiestniť text v riadku systémom podobným tabulátorom na písacích strojoch (pozri múzeum), t. j. písať text do rovnakých pozícií v riadkoch. Tabulátory vo vnútri prostredia `tabbing` sa definujú príkazom `\=` a posun jednotlivých častí textu na pozíciu tabulátora zabezpečuje príkaz `\>`. Prechod na nový riadok sa zabezpečí príkazom `\\[výška]` (nepovinný parameter [výška] udáva veľkosť medzery medzi riadkami). Ak riadok končí príkazom `\kill`, potom sa riadok nezobrazí, nastavenie tabulátorov sa zachová a sadzba sa nastaví od začiatku riadku. Pri nastavení tabulátorov si musíme dať pozor, aby sa text zmestil do vymedzeného priestoru na šírku. Nasledujúci príklad ilustruje správne aj nesprávne použitie tabulátorov. Výsledok je zobrazený v tabuľke 3.

```
\begin{tabbing}
\= {\bf Tovar} \= {\bf Cena} \= {\bf Množstvo} \= {\bf Celkom} \\
\> Kolobežka \> 22,00 \euro \> 11 ks \> 242,00 \euro \\[-.2em]
\> Lopta \> 12,21 \euro \> 2 ks \> 24,42 \euro \\[.5em]

\= {\bf XXXXXXXXX} \= {\bf XXXXXXX} \= {\bf XXXXXXXX} \= \kill
\> {\bf Tovar} \> {\bf Cena} \> {\bf Množstvo} \> {\bf Celkom} \\
\> Kolobežka \> 22,00 \euro \> 11 ks \> 242,00 \euro \\[-.2em]
\> Lopta \> 12,21 \euro \> 2 ks \> 24,42 \euro
\end{tabbing}
```

Jednotlivé prostredia `tabbing` sa nemôžu do seba vnárať. Toto prostredie poskytuje oveľa viac možností, ale ich vysvetlenie presahuje rozsah tohto článku (pozri [3–5]).

Tovar	Cena	Množstvo	Celkom
Kolobežka	22,00 €	11 ks	242,00 €
Lopta	12,21 €	2 ks	24,42 €

Tabuľka 2: Prostredie `tabbing`

Na tvorbu tabuliek je vhodnejšie prostredie `tabular`. Toto prostredie má dva varianty:

³Na rozdiel od nemenovaných klikacích programov, v ktorých zmeniť veľkosť alebo štýl písma síce nie je veľký, ale veľmi pracný problém.

<code>\begin{tabular}[p]{formát_stĺpcov}</code>	<code>\begin{tabular*}[p]{šírka}{formát_stĺpcov}</code>
Vlastný text tabuľky \dots	Vlastný text tabuľky \dots
<code>\end{tabular}</code>	<code>\end{tabular*}</code>

V modifikácii `tabular` sa šírka tabuľky určí podľa jednotlivých stĺpcov a v modifikácii `tabular*` sa šírka tabuľky definuje povinným parametrom⁴ `{šírka}` (napr. `{12em}`, `{5cm}`, resp. `{.8\linewidth}`), t. j. 80 % aktuálnej šírky riadku na strane). Nepovinný parameter `[p]` určuje spôsob pripojenia tabuľky k okolitému textu, môže nadobúdať hodnoty `t` (top – tabuľka sa k textu pripojí horným okrajom), `b` (bottom – tabuľka sa k textu pripojí dolným okrajom) a `c` (center – tabuľka sa k textu pripojí stredom – implicitné nastavenie).

V povinnom parametri `{formát_stĺpcov}` sa definuje spôsob zarovnávanía pre každý stĺpec tabuľky. Používajú sa na to symboly `l` (stĺpce zarovná k ľavému kraju), `r` (stĺpce zarovná k pravému kraju), `c` (stĺpce vycentruje), `p{šírka_stĺpca}` (text vypíše do bloku širokého `{šírka_stĺpca}`, širší text rozdelí na viac riadkov, pričom nedelí slová a zarovnáva na ľavú zástavku). Symbol `|` definuje zvislú čiaru medzi stĺpcami a `@{tvar_medzery}` definuje tvar medzery susediacich stĺpcov s týmto parametrom.

Ak použijeme `@{}`, potom medzi týmito stĺpcami nebude žiadna medzera, ale napríklad po použití `@{\quad=X=\quad}` sa v tabuľke v každom riadku medzi príslušnými stĺpcami objaví text `=X=` ohraničený širokými medzerami. Ak v tabuľke nasleduje za sebou niekoľko rovnako definovaných stĺpcov, potom môžeme použiť `*{počet}{formát_stĺpcov}`, kde číslo `{počet}` udáva koľkokrát sa má definícia `{formát_stĺpcov}` zopakovať (napr. `\begin{tabular}{|r||l|c|l|c|}` a `\begin{tabular}{|r||*2}{l|c|}` sú ekvivalentné príkazy). Príkaz `\setlength{\doublerulesep}{medzera}` zmení šírku medzery medzi dvojitými vertikálnymi čiarami (`||`) na hodnotu `{medzera}`. Po nastavení `\setlength{\doublerulesep}{0pt}` sa táto medzera zruší a z dvojitej čiary sa stane tučnejšia čiara. Preddefinovaná vzdialenosť medzi vnútróm stĺpca a jeho okrajom je 6 pt a zmeniť ju môžeme predefinovaním registra `\tabcolsep` (napr. `\tabcolsep=3pt`).

Údaje sa zapisujú po riadkoch, pričom jednotlivé stĺpce sa oddeľujú symbolom `&` a prechod na nový riadok sa zabezpečí príkazom `\\[výška]` (pozri str. 231). Horizontálna čiara cez celú šírku tabuľky medzi riadkami tabuľky sa definuje príkazom `\hline`. Príkaz `\cline{x-y}` definuje vodorovnú čiaru, ktorá začína pod stĺpcom s poradovým číslom `x` (počítané zľava) a končí pod stĺpcom s poradovým číslom `y`. Medzery medzi riadkami v tabuľke meníme predefinovaním príkazu `\arraystretch` (napr. `\arraystretch=1.5` ich zväčší na 1,5-násobok pôvodnej hodnoty).

⁴Celková šírka stĺpcov by mala korešpondovať so šírkou tabuľky.

Príkazom `\multicolumn{počet_stĺpcov}{formát_stĺpca}{text}` umiestnime text do viacerých stĺpcov (t. j. ich previažeme na šírku). Parameter `{počet_stĺpcov}` udáva koľko stĺpcov máme previazať, v parametri `{formát_stĺpca}` sa definuje formát tohto stĺpca a v parametri `{text}` sa nachádza vlastný text.

Balíček `array` ďalej rozširuje možnosti tvorby tabuliek na vertikálne prepojenia textu, nové stĺpce... Definuje sa v preambule príkazom `\usepackage{array}` (pozri [2–5]). Existujú aj balíčky (napr. `supertab`), ktoré riešia sadzbu „veľkých“ tabuliek presahujúcich niekoľko strán (prostredia `supertabular`, `longtable` – pozri [2, 3, 5]).

Keďže sa nám mnohokrát nepodarí naformátovať text tak, aby sa tabuľky nelámali na konci strany, ukladáme ich (analogicky aj obrázky) ako tzv. **plávajúce objekty**. To znamená, že ich L^AT_EX uloží na prvé vhodné miesto. Toto zabezpečuje plávajúce prostredie `table` (pre obrázky prostredie `figure`). L^AT_EX zabezpečí ich automatické číslovanie a vhodné umiestnenie. Syntax pre vytvorenie plávajúcej tabuľky má tvar:

```
\begin{table}[p]\label{návestie}      resp.  \begin{table}[p]\label{návestie}
    vlastná tabuľka                    \caption{popis_tabuľky}
\caption{popis_tabuľky}              vlastná tabuľka
\end{table}                           \end{table}
```

Voliteľný parameter `[p]` určuje miesto, kde sa má prednostne plávajúca tabuľka uložiť (pokiaľ je to možné). Tento parameter môže nadobúdať hodnoty (aj všetky naraz) `h` (here – prioritne na toto miesto), `t` (top – prioritne na hornú časť strany), `b` (bottom – prioritne na dolnú časť strany) a `p` (page – na samostatnú stranu, kde sú iba plávajúce objekty).

Príkaz `\caption[názov_do_obsahu]{popis_tabuľky}` definuje popis tabuľky pod alebo nad tabuľku (podľa jeho umiestnenia) a zabezpečuje číslovanie (počítadlo `table`). Popis má tvar `Tabuľka číslo: popis_tabuľky`. Ak tento príkaz chýba, potom sa tabuľka nečísluje. Ak je prázdny, t. j. `\caption{}`, potom sa vypíše iba `Tabuľka číslo:`. Obsah tabuliek, resp. obrázkov sa inicializuje (identicky ako normálny obsah) príkazmi `\listoftables`, resp. `\listoffigures`.

Do prostredia `table` môžeme vložiť aj externý obrázok (napr. `*.eps`, pozri [3–5]). Takto vložená tabuľka sa sádza od ľavého okraja strany, preto je vhodné ju centrovat'. Na záver ukážeme (v tomto prípade obsahovo nelogické) niektoré možnosti tabuliek. Dve plávajúce tabuľky vedľa seba umiestnime pomocou prostredia `minipage` (pozri tabuľky 4 a 5, 6).

```
\begin{tabular}{|l|l|l|l|} & & & \hline\hline
{\bf Tovar} & {\bf Cena} & {\bf Množstvo} & {\bf Celkom} \\ \hline\hline
Kolobežka veľká & 22,00 \euro & 11 ks & 242,00 \euro \\ \hline
Lopta & 12,21 \euro & 2 ks & 24,42 \euro \\ \hline\hline
\end{tabular}
```

```

\begin{table}[ht]\begin{center} \begin{minipage}[t]{.48\linewidth}
  \setlength{\doublerulesep}{0pt}
\begin{tabular}[t]{|@{}p{5em}||*{3}{l}|} \hline\hline
{\bf Tovar} & & {\bf Cena} & & {\bf Množstvo} & & {\bf Celkom} & \\\hline\hline
Kolobežka veľká & 22,00 \euro & & 11 ks & & & 242,00 \euro & \\\hline
Lopta & 12,21 \euro & & 2 ks & & & 24,42 \euro & \\\hline\hline
\end{tabular}\caption{Prvá tabuľka}\end{minipage}
  \hfill
\begin{minipage}[t]{.48\linewidth} \caption{Druhá tabuľka}
  \setlength{\doublerulesep}{0.5pt}
\begin{tabular}[t]{|p{4em}||l|c|@{}r|} \hline\hline
{\bf Tovar} & & {\bf Cena} & & {\bf Množstvo} & & {\bf Celkom} & \\\hline\hline
\cline{1-2}\cline{4-4}
\multicolumn{2}{|c|}{Kolobežka veľká} & & 11 ks & & & 242,00 \euro & \\\hline
Lopta & 12,21 \euro & & \multicolumn{1}{l}{2 ks} & & & 24,42 \euro & \\\hline\hline
\end{tabular}\end{minipage} \end{center}\end{table}

```

Tovar	Cena	Množstvo	Celkom
Kolobežka veľká	22,00 €	11 ks	242,00 €
Lopta	12,21 €	2 ks	24,42 €

Tabuľka 3: Prostredie tabular

Tovar	Cena	Množstvo	Celkom
Kolobežka veľká	22,00 €	11 ks	242,00 €
Lopta	12,21 €	2 ks	24,42 €

Tabuľka 4: Prvá tabuľka

Tabuľka 5: Druhá tabuľka

Tovar	Cena	Množstvo	Celkom
Kolobežka veľká		11 ks	242,00 €
Lopta	12,21 €	2 ks	24,42 €

6 Záver

V zdrojových textoch sa používa symbol tilda `~`, ktorý zakazuje v danom mieste rozdelenie slov na nový riadok. V texte sa vyskytuje aj symbol pre euro (€). Keďže Eurozóna je podstatne mladší produkt ako \LaTeX , euro nie je jeho súčasťou. Tento problém rieši balíček `eurosym` a príkaz `\euro` (v preambule musíme deklarovať `\usepackage{eurosym}`).

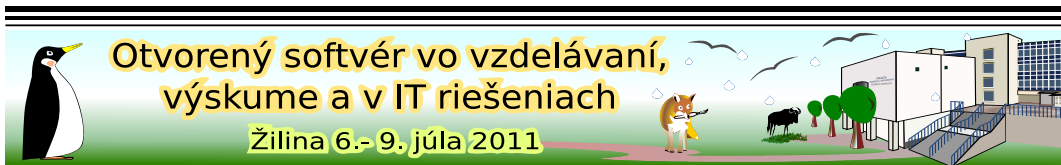
Ďakujem všetkým nadšencom, ktorí sa venujú rozvíjaniu \TeX u. Táto práca vznikla na propagáciu \LaTeX u.

Literatúra

- [1] BLAŠKO, R.: *\LaTeX nie je farba na maľovanie*, Otvorený softvér vo vzdelávaní, výskume a IT riešeniach, zborník medzinárodnej konferencie OSSConf 2010, Žilina, 1.–4. júla 2010, str. 43–52, ISBN 978-80-970457-0-8,
<http://sospreskoly.org/files/OSSConf2010/ossconf2010-Blasko.pdf>;
resp. Elektronický časopis openMagazin 09/2010, Liberix, str. 23–29, ISSN 1804-1426,
<http://www.openmagazin.cz/pdf/2010/openMagazin-2010-09.pdf>.
- [2] KNUTH, D. E.: *The $T_{\text{E}}X$ book*, Volume A of *Computers and Typesetting*, Addison-Wesley Publishing Company (1984), ISBN 0-201-13448-9.
- [3] KOPKA, H. – DALY, P. W.: *\LaTeX – Podrobný průvodce*, Brno, Computer Press, 2004, ISBN 80-722-6973-9.
- [4] OETIKER, T. – PARTL, H. – SCHLEGL, E. – HYNA, I. – KOČER, M. – SÝKORA, P.: *Ne příliš stručný úvod do systému $\LaTeX 2_{\epsilon}$* , „public domain“ dokument, 1998,
<http://www.penguin.cz/~kocer/texty/lshort2e/lshort2e-cz.pdf>, resp.
<http://ftp.cstug.cz/pub/tex/CTAN/info/lshort/slovak/Slshorte.pdf>
(Buša J. ml. a st., slovenský preklad).
- [5] RYBIČKA, J.: *\LaTeX pro začátečníky*, Brno, KONVOJ 2003, ISBN 80-7302-049-1.

Kontaktná adresa

RNDr. Rudolf BLAŠKO, PhD., Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská Republika, beerb@frcatel.fri.uniza.sk, <http://frcatel.fri.uniza.sk/~beerb/>.



DOI: 10.5300/2011-OSSConf/239

OTVORENÝ FORMÁT TEXTOVÝCH DOKUMENTOV V KANCELÁRII

FOLTIN, Martin, (SK); BLAHO, Michal (SK); FODREK, Peter(SK)

Abstrakt. *Pred dvoma rokmi sme sa zaoberali myšlienkou využitia otvorených formátov v kancelárskych dokumentoch [1]. Porovnaním sme zistili, že nasadenie ODF (Open Document Format) do kancelárskeho prostredia získavajú prakticky všetci. Používatelia nemusia riešiť problém s kompatibilitou formátov a zavedenie tohto štandardu nemusí predstavovať pre organizáciu žiadne finančné zaťaženie. Ukázalo sa, že najväčší problém so spracovaním otvoreného formátu má balík od spoločnosti Microsoft – MS Office 2007 napriek použitiu špeciálneho zásuvného modulu pre prácu s ODF. Skúseneho používateľa nekompatibilita neprekvapí, nakoľko Microsoft má problém s kompatibilitou aj vo svojich formátoch OOXML. Vo svojej nekompatibilite šiel dokonca tak ďaleko, že jeho odnož pre operačný systém Mac OS X (Microsoft Office 2008) nedokázal korektne otvoriť dokument .docx, ak obsahoval vzorce. Práve v tomto prípade sa ako vhodné riešenie ukazuje nasadenie otvoreného formátu pre kancelárske dokumenty. V článku sa budeme zaoberať tým, čo sa v tejto oblasti zmenilo za posledné dva roky.*

Kľúčové slová. *OpenOffice.org, LibreOffice, Microsoft Office 2010/2011, Office Open XML, Open Document Format, porovnanie, použiteľnosť.*

OPEN DOCUMENT FORMAT FOR PRODUCTIVITY SUITE'S TEXT PROCESSORS

Abstract. *We have been to deal of open document formats in our paper[1] two years ago. During Comparison we were to find out that usage of ODF (Open Document Format) in everyday office use helps everyone to better hit their goals. Users are not to solve document format compatibility issues and therefore converting to this format may not cost any company substantial amount of money. It has been found out that most and only ODF compatibility issues were caused by using Microsoft's productivity suite especially Microsoft Office 2007. It's incompatibility is horrible. We were to try open Microsoft's Windows based Office Word 2007 document (.docx) using it's own Mac OS X version named Microsoft Office Word 2008. and it failed when we were to use formulas in the document. This is case when it is suitable to use open documents format for productivity suite documents. This paper deals changes in the state of art in this case during last two years.*

Key words and phrases. *OpenOffice.org, LibreOffice, Microsoft Office 2010/2011, Office Open XML, Open Document Format, comparison, usability.*

1 Úvod

Významnou udalosťou v oblasti kancelárskeho softvéru bolo určite uvedenie nového balíku od spoločnosti Microsoft. Najskôr pre operačný systém Windows (MS Office 2010) a o pár mesiacov neskôr aj pre operačný systém Mac OS X (MS Office 2011). Otázkou, ktorou sa budeme v článku zaoberať je, ako si tieto balíky poradia s dokumentami ODF. Ďalšie udalosti sa odohrali pri vývoji lídra na poli balíkov primárne určených pre generovanie ODF – OpenOffice.org (<http://www.openoffice.org/>). Práve tomuto balíku pribudol významný konkurent v podobe LibreOffice (<http://www.libreoffice.org/>). Oba balíky sa tešia značnej popularite používateľov na všetkých obvyklých platformách. Svojim životom si naďalej existuje aj klon OpenOffice pre Mac OS X – NeoOffice (<http://www.neooffice.org/>). Svojich používateľov láka na aktuálnu verziu 3.2, ktorá sa už pripravuje na príchod nového operačného systému Mac OS X 10.7 (kódové meno Lion). Benefitom proti zavedeným balíkom je uspôsobený dizajn do podoby Mac OS X, ale aj podpora mobilných zariadení iPhone a iPad v podobe NeoOffice Mobile.

V roku 2010 došlo k rozdeleniu projektu OpenOffice.org. Existovala skupina vývojárov dlhodobo nespokojná s postupom hlavného donora projektu – spoločnosťou Sun. Rozpory vyústili až do odchodu kľúčových vývojárov (autori až 85% kódu OpenOffice.org) po tom ako spoločnosť Sun vzal pod svoje krídla Oracle. Táto skupina vývojárov si preto založila vlastnú nadáciu The Document Foundation. Nadácia si registrovala značku LibreOffice a pod touto značkou dnes ponúka vlastný kancelársky balík. Na svoju stranu sa im podarilo získať kľúčových podporovateľov OpenOffice.org ako RedHat, Canonical, alebo Novell. Po týchto udalostiach spoločnosť Oracle stratila záujem o ďalší vývoj OpenOffice.org a prenechala celý projekt komunite Apache. Medzi podporovateľov LibreOffice patrí už aj vývojový tím NeoOffice. Budúca verzia obľúbeného balíka pre Mac OS X už bude vychádzať práve z neho a nie z OpenOffice.org.

2 Štandardy ODF a OOXML

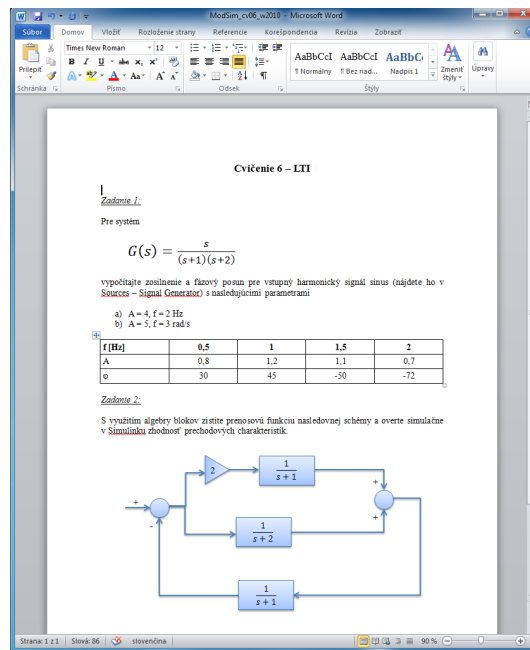
O výhodách používania otvorených formátov v elektronických dokumentoch sme sa zmienujeme v článku [1]. V súčasnosti sa ISO certifikátom v oblasti kancelárskych dokumentov môže pochváliť otvorený ODF. Formát ODF je implementovaný vo viacerých kancelárskych balíkoch ako LibreOffice, alebo OpenOffice.org.

Ďalším významným štandardom na poli kancelárskych dokumentov je Office Open XML (ďalej OOXML). O tento štandard sa opierajú súčasné verzie kancelárskeho balíka spoločnosti Microsoft – MS Office. Často sa stretávame s názorom, že súčasný súbor .docx (predstaviteľ štandardu OOXML) plní ISO štandard. V tomto smere však nie je situácia taká priaznivá, ako v prípade ODF. OOXML síce taktiež disponuje ISO certifikátom, žiaľ jedná sa o verziu, ktorá doposiaľ nebola implementovaná do súborov ako .docx. OOXML, ktorý skutočne má ISO sa pravdepodobne objaví až vo verzii MS Office 2013, alebo 2015. Aktuálne verzie súborov .docx teda týmto certifikátom nedisponujú [5].

Keďže Microsoft vydáva novú verziu balíka Office približne raz za tri roky, plná podpora Open XML je očakávaná najskôr v roku 2013, až päť rokov po schválení ISO štandardu Open XML [5].

3 Otvorené štandardy v praxi

Sústredíme teraz svoju pozornosť na aktuálne verzie populárneho balíka od spoločnosti Microsoft – MS Office 2010 a 2011. Microsoft sa vo vodách otvorených štandardov snaží presadiť svoj štandard Office Open XML. Aby sme zistili nakoľko sú dokumenty v tomto formáte čitateľné aj pre ostatné produkty, vytvorili sme dokument obsahujúci text, vzorec, tabuľku a obrázok (obr. 1). Uložili sme ho v tomto formáte (.docx).



Obr. 1: Dokument vytvorený ako .docx v MS Word 2010 v operačnom systéme MS Windows 7

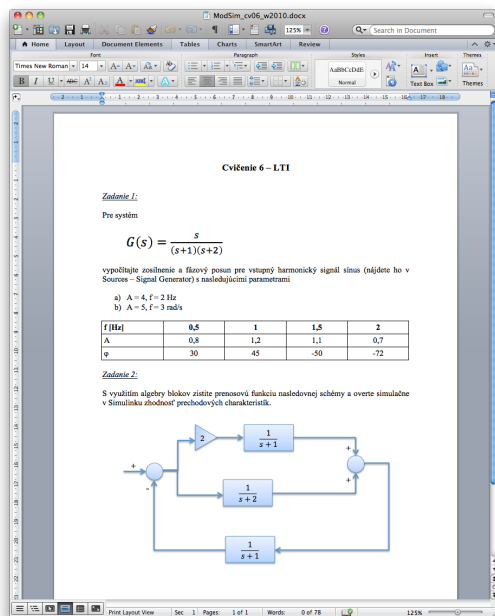
Tento dokument sme sa pokúsili otvoriť v balíku tej istej spoločnosti pre operačný systém Mac OS X (obr. 2). V minulosti práve vzorce prestavovali problém pre MS Word 2008.

Môžeme konštatovať, že s príchodom novej generácie balíka MS Office prichádza aj k zlepšeniu kompatibility medzi operačnými systémami Windows a Mac OS X. Jednotlivé verzie si dokážu medzi sebou bezproblémovo vymieňať dokumenty bez toho, že by nejakým obmedzovali používateľa. V prípade, že máte vo svojom okolí iba používateľov s operačným systémom Windows a Mac OS X a zároveň všetci majú aktuálnu verziu balíka MS Office, je možné bezproblémovo komunikovať. Problém nastáva, ak by ste do pracovného reťazca

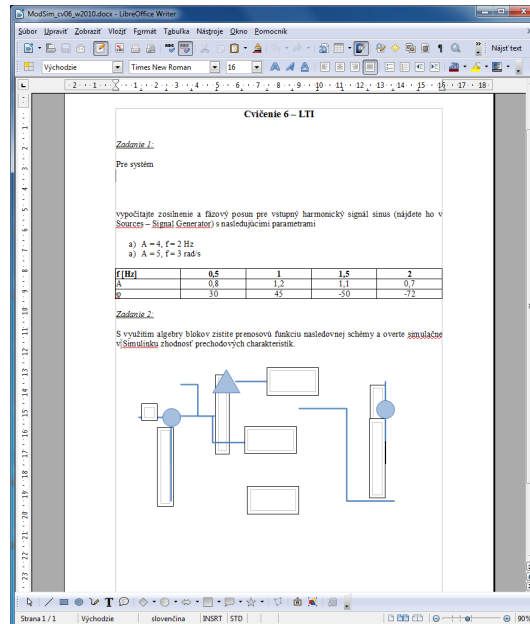
chceli zaradiť niekoho s operačným systémom Linux, alebo niekoho, kto nie je ochotný investovať nemalé finančné prostriedky do licencie MS Office. Problém implementácie korektného importu formátu .docx do programov z rodiny OpenOffice.org, alebo LibreOffice stále pretrváva. Problémom je práve implementácia OOXML vo vydaní, ktoré zatiaľ neplní ISO štandard. Preto nie je možné dostatočne kvalitne implementovať tento formát do programov iného producenta ako Microsoft. Jedná sa teda o jednoznačné obmedzovanie používateľa. Ak sa teda chcete zaradiť do tímu pracovníkov, ktorí využívajú na komunikáciu .docx nemáte inú možnosť ako kúpiť si počítač s operačným systémom Windows, alebo Mac OS X a kancelársky balík MS Office. Ako demonštráciu komplikácií uvádzame obrázok dokumentu .docx v balíku LibreOffice v operačnom systéme openSUSE (obr. 3).

Vidíme, že použitie formátu OOXML nie je vhodné, ak chceme naše dokumenty zdieľať naprieč operačnými systémami, alebo pokiaľ, nie všetci zúčastnení disponujú kancelárskym balíkom MS Office.

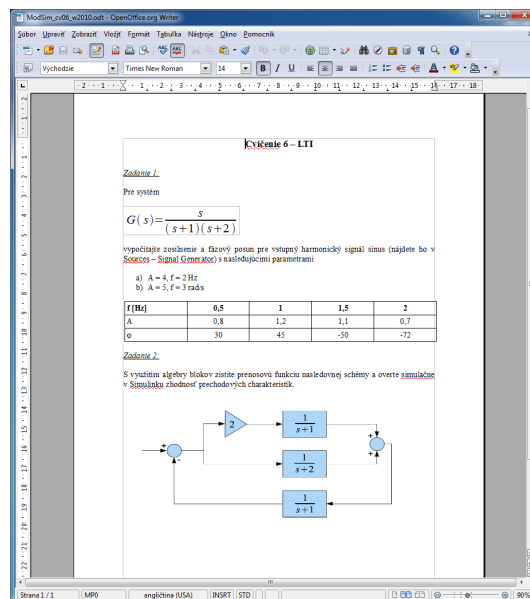
Sústredíme sa teda na druhú možnosť a tou je použitie ODF. Tento formát aj v súčasnej podobe plní ISO štandard a jeho implementácia by mala byť bezproblémová v každom operačnom systéme. Taktiež je na trhu značný počet kancelárskych programov, ktoré s týmto formátom dokážu narábať. Vytvorili sme obdobný dokument ako v predošlom prípade v kancelárskom balíku OpenOffice.org v operačnom systéme Windows 7 (obr. 4).



Obr. 2: Dokument .docx otvorený v MS Word 2011 v operačnom systéme Mac OS X 10.6 Snow Leopard

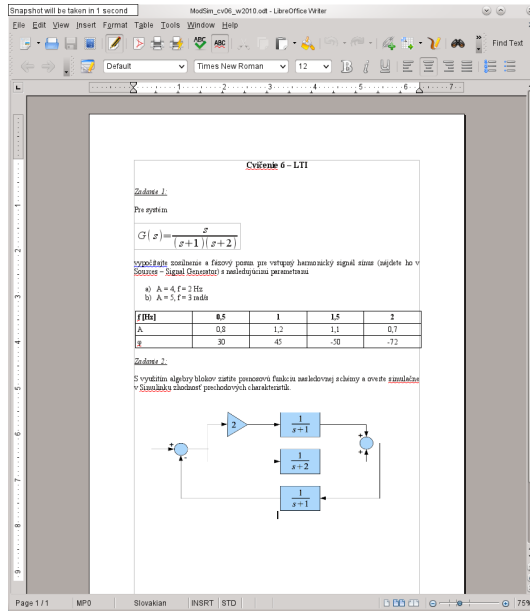


Obr. 3: Textový dokument .docx otvorený v LibreOffice v systéme openSUSE 11.4

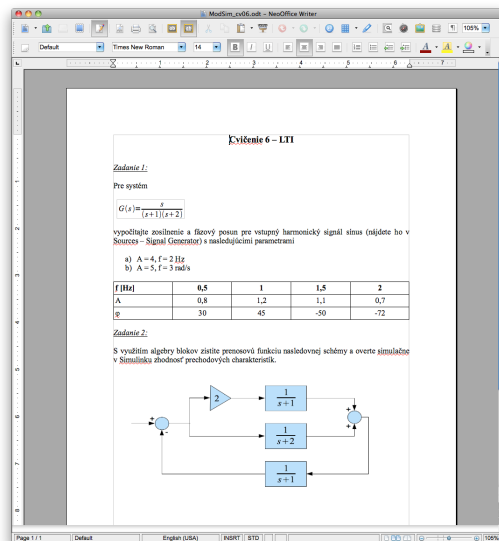


Obr. 4: Dokument .odt vytvorený v OpenOffice.org v operačnom systéme Windows 7

Tento .odt dokument môžeme bezproblémovo zdieľať s kolegami, ktorí využívajú napr. LibreOffice v OpenSUSE (obr. 5), alebo NeoOffice v Mac OS X (obr. 6).



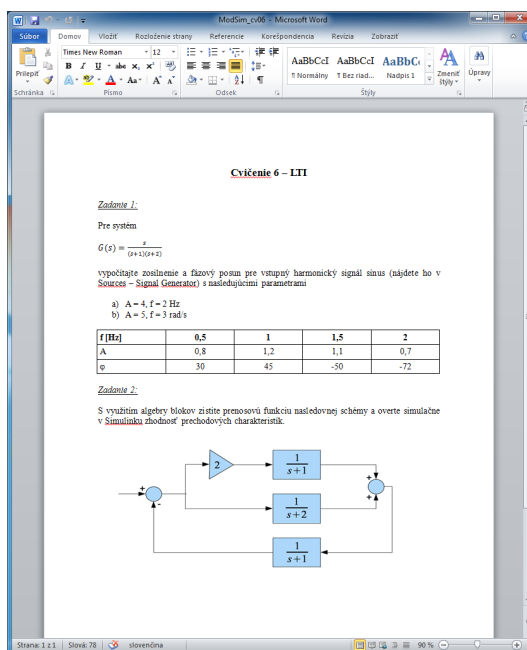
Obr. 5: .odt otvorený v LibreOffice v operačnom systéme openSUSE



Obr. 6: .odt otvorený v NeoOffice v operačnom systéme Mac OS X Snow Leopard

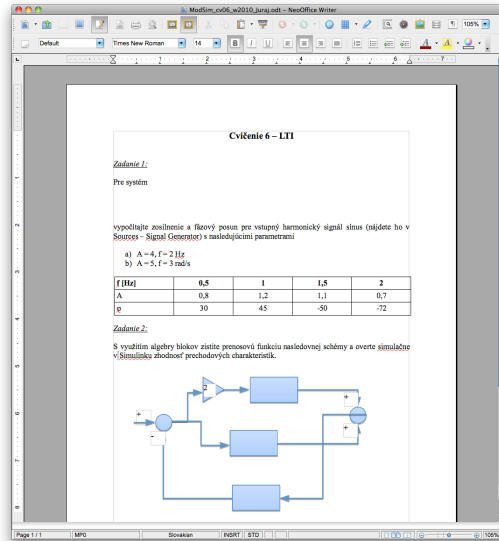
Vidíme, že použitie formátu .odt je bezproblémové vo viacerých operačných systémoch a aj v rôznych kancelárskych programoch. Používateľ si tak môže vybrať a nie je limitovaný. Nespornou výhodou je, že nami uvedené programy je možné využívať bezplatne aj na komerčné účely.

Spoločnosť Microsoft reaguje na zvyšujúcu sa popularitu formátu ODT a preto je možné tento formát editovať napr. aj v programe MS Word 2010 (obr. 7). Podpora vo Worde 2011 pre Mac však chýba. Je to napriek tomu, že ODF podporuje Mac OS X natívne vo svojich aplikáciách napr. v programe TextEdit, ktorý je obdobou programu Notepad (Poznámkový blok) v Microsoft Windows, alebo gedit/kate v dominantných prostrediach v GNU/Linux. V Mac OS X sú dokonca zabudované, od verzie 10.5, knižnice na prácu s ODF.



Obr. 7: .odt otvorený v MS Office 2010 v operačnom systéme MS Windows 7

Zdalo by sa, že sme sa vlastne dostali do štádia, kedy postačuje používať otvorený formát .odt a nemôže nastať problém. Používateľ si môže slobodne zvoliť operačný systém, alebo kancelársky balík. Dokonca už aj MS Word 2010 dokáže korektne otvoriť tento formát. Žiaľ práve implementácia ODF do MS Office nie je bezchybná. Otvoriť a editovať .odt vo Worde 2010 skutočne môžete a nepredstavuje to žiaden problém. Komplikácie sa objavia keď dokument uložíte a chcete ho editovať mimo Word 2010. V takomto prípade je už dokument zdeformovaný (obr. 8). Problematické sú najmä vzorce a vektorové obrázky. Zaujímavosťou je, že Word si s takýmto súborom bezproblémovo poradí. Musíme teda konštatovať, že populárny MS Office je stále problematický v prípade použitia ODF.



Obr. 8: Zdeformovaný .odt súbor Wordom 2010 (ovorený v NeoOffice)

4 Záver

S dvojročným odstupom môžeme hodnotiť situáciu v oblasti otvorených formátov tak, že sa na trhu objavuje nový významný hráč LibreOffice. Populárny OpenOffice.org sa vracia komunita. Microsoft implementuje do MS Office 2010 podporu ODF. Nejedná sa však o úplnú podporu. Čítanie a editovanie ODF súborov je bezproblémové. Komplikácie spôsobuje ukladanie v tomto formáte. Dokument .odt uložený v MS Word 2010 sa stáva pre ostatné programy zdeformovaný. V prípade, že sa rozhodnete využívať otvorený formát ODF na elektronickú komunikáciu, je stále výhodnejšie spoľahnúť sa na zavedené balíky v tejto oblasti ako OpenOffice.org, LibreOffice, alebo NeoOffice. Je pozoruhodné, že komunita zodpovedná za vývoj týchto balíkov nemá problém zaradiť podporu ODF a to ako pri editovaní, tak pri ukladaní dokumentov. Naproti tomu softvérový gigant, akým Microsoft bezosporu je, nedokáže vo verzii MS Office 2010 korektne zaradiť podporu tohto formátu. Verzia pre operačný systém Mac OS X – MS Office 2011 podporou ODF nedisponuje vôbec.

5 PodĎakovanie

Táto práca bola podporovaná Agentúrou KEGA na základe zmluvy č. 032STU-4/2011
 This work was supported by the KEGA Agency under the contract No. 032STU-4/2011
 Táto práca bola podporovaná aj Agentúrou VEGA v rámci zmluvy č. 1/0592/10
 This work was supported by the VEGA Agency under contract number 1/0592/10.

Literatúra

[1] FOLTIN, M., FODREK, P., BLAHO, M.: *Otvorený formát v kancelárskych dokumentoch*, [online] dostupné na: <http://www.posterus.sk/?p=161>, citované 16. 5. 2011, Posterus, Bratislava, 2009

[2] EFFENBERGER, F.: *Thousands of donors contribute €50,000 in just eight days to The Document Foundation*, [online] dostupné na: <http://blog.documentfoundation.org/2011/02/25/thousands-of-donors-contribute-e50000-in-just-eight-days-to-the-document-foundation>, citované 17. 5. 2011, The Document Foundation, Spolková republika Nemecko, 2011

[3] EFFENBERGER, F.: *LibreOffice Community starts 50,000 € challenge for setting-up its foundation*, [online] dostupné na: <http://blog.documentfoundation.org/2011/02/16/>, citované 17. 5. 2011, The Document Foundation, Spolková republika Nemecko, 2011

[4] Wikimedia: *The Document Foundation*, [online], dostupné na: https://secure.wikimedia.org/wikipedia/en/wiki/The_Document_Foundation, citované 17. 5. 2011, Wikimedia Foundation, San Francisco, CA, USA, 2011

[5] DSL.sk, *MS nebude plne podporovať Open XML ani v Office 2010*, [online], dostupné na: <http://www.dsl.sk/article.php?article=8966>, citované 17. 5. 2011, Dsl.sk, Bratislava, 2010

Kontaktná adresa

Martin FOLTIN (Ing., PhD.),

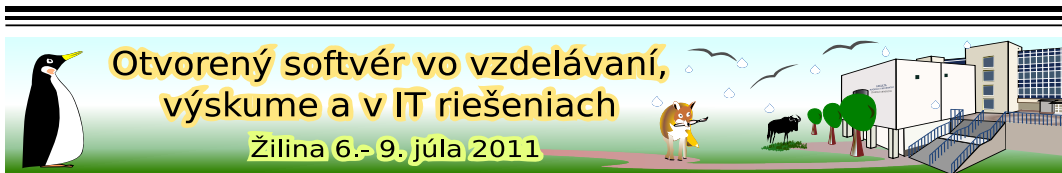
Vedúci, Oddelenie informačných a komunikačných systémov,
Ústav riadenia a priemyselnej informatiky,
Prodekan pre rozvoj informatizácie a vzťahy s verejnosťou FEI STU v Bratislave,
Ilkovičova 3, 812,19 Bratislava, martin.foltin@stuba.sk

Michal BLAHO (Ing.),

Pedagogický pracovník,
Oddelenie informačných a komunikačných systémov,
Ústav riadenia a priemyselnej informatiky FEI STU v Bratislave,
Ilkovičova 3, 812,19 Bratislava, michal.blaho@stuba.sk

Peter FODREK, ml. (Ing., PhD.),

Výskumný pracovník,
Oddelenie informačných a komunikačných systémov,
Ústav riadenia a priemyselnej informatiky FEI STU v Bratislave,
Ilkovičova 3, 812,19 Bratislava, peter.fodrek@stuba.sk



DOI: 10.5300/2011-OSSConf/249

PREZENTAČNÉ MATERIÁLY V TRIEDE BEAMER

KOZUBÍK, Aleš, (SK)

Abstrakt. *Tento príspevok prináša úvod do triedy `beamer`. Sú tu predstavené základné nástroje pre riadenie vzhľadu a štruktúry prezentačných materiálov.*

Kľúčové slová. *\LaTeX , prezentácia, trieda `beamer`.*

PRESENTATION MATERIALS WITH THE BEAMER CLASS

Abstract. *This paper presents an introduction into the `beamer` class. Here are described simple tools of the `beamerclass` managing the themes and content of the presentation.*

Key words and phrases. *\LaTeX , presentations, `beamer` class.*

1 Úvod

Pri vystúpeniach na konferenciách, resp. pri ich príprave sme často konfrontovaní s úlohou vytvorenia vhodných prezentačných materiálov. Pri riešení tejto úlohy máme na výber niekoľko možností. Jednou alternatívou je, že siahneme po produktoch z kancelárskych balíkov, či už je to Impress z OpenOffice.org alebo jeho komerční konkurenti. Druhou možnosťou, pre užívateľov oboznámených s typografickým systémom \TeX je využitie niektorého z balíčkov, podporujúcich tvorbu prezentácií.

Ako je pri otvorenom softvéri zvykom, ani tu neostáva na výber jediná cesta, ale je k dispozícii opäť viacero riešení. Sám mám osobnú skúsenosť s triedami `prosper` a `beamer`, ďalšie možnosti sú spomenuté napr. v [4], kde sú aj ukážky tej istej prezentácie s využitím viacerých tried dokumentov.

V tomto príspevku si v krátkosti predstavíme triedu `beamer` a uvedieme niekoľko jednoduchých príkladov na jeho použitie. Pri tom predpokladáme, že čitateľ má základné znalosti a zručnosti pre prácu so systémom \TeX resp. \LaTeX . V opačnom prípade je možné sa so základmi práce oboznámiť napr. v [1], alebo detailnejšie vo vydarenej učebnici [5].

2 Začínáme

Trieda `beamer` je, ako sa uvádza aj v [3], už štandardnou súčasťou distribúcie \TeX u, a to ako pre OS Windows, tak aj pre operačné systémy Unix/Linux. Užívatelia, ktorí nemajú možnosť využívať výhody niektorej z distribúcií, (najrozšírenejšia je asi \TeX Live) nemajú iné východisko, ako si stiahnuť a doinštalovať `beamer` z archívu <https://bitbucket.org/rivanvx/beamer/downloads>.

Ak teda budeme vychádzať z predpokladu, že máme `beamer` úspešne nainštalovaný, môžeme sa pustiť do prípravy svojej prvej prezentácie. Prvým krokom je, že musíme v preambuli označiť dokument ako prezentáciu v `beameri`, čo docielime príkazom:

```
\documentclass [<voľby>]{beamer}
```

Nepovinné parametre [<voľby>] slúžia na úpravu niektorých vlastností získaného dokumentu. Ako príklad uveďme:

- `compress` minimalizuje veľkosť navigačných symbolov
- `t` zarovnáva obsah stránky na vrchol stránky
- `c` zarovnáva obsah stránky na stred stránky
- `xcolor=dvipsnames` definuje, aké názvy farieb budú v dokumente používané. `Beamer` automaticky nahráva aj balíček `xcolor`, ak však chceme použiť rozšírené názvoslovie farieb, je potrebné to deklarovať. Inými možnosťami sú voľby `xcolor=svgnames` alebo `xcolor=x11names`.

2.1 Základná štruktúra prezentácie – frame

Základnou stavebnou (a zobrazovanou) jednotkou prezentácie v `beameri` je **frame**, ktorý sa môže skladať z niekoľkých slajdov. Každý `frame` vytvoríme pomocou prostredia `frame`, teda postupnosťou príkazov:

```
\begin{frame}
... obsah framu...
\end{frame}
```

Každý z `framov` možno pomenovať a prípadne mu aj priradiť podnadpis, čo docielime príkazmi `\frametitle{Názov}` a `\framesubtitle{Podnadpis}`.

Takisto ako celá trieda `beamer`, aj jednotlivým `framom` je možné priradiť nepovinné voľby, ktorými sa upravujú vlastnosti jednotlivých `framov`. Z mnohých alternatív uveďme aspoň niekoľko najfrekvencovanejších:

- `plain` potláča hlavičky, pätičky, okrajové informácie na slajde a pod., výstupom je len nadpis a samotný text,

- `fragile` táto voľba je potrebná v prípade, že chceme na stránke používať prostredie `verbatim` (napr. pre výpisy programov),
- `t` umiestňuje obsah framu od vrcholu stránky
- `c` vertikálne centruje obsah framu na stránke.

Kompletná konštrukcia našej prvej prezentácie, ktorú si môžeme uložiť do nejakého cvičného súboru napr. `skuska.tex`, by teda mohla vyzerat' napríklad takto:

```
\documentclass[compress,t]{beamer}
\usepackage[slovak]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\begin{document}
\begin{frame}[c]
\frametitle{Môj prvý frame}
\framesubtitle{Môj prvý slajd}
Text môjho prvého slajdu, zarovnaný pekne na prostriedok.
\end{frame}
\end{document}
```

Ak si chceme pozrieť výsledok, je potrebné prezentáciu skompilovať a to buď pomocou príkazu

```
pdflatex skuska.tex
```

priamo do formátu pdf, alebo postupnosťou príkazov:

```
latex skuska.tex
dvips skuska.dvi
ps2pdf skuska.ps
```

v prípade, že využívame nástroje `PSTricks`. Trieda `beamer` však nepodporuje `dvipdf`.

2.2 Riešenie vzhľadu celej prezentácie

Keď si skompilujeme našu prvú prezentáciu a pozrieme si výsledok, uvidíme, že je síce funkčná, avšak nie veľmi príťažlivá. Celkový dizajn stránok prezentácie sa riadi pomocou tzv. tém, ktoré ovplyvňujú globálne správanie celej prezentácie. Prehľad všetkých tém, ktoré sú k dispozícii obvykle nájdeme v adresároch:

```
/usr/share/texmf/tex/latex/beamer/themes/theme/
/usr/share/texmf/tex/latex/beamer/themes/inner/
/usr/share/texmf/tex/latex/beamer/themes/outer/
/usr/share/texmf/tex/latex/beamer/themes/color/
/usr/share/texmf/tex/latex/beamer/themes/font/
```

V prípade OS Windows obvykle v adresári

```
C:\texlive\2010\texmf-dist\tex\latex\beamer\themes
```

Základnú tému prezentácie, ktorá riadi všetky prvky prezentácie, nastavíme v preambuli dokumentu príkazom:

```
\usetheme{<názov témy>}
```

Do nášho cvičného súboru si teda môžeme pridať napr. `\usetheme{Warsaw}` a pozrieť si výsledok.

Dekorácie jednotlivých framov v podobe záhlaví, pätičiek príp. okrajov je možné centrálné ovládať prostredníctvom tzv. vonkajších tém príkazom:

```
\useoutertheme{<názov témy>}
```

Podobne, správanie samotného obsahu framu sa centrálné riadi pomocou vnútorných tém, teda príkazom:

```
\useinnertheme{<názov témy>}
```

Posledné dve témy, ktorých pomenovanie je veľmi intuitívne, riadia farebnú schému celej prezentácie resp. použité písmo. Zadáваме ich príkazmi:

```
\usecolortheme{<názov témy>}
\usefonttheme{<názov témy>}
```

Ako preddefinované farebné témy možno použiť `albatross`, `beetle`, `crane`, `default`, `dolphin`, `dove`, `fly`, `lily`, `orchid`, `rose`, `seagull`, `seahorse`, `sidebartab`, `structure`, `a whale`. Pokiaľ sa týka tém pre voľbu použitých fontov, ako základné témy máme na výber `default`, `professionalfonts`, `serif`, `structurebold`, `structureitalicserif` a `structuresmallcapserif`. Poznamenajme, že týmto spôsobom určujeme len základný tvar použitého písma. Jeho veľkosť je možné ovplyvniť v rámci volieb `\documentclass`, napr.

```
\documentclass[17pt]{beamer}
```

2.3 Titulná stránka prezentácie a osnova

Zatiaľ sme si ukázali, ako možno vytvoriť jednoduchý prezentačný materiál a meniť vzhľad celej prezentácie. Jej dôležitou súčasťou je však aj titulná stránka. Samotný obsah titulnej stránky potom predpíšeme taktiež obvyklým spôsobom, spravidla ešte v preambule dokumentu. Celkovo môžeme zadať až päť údajov s ich plnou a skrátanou formou:

```
\title[<krátky názov>]{<Plný (dlhý) názov>}
\subtitle[<krátky podnadpis>]{<úplný zápis podnadpisu>}
\author[<skrátané meno autora>]{<plné meno autora>}
\institute[<skratka inštitúcie>]{<plný názov organizácie>}
\date{<dátum>}
```


Pripomeňme, že ako dátum je možné uviesť aj `\today`, ktorý zabezpečí pri kompilácii vloženie aktuálneho dátumu.

Titulnú stránku potom do prezentácie zaradíme konštrukciou, ktorá užívateľom \TeX u nie je neznáma, teda:

```
\begin{frame}
\titlepage
\end{frame}
```

Druhou dôležitou súčasťou každej prezentácie býva osnova, ktorá poslucháčov hneď v úvode stručne oboznámi s priebehom a obsahom celého vystúpenia. To je možné docieľiť rozdelením prezentácie do sekcií a podsekcí. V samotnej prezentácii sa názvy sekcií a subsekcí nezobrazujú, je z nich však možné štandardným spôsobom vygenerovať obsah celej prezentácie. Názvy sekcií sa potom v obsahu objavia ako očíslované položky, názvy subsekcí len v bodoch. Osnovu prezentácie teda vygenerujeme pridaním framu:

```
\begin{frame}
\frametitle{Osnova}
\tableofcontents[pausesections]
\end{frame}
```

Nepovinný parameter `pausesections` spôsobuje postupné zobrazenie jednotlivých sekcií na vstup z myši alebo klávesnice. Bez tohto prepínača je celá osnova zobrazená naraz. Podobný efekt je možné docieľiť aj pre podsekcie pridaním `pausesubsections`, prípadne kombinovať obidva parametre oddelené čiarkou.

3 Vnútoraná štruktúra jednotlivých slajdov

V rámci prípravy prezentačného materiálu je možné využívať štandardné prostredia \LaTeX -u, ako sú `itemize`, `enumerate` alebo `description`. Ich vzhľad sa upravuje pomocou tém, už spomenutých v predchádzajúcom odstavci. Takisto prostredie `equation` pre sadzbu matematiky funguje bez akýchkoľvek úprav. Popri týchto známych prostrediach ponúka trieda `beamer` viacero štruktúr pre úpravu vzhľadu slajdu. Tu sa zameriame na dve hlavné štruktúry a to na:

- stĺpce (`columns`) a
- zvýraznenie do blokov (`block`).

3.1 Stĺpce

Pre rozdelenie obsahu slajdu do viacerých stĺpcov obsahuje trieda `beamer` prostredie `columns`. Toto umožňuje rozdeliť slajd na ľubovoľný počet stĺpcov, treba však vždy pamätať aj na to, či má zmysel vytvárať príliš vysoký počet stĺpcov. Syntax pre stĺpcové prostredie vyzerá takto:

```

\begin{columns}
\begin{column}[pozícia]{šírka}
Obsah stĺpca~1
\end{column}
\begin{column}[pozícia]{šírka}
Obsah stĺpca~2
\end{column}
... atď. ...
\end{columns}

```

Pri jednotlivých stĺpcoch môžeme voliť pozíciu ich vzájomného zarovnávania. Na výber máme ako obvykle tri alternatívy t, c alebo b, teda zarovnanie na vrchol, stred alebo dno stĺpca. Parameter šírka potom slúži na úpravu šírky sadzby daného stĺpca. Súčet všetkých širok by pochopiteľne nemal presahovať šírku stránky. Máme na výber voľbu medzi pevným nastavením v dĺžkových jednotkách (napr. 25mm) alebo relatívny rozmer v násobkoch parametra `\textwidth` (napr. `0.3\textwidth`).

3.2 Bloky

Bloky je možné charakterizovať ako nástroje na výrazné oddelenie určitej špecifickej časti textu (definície, vety, poučky, citáty a pod.) od ostatného obsahu slajdu. Syntax pre zaradenie bloku je nasledujúca:

```

\begin{block}{Názov bloku}
Text zvýraznený v~bloku.
\end{block}

```

Celkový vzhľad môžeme opäť ponechať na nastavených témach, aj keď je v prípade potreby možné meniť vzhľad (najmä farebný) jednotlivých blokov. To je možné doceliť postupnosťou príkazov:

```

\setbeamercolor{horny}{fg=<farba textu 1>,bg=<farba pozadia 1>}%
\setbeamercolor{dolny}{fg=<farba textu 2>,bg=<farba pozadia 2>}%
\begin{beamerboxesrounded}[upper=horny,lower=dolny,shadow=true]{Názov}
Obsah bloku
\end{beamerboxesrounded}

```

Ako je vidieť, tu najskôr definujeme farebnosť hornej a dolnej časti vytvoreného bloku. Pri tom fg je ako obvykle farba popredia, teda zobrazeného textu a bg farba pozadia. Pomenovania farieb volíme v rámci nastaveného farebného priestoru, teda dvips alebo svg. Pomocou shadow nastavujeme, či má byť blok zobrazený s ilúziou tieňa alebo nie.

4 Práca s grafikou

Grafické objekty je zvykom vkladať do prezentácie v dvoch prípadoch:

- ako obrázky, ktoré sú súčasťou prezentácie,
- ako obrázky, resp. textúry na pozadí slajdu.

V prvom prípade postupujeme štandardným postupom, teda pomocou prostredia `figure` a pomocou balíčka `graphicx` a príkazu `\includegraphics`. Vzhľadom na to, že väčšinou prekladáme pomocou `pdflatex`, používame obvykle obrázky vo formáte `jpg`, `png` a `pdf`. Budeme sa teda viac venovať druhej alternatíve, ktorou je vytváranie textúr na pozadí slajdov.

Najjednoduchšou operáciou je nastavenie farby pozadia. To vykonáme pomocou príkazu

```
\setbeamercolor{normal text}{bg=farba}
```

kde hodnota `farba` sa nastaví buď priamo jej pomenovaním alebo zmiešavaním farieb, spôsobom bežným v systéme \LaTeX . Tak napríklad konštrukciou

```
\setbeamercolor{normal text}{bg=green!12}
```

```
\begin{frame}{Nastavenie farby pozadia}
```

Táto stránka má zelené pozadie s $\sim 12\%$ intenzitou zelenej farby.

```
\end{frame}
```

vytvoríme slajd, ktorý bude mať na svojom pozadí svetlo zelenú farbu (úroveň jej intenzity v $\%$ je daná číslom za výkričníkom). Toto nastavenie farebnosti pozadia platí až do najbližšieho príkazu `\setbeamercolor`, ktorý ho zmení. Ak ho chceme aplikovať len na jednu stránku, je potrebné celú konštrukciu uzavrieť do zložených zátvoriek. Naopak, umiestnením príkazu do preambuly dokumentu nastavíme túto vlastnosť pre celú prezentáciu s výnimkou slajdov, na ktorých ju tým istým príkazom zmeníme.

Takisto je možné na pozadí nastaviť pomocou príkazu `\setbeamertemplate` postupný farebný prechod z jednej farby do druhej. Ak chceme napríklad nastaviť prechod z 30% -nej zelenej farby na vrchole stránky do 20% -nej červenej na spodku stránky, použijeme konštrukciu:

```
{
```

```
\setbeamertemplate{background canvas}[vertical shading]
```

```
[bottom=red!20,top=green!30]
```

```
\begin{frame}[c]{Nastavenie farebného prechodu na pozadí}
```

Pozadie prechádza z $\sim 12\%$ červenej dole do 30% zelenej hore.

```
\end{frame}
```

```
}
```

Pripomeňme si, že uzavretím do zložených zátvoriek sa toto nastavenie vzťahuje len na jeden slajd. Prechod možno okrem volieb `bottom` a `top` doplniť ešte nastavením tretieho parametra `middle`, ktorá doplní farbu, cez ktorú sa má celý prechod realizovať. Niekedy je žiaduce (napr. na zabezpečenie mierky) pridať do pozadia aj mriežku. To zariadime príkazom:

```
\setbeamertemplate{background}[grid][step=velkosť,color=farba]
```

Parametrom *velkosť* je numerický údaj s vyznačením dĺžkovej jednotky a udávajúci rozostup jednotlivých priamok, vytvárajúcich mriežku a *farba* je názov farby mriežky (opäť je možné aj štandardné nastavenie intenzity farby, príp. zmiešanie farieb).

Ukážme si ešte, ako je možné na pozadie slajdu umiestniť obrázok. Ten si však najskôr musíme pripraviť v adekvátnom grafickom formáte. Potom ho už vložíme príkazom `\setbeamertemplate` ako jeho druhý povinný parameter `{background canvas}`. Ak teda máme k dispozícii obrázok `meno_obrazka.jpg` a chceme ho vložiť na pozadie nášho slajdu, postupujeme takto:

```
\setbeamercolor{structure}{fg=farba}
\setbeamertemplate{background canvas}{\includegraphics
[width=\paperwidth,height=\paperheight]{meno_obrazka.jpg}}
\begin{frame}[plain]{}
\textcolor{farba}{Text slajdu}
\end{frame}
```

Poznamenajme, že nastavenie farieb pre štruktúry je nepovinnou súčasťou, ale je výhodné pamätať na zmeny farebnosti prvkov na slajde tak, aby boli na pozadí obrázka dobre viditeľné. Takisto farbu textu je dobré ovplyvniť pomocou `\textcolor{farba}{text}` tak, aby bol text dobre čitateľný na farebnom pozadí obrázku.

Pri vkladaní obrázkov je tiež dobré si uvedomiť, že základný formát slajdu má pomer strán 4 : 3 a ak náš obrázok má iný pomer strán, môže dôjsť ku jeho značnej deformácii.

5 Prekrývania na slajde

V tejto záverečnej sekcii sa budeme venovať prístupom ku niektorým základným efektom v rámci prezentácie, ako sú prekrytia slajdov v rámci jednej stránky.

Základným efektom na stránke prezentácie je postupné objavovanie sa jednotlivých prvkov stránky, prípadne náhrada jedného prvku iným bez zmeny ostatných prvkov stránky. Základným ovládacím nástrojom je príkaz `\pause`, ktorý označuje miesta na slajde, kde sa má zobrazovanie prerušiť až do ďalšieho pokynu na pokračovanie. Tak napríklad konštrukcia:

```
\begin{frame}[c]{Postupný text}
Tento \pause text \pause sa \pause objavuje \pause slovo \pause
za \pause slovom.
\end{frame}
```

vytvára slajd, ktorého text sa bude postupne objavovať na slajde po kliknutí myšou alebo použitím navigačných kláves. Tento postup je užitočný napr. pri vytváraní zoznamu na stránke, ktorý potom postupne zobrazujeme. Sústavné opakovanie si potom môžeme zjednodušiť pridaním voľby `[<+>]` na začiatku prostredia, teda napr.:

```

\begin{itemize}[<+>]
\item prvá položka
\item druhá položka
\item atď.
\end{itemize}

```

vytvorí zoznam, v rámci ktorého sa budú nové položky objavovať vždy na kliknutie myšou alebo v reakcii na stlačenie navigačných kláves.

Môže sa stať, že niektoré časti textu chceme na slajde zobrazit' len dočasne, prípadne jeden prvok na slajde nahradiť iným. V prvom prípade si môžeme pomôcť príkazom `\onslide<n> text`, ktorý zobrazuje text iba na špecifikovaných slajdoch. V prípade postupne odkrývaných zoznamov je možné jednotlivé položky ošetriť pridaním voľby `<n>`, ktorá taktiež špecifikuje poradové číslo, resp. čísla slajdov na stránke, na ktorých sa má daný text objaviť. Zdrojový kód takýchto stránok potom môže vyzerat' napr. takto:

```

\begin{frame}[c]{Text na vybraných slajdoch}
\onslide<1> Od prvého slajdu do konca
\onslide<2> od druhého do konca
\onslide<-2> len po druhý slajd
\onslide<3> len tretí slajd.
\end{frame}
\begin{frame}[c]{Text na vybraných slajdoch}
\begin{itemize}
\item<1-2> Prvý a druhý slajd
\item<2> Len druhý slajd
\item<2-3> Druhý a tretí slajd
\item<1,3> Prvý a tretí slajd
\end{itemize}
\end{frame}

```

Z tohto kódu je zrejme aj to, aké sú prípustné voľby špecifikácií, na ktorom slajde sa má daný text, príp. iný objekt objaviť. Ak chceme na tom istom mieste nahradiť daný objekt iným, použijeme príkaz `\only<n>`, ktorý špecifikuje číslo slajdu, na ktorom sa objaví. Tak napríklad príkazmi:

```

\begin{block}{Zmena textu v~bloku}
\only<1>{Text prvej poučky}
\only<2>{Text druhej poučky}
\end{block}

```

dosiahneme efekt, kedy sa vymenia dva texty v spodnej časti daného bloku.

Tieto konštrukcie je možné využiť taktiež na postupné zvýrazňovanie jednotlivých častí textu, pričom všetky ostávajú viditeľné. Dosiahneme to spojením s príkazom `\alert`. Použitie si môžeme ilustrovať kódom:

```

\alert{Zvýraznené stále}\\
\alert<2>{Zvýraznené len na slajde~2}\\
\alert<3>{Zvýraznené len na slajde~3}\\
\alert<1,3>{Zvýraznené len na slajdoch~1~a~3}\\
\alert<-2,4>{Zvýraznené na slajdoch~1,2~a~4}\\

```

ktorý názorne ilustruje aj špecifikácie poradia zvýraznenia jednotlivých položiek ako aj trvanie týchto zvýraznení na jednotlivých slajdoch.

S postupným zobrazovaním obsahu súvisí ešte jedna dôležitá voľba, ktorú nastavujeme príkazom `\setbeamercovered{transparent}`. Táto umožňuje zobrazit' priesvitne celý obsah slajdu, ktorý je viditeľný pre rečníka na obrazovke avšak pri projekcii ostane neviditeľným až do okamihu jeho plného zobrazenia.

6 Záver

Tento príspevok poskytuje popis len niekoľkých základných techník pre prípravu prezentačných materiálov za pomoci typografického systému $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. V uvedenej podobe môže poslúžiť pre prípravu jednoduchých, plnohodnotných prezentácií. Vzhľadom na svoj rozsah nemôže byť vyčerpávajúcim návodom popisujúcim bohaté možnosti tejto triedy a už vôbec nie prehľadom alternatívnych nástrojov, ako je napr. *prosper*. Niektoré pokročilejšie techniky a triky môžeme predstaviť zasa niekedy nabadúce. Tí menej trpezliví môžu siahnuť po manuáli triedy *beamer*.

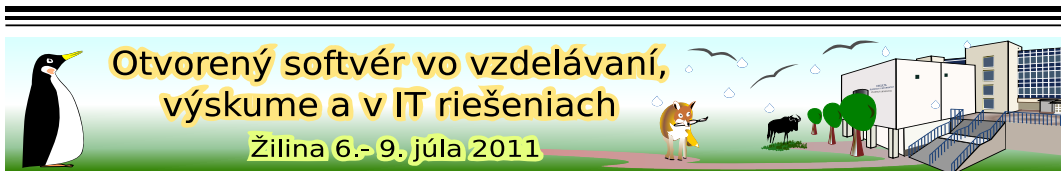
Literatúra

- [1] BLAŠKO, R.: *L^AT_EX nie je farba na maľovanie*. Zborník príspevkov medzinárodnej konferencie OSSConf 2010, 1.–4. júla 2010, Žilina, str. 43. ISBN 978-80-970457-0-8.
- [2] KOPKA, H. – DALY, P. W.: *L^AT_EX – podrobný príručka*. Computer Press, Brno, 2004. ISBN 80-722-6973-9.
- [3] PRCHAL, L. – SCHLESINGER, P.: *Prezentace v T_EXu*. Zpravodaj Československého sdružení uživatelů T_EXu, ročník 18 (2008), číslo 1–2, str. 56. ISSN 1211-6661.
- [4] STRÍŽ, P. – POLÁŠEK, M.: *Ukázky prezentací*. Zpravodaj Československého sdružení uživatelů T_EXu, ročník 18 (2008), číslo 1–2, str. 63. ISSN 1211-6661.
- [5] RYBIČKA, J.: *L^AT_EX pro začátečníky*. KONVOJ, Brno, 2003. ISBN 80-7302-049-1.

Kontaktná adresa

Aleš KOZUBÍK (RNDr., PhD.),

Katedra matematických metód FRI ŽU v Žiline, Univerzitná 8215/1,
010 26 Žilina, alesko@frcatel.fri.uniza.sk



DOI: 10.5300/2011-OSSConf/259

HISTÓRIA A SKÚSENOSTI Z POUŽÍVANIA FREE/OPEN-SOURCE TECHNOLÓGIÍ V SAMOSPRÁVE MESTA BANSKÁ BYSTRICA prípadová štúdia z osobného pohľadu – prvá časť, desktopové aplikácie

TUHÁRSKY, Peter, (SK)

Abstrakt. *Samospráva mesta Banská Bystrica je známa svojím zameraním na free/open-source (FOS) technológie; ich dlhodobým reálnym nasadením vo svojej informačnej infraštruktúre sa v kontexte samospráv na Slovensku dostala do pozície lídra na tomto poli IT. V prostredí verejnej správy, kde vo všeobecnosti stále hrozí prevládnutie politických kritérií nad odbornými, a kde tieto i finančné dôvody nevytvárajú priaznivé podmienky pre získanie a udržanie kvalitných odborníkov v akejkoľvek oblasti, môžeme hovoriť o zaujímavosti, možno aj úspechu, ktorého genézu predstavuje tento materiál z pohľadu autora. V tejto prvej časti sú zhrnuté dôvody, spôsob a niektoré skúsenosti z nasadenia FOS technológií v oblasti desktopových aplikácií.*

Kľúčové slová. *Samospráva, informačná infraštruktúra, free/open-source (FOS) technológie.*

HISTORY AND EXPERIENCE GAINED BY USING FREE/OPEN-SOURCE TECHNOLOGIES IN THE MUNICIPALITY OF BANSKÁ BYSTRICA a case study from the personal point of view – first part, desktop applications.

Abstract. *The municipality of the city of Banská Bystrica is well-known for its orientation towards the free/open-source (FOS) technologies; owing to the longstanding real employment of FOS in its information infrastructure this municipality became a leader in the context of other Slovak cities. In the environment of public administration, where the danger of prevailing political criteria over professional ones exists in general, and where these as well as financial reasons make acquisition of high-quality specialists hard, we can talk about curiosity, or eventually about success, the genesis of which is presented in this paper from the point of view of the author. In this first part, reasons, means and experience from the onset of FOS technologies in desktop applications are further summarized.*

Key words and phrases. *Municipality, information infrastructure, free/open-source technologies.*

1 Úvod

Banská Bystrica má približne 80 000 obyvateľov a ako samosprávny subjekt zriaďuje, zastrešuje alebo financuje desiatky menších subjektov v rámci rôznych právnych vzťahov, napríklad mestskú políciu, 14 základných a 27 materských škôl, zariadenia sociálnych služieb a krízovej intervencie atď. Rozpočet mesta je v súčasnosti približne 40 mil. € ročne, z čoho polovicu tvoria podielové dane od štátu, zvyšok tvoria miestne dane, administratívne poplatky a iné zisky mesta. Samosprávne funkcie mesta zastrešuje predovšetkým mestský úrad (MsÚ), ktorý samotný má v dlhodobom priemere vyše 200 zamestnancov.

V priebehu minulého desaťročia štát preniesol na mestá množstvo kompetencií, z čoho plynie veľmi rozsiahla agenda MsÚ – matrika a evidencia obyvateľstva, podnikateľských subjektov, výber miestnych daní a poplatkov, voľby, sociálna pomoc, školský úrad, stavebný úrad, dopravné a komunálne veci vrátane údržby cestnej siete, správa majetku mesta, životné prostredie – údržba zelene a nakladanie s odpadom, zasadnutia mestského zastupiteľstva a jeho komisií, atď. Celá agenda musí byť pokrytá z pohľadu registratúrneho, právneho, prevádzkového, ekonomického, informačného – a to nielen v zmysle technológie, ale aj povinného zverejňovania veľkého množstva informácií.

Ide teda o prostredie mimoriadne pestré, a vďaka neustálym legislatívnym a politickým zmenám aj premenlivé, a to tak z pohľadu agendy, personalistiky ako aj právnych vzťahov, a bezpochyby je preto z hľadiska IT výzvou. Oddelenie informatiky (OIT) dostalo do správy nielen IT na MsÚ, ale aj na materských školách, mestskej polícii a ďalších zariadeniach, čo predstavuje spolu približne 450 PC/notebookov s príslušenstvom, z čoho tretina je rozmiestnená po celom meste. Ďalšie povinnosti zahŕňajú zabezpečovanie akcií vo Veľkej sieni MsÚ, správu kopírokov, telekomunikačných služieb vrátane telefónov a mobilov, pretože slovami vtedajšieho prednostu, „nikto iný na úrade by to nezvládol“. Kiežby sa táto dôvera raz preniesla aj do ocenenia našej práce...

V súčasnosti OIT pozostáva zo šiestich ľudí – dvaja zabezpečujú technickú podporu, jeden zodpovedá za zverejňovanie informácií na webstránke, kolegyňa za informačný systém a mobilné telefóny, ja za sieť, sieťové služby a koncepcie, a vedúci za riadenie a administratívu. Hospodárime s rozpočtom cca 280 000 € ročne, z toho na potreby informatiky pripadá cca 200 000 €. Týmto sú snád' v hrubých rysoch ilustrované podmienky našej práce, aké sú mimochodom v mestských samosprávach na Slovensku bežné.

2 História IT

Ako je už asi zrejmé, nasadzovanie FOS technológií sa odohralo na MsÚ BB za špecifických okolností a súvislostí, ktoré boli ovplyvnené nielen vývojom technológií, legislatívy a požiadaviek používateľov, ale aj dostupnými financiami, personálnymi možnosťami a koncepciou OIT; museli sme vždy predvídať ďalší vývoj, usmerňovať ho a riešiť požiadavky s predstihom.

Krokom, ktorý asi najviac poznamenal vývoj, bolo zavedenie integrovaného Informačného Systému Samosprávy (ISS) firmy Cora LTJ (dnes Corageo) okolo roku 2000. Spočiatku išlo najmä o moduly na evidenciu obyvateľstva, dane a ekonomiku. Systém je založený na technológii Visual Foxpro, čiže pre svoj beh potrebuje Microsoft Windows na strane klienta, spúšťa sa zo súborového servera vo Windows sieti a databázu zabezpečuje Oracle. Hoci z koncepčného hľadiska sme voči ISS mali vždy vážne výhrady, systém si bezpochyby zaslúži rešpekt: pre porovnanie, centrálny register obyvateľstva ministerstva vnútra svojou zložitou približne zodpovedá jedinému z niekoľkých desiatok modulov ISS, pričom absolútne nemá význam porovnávať ceny obstarania týchto dvoch produktov (samozrejme aj s prihliadnutím na rozdiel v objeme databáz).

Aby sme pochopili podmienky a dôvody nášho postupu v nasadzovaní FOS technológií, priblížme si východiskový stav niekedy v roku 2002. V hlavnom krídle budovy MsÚ bola štruktúrovaná kabeľáž, ale sieťové prepínače boli inštalované len v nevyhnutnom rozsahu, a to s rýchlosťou 10MBit/s. Kabeľáž v druhom krídle sme museli riešiť v nasledujúcich rokoch. Na MsÚ bolo približne 40 tlačiarň a 110 počítačov, z toho cca 40 triedy i586 až i686 (100–300 MHz) so systémom Windows 95. Veľká časť slúžila len ako písacie stroje, a tak neboli pripojené k sieti. Ostatné počítače boli prevažne triedy Intel Celeron-PIII a AMD K7 Duron (600–1300 MHz) s Windows 98, ktoré už boli častejšie pripojené k sieti kvôli práci s ISS. Databáza Oracle8i bežala na serveri Windows NT4, ktorý zabezpečoval aj služby doménového radiča a súborového a tlačového servera. Doména sa správala nevyspytateľne a približne po roku vždy prestávala fungovať a vyžadovala reinštaláciu servera.

3 Prvé využitie Linuxu

Vedúci referátu informatiky, Ing. Hlavatý, nebol spokojný s filozofiou sieťových služieb Microsoftu a reálnymi skúsenosťami s ich prevádzkou a jej cenou. Preto so záujmom sledoval rozmach Linuxu, ktorý mu nebol celkom cudzí, pretože s UNIXovými systémami už predtým prišiel do kontaktu.

Prvým podnetom k reálnemu nasadeniu Linuxu v prostredí MsÚ bol dopyt po prístupe k Internetu a e-mailu medzi niekoľkými zamestnancami. Mesto malo vlastnú doménu a 64kbit/s, neskôr 128kbit/s konektivitu za cca 16000Sk mesačne. Aby tieto vymoženosti mohli využiť zamestnanci, bolo potrebné zriadiť firewall/proxy a poštový server. Linux (distribúcia Mandrake) túto funkciu spoľahlivo zabezpečil, hoci účelu muselo postačiť obyčajné staršie PC (300 MHz, 64 MB RAM). Používateľské účty boli lokálne, nemali žiadnu väzbu s doménou, čo obnášalo nutnosť viacnásobného zadávania používateľov a ich hesiel.

4 Moje začiatky na MsÚ

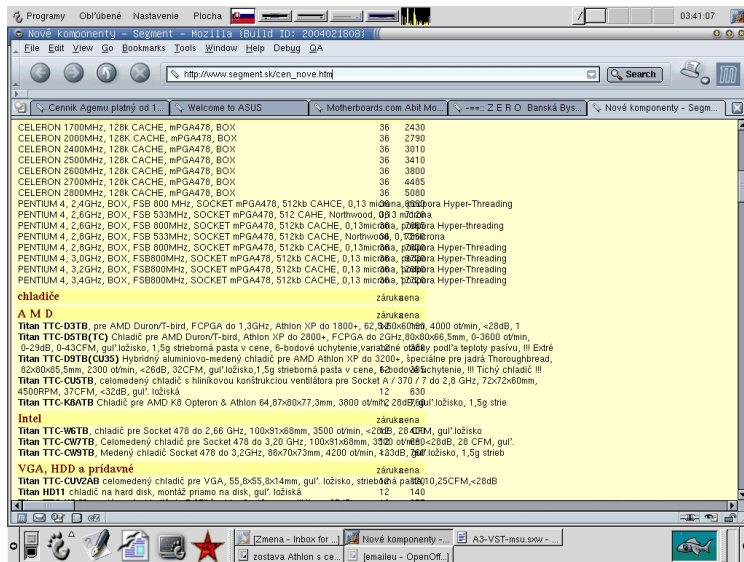
Môj nástup v roku 2003 sa niesol v duchu nevelkých skúseností no značného záujmu o sieť, Linux a súvisiace technológie. Na svoj pracovný desktop som si, pred zvedavými zrakmi

kolegov, nainštaloval popri Windows 98 aj Linux a takto som sa prehrýzal cez svoje nové pracovné úlohy, ktoré spočívali v starostlivosti o webstránku a ISS, príležitostnej technickej podpore a ďalších veciach. Mal som potrebné nadšenie a potrebnú dávku tvrdohlavosti, a dostal som sa do kolektívu, kde panoval záujem o nové technológie, pod vedením vedúceho s veľkým rozhľadom, skúsenosťami, pragmatickým prístupom a sympatiami k otvoreným štandardom. V takomto (vtedy 4-člennom) zložení môj príchod znamenal posilu a možnosť rozvoja takým spôsobom, na aký predtým nebol čas a sily. Stal som sa nadšeným pokusným králikom pre FOS technológie a nakoľko som používal GNU/Linux a iné FOS programy doma i pri bežnej práci, mohol som spolu s kolegami „v priamom prenose“ sledovať vývoj jeho možností a použiteľnosť jednotlivých programov pre konkrétne úlohy. Takto boli vlastne dlhodobo naživo testované aj OpenOffice.org a Mozilla Suite. Tento výber nebol náhodný – nakoľko som už mal nejaké skúsenosti s heterogénnym prostredím, volil som vždy, keď sa dalo, multiplatformné programy, a tento dobrý zvyk som priniesol so sebou na MsÚ a s pomocou vedúceho ho presadil do bežnej praxe. Od začiatku som sa zapájal do systémov hlásenia chýb a získal tak realistickú predstavu o rýchlosti resp. pomalosti vývojového cyklu. Toto nám umožnilo v kritickom čase dobre odhadnúť zrelosť jednotlivých programov pre plošné nasadenie.

Po otestovaní viacerých distribúcií Linuxu som dal prednosť distribúcii Debian. Popri nízkych nárokoch na hardvér mala dobrú reputáciu aj z pohľadu spoľahlivosti, bezpečnosti a FOS filozofie. Hoci z pohľadu desktopu bola spomedzi všetkých najviac pozadu a „nič nefungovalo samo od seba“, zbral som to ako výzvu a rozhodol som sa, že najlepšou investíciou času a síl bude pokúsiť sa dobre zvládnuť jednu distribúciu a všestranne ju využiť, než učiť sa rôzne nuansy rôznych distribúcií pre desktop a pre server. Napokon, v prípade siete tenkých klientov by som tak či tak musel tieto dva odlišné svety spojiť, a na túto možnosť som chcel byť pripravený.

Moja premiéra v serverovej oblasti prišla čoskoro, popri plnení bežných povinností a zdolávaní nástrah Linuxového desktopu. Dostal som priestor, vyriešiť sieťový problém a naučiť sa pritom viac o sieti i Linuxe. Vzhľadom k pôvodne malému počtu počítačov pripojených do siete sa IP adresy pridelovali len lokálne/staticky, čo sa postupne ukázalo ako neprehľadné, neefektívne a navyše výrazne limitujúce pri zamýšľanom zavádzaní nových sieťových služieb. Na nákup servera neboli žiadne financie, ale v sklade ležalo 5 vyradených a ináč nepoužiteľných HP PA/RISC serverov. Distribúcia Debian Woody našťastie podporovala túto architektúru, takže sa mi 2 z nich podarilo spojazdniť do úlohy DHCP/DNS serverov v redundantnom zapojení. Napriek tomu, že tieto stroje mali menší výkon než najslabšie funkčné PC na úrade, svoje úlohy plnili spoľahlivo a s ľahkosťou (zátťaž CPU 1%). Tento prvý malý úspech bol pre všetkých potrebným povzbudením, hoci so sebou niesol potrebu fyzicky obísť a nastaviť všetky PC. Odmenou bola dovtedy nevídaná flexibilita siete a silný nástroj k jej reformám. Pre zaujímavosť, servre boli konfigurované s výhľadom, že raz budú zastrešovať bootovaciu službu pre sieť tenkých klientov, o ktorej sme uvažovali ako o cieľovom stave.

Na jednom z týchto dvoch strojov sme 30. 1. 2004 spustili jednoduchú statickú stránku Intranet, ktorá slúžila najmä na informovanie zamestnancov o servisných odstávkach, ktoré boli kvôli ISS pomerne časté. Nastavovali sme ju v prehliadači ako predvolenú stránku a postupne sa stala oficiálnym a povinným miestom vnútorných oznamov, akousi nástenkou pre zamestnancov, a rázcestím pre čoraz viacero ďalších sieťových služieb a informácií. Bočným efektom bolo, že bol takto zabezpečený dohľad nad funkčnosťou servera ako takého, ktorý sa od začiatku javil ako menej spoľahlivý. Po cca roku nepretržitej prevádzky tento staručký stroj začal vykazovať nestabilitu a bol definitívne vyradený. Nahradený bol najskôr starým PC (Pentium 133 MHz) a neskôr väčším (PentiumII 133 MHz), ktorý začal slúžiť aj ako úložisko pravidelných záloh zo súborového servera.



Obr. 1: Február 2004. Kostřbaté fonty, žiadny antialiasing, nesúrodý dizajn, problémy s diakritikou, klávesnicou, myšou. Prvý reálne používaný linuxový desktop na MsÚ BB (môj) vyžadoval sebazaprenie a tvrdohlavosť. Systém: Debian Woody, GNOME 1.2. Červená hviezda s drakom bola štandardnou ikonou prehliadača Mozilla. Stránka je rozhodená preto, lebo išlo o export z Excelu do HTML

5 Desktop

Doprajeme si malé odbočenie od siete a zameriame sa na desktopy, pretože mnoho ďalších krokov v infraštruktúre bolo poznačených snahou o koncepčné riešenie niektorých klasických desktopových problémov a efektívnejšie zvládnutie ohromného nárastu počtu zamestnancov používajúcich PC. Išlo o nárast v niekoľkých rovinách – nielen kvantitatívny, ale aj kvalitatívny.

V rokoch 2004-2008 sa podarilo presadiť do rozpočtu prostriedky na nákup približne 40 nových PC ročne, ktoré boli využité na výmenu starých PC a na vybavenie zamestnancov, ktorí dovtedy žiadne PC nemali. Naším úmyslom bolo pokračovať týmto tempom a zabezpečiť tak kontinuálnu obnovu techniky v približne 5-ročnom cykle. Ustanovili sme štandard: každý nový počítač bude pripojený k lokálnej sieti a jeho používateľ získa prístup na Internet a svoju vlastnú e-mailovú schránku. Tieto veci sa dnes zdajú triviálne, ale treba si uvedomiť, že dovtedy len časť počítačov bola vôbec pripojená k lokálnej sieti a len hŕstka najpokročilejších zamestnancov mala a aj používala prístup k Internetu a vlastnú e-mailovú schránku.

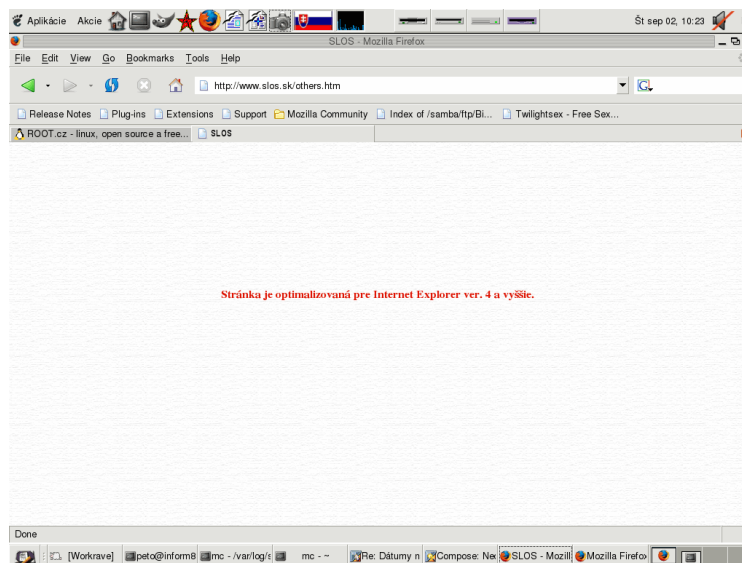
Pripájanie počítačov k Internetu bolo našou iniciatívou, nikto to od nás nepožadoval, niektorí e-mailovú schránku ani nechceli a dostali ju „príkazom“. Naše rozhodnutie bolo z dlhodobého pohľadu logické a koncepčné, hoci sme si tým vedome skomplikovali správu a urýchlili potrebu ucelene riešiť bezpečnosť celej siete, ak sme sa chceli vyhnúť katastrofic-kým epidémiám a výpadkom.

Samotný manažment rozmiestňovania toľkých počítačov bol náročný – odohrávalo sa za plnej prevádzky, popri obvyklých pracovných povinnostiach, pričom bolo spomaľované prenosom lokálnych dát zo starých vyradovaných počítačov. Popri tom bolo treba systematicky rozširovať aktívne sieťové prvky a komplexne riešiť celú serverovú infraštruktúru tak, aby nielen zvládala tento nárast, ale aby sa veci diali aj koncepčne. Naším cieľom bolo, aby sa vývoj nastavil smerom ktorý bude nadčasový a dlhodobo udržateľný, a aby nám tieto technológie, podľa možnosti, aj ušetril námahu pri správe. Samozrejme, bolo nutné aj zabezpečiť technickú podporu pre používateľov, ktorí boli často počítačovo celkom negramotní, a to všetko za situácie minimálnych finančných prostriedkov. Zvládnutie tejto výzvy považujem za najväčší úspech nášho tímu.

Nasadenie obligátnej trojice programov OpenOffice.org, Mozilla Thunderbird a Mozilla Firefox sa v dnešnej dobe javí ako bežné, možno až konzervatívne. V čase, keď sme tieto programy nasadzovali my, boli pomerne neznáme, ešte sa o nich len začínalo písať a hovoriť. Ich voľba nebola pre nás vecou prevzatia nejakého trendu alebo imidžu (pretože vtedy neexistoval), ale pragmatických kritérií, predovšetkým otvorenosti a multiplatformnosti, a dôkladného otestovania v dlhodobom používaní. Aby sme čo najviac podporili proces nasadenia týchto programov, vyhľadali sme a prostredníctvom Intranetu sprístupnili zamestnancom množstvo dokumentácie a návodov v elektronickej podobe a zakúpili aj niekoľko kníh.

5.1 Mozilla

Vzhľadom k zámeru vybaviť každého zamestnanca prístupom k Internetu, jedným z prvých našich rozhodnutí v smere k FOS bolo nasadenie dvojice Mozilla Thunderbird + Firefox. Prakticky všetok škodlivý kód totiž využíval zraniteľnosti Internet Explorera a Outlooku, takže týmto sme zabezpečili výrazné zníženie rizika. Okrem toho sme lokálne nasadzovali NOD32 antivírus a postupne robili bezpečnostné opatrenia aj na serveroch, k čomu sa vrátim.



Obr. 2: September 2004. Aj takto sa webové stránky správali v čase začiatkov Mozilla Firefox. Firmu som kontaktoval a stránku opravil. Na okraj: prostredie GNOME 2 prinieslo podstatné zlepšenie dizajnu i ergonomie desktopu

Firefox bol logickou voľbou aj preto, že v danom čase neexistoval iný prehliadač, ktorý by bol FOS, bezplatný, multiplatformný a k tomu veľmi rýchly aj na slabšom hw (áno, vtedajšie verzie boli skutočne také! ;-)

Presadiť Thunderbird namiesto Outlook Express nebolo až také ťažké, pretože bol o poznanie stabilnejší. Oproti tomu nasadenie Firefoxu bolo zložitejšie. Treba si uvedomiť, že sme ho plošne zaviedli v roku 2004, akonáhle začali byť verzie 0.9.x dostatočne stabilné; v tom čase ho používalo možno menej než 1% návštevníkov internetových stránok. Absolútnym monopolom bol vtedy Internet Explorer 4 a 5 a viacerí tvorcovia prispôbovali stránky výlučne tomuto nešťastnému prehliadaču, ktorý ignoroval a krivil webové štandardy. Preto niektoré stránky pod Firefoxom nefungovali a takéto incidenty sme potom museli riešiť, nielen v smere k našim používateľom, ale aj k webmasterom. Vybrať si slobodne webový prehliadač dnes považujeme za samozrejmé, ale nestalo sa to samo od seba; na začiatku musel byť sústredený a neúnavný tlak „zdola“, na ktorého vytváraní sme sa spolupodieľali v čase, keď takáto snaha mohla vyzerat' celkom márna. Keď vezmeme do úvahy veľmi malý podiel Firefoxu na Internete a malú rozšírenosť internetového pripojenia v tej dobe, naše rozhodnutie mohlo v tom čase dosť viditeľne ovplyvniť štatistiky na regionálnej úrovni. Dnes už drvivá väčšina stránok počíta s rozmanitými prehliadačmi a dodržiava štandardy, hoci nájdu sa aj hanebné výnimky, akou je napríklad mapová časť katasterportálu, ktorá v rozpore so zákonom ešte aj v roku 2011 vyžaduje Windows a Internet Explorer.

5.2 OpenOffice.org

Ďalším rozhodnutím, ktoré kvôli nasadzovaniu množstva nových počítačov neznieslo odklad, bola voľba kancelárskeho balíku. Dovtedy sa na MsÚ používalo niekoľko inštalácií Microsoft Office 97 a plošne predovšetkým PCSuite od Software602, čo bol nenáročný kancelársky balík pre Windows, ktorého zásadnou nevýhodou bol neštandardný formát súborov. Do istej miery dokázal súbory exportovať aj do formátov MSOffice, ale väčšinu dokumentov vytvorili používatelia v natívnom formáte.

Krátkodobým riešením mohol byť nákup licencií na novšiu verziu PCSuite, lenže tento produkt už nemal perspektívu ďalšieho vývoja. Keďže sme sledovali napredovanie OpenOffice.org (OOo) a jeho odnoží, bolo nám jasné, že Software602 by nemohol dlhodobo tomuto vývoju konkurovať. Ďalšia možnosť – celoplošný nákup MSOffice – bola finančne absolútne nereálna. MSOffice sa nám zdal hrozbou aj kvôli predpokladanej nutnosti pravidelných nákupov nových verzií spolu s hardvérom a operačným systémom, a to ani nie tak kvôli pokrokom vo funkcionalite, ako skôr kvôli tlaku výrobcu prostredníctvom nekompatibility formátov súborov a požiadavkám ohľadom operačného systému. Takto vynaložené prostriedky by nám pripadali absolútne neefektívne vzhľadom k potrebám zamestnancov MsÚ. Inštalácia MSOffice bez zakúpených licencií neprichádzala do úvahy.

Okrem toho, našou požiadavkou v tom čase už bola multiplatformnosť a štandardný formát dokumentov. Horúcim kandidátom bol preto OOo. Prinášal aj výhodu v podobe odolnosti voči červom zneužívajúcim slabiny MSOffice a Visual Basic. Starosti nám robili niektoré jeho chyby, vyššia náročnosť na hardvér, ktorá vymedzovala jeho inštaláciu len na nové kúpené PC, a nemožnosť otvárať dokumenty PCSuite. Chvíľkovú nádej vzbudilo, keď Software602 uviedol na trh kancelársky balík 602Office založený na OOo. Po konzultáciách s výrobcom nám však začalo byť jasné, že importný filter pre staré dokumenty PCSuite nie je v pláne, a že multiplatformnosť tiež nie je dlhodobo zaručená. 602Office pre nás nepredstavoval žiadnu reálnu výhodu oproti bezplatnému OOo.

Zvíťazil teda OOo vo verzii 1.1 a začali sme ho plošne inštalovať. Zamestnanci, ktorí predtým pracovali s PCSuite, však mali k dispozícii aj tento program, aby mohli otvoriť staršie dokumenty a podľa potreby ich konvertovať. Na tvorbu nových dokumentov už mali používať OOo, ale niektorí aj tak ďalej používali PCSuite, až kým sme ho definitívne neodstavili. Poslednou možnosťou bolo napísať dokument nanovo (čo zodpovedalo nedávnej úrovni práce na písacích strojoch).

OOo verzie 1.1.5 priniesol podporu pre OpenDocument (ODF). Testovali sme vstavaný nástroj OOo na automatickú konverziu z dokumentov MSOffice, ale bol príliš nestabilný a chybový, takže sme upustili od zámeru, previesť naraz všetky úradné dokumenty do ODF. Preto niektorí zamestnanci narazili na chybu diakritiky v prastarých dokumentoch, ktoré OOo pri ukladaní do formátu Word 6 poškodil. Ponechanie dokumentov v pôvodnom stave však malo aj výhodu – dalo sa predpokladať, že pri sústavne sa zlepšujúcich importných filtroch OOo bude vlastne pre zamestnancov výhodou, ak sa s ich starými dokumentami bude trápiť novšia verzia OOo, než tá, ktorá bola k dispozícii v čase snahy o hromadnú

konverziu – šance na bezchybný import sa naozaj s pribúdajúcim časom zlepšovali. Za pomoci prednostu sme presadili nariadenie, vďaka ktorému sa ODF stal povinným pre MsÚ; hoci ďalší prednostovia sa k nemu stavali rôzne, platí dodnes.

Nasadenie OOo bolo asi najväčším bojom spomedzi všetkých FOS programov na MsÚ. Jedným problémom bol návyk používateľov na PCSuite alebo MSOffice, ďalším boli viaceré chybičky, o ktorých síce väčšina zamestnancov nevedela, ale pre tých niekoľkých pokročilejších, ktorí ich objavili, predstavovali problém. Aj z týchto dôvodov sme skúmali možnosti získania platenej podpory, napríklad u firmy blue.point Solutions, ale žiaľ vzhľadom na rozpočet sme tento projekt nemohli rozbehnúť. Ďalšiu skupinu problémov vytvorili pracovníčky referátu daní, ktoré namiesto v ISS robili časť svojej agendy pomocou excelovských tabuliek, a vzhľadom k mierne odlišnému ovládaniu, nedokonale podpore súborov MSOffice, a funkčnému zaostávaniu za Excelom, videli v OOo svojho nepriateľa. Toto sa mohlo javiť ako neúspech OOo, ale v skutočnosti išlo o nepochopenie, že na prácu s databázou a generovanie formulárov slúži v skutočnosti ISS, a že Excel nie je náhrada integrovaného informačného systému. Jediným správnym riešením nedostatkov ISS mal byť tlak na dodávateľa ISS, nie jeho suplovanie Excelom.

Mali sme aj pozitívnu skúsenosť so skúsenými zamestnancami na útvare kontroly. Keď dostali nové PC s OOo a nevedeli si s niektorými vecami rady, samozrejme sme pomohli, a tiež upozornili na dostupnosť návodov a kníh. Jednu knihu si od nás hneď požičali, našťudovali, po týždni vrátili a ďalej už pracovali prakticky bez akýchkoľvek ďalších otázok. Pre nás to bolo veľmi povzbudivé, pretože vzhľadom na malé personálne možnosti sme nemohli každého zaškoľovať, a pre nedostatok financií sa nám nepodarilo presadiť celoplošné školenie.

Najväčším problémom pri používaní OOo boli samozrejme súbory vo formáte MSOffice, ktoré nám neúnavne posielali štátne orgány a inštitúcie. Nič sa v tom nezmenilo ani po roku 2006, kedy bol ODF schválený ako norma ISO 26300, a kedy vstúpil do platnosti aj zákon č.275/2006 o informačných systémoch verejnej správy, ktorý jasne určil, že používané štandardy musia byť otvorené a technologicky neutrálne. Spolu s výnosom MDPaT č.1706/2006 o štandardoch bola legislatíva jasne na našej strane. Zamestnancom sme vysvetľovali, že kým používajú ODF, sú v práve a nikto im nemôže nanucovať neštandardné formáty. Lenže v rámci obrovskej štátnej zakázky všetky ministerstvá dostali MSOffice a naši mestskí úradníci sa voči nim stavali vždy do submisívnej pozície, hoci v skutočnosti ministerstvá nie sú nijako nadradené mestskej samospráve. Situácia sa pozvoľne začala zlepšovať až okolo roku 2010, keď ministerstvo financií zintenzívnilo kontroly zamerané na dodržiavanie štandardov. Po niekoľko rokov sme však boli nútení v niekoľkých prípadoch zakúpiť licencie MSOffice, a to výlučne z dôvodu komunikácie v jeho uzavretom neštandardnom formáte s vonkajšími subjektami.

5.3 Linux na desktope

Naším pôvodným zámerom bolo doviest' postupne celú informačnú infraštruktúru na technológiu tenkých klientov a linuxových serverov. Nemali sme financie na potrebné serverové

vybavenie, ani na profesionálne zariadenia pre tenkých klientov, a obávali sme sa problémov s behom ISS v prostredí Wine, ale všetky kroky sme robili so zreteľom na tento náš cieľ. Preto keď sme v roku 2005 organizovali verejnú súťaž na nákup počítačov a tlačiarňí, kritériá sme stanovili tak, že dodaným operačným systémom nemusel byť nutne Windows XP, ale mohol ním byť aj Linux v prípade, že bude dodržaná kompatibilita s Visual Foxpro 7. Počítače zároveň museli byť schopné bootovať cez sieť, a spolu s tlačiarňami museli byť plne kompatibilné s Linuxom, a to aj bez použitia uzavretých ovládačov. Toto kritérium museli súťažiaci preukázať pomocou ľubovoľnej distribúcie Linuxu. Vtedajšie modely procesora Intel Celeron-P4 sa pri linuxovom multitaskingu správali rozpačito – výkonovo nevyrovnane, často na úrovni AMD Duronov s polovičným taktom, čo sme pripísali malej vyrovnávacej pamäti procesora Celeron-P4 (architektúra netburst). Jednou zo súťažných podmienok bola preto aj veľkosť vyrovnávacej pamäte. Ďalšou podmienkou bolo, že operačný systém musel podporovať prácu v doméne, čím sme zo súťaže vyradili problematický Windows 98 s končiacou podporou.

Bol to v podstate pokus o prechod na Linuxové terminály s medzikrokom – ak sa nájde dodávateľ, ktorý spojzdni ISS pod Linux/WINE, bez problémov vyhrá súťaž vďaka cenovej výhode oproti zostavám s Windows. V prípade, že by migrácia z nejakého dôvodu zlyhala, tieto PC s definovanými výkonovými parametrami a ultranízkou cenou (ekvivalentnou tenkým klientom) by sa dali poľahky preinštalovať na bežné PC s Windows XP. Ak by sa nenašiel dodávateľ schopný sprevádzkovať Linux, a súťaž by teda vyhrala zostava s Windows XP, povinná kompatibilita zostáv s Linuxom zaručovala, že by sme migráciu na Linux mohli urobiť podľa uváženia niekedy neskôr. Kompatibilita hw s Linuxom bola v tom období vo všeobecnosti slabšia a pre niektoré modely hw neboli pre Linux dostupné ovládače; týmto sme zároveň ako zákazníci aspoň trochu prispeli k tlaku na výrobcov.

Súťaž vyhrala žilinská firma ESMO, ktorá dokázala podmienky splniť a vyhrala súťaž na základe najnižšej ceny. ISS pod Wine bežalo dokonca rýchlejšie než natívne pod Windows, zrejme vďaka efektívnejšej sieťovej vrstve v Linuxe. Keďže na terminálový server sme stále nemali peniaze, nasadili sme takto približne 50 plných linuxových desktopov založených na distribúcii Debian Sarge. Inštalácie prebiehali veľmi rýchlo jednoduchým klonovaním, desktopové prostredie bolo IceWM kvôli jednoduchosti, prehľadnosti, malej hw náročnosti, a „robustnosti“ - prostredie sa ovládalo konfiguračným súborom a používateľ prakticky nemal možnosť ho znefunkčnit'. Programátori Corageo nám vyšli v ústrety vo forme informácií potrebných k automatizovanej inštalácii knižníc VFP potrebných k chodu ISS.

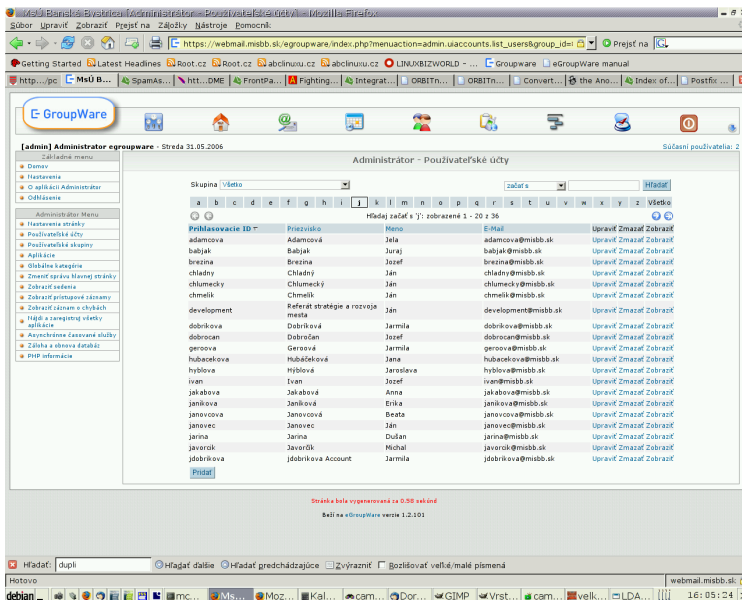
Analyzovali sme, aké nároky môžu byť na PC ako pracovný nástroj a vytipovali sme programy, ktoré tieto nároky splnia. S výhodou sme využili naše doterajšie pragmatické rozhodnutia v podobe multiplatformných aplikácií, takže používatelia dostali OOo, FF, TB ktoré už poznali z Windows, čo značne zmiernilo problémy s prechodom – v podstate mali k dispozícii to isté ako všetci ostatní, vrátane právneho systému ASPI (pomocou Wine) a keďže išlo o nové počítače s dobrým výkonom a na svoju dobu nadpriemernou ergonomiou (podložka zápästia pre klávesnicu i myš, 100Hz monitor), nemuseli mať pocit, že dostávajú niečo „menej hodnotné“. Priečinok dokumentov používateľa sa pomocou smbmount pripájal

z toho istého servera ako u staníc s Windows – snažili sme sa udržať naše sieťové prostredie čo najviac homogénne bez ohľadu na to, že sme vniesli heterogénnosť do operačných systémov na klientských PC. Používateľ mohol v prípade problémov kliknutím na ikonu spustiť VNC a mohli sme mu pomôcť vzdialene.

Správa týchto strojov bola efektívna, ich aktualizácia bola automatizovaná pomocou skriptov. Problém nastal začiatkom roku 2006, keď dodávateľ ISS (fa Corageo) zmenil verziu knižníc VisualFoxpro zo 7 na 8. Tento krok mal zlepšiť tlač, v skutočnosti však úplne prestala fungovať pod Windows 98 aj pod Wine, a tak jediným systémom, kde aspoň s problémami fungovala, bol Windows XP. Pokusy vyriešiť tento problém komunikáciou s dodávateľom zlyhali, a riešenia na úrovni Wine v spolupráci s firmami IFNE a Codeweavers, napriek ich ochote, viedli len k čiastkovým úspechom a odhalienu zložitosti problému. V tejto situácii, bez záruky rýchleho úplného vyriešenia, sme napokon v záujme rýchleho obnovenia použiteľnosti systému museli nakúpiť desiatky licencií na Windows XP a rýchlo premigrovať všetky systémy na úrade na ten „jediný správny“. Celoplošné nasadenie Linuxu teda zmarila predovšetkým závislosť kritickej aplikácie od Windows. Ironiou je, že firma Corageo napokon obnovila možnosť tlače pomocou knižníc VFP7, pretože problémy sa vyskytli u viacerých miest. V tom čase sme však boli už príliš zahľtení inou prácou, ktorá dovtedy musela počkať kvôli riešeniu tejto akútnej krízy, okrem toho licencie na Windows XP už boli zakúpené, takže na migráciu späť na Linux sme už nemali sily, ani reálne dôvody.

Tento prípad poukázal na veľké rozdiely medzi našou koncepciou a koncepciou dodávateľa (Corageo), ktorý sa silno držal vybranej architektúry a aplikačného modelu a z toho samozrejme vyplývali tlaky tretích strán – či už výrobcu operačného systému (Microsoft) alebo databázy (Oracle). Dodávateľ, aby bol „krytý“, sa správal podľa vyhlásení tretích strán o „podpore“ a tým prenášal ich tlak na nás. Takže keď Microsoft ukončil podporu pre VFP7, dodávateľ vykonal prechod na VFP8, aj za cenu narušenia tlače pod Windows 98 a Linuxom. Nedokázali sme mu vysvetliť, že radšej by sme venovali pol milióna SKK na vývoj ISS, než na licencie Windows XP, z ktorých oni nič nemajú. Podobne sme boli tlačení do upgradu databázy Oracle na verziu 10. Samosprávy celkovo nepatria medzi tých klientov, ktorí by dokázali poľahky pokrývať takéto požiadavky, či už po stránke finančnej alebo personálnej. Určité napredovanie technológií sa pochopiť dá, ale z pragmatického hľadiska, malo by priniesť konečným používateľom aspoň nejaké výhody, nielen výdavky a nadbytočnú prácu.

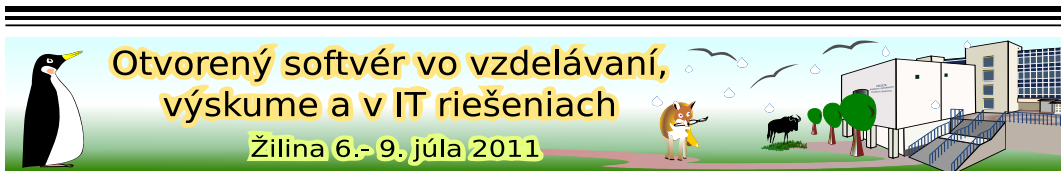
Hoci tento pokus o migráciu na Linux bol nakoniec neúspešný, získali sme veľa skúseností a predstavu o slabších i silných stránkach linuxového desktopu a našich používateľoch. Najčastejšími problémami boli periférie – najmä tlačiarne so slabou podporou pre Linux. Preto je pragmatickým prístupom, požadovať kompatibilitu s Linuxom aj v čase, keď sa prevádzka pod Linuxom bezprostredne neplánuje. Každé rozhodnutie v informatike má tendenciu dlhodobo ovplyvňovať naše možnosti ešte aj v prekvapivo vzdialenej budúcnosti (5 rokov je v informatike veľmi dlhá doba, predstavuje posun o dve generácie sw i hw), preto treba rozhodnutia robiť tak, aby sme si aj pre budúcnosť zachovali slobodu rozhodovania. Nedokážeme totiž predvídať vývoj do tak vzdialenej budúcnosti a dnešná ľahkovážnosť sa raz môže vypomstiť.



Obr. 3: Rok 2006, jedna z mála zachovaných snímok pracovného prostredia IceWM v podobe nasadenia na linuxových desktopoch MsÚ. Vľavo dole vidno tlačítka na spustenie programov a panel úloh (zvyčajne poloprázdny), a vpravo indikátor záťaže systému, čas a tlačítko na skrytie panelu. Kliknutím na tlačítko debian sa otváralo menu; neskôr bolo nahradené zlatistým trojuholníkom a pribudol indikátor národnej klávesnice. Otvorený je tu Firefox zrejme verzie 1.5.x s webovou aplikáciou EGroupware. Snímka pochádza z môjho PC – používal som totožné prostredie ako ostatní, aby som včas odhalil nedostatky a pomocou aktualizáčnych skriptov ich opravil naraz u všetkých

Pokračovanie Článok pokračuje druhou časťou, ktorá je venovaná serverovej časti infraštruktúry.

Peter Tuhársky (Mgr.),
peter.tuharsky@banskabystrica.sk



DOI: 10.5300/2011-OSSConf/271

HISTÓRIA A SKÚSENOSTI Z POUŽÍVANIA FREE/OPEN-SOURCE TECHNOLÓGIÍ V SAMOSPRÁVE MESTA BANSKÁ BYSTRICA prípadová štúdia z osobného pohľadu – druhá časť, serverové aplikácie

TUHÁRSKY, Peter, (SK)

Abstrakt. *Samospráva mesta Banská Bystrica je známa svojím zameraním na free/open-source (FOS) technológie; ich dlhodobým reálnym nasadením vo svojej informačnej infraštruktúre sa v kontexte samospráv na Slovensku dostala do pozície lídra na tomto poli IT. V prostredí verejnej správy, kde vo všeobecnosti stále hrozí prevládnutie politických kritérií nad odbornými, a kde tieto i finančné dôvody nevytvárajú priaznivé podmienky pre získanie a udržanie kvalitných odborníkov v akejkoľvek oblasti, môžeme hovoriť o zaujímavosti, možno aj úspechu, ktorého genézu predstavuje tento materiál z pohľadu autora. V tejto prvej časti sú zhrnuté dôvody, spôsob a niektoré skúsenosti z nasadenia FOS technológií v oblasti desktopových aplikácií.*

Kľúčové slová. *Samospráva, informačná infraštruktúra, free/open-source (FOS) technológie.*

HISTORY AND EXPERIENCE GAINED BY USING FREE/OPEN-SOURCE TECHNOLOGIES IN THE MUNICIPALITY OF BANSKÁ BYSTRICA a case study from the personal point of view – second part, server applications.

Abstract. *The municipality of the city of Banská Bystrica is well-known for its orientation towards the free/open-source (FOS) technologies; owing to the longstanding real employment of FOS in its information infrastructure this municipality became a leader in the context of other Slovak cities. In the environment of public administration, where the danger of prevailing political criteria over professional ones exists in general, and where these as well as financial reasons make acquisition of high-quality specialists hard, we can talk about curiosity, or eventually about success, the genesis of which is presented in this paper from the point of view of the author. In this first part, reasons, means and experience from the onset of FOS technologies in desktop applications are further summarized.*

Key words and phrases. *Municipality, information infrastructure, free/open-source technologies.*

6 Serverová infraštruktúra

V prvej časti príspevku o histórii používania FOS technológií v samospráve mesta Banská Bystrica sme sa venovali problematike desktopov. V druhej časti sa pozrieme na serverovú infraštruktúru, ktorá tvorila dôležitú oporu rozširovaniu vybavenia zamestnancov MsÚ nielen o samotné počítače, ale aj o ponuku sieťových služieb.

6.1 Súborový server

Pri výmene starých počítačov, ako aj pri servise, bolo oddávna nutné presúvať dokumenty používateľov, alebo ich zachraňovať z poškodeného disku, čo nie vždy bolo úspešné. Aby sa presun robil len raz, a aby sa predišlo ďalším stratám údajov, zriadili sme približne koncom roku 2004 súborový server, ktorý poskytoval používateľom domovský priečinok so všetkými ich dokumentami. Pri poruche PC už stačilo, aby sa používateľ prihlásil na inom PC svojím menom a heslom, a všetky dokumenty mal poruke. Experimentovali sme aj s mobilnými profilmi Windows, čo by znamenalo, že používateľ by mal po prihlásení na ktoromkoľvek PC k dispozícii svoju pracovnú plochu so všetkými nastaveniami, ale kvôli problémom s právami a synchronizáciou sme od tohto zámeru upustili, a neskôr sa oň už nepokúšali. Postupným vylepšovaním inštalácie Windows sme totiž dospeli k stavu, keď už po prvom prihlásení na danom stroji má používateľ k dispozícii všetko potrebné a stačí už len nastaviť Thunderbird a niekoľko drobností. Správa sieťových profilov by zrejme bola skôr komplikáciou než prínosom.

Na tento súborový server sme popri ďalších veciach presunuli aj spustiteľnú časť ISS, čím sme značne odľahčili hlavný server s Windows NT4, zabezpečujúci Windows doménu a databázu Oracle. Na strane dodávateľa ISS tento krok nevyvolal nadšenie a priniesol aj problémy, pretože v tom čase si zjednodušoval prácu s aktualizáciami tak, že sa spúšťali priamo na serveri – dodávateľ napevno predpokladal, že jediný stroj zabezpečuje funkciu súborového i databázového servera a že na ňom beží Windows. Tieto podmienky už u nás neplatili. ISS ako aplikácia klient-server samozrejme dovoľovala riešiť aktualizáciu aj elegantnejšie, a výrobca postupne prešiel na nový štýl aktualizácií, ktoré sa dajú spustiť z ľubovoľného klienta, čím sa problém vyriešil.

Vzhľadom k chýbajúcim financiám, súborovým serverom bol opäť len značne starý PC (PII/166 MHz, neskôr AMD Duron 700 MHz) so 120 GB IDE diskom, ktorý sa zálohoval na iný IDE disk, na ďalší server (Intranet) fyzicky umiestnený v inej miestnosti, a na pásky. Operačným systémom bol samozrejme Debian GNU/Linux, najskôr vo verzii Woody, potom Sarge. Stroj plnil svoju funkciu vynikajúco, ale nárast požiadaviek systému samotného, ako aj stúpajúci podiel používateľov ISS, vyžadoval rozširovanie operačnej pamäte postupne „až“ na 384MB, pretože každý pripojený používateľ znamenal z pohľadu Samby samostatný proces smbD s pamäťovými nárokmi. Lenže ani to už čoskoro nestačilo; server síce dokázal obslúžiť všetky požiadavky, ale nebolo to „dost' rýchlo“ (najmä v rannej špičke), pretože knižnice Visual Foxpro na klientoch boli pri spustení ISS mimoriadne náročné na to, aby

dostali všetky potrebné súbory v priebehu milisekúnd, ináč vznikali podivné chyby („error 1104“). Podieľal sa na tom aj fakt, že súbory ISS mali spolu niekoľko stoviek MB a nezmestili sa do vyrovnávacej diskovej pamäte Linuxu v malej RAM tohto stroja, takže museli byť neustále čítané z disku.

Preto bol neskôr nasadený vyslúžilý databázový server (AMD Athlon 3000+ a SCSI RAID5) ktorý mal 2GB RAM a gigabitové sieťové rozhranie pripojené na gigabitový port v koreňovom prepínači, čím sa podarilo odstrániť všetky „úzke hrdlá“. Neskôr ho nahradil stroj 2x AMD Opteron so serverovými komponentami, vrátane SCSI diskového poľa RAID5 pod taktovkou 64-bit PCI-X radiča Adaptec.

6.2 Doména

Úspech s DHCP/DNS nás na začiatku povzbudil k tomu, aby sme nespoľahlivosť NT4 domény riešili migráciou doménového servera na Linux s použitím služieb Samba a LDAP. Na túto úlohu sa podujal vedúci, Ing. Hlavatý, ktorý mal vďaka riešeniu problémov už značné znalosti Windows domény. Kvôli jeho pracovnej vyťažnosti, ako aj technologickým problémom, bol tento prechod veľmi namáhavý a jeho príprava trvala približne rok. Samba bola práve v období prerodu z vetvy 2.x na 3.x, a tak aj medzi malými verziami boli veľké rozdiely, sprevádzané množstvom chýb a nejednotnou dokumentáciou. Tieto veci sa podstatne zlepšili približne do 2 rokov po našej migrácii, čo dokazuje, že sme sa vtedy pohybovali na špičke dostupnej technológie. Pretrpeli sme tak jej „detské problémy“, ktorými sa už nemuseli zaoberať tí, ktorí túto technológiu prevzali až keď sa stala hlavným prúdom.

Problémy nám spôsobovalo aj samotné „vytiahnutie“ doménových údajov z Windows NT4 servera. Zistili sme, že tento krok nie je taký jednoduchý; za normálnych okolností sa Windows delil o doménové údaje len s iným strojom vo funkcii záložného doménového radiča, čo však Samba nevedela robiť. Akékoľvek pokročilejšie manipulácie s údajmi domény vyžadovali zakúpenie ďalších utilít od Microsoftu. Funkcia „vampír“, ktorá mala Sambe zabezpečiť schopnosť „vytiahnuť“ doménové údaje z Windows servera, nám nefungovala; ťažko odhadnúť dôvod, lebo Windows doména už bola opäť v štádiu začínajúcich problémov a Samba zase mala v tom čase s každou verziou svojské konštelácie funkčnosti/nefunkčnosti.

Migrácia nakoniec prebehla metódou „čistý stôl“ – najprv sme vytvorili celkom novú doménu a v nej duplicitne záznamy používateľov a počítačov, a postupne sme nastavovali počítače pre členstvo v novej doméne. V tomto období sme museli duplicitne udržiavať a aktualizovať všetky súborové služby, čo bolo náročné. Potom ku „dňu D“ sme čo najrýchlejšie vykonali konfigurácie zvyšných počítačov tak, aby sa hlásili už do novej domény, a starú doménu vyplli. V tom čase sme sa naučili mnoho o nevyspytateľnosti správania jednotlivých na pohľad rovnakých inštalácií Windows klientov. Postupne som sa zaučil a správu domény od vedúceho prevzal.

Po migrácii z domény Windows NT4 na doménu Debian Woody GNU/Linux + OpenLDAP + Samba 3.0.x sme konečne získali jednotnú bázu používateľov pre všetky vtedajšie i budúce služby. Pokiaľ ide o fyzické prevedenie, doménové služby zabezpečovala, ako zvy-

čajne, dvojica vyradených PC (100 MHz), ktoré sa vedeli zastúpiť v prípade výpadku. Po odsledovaní výkonových a inštalčných nárokov, prax postupne viedla k tomu, že DHCP/DNS služby sme tiež previedli na tieto servery.

Počas dlhoročného používania domény na technológii Linux/LDAP/Samba sa ukázala táto časť infraštruktúry ako najkrehkejšia. Stalo sa nám, že jedna bežná bezpečnostná aktualizácia Samby spôsobila znefunkčnenie domény. Odvtedy sme s aktualizáciami v tejto oblasti veľmi opatrní. Tiež sa stalo, že hw porucha jedného z prvých provizórnych doménových „serverov“ spôsobila poškodenie LDAP databázy, ktorú sa nám nepodarilo opraviť a museli sme si vypomôcť zálohou; odvtedy doménové záznamy pravidelne zálohujeme. V začiatkoch sa tiež občas stalo, že iný linuxový server zo záhadných dôvodov „vypadol“ z domény a museli sme ho tam znovu pridať. Stalo sa tiež, že niektoré PC prevzalo úlohu master browsera a ak sme chceli posadiť na trón opäť server, museli sme Sambe ručne zvýšiť „číslo OS“, na základe ktorého počítače vo Windows sieti volia svojho hlavného informátora. Okrem toho, zatiaľ prakticky každá migrácia na novšiu verziu systému Debian a teda aj novšiu verziu služieb Samba a OpenLDAP, priniesla nutnosť riešiť menšie či väčšie problémy. Je to o.i. aj známkou neustáleho vývoja v tejto oblasti, ktorý je čiastočne vynútený aj zmenami v každej verzii Windows. Výsledkom je, že stále treba počítať s pribúdaním nových parametrov, alebo so zmenami predvolených hodnôt používaných parametrov, pričom oboje treba pri prechode na novú verziu zohľadniť v konfiguračných súboroch.

Napriek týmto problémom, takto postavená doména sa javí ako stabilnejšia než bol jej Windows NT predchodca, za predpokladu, že sa za chodu nemia verzie softvéru, ktorý ju zabezpečuje. Je omnoho flexibilnejšia, pretože jednak voči LDAP vie fungovať veľké množstvo sieťových služieb, jednak sa dá ľubovoľne dopĺňať o nové údaje, ktoré môžu tieto sieťové služby využiť, ale najmä, dá sa jednoducho zálohovať a v prípade potreby spojzduť na celkom inom stroji. V súčasnosti beží doména na serveri HP 465C so systémom Debian Squeeze, ktorý poskytuje aj DHCP, DNS, časovú synchronizáciu a je hlavným súborovým serverom.

6.3 Oracle

Nasadzovanie ďalších a ďalších modulov ISS (najmä v rokoch 2004-6), pokrývajúcich čoraz väčšiu časť agendy MsÚ, znamenalo pripájanie nových zamestnancov do siete i do ISS. Pôvodní používatelia ISS už boli skúsení a využívali ho intenzívnejšie. Rástli legislatívne požiadavky na registratúru, takže vďaka modulu pre evidenciu písomností sa ISS stalo potrebným aj pre tých zamestnancov, ktorí ho predtým nepoužívali – postupne prakticky pre všetkých. Tieto okolnosti spolu spôsobili veľmi rýchly rast počtu aj „intenzity“ používateľov ISS.

Databáza Oracle8i bežala stále v prostredí Windows NT4 a čoraz častejšie sa prejavovali nedostatky – jediný rozsiahlejší dopyt spôsoboval úplné vytáženie servera a dokázal znemožniť prácu ostatných. V roku 2004 sme sa preto rozhodli skúsiť migrovať databázu na Linux. Za týmto účelom sme kúpili „silnejší PC“ do úlohy servera, keďže na značkový



Obr. 4: Takto vyzerala flotila serverov MsÚ BB okolo roku 2006. Väčšina z nich ešte stále plní záložné a testovacie úlohy

server, ako zvyčajne, neboli v rozpočte peniaze. Išlo o AMD Athlon 3000+ (2GHz) s 2GB RAM a SCSI radičom Adaptec s podporou RAID5. Po dlhšom boji sa začiatkom roku 2005 podarilo spojzduť Oracle8i pod distribúciou Debian Woody. Pre tento úspech bolo kľúčovým poznanie, že verzie knižníc a najmä Javy musia byť podľa možnosti presne také, ako predpisuje Oracle, a v žiadnom prípade novšie.

Hoci išlo v podstate o výkonnejší desktopový počítač, databáza bežala s nebývalou ľahkosťou a ani pri dvoch súbežných veľkých dopytoch ostatní používatelia nepozorovali spomalenie práce či nedostupnosť systému ako predtým. Pre objektivnosť treba dodať, že pôvodný NT4 server pre databázu bol o generáciu starší (Intel PIII >1GHz, 640MB RAM), pričom ale išlo o skutočný server s 64-bitovou PCI-X zbernicou, a jeho problémy sa nám pri analýze javili skôr ako neefektívnosť multitaskingu v podaní NT4, než ako nedostatočnosť výkonu alebo kapacity servera.

Neskôr v roku 2005 sa podarilo zakúpiť stroj poskladaný zo serverových komponentov triedy 2xAMD Opteron s 2GB RAM a 64-bitovým PCI-X radičom SCSI diskového poľa. Server bol pôvodne zamýšľaný ako terminálový, ale nakoniec sme naň presídlili databázu. Dodávateľ ISS, firma Corageo, mal už veľké starosti z našich experimentov s „nepodporovanou platformou“ (na všetkých ostatných mestách bežala databáza i súborový server na Windows, linuxový server bol niečím celkom exotickým), a pri niektorých chybách ISS mali niektorí vývojári sklony vyhovárať sa, že „to preto, lebo tam máte ten Linux“. Aby sme zamedzili takýmto incidentom, v októbri 2006 sme začali pracovať na migrácii na Suse Linux Enterprise Server 10, ktorý Oracle oficiálne uvádzal ako podporovaný systém, a pri príležitosti migrácie databázy na nový server sme prešli priamo na Oracle verzie 10. Potom sme pozvali niektorých predstaviteľov Corageo na priateľskú exkurziu. V roku 2008 sme databázu migrovali na nový server HP 685C (2x dual Opteron, 4GB RAM) a systém Red Hat Enterprise Linux 5.

6.4 Proxy

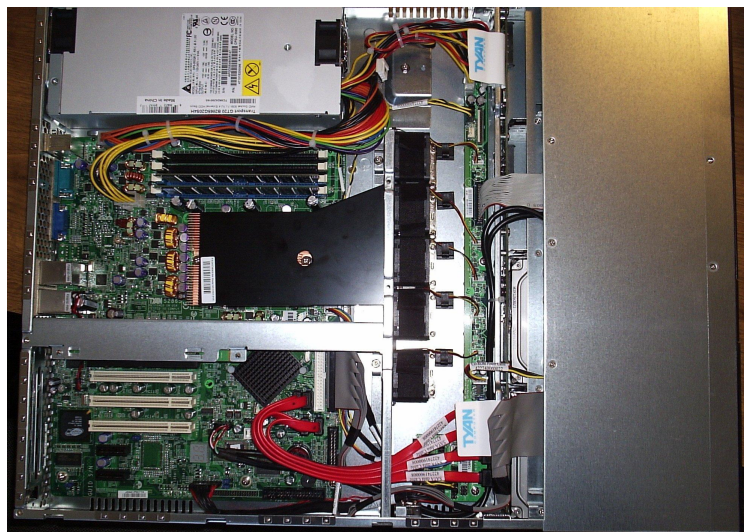
Intenzívne rozširovanie internetového pripojenia medzi zamestnancov sme z hľadiska bezpečnosti neponechali na náhodu. Bolo zrejme, že Internetom sa nové vírusy a červy môžu rozšíriť v priebehu hodín, takže z pohľadu aktuálnosti databázy, antivírus vždy ťahá za kratší koniec. Okrem toho, antivírusy v tom čase riešili len vírusy a červy, nie spyware. Aby bola bezpečnosť podporená aj nejakým účinným generickým opatrením a škodlivý kód sa podľa možnosti vôbec nedostával na naše počítače, v roku 2004 sme využili, ako ináč, staré PC (166 MHz, 64 MB RAM, 4 GB disk) a s pomocou Linuxu (Debian Woody) sme z neho spravili firewall/proxy (Netfilter + Squid). Pomocou rozšírenia Squidguard sme zabránili používateľom v sťahovaní spustiteľných súborov a blokovali pornografické stránky, ktoré sú najčastejším zdrojom škodlivého softvéru. Zakázali sme ešte používanie takých stránok, ktoré najviac vyťažovali naše skromné pripojenie v pracovnej dobe, ináč sme prístup prakticky neobmedzovali.

Upgrade na Debian Sarge si vyžiadal doplnenie RAM „až“ na 160MB. Vďaka dlhodobému sledovaniu štatistík a postupnému ladeniu parametrov Squid proxy/cache sme dosiahli nielen dostatočný výkon servera napriek archaickému hw, ale aj veľkú úspešnosť vyrovnávacej pamäte, ktorá dlhodobo vykryla až 30% dopytov. To predstavovalo nielen značné zrýchlenie pre používateľov, ale aj úsporu nákladov na pripojenie, keďže prekračovanie predplateného prenosu dát bolo spoplatnené.

Pomocou funkcie Squid delay pools sme dosiahli spravodlivé rozdeľovanie voľného pásma, pretože niektorí používatelia by ho ináč boli schopní úplne zahltiť sťahovaním rôznych súborov alebo počúvaním internetového rádia. Tieto problémy sa zmiernili po prechode zo 128kbit/s pripojenia na stabilné 3Mbit/s cez DSL, ale ukázalo sa, že aj takéto pásmo boli používatelia schopní pohotovo zahltiť, takže sme reguláciu pridelovania pásma a zvýhodňovanie/znevýhodňovanie niektorých stránok a typov súborov ponechali a platí dodnes.

Filter Squidguard riešil filtrovanie podľa typov súborov len okrajovo a vôbec neriešil antivírusové filtrovanie, ktoré sme do Proxy chceli zakomponovať. Preto sme postupne skúšali viacero FOS rozšírení pre Squid, avšak všetky mali dosť značné problémy so vzájomným prepojením. Integroujúcou platformou, ktorá tieto problémy vyriešila a sama osebe kvalitatívne veľmi zlepšila možnosti kontroly obsahu, sa stal filter obsahu Dansguardian, ktorý popri svojich vlastných kontrolách a filtroch dokáže odovzdávať súbory na kontrolu antivírusu ClamAV.

Samozrejme, takéto rozsiahle online filtrovania kladie vysoké požiadavky na výkon, a tak sme toto riešenie otestovali a reálne nasadili až v roku 2006 na systéme Debian Etch s pomocou nového serveru. Ten sme tentoraz obstarali v štýle barebone – rackovú skrinku so serverovou základnou doskou Tyan sme osadili dvoma Opteronmi, 4GB RAM a 2-portovou sieťovou kartou. Takýmto spôsobom sme postavili plnohodnotný server za približne polovičnú cenu oproti značkovým zostavám. Server slúži dodnes a ani pri 20Mbit/s pripojení nie je pocitovaný ako úzke hrdlo, s výnimkou videa, ktoré sa prehrá až potom, ako je serverom najskôr celé stiahnuté a skontrolované.



Obr. 5: Tyan barebone pri pohľade zhora – v roku 2006 to bol dobrý spôsob, ako získať profesionálny server za výrazne lepšiu cenu. Celkovo sme kúpili 3 takéto stroje. Nevýhodou 1U rack prevedenia je hluk 11000-otáčkových ventilátorov

Spomedzi zriedkavých problémov spomením nečakaný výpadok spôsobený zablokovaním ClamAV, ktorý vznikol preto, lebo distribúcia Debian nezaradila do štandardných aktualizácií zvýšenie verzie jadra ClamAV, hoci jeho vývojári signalizovali, že od istého dátumu starú verziu prestanú podporovať. Druhý problém bol hardvérový – kúpili sme 4-portovú sieťovú kartu Intel, ale pod žiadnou distribúciou Linuxu nefungovala, hoci ovládač v jadre existoval a všetky porty karty boli rozpoznané (a overili sme, že nešlo o chybu kusu). Vďaka ohláseniu v systéme ohlasovania chýb kernelu nás kontaktovali vývojári Intelu. Zriadili sme im testovací server oddelený od siete, aby mohli chybu identifikovať a ladiť. Žiaľ, v ich tíme došlo k tragickej udalosti, a navyše terminálový prístup cez naše DSL pripojenie mal spoza Atlantiku pomalú odozvu. Tieto okolnosti výrazne skomplikovali ich prácu, takže pokým vývojár zistil, že ide o nejaký záhadný problém v komunikácii so zbernicou, medzitým sa pri výrobe tohto typu karty už začal používať novší čip. Intel nám teda láskavo bezplatne poslal novú kartu, ktorá už fungovala, a týmto sme problém uzavreli.

6.5 Mailserver

Ako už bolo uvedené, poštový server bol jedným z prvých využití GNU/Linuxu na MsÚ, a dlho aj ostal v pôvodnom prevedení a funkcionalite – obsahoval POP3 účty, takže používatelia si správy sťahovali do svojich lokálnych klientov. Pomocou jednoduchého skriptu sme počas epidémií dokázali odstraňovať aktuálne rozšíreného červa z poštových správ, takže po stiahnutí používateľom už správa nebola nebezpečná. Išlo o primitívnu, ale v núdzi aspoň nejakú metódu do času, kým sa nový červ objavil v antivírusovej databáze jednotlivých PC.

S pribúdajúcim počtom schránok, ako aj pošty v nich, sa skript stal počas roku 2004 nepoužiteľným z hľadiska výkonu, a celkovo bolo jasné, že treba prijať účinnejšie a najmä automatizované riešenie. Vyskúšali sme niekoľko FOS rozšírení poštového systému, až kým sme nenašli prvý reálne použiteľný, ktorým bol Anomy sanitizer. Zneškodňoval chybné hlavičky správ, zahadzoval spustiteľné súbory, a „odzbroyoval“ také prílohy (ako napr. súbory MSOffice), ktoré síce neboli priamo spustiteľné, ani sme si nemohli dovoliť ich priamo zakázať, ale predsa len predstavovali bezpečnostné riziko. Filter ich premenoval tak, že Windows ich už nerozoznal podľa prípony, takže používateľ ich najprv musel uložiť so správnu príponou na disk. Počas tohto kroku došlo k automatickej kontrole antivírusom klienta. Takéto riešenie však bolo značne nepohodlné, a tak sme od neho čoskoro upustili.

Niekedy v tomto čase sa nám pritrafil prvý „bezpečnostný incident“, keď kvôli chybe v konfigurácii sa SMTP server (Postfix) začal správať ako open-relay, čo si „spameri všetkých krajín“ všimli do niekoľkých hodín a začali cez náš server preposielať spam. Na druhý deň nám už volali z firmy Nextra, ktorá nám poskytovala pripojenie. Naše prvotné obavy, že niekto ovládol náš poštový server, sa našťastie nepotvrdili, a po jednoduchej zmene jediného riadku v konfiguračnom súbore prestal poskytovať služby „náhodným okoloidúcim“. Táto situácia však priniesla dávku adrenalínu a potvrdila správnosť nášho prístupu – spojzdaňovať si dôležité služby vo vlastnej réžii. Je síce menej pohodlný ako nechať to implementovať na zákazku externej firme, ale okrem finančnej má svoju výhodu aj v tom, že sme nútení tieto služby aj do istej miery ovládať, čo sa viackrát ukázalo ako rozhodujúce pri riešení problémov. Zároveň sme týmto nútení k profesionálnemu rastu aj napriek absencii odborných školení zo strany zamestnávateľa.

Zabezpečenie klientských PC voči škodlivému kódu bolo len jedným vážnym poštovým problémom. S rastúcim množstvom pošty pribúdali aj prípady fatálneho poškodenia lokálnych poštových súborov vinou nestability programu Outlook Express na počítačoch, ktoré sme ešte nestihli vymeniť za nové s Thunderbirdom. Každá oprava PC vyžadovala kopírovanie poštového priečinku na náhradný počítač zamestnanca, čo pri rozsahu nezriedka niekoľko GB bolo veľkým zdržaním. Navyše pribúdali poruchy pevných diskov, a to nielen na starých počítačoch, ale aj u niektorých sérií nových. Všetky tieto okolnosti nás už pomerne skoro viedli k úvahám o výhodách centralizovaného úložiska poštových priečinkov podľa štandardu IMAP – v prípade poruchy PC by stačilo používateľa pripojiť k jeho priečinku z iného PC, a aj samotné riziko straty dát by sa týmto dalo znížiť, jednak vďaka spoľahlivejšiemu úložisku na strane servera, jednak vďaka zálohovaniu.

Naše pokusy s grupvérovými balíkmi, najmä Egroupware, ukazovali na ešte širšie možnosti v prípade centralizovaného úložiska – dalo by sa sprístupniť poštovú schránku cez webové rozhranie, bez nutnosti inštalácie a údržby poštového klienta na každom PC, a to dokonca aj mimo lokálnej siete (cez Internet).

Už v roku 2005 sme preto kúpili silnejšie PC (AMD Athlon 3000+, 2GB RAM, 2x120GB IDE disk pre sw RAID), no poštové služby sme v roku 2006 presunuli na pôvodný DB server, výkonovo totožný, ale s hw SCSI RAID 5. Popri IMAP sme prechodne museli ponechať aj POP3, kvôli množstvu takto nastavených klientov. Keďže na ich skokovú migráciu na IMAP sme nemali personálne kapacity, rozhodli sme sa pre migráciu postupnú.

Po zavedení tejto novinky – IMAP na strane serverovej infraštruktúry – sa teda servisný postup zmenil: keď sa k nám dostalo PC kvôli poruche alebo sme migrovali používateľa zo starého stroja na nový, namiesto kopírovania poštového priečinku z PC do PC sme už rutinne vykonávali prenos lokálneho priečinku priamo na IMAP server. Tento úkon prebiehal buď jednoduchým ťahaj-a-pusti úkonom v Thunderbirde, alebo tomu predchádzal ešte prípravný krok v podobe importu priečinku Outlook Express do Thunderbirdu (u tých starých PC, ktoré ešte neboli konvertované na Thunderbird). Táto operácia bola časovo i výkonovo náročná a u poruchových diskov aj dosť napínavá.

Teoretické výhody riešenia IMAP boli jasné, ale v praxi nás zaskočila veľkosť poštových schránok používateľov; niektoré sa museli prenášať na etapy, pretože ináč dochádzalo k zahlteniu PC alebo k vypršaní limitu na obsluženie požiadavky na strane IMAP servera. Skúsenosti ukázali, že pri rastúcom podiele správ s čoraz väčšími prílohami sa dalo spoľahlivo preniesť len cca 500 správ naraz, potom bolo treba označiť ďalších 500 a úkon opakovať. Navyše, ohromné nahromadené priečinky zamestnancov spôsobili nečakane rýchlo zaplnenie 36GB SCSI diskového poľa. Keďže na jeho rozširovanie neboli financie, ďalšie priečinky putovali už na bežný 200GB IDE disk. Približne po roku sa tento disk nielen zaplnil, ale aj opotreboval a začal kolabovať, s čím sme samozrejme počítali, ale zaskočilo nás, že sa jeho stav zhoršil náhle, bez varovných príznakov v SMART atribútoch ktoré sme sledovali. Čítanie a zápis na pribúdajúcich opotrebovaných častiach povrchu spôsobovalo zahltenie a výpadky servera. Z tejto krízy nás vyslobodil len urýchlený prechod na čerstvo zaobstaranú serverovú blade platformu HP začiatkom roku 2008. V tomto čase sme už službu POP3 vypli, lebo všetci zamestnanci už mali svoj priečink uložený v úložisku IMAP spôsobom Maildir (každá správa v samostatnom súbore).

Skúsenosti ukázali, že uloženie poštových priečinkov na serveri je dosť náročné, a to nielen na kapacitu (všetky priečinky sú na jednom diskovom priestore), ale aj na rýchlosť diskového systému a na voľnú operačnú pamäť; oproti použitiu POP3 je to nárast záťaže o 1-2 rády. Keďže sme od začiatku volili šifrovaný spôsob pripojenia IMAP/SSL, požiadavky na výkon servera boli ešte asi 3-násobne vyššie oproti nešifrovanému spojeniu. V apríli 2011 (pred zavedením kvót) bolo na serveri 336 poštových priečinkov s priemernou veľkosťou 1,5GB. Server HP 465C mal síce 2-jadrový procesor Opteron 2,6GHz, ktorého záťaž bola zvyčajne len okolo 20%, ale obmedzujúcou sa ukázala veľkosť RAM – len 2GB, z čoho významnú časť zaberali procesy súvisiace so spracovaním pošty a filtrovaním správ. Kvôli tomu ostávalo len niekoľko stoviek MB k dispozícii ako disková vyrovnávací pamäť. Toto sa ukázalo ako jedno z úzkych hrdiel, takže napriek výkonnému diskovému poľu RAID5 z 15k-otáčkových SAS diskov sa občas stávalo, že server prestal byť dostupný novým požiadavkám, napríklad kvôli tomu, že niektorý používateľ presúval niekoľko gigabajtov správ. Záťaž vtedy poľahky presiahla $load=20$ (cca 20-násobok normálnej kapacity servera) a keďže používatelia mali sklon opakovať neúspešné úkony, v extrémnych prípadoch následne stúpala až na 50, kedy bol server už absolútne zahltený. Práca s desiatkami správ v každom zo stoviek priečinkov jednoducho vyžaduje v pamäti určitý priestor „na dýchanie“, okrem toho súborový systém ext3 je pomerne neefektívny pre takéto situácie. Preto sme pri nedávnej

migrácii na Debian Squeeze mysleli aj na posilnenie pamäte na 4GB a súčasne sme preniesli poštové priečinky na ext4, ktorý by mal byť pri práci s veľkým množstvom súborov omnoho výkonnejší. Zatiaľ sú skúsenosti dobré.

Pokiaľ ide o technológiu z hľadiska sw, používame čisto FOS kombináciu Postfix (SMTP server) + Courier (IMAP server) + Amavis (platforma pre filtrovanie obsahu) + Spamassassin (spamový filter) + ClamAV (antivírus), a samozrejme Debian GNU/Linux, pôvodne vo verziách Sarge, Etch a v súčasnosti Squeeze. Z používania máme niekoľko postrehov. Niekoľko mesiacov sme používali poštový filter od ESETu. Bol veľmi rýchly, ale jeho spamový filter bol príliš citlivý, čo sa nedalo nastavovať, a od istého času nám preto prestal vyhovovať. Keď sme teda chceli skombinovať aspoň antivírus od Esetu s filtrom Spamassassin, túto kombináciu sa nám na platforme Amavis nepodarilo spojzdnúť. Je možné, že problém sa dal riešiť, ale nakoľko sme museli akútne riešiť odchod od spamového filtra ESET, a nechceli sme ustúpiť od požiadavky na prítomnosť všetkých troch vrstiev ochrany – generického filtra správ (Amavis), spamového filtra (Spamassassin) a antivírusového filtra, tak v časovej tiesni sme nakoniec upustili aj od antivírusu ESET a ostali sme pri FOS antivíru ClamAV. Kládie síce podstatne vyššie nároky na výkon servera a možno nemá takú úspešnosť detekcie vírusov, ale antivírus od ESETu máme plošne nainštalovaný na klientoch, takže použitie odlišného antivírusu na strane servera vytvára skôr žiadúcu diverzitu, ktorá môže znamenať vyššiu šancu zachytenia vírusu než keby sme použili rovnaké vyhľadávacie jadro na serveri i na desktopoch. Okrem toho, veľkú časť vírusov dokáže zneškodniť už samotný Amavis vďaka odmietaniu spustiteľných príloh a správ s neštandardnou hlavičkou. Napriek týmto okolnostiam, použitie antivírusu ESET do budúcnosti nevylučujeme, a možno už medzitým vylepšili aj spamfilter. V súčasnosti nám FOS riešenie vyhovuje aj z finančných dôvodov.

Kvôli korektnosti voči svetu, filtrom správ prechádza nielen prichádzajúca pošta, ale aj všetky odchádzajúce správy.

Spamový filter na serveri funguje na základe heuristiky Spamassassinu – správa sa vyhodnocuje podľa zabudovaného systému niekoľko tisíc kritérií – zisťuje sa napríklad porušovanie určitých štandardov, neobvyklé charakteristiky správy, výskyt určitých kombinácií slov a podobne. Správy s podlimitným bodovým hodnotením sa považujú za bežnú poštu, správy ktoré presiahnu základný prah sú automaticky presunuté do priečinku Spam adresáta, kým správy s veľkým počtom spamových bodov sú rovno odmietnuté. Okrem toho náš SMTP server (Postfix) je nastavený, aby pri porušení základných pravidiel elektronickej pošty (podľa relevantných RFC) správu okamžite odmietol. Týmto opatrením sa zbavíme približne štvrtiny spamu a vírusov skôr, než by sa nimi vôbec začali zaoberať filtre, čím sa šetrí výkon servera.

Odmietnutie správy (či už zo strany Postfixu alebo Spamassassinu) sa u nás uplatňuje už počas SMTP relácie, čo je podľa RFC správnejšie, než správu v rámci SMTP prijať, a až potom ju kontrolovať, zahadzovať a následne posilať správy o nedoručení. Paradoxne nás však korektné správanie dostalo do konfliktu s bývalým poskytovateľom pripojenia, GTS Nextra. Kvôli zvýšeniu bezpečnosti a odľahčeniu našich liniek a poštového servera sme si u nich platili službu – naša prichádzajúca pošta bola smerovaná cez ich server, ktorý ju kontroloval

proti vírusom. Ten pracoval v tandemovom režime – všetky správy v rámci SMTP prijal, zaradil do svojej fronty a až potom priebežne kontroloval a posielal nášmu serveru. Toto zapojenie má veľkú výhodu v rozložení záťaže (kontrola správ nemusí prebehnúť okamžite v rámci SMTP relácie, ale priebežne podľa dostupného výkonu), ale problém nastal, keď náš server niektoré správy odmietal prijať. Server na strane Nextry sa dostával do chýlostivej situácie, pretože on už tie správy prijal a musel sa odosielateľom dodatočne „ospravedlňovať“ (posielať správy o nedoručení). Keďže náš server odmietal predovšetkým spam, išlo zvyčajne o fiktívne adresy, alebo o zneužitie adresy nič netušiacich ľudí, a vzhľadom ku veľkým množstvám takýchto zbytočne vygenerovaných hlásení o nedoručení, sledovacími systémami na Internete to začalo byť považované za spamovanie a hrozilo, že nielen naša doména, ale aj server GTS Nextra sa dostane na zoznam spamových domén. Nechceli sme prísť o dôležitú úroveň ochrany pred nežiaducimi správami, ktorou bolo odmietanie na základe RFC pravidiel, a tak jediným východiskom bolo zrušenie antivírusovej služby a úplné „osamostatnenie“ nášho poštového servera, ktorý týmto už prestal byť krytý filtrom GTS Nextry a musel si so všetkým poradiť už sám. Na tomto príklade vidno, že niekedy sa aj vďaka svedomitému dodržiavaniu pravidiel a štandardov môžete dostať do problémov.

Keďže sme úrad, ktorý komunikuje s verejnosťou a štátnou správou, spamový filter na serveri je nastavený pomerne konzervatívne, aby radšej prepustil podozrivú správu, než „stratil“ niečo dôležité. Tento kompromis má jednoduchý dôsledok – server odchyť len niečo vyše polovicu nevyžiadanych správ. Zvyšok je ponechaný na úsudku Bayessovského filtra poštového klienta Thunderbird, ktorý si môžu jednotliví zamestnanci natrénovať tak, aby si poradil s ostatnými správami, ktoré cez serverový filter preklázli.

6.6 EGroupware

Potreba efektívneho manažmentu úloh ma viedla už v roku 2004 k testovaniu grupvérového balíku PHPgroupware, ktorý obsahoval helpdeskovú aplikáciu Trouble Ticket System. Dúfal som, že najprv si doň budeme značiť úlohy my, informatici, a postupne naučíme zamestnancov, aby do neho sami zapisovali svoje požiadavky a problémy. Veľkou brzdou efektivity našej práce totiž boli tí zamestnanci, ktorí buď svoje požiadavky oznamovali telefonicky, často aj viacerým kolegom a opakovane, čím vznikalo množstvo duplicitnej práce, alebo ich oznamovali ústne „cez plece“ na chodbe, a potom sa hnevali, že sa na nich zabudlo.

Na PHPgroupware ma oslovila nielen FOS filozofia tohto balíku, ale aj čistá webová architektúra (Apache + PHP + MySQL), ktorá bola príslubom jednoduchého nasadenia bez nutnosti inštalácie klientskeho softvéru na stovkách PC. Ďalšou výhodou bolo, že používatelov a skupiny dokázal čerpať z domény LDAP. Keďže neexistoval slovenský preklad, neostávalo mi nič iné len si ho napísať (a samozrejme ponúknuť komunite). Vývoj PHPgroupware však postupne uviazol a keď približne rok nevyšla žiadna nová verzia, vznikol odštiepený projekt EGroupware, ktorý som doprekladal a v roku 2006 prostredníctvom Intranetu sprístupnil všetkým zamestnancom MsÚ ako dobrovoľnú pracovnú pomôcku. Na našom oddelení sme si do neho napríklad značili obchodné kontakty pomocou aplikácie Adresár.

Pôvodne sme sa zahrávali s myšlienkou, využiť webmailovú aplikáciu v EGroupware ako štandardný a jediný spôsob prístupu k poštovej schránke na MsÚ, čo by nám ušetrilo inštaláciu a správu plných poštových klientov. Ako bolo našim zvykom, najprv sme tento nápad vyskúšali na sebe a zistili sme, že webová aplikácia neponúka dostatočný komfort a nie je plnohodnotnou náhradou Thunderbirdu, môže však poslúžiť pre núdzový prístup k poštovej schránke v prípade poruchy PC alebo pri prístupe zvonku. Thunderbird teda ostal na svojom mieste.

Ako prví pochopili výhody EGroupware tí zamestnanci, ktorí boli vyspelejší v používaní PC a potrebovali prehľadne organizovať svoje pracovné stretnutia, čomu poslúžila aplikácia Kalendár. Cez EGroupware sa postupne začala riešiť rezervácia zasadačiek a Veľkej siene pomocou aplikácie Zdroje – už nebolo treba telefonovať sekretárkam a pátrať v zošitkoch, kedy bude ktorá miestnosť voľná, už si to mohol každý zistiť na niekoľko klikov a rovno si ju rezervovať pre svoju akciu. Samozrejme, nešlo to vždy hladko, stávalo sa najmä, že zamestnanec udalosť naplánoval len vo svojom kalendári a zabudol odkliknúť, že si rezervuje aj miestnosť, a potom vznikol konflikt keď si ju niekto iný naozaj zarezervoval na ten istý termín. Alebo sa niekto vôbec neunúval rezervovať miestnosť, jednoducho tam prišiel a vyhodil tých, ktorí v nej boli oprávnené. Avšak táto oblasť vzťahov leží mimo možnosti technológie a mimo sféry informatiky.

Po voľbách v roku 2006 prišla od vedenia požiadavka na riešenie spoločného plánovania stretnutí na sekretariáte primátora a prednostu. EGroupware bol logickým riešením, hoci nastaviť prístupové práva podľa požiadaviek a podľa skutočného režimu práce bolo zložité. Takto sme sa zároveň zbavili posledných inštalácií Microsoft Outlook, ktoré nijako nezapadali do celkovej koncepcie založenej na otvorených štandardoch. Primátor sledoval svoj program stretnutí cez mobil s internetovým prehliadačom.

Postupne vyvstala logická požiadavka, aby sa aj služobné vozidlá mohli sledovať a rezervovať cez EGroupware. V tomto čase sa objavil nový druh hádok o automobily, z ktorých bol najskôr obviňovaný EGroupware, pokým sme príčinu nevystopovali až k fenoménu „ignorovať konflikty sa nevypláca“. Vznikali tak, že používateľ sa snažil zarezervovať auto a keď ho EGroupware upozornil, že je v danom čase už obsadené, tak používateľ v rámci zásady „nečítať, čo počítač píše“ odklikol „ignorovať konflikt“.

Neskôr prednosta MsÚ, Ing. Sihelský, vyslovil požiadavku na vyriešenie elektronického systému sledovania plnenia úloh plynúcich z porád, a v rámci neho sme presadili riešenie založené na aplikácii Záznamník v EGroupware. V súčasnosti každého nového zamestnanca školíme aj o možnostiach EGroupware.

Vidno tu odlišný prístup – kým v prípade desktopových aplikácií sme striktne stanovili, aké programy sa budú plošne používať, kvôli zabezpečeniu kompatibility a bezpečnosti, v prípade EGroupware sme len sprístupnili a ukázali technológiu a nechali sme používateľov, aby sa s ňou oboznámili a aby ju využili podľa vlastných schopností a potrieb a sami presadzovali na svojom oddelení. Išlo o prístup „revolúcie zdola“. Na inú, rozsiahlejšiu revolúciu sme už nenašli sily, pretože z našej pozície nie je reálne tlačiť zamestnancov do používania akejkolvek technológie, pokiaľ sami nechápu jej výhody pre zefektívnenie

svojej práce. Takýto benevolentný prístup bol možný len vďaka bezplatnosti EGroupware, ktorá nám dovolila sprístupniť ho každému zamestnancovi bez potreby riešiť počty licencií; keby išlo o komerčný produkt, v prvých rokoch by jeho nepoužívanie robilo investíciu neudržateľne neefektívnou.

Webová architektúra EGroupware sa opakovane ukázala ako veľká výhoda. Umožnila nám bez problémov vyriešiť požiadavky používateľov na prístup k e-mailom mimo siete MsÚ. Navyše, po rozhodnutí vedenia o začlenení materských škôl a jedální sme im mohli pružne sprístupniť naše poštové služby, čo by sme ináč nedokázali v reálnom čase zabezpečiť, pretože nemáme ľudské zdroje na to, aby sme obchádzali tieto zariadenia a nastavovali im lokálnych poštových klientov.

Paradoxne, aplikácia, kvôli ktorej som začal aktivity ohľadom grupvéru – systém pre helpdesk – nakoniec ostala jednou z mála nevyužívaných aplikácií EGroupware. Kolegovia uprednostnili podobnú aplikáciu Magicdesk z balíka PCInfo, a hoci koncepčne a technologicky nezapadla do našej infraštruktúry, spolu s vedúcim sme boli radi, že sa začal používať aspoň nejaký helpdesk, a akceptovali sme ju napriek našej preferencii k aplikácii Sledovač v EGroupware.

6.7 GIS

Územný plán a mapové podklady pre rozhodovanie mesta spravuje Útvár hlavného architekta (ÚHA). Technickú stránku vecí, vrátane GIS, zabezpečoval Bc. Voskár, ktorý bol tiež nadšencom FOS, takže medzi nami vznikla spolupráca. Podarilo sa mu naštudovať, navrhnuť a v spolupráci s firmou Sw@gis [čítať: swagis] zabezpečiť spustenie GIS portálu mesta, a to nielen pre potreby MsÚ, ale aj verziu pre širokú verejnosť. Operačný systém bol Debian Linux, databáza PostgreSQL s nadstavbou PostGIS a samotný mapový server UMN Mapserver (Univerzita Minnesota). Klientom bol Cartoweb (Camptocamp). Išlo o unikátne riešenie, pretože z pohľadu klienta bolo nezávislé na platforme; jedinou podmienkou bola prítomnosť knižníc Javy (porovnajme s katasterportálom, ktorý pre mapu vyžaduje, aby si občan kúpil PC s Microsoft Windows a inštaloval proprietárny ActiveX komponent).

Náklady na nasadenie tohto FOS riešenia pre GIS spočívali predovšetkým v príprave a spracovaní dát. Už samotné vstupné náklady boli približne o dva rády nižšie, než u hrozivacieho komerčného riešenia, nehovoriac o ďalších nákladoch na obnovu licencií atď u komerčného produktu.

Toto GIS riešenie bolo ocenené v roku 2007 cenou Zlatý erb. V roku 2011 bolo pod správou Mgr. Fekiača, (nástupcu Bc.Voskára) nahradené podobným FOS riešením firmy Gista.

7 Záver

Nasadenie FOS technológií na MsÚ BB bolo podmienené viacerými faktormi. Prvým bolo stretnutie správneho kolektívu v správnom čase; bez osobného záujmu konkrétnych ľudí

a ich cieľavedomého postupného napredovania by sa mesto Banská Bystrica pravdepodobne nijako nelíšilo od ostatných krajských miest; vonkajšie okolnosti sú k FOS skôr nepriaznivé. Financie neboli hlavným dôvodom nasadzovania, pôsobili len ako definitívny argument pri aktuálnych (ne)možnostiach rozpočtu. Úspora vďaka FOS technológiám predstavuje v hrubom odhade spolu cca 800 000 € v porovnaní s bežnými komerčnými produktami v porovnateľnom počte licencií. Zásadným dôvodom nasadenia bola preferencia otvorených štandardov ako prevencia závislosti na konkrétnom dodávateľovi. Veľkou výhodou bola flexibilita (možnosť nasadiť kedykoľvek a v akomkoľvek rozsahu).

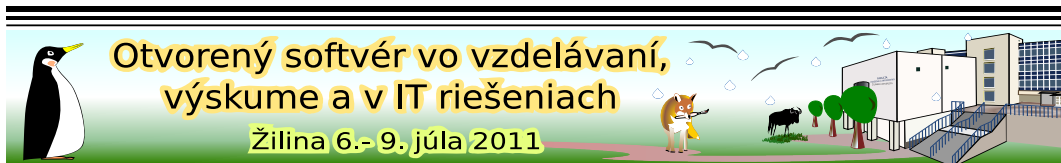
Mesto Banská Bystrica rutinne používa viaceré FOS technológie – na desktopoch Open-Office.org, Mozilla Thunderbird a Firefox, na serveroch sú nimi zastrešené základné prvky infraštruktúry – doménové a súborové služby, elektronická pošta, web, grupvér. V príspevku sme opísali viaceré ťažkosti, s ktorými sme sa stretli pri ich nasadzovaní. Mnohé z nich vyplynuli z nedostatočného hardvéru, ale aj z neustáleho vývoja technológií. Avšak práve vďaka ich vývoju a zlepšeniu situácie na trhu, v súčasnosti sú mnohé veci už podstatne jednoduchšie. Aj tak sa však pri ich nasadzovaní vyžaduje aspoň základná úroveň znalostí u správcov, s výnimkou riešení plne outsorcovaných alebo na kľúč.

Najväčším faktorom nasadenia akéhokoľvek softvéru je úroveň počítačovej gramotnosti používateľov. Pri nasadzovaní je veľmi vhodné mať vypracovaný plán a získať podporu vedenia. Inou možnosťou je premýšľať vopred o možnostiach v rámci koncepcie a mať pripravené riešenia pre predpokladané budúce požiadavky. Dôležitou podmienkou budúceho úspechu pri nasadení FOS technológií je zvažovanie už každého dnešného rozhodnutia a každého nákupu hardvéru alebo softvéru z pohľadu jeho podpory otvorených štandardov. Ak sa raz pripustia kompromisy voči otvoreným štandardom, zbavenie sa závislosti na konkrétnom dodávateľovi (vendor lock-in) môže byť veľmi náročné až nereálne. Popri závislosti na konkrétnom proprietárnom dodávateľovi, najväčšou potenciálnou hrozbou pre používanie FOS technológií by mohli byť neodborné politické rozhodnutia.

Peter Tuhársky (Mgr.),

peter.tuharsky@banskabystrica.sk

**ABSTRAKTY UKÁŽOK
A PREZENTÁCIÍ**



OTVORENÝ SOFTVÉR VO VYUČOVANÍ ASTRONÓMIE A ASTROFYZIKY

CSATÁRYOVÁ, Mária (SK); ŠECHNÝ, Martin, (SK)

Astronómia patrí medzi obľúbené predmety žiakov, no jej výučba je náročná na predstavivosť. Počítačové programy nám umožňujú vizualizáciu nočnej oblohy, astronomických objektov a jednotlivých astronomických úkazov. V článku predstavujeme slobodné počítačové produkty Aladin, Stellarium, Celestia na výučbu astronómie a astrofyziky a možnosti využitia projektov virtuálnych observatórií, napr. EURO-VO.

PRAKTICKÉ ZKUŠENOSTI Z NASAZENÍ LTSP A DALŠÍHO OSS NA ZŠ VALČÍKA V OSTRAVĚ Z POHLEDU EXTERNÍHO SPRÁVCE SÍTĚ

DVOŘÁK, David, (CZ)

V současné počítačové síti na ZŠ Valčíka v Ostravě je zapojeno přes 80 počítačů. Z tohoto počtu je cca 55 tzv. tenkých klientů (LTSP), zbytek tvoří plnohodnotné počítače s OS Windows 7 a OS Ubuntu 10.04, resp. dualbootem. Vedle zmíněné sítě tenkých klientů škola používá mj. následující open-source software : LibreOffice, Moodle, eGroupware, Samba.

Příspěvek se zaměří na čtyřleté praktické zkušenosti externího správce sítě s důrazem na úskalí budování a provozu školní sítě s větším počtem tenkých klientů a open-source softwarem. Jelikož provoz školy má svá specifika – např. školní matrika, účetnictví s výstupem směrem k zřizovateli atd., bude přednáška rovněž zahrnovat praktické poznatky s provozem tohoto nutného proprietárního software v převážně linuxovém prostředí.

VEKTOROVÉ GEOÚDAJE VO WEBOVOM PROSTREDÍ

CIBULKA, Dušan, (SK)

Príspevok v prvej časti popisuje vybrané spôsoby publikovania a spracovania vektorových geodajov na internete. Zameriava sa na služby a formáty štandardizované OGC (Open

Geospatial Consortium). Hlavne na WFS (Web Feature Service), ako na službu slúžiacu na poskytovanie a editáciu geografických údajov a na WPS (Web Processing Service), ako na službu umožňujúcu spracovanie geografických údajov.

V druhej časti sa môže čitateľ oboznámiť s Open Source nástrojmi, ktoré umožňujú prácu s vektorovými geoúdajmi vo webovom prostredí. Popisuje ich možnosti, štandardy ktoré podporujú a princíp práce s nimi. Jedná sa hlavne o dva nástroje, mapový server GeoServer a knižnicu OpenLayers. GeoServer je mapový server, ktorý umožňuje publikovať rôzne dáta prostredníctvom služieb WMS (Web Map Service), WFS, WCS (Web Coverage Service) a od poslednej verzie umožňuje aj spracovanie geoúdajov prostredníctvom služby WPS. GeoServer je sprístupnený pod GPL (General public license) licenciou. OpenLayers je JavaScriptová knižnica respektíve API (Application Programming Interface), ktoré slúži na vkladanie dynamických máp do ľubovoľných webových stránok. Dokáže zobrazovať údaje zo služieb typu WMS, editovať údaje sprístupnené cez transakčné WFS alebo zobrazit' formáty ako GML (Geography Markup Language), GeoJSON (Geospatial JavaScript Object Notation) či KML (Keyhole Markup Language).

Článok popisuje kombináciu spomenutých technológií za účelom transparentného a štandardizovaného sprístupňovania a spracovania geoúdajov vo webovom prostredí.

KONVERZIA FORMÁTU HTML DO TEX

ČANECKÝ, Peter, (SK)

Cieľom príspevku je predstaviť konvertor z formátu HTML do LaTeXu, pri ktorom je možné vybrať kódovanie vstupného súboru. Uvedený konvertor má podporu práce s obrázkami a dokáže spracovať aj tabuľky. Program sa od existujúcich riešení odlišuje najmä podporou voľby vstupného kódovania webových stránok, podporou základného spracovania CSS, podporou sťahovania dokumentov z internetu, spracovaním HTML tabuliek.

VÝVOJ WEBOVÝCH APLIKÁCIÍ V RUBY

FOJTÍK, Michal, (SK)

Stručný úvod do vývoja aplikácií s použitím programovacieho jazyka Ruby. V prezentácii budú predstavené rôzne frameworky, ktoré uľahčujú samotný návrh, vývoj a testovanie. Okrem všeobecne známeho frameworku Ruby on Rails, sa budem venovať aj iným, menej známym no o to zaujímavejším frameworkom ako Sinatra, Padrino a podobne. Okrem prezentácie samotného vývoja, by som rád predstavil aj najnovšie trendy vývoja web aplikácií ako sú rôzne mashupy (spájanie rôznych služieb pomocou API).

AEOLUS PROJEKT

HRČKA, Tomáš, (SK)

Projekt Aeolus je nástroj, umožňujúci manažment cloudoveho prostredia typu IaaS, vytváranie inštancií virtuálnych strojov a monitorig. Pri práci na tomto projekte sa stretávame s množstvom zaujímavých problémov, napríklad zostavovanie obrazov operačných systémov pre rôznych poskytovateľov cloudových služieb. V prezentácii by som rád predstavil tento projekt a jeho subprojekty, ako aj možnosť jeho nasadenia.

RUBY ON RAILS ALEBO POTREBUJEME ĎALŠÍ WEBOVÝ FRAMEWORK?

HRČKA, Tomáš, (SK)

Ruby on Rails je webový framework založený na jazyku Ruby. Využíva architektúru MVC, vďaka čomu umožňuje rýchly a jednoduchý vývoj webových aplikácií. Ruby on Rails sa skladá z viacerých komponentov, ako napr. ActiveRecord, ActiveSupport, ActionController, ActionView apod., ktoré uľahčujú prístup k databáze, spracovávanie požiadaviek, zobrazovanie obsahu atď. . .

V prezentácii by som chcel predstaviť framework a jeho jednotlivé časti, demonštrovať jednoduchosť vývoja prostredníctvom dema a porovnať ho s konkurenčnými technológiami.

NEBOJTE SE VIRTUALIZACE S KVM, LIBVIRT A CGROUPS

HRUŠECKÝ, Michal, (CZ)

Stručný úvod do virtualizace pomocí kvm ve spolupráci s libvirt a cgroups. Jak začít a jaké zajímavé vlastnosti zmiňované technologie poskytují uživateli. A že i když to vše vypadá nepřívětivě a složitě na první pohled, ve skutečnosti to zas tak špatné není.

AKO VZNIKALA STRÁNKA OSSCONF

KAUKIČ, Michal, (SK)

Rozprávanie o histórii webstránok konferencie a o tom, ako sme sa rozhodli vyvinúť vlastný systém. Potom niečo o konkrétnych nástrojoch, ktoré sme použili a o spoločnej tvorbe stránky so študentami v rámci predmetu „Open source softvérové nástroje pre inžinierov“.

TVORBA FLASHOVÝCH HIER POMOCOU OPEN SOURCE TECHNOLÓGIE FLIXEL

KOSTOLNÝ, Jozef, (SK)

Príspevok sa zaoberá predstavením open-source knižnice „Flixel“, ktorá umožňuje tvorbu 2D flashových hier. Je napísaná v jazyku ActionScript 3, čo je Adobe verzia JavaScriptu. Flixel umožňuje riešiť bežné veci, s ktorými sa vývojár stretáva pri tvorbe hier. Napríklad patrí k tomu detekcia kolízií objektov, prepínanie hry s herným menu alebo animácia postavičiek. V článku predstavím aplikáciu tejto knižnice s použitím Adobe open-source frameworku Flex, ktorý je založený na technológii Adobe Flash.

INSPIRE A INFRAŠTRUKTÚRA PRIESTOROVÝCH INFORMÁCIÍ VYUŽITÍM OPEN SOURCE SOFTWARE

KOŠKA, Martin, (SK)

Implementácia smernice INSPIRE predstavuje okrem samotných údajov a ich metaúdajov aj implementáciu sieťových služieb – vyhľadávacích, zobrazovacích, ukladacích a transformačných. Existuje celý rad riešení na vytvorenie efektívnej infraštruktúry, ktoré budú spĺňať aj jednotlivé kritériá definované v jednotlivých nariadeniach INSPIRE. Jednou z možností je využívanie riešení s otvoreným kódom, ktoré sú v súčasnosti výkonnostne a kvalitatívne porovnateľné s komerčnými riešeniami.

Príspevok predstaví možnosť vytvorenia infraštruktúry pre priestorové informácie využitím bežne dostupných open source riešení ako – postgress, postgis, geoserver, deegree, geotools, openlayers, geomajas, opengeo suite a ďalších so zameraním sa na výhody a nevýhody takého riešenia.

OPENJDK 7 REVOLÚCIA, ALEBO PLÁN B ?

LACKO, Michal, (SK)

Java slúži ako základ pre mnohé open-source projekty, ale sama nemá otvorený kód. Na internete však existuje aj zopár open-source implementácií Java SE platformy. Jedným z nich je OpenJDK. V blízkej dobe vychádza jeho nová verzia 7, pri návrhu ktorej sa predstavilo asi dvesto vylepšení, do finálnej verzie sa však všetky nedostali.

Dôležité je však, že verzia prináša niektoré zmeny, ktoré môžu pozitívne ovplyvniť vývoj projektov a vývojári na ne čakali niekoľko rokov. Článok predstaví niektoré z týchto noviniek, projekty spojené s vývojom a aj prehľad open-source implementácií JDK.

OTVORENÉ DÁTA A SOFTVÉ V MODERNEJ ASTROFYZIKE

KOCKA, Matúš, (SK)

Všetky odvetvia modernej vedy produkujú obrovské množstvá dát a to nesie so sebou problém, ako tieto dáta spracovávať a tiež ich zdieľať či už vo forme „raw“ alebo výsledkov. Špeciálne, fyzikálne odbory ako je časticová fyzika CERN (15TB/rok) či astrofyzika poznajú projekty, generujúce kvantum dát. Práve preto je väčšina softvéru verejná, dokonca pod licenciou GPL a dáta sú verejne prístupné, aby mohli byť spracovávané širokou komunitou. Toto je tiež neuveriteľná príležitosť pre študentov, učiteľov, ale aj nadšencov bez príslušného vzdelania, schopností a financií, zapojiť sa do supermoderných vedeckých projektov. V mojej prezentácii načrtnem, ako na to.

ČO MÔŽEME UROBIŤ PRE BUDOVANIE OPENGIS KOMUNITY NA SLOVENSKU

OFÚKANÝ, Miloslav, (SK)

- mať záujem využívať FOSS4G
- sledovať domáce a zahraničné dianie vo FOSS4G
- čítať stránku opengis.sk a posielat' príspevky na uverejnenie
- zoznam ďalších tematických webov
- stať sa členom SOIT v odbornej skupine OpenGIS
- šíriť osvetu o OpenGIS a aktivitách SOIT

PŘÍKLADY NASAZENÍ OPEN SOURCE V ČR A SR

OTT, Vlastimil, (CZ)

V České i Slovenské republice se open source či nějaký linuxový operační systém používají stále častěji, i když jsou na tom jiné evropské země asi lépe. Přesto u nás nalezneme příklady dobré praxe, které jsou dokladem toho, že open-source software je po technologické stránce dostatečně zralý a vhodný pro nasazení téměř kdekoliv. Přednáška představí několik organizací v ČR a SR, které používají otevřené technologie.

PROJEKTOVÉ ŘÍZENÍ POMOCÍ REDMINE – ZKUŠENOSTI Z PRAXE

OTT, Vlastimil, (CZ)

Redmine patří mezi aplikace pro správu událostí, v užším smyslu "úkolů". Uživatelé v tomto informačním systému zadávají nebo přijímají úkoly, které následně řeší. Každý úkol disponuje prioritou, kategorií, termínem a dalšími informacemi. Správce projektu snadno sleduje využití jednotlivých kolegů a může jim úkoly operativně přeskupovat, aby dosáhl zamýšleného splnění.

Úkoly se ukládají včetně související diskuze a příloh do historie a tvoří tak postupem času cenné know how. Aplikace je open source, její implementace vyžaduje zaškolení, nicméně následné použití je velmi efektivní. Aplikaci používáme v Liberixu a zkušenosti pocházejí z reálného provozu několika měsíců.

FREEMAP.SK – ČO SME A ČO JE NOVÉ

PÁLENÍK, Michal, (SK)

Freemap.sk je komunitný portál slovenskej časti OpenStreetMap.org. Prevádzkuje mapový portál s rôznymi službami, združuje slovensku komunitu okolo projektu OpenStreetMap.org, propaguje myšlienku slobodného softvéru a dát, a vzdeláva študentov na slovenských školách. Oproti minulosti sme na portáli urobili niekoľko zmien pod kapotou (zrýchlenie renderovania, rýchlejšie načítanie, využívanie viacerých serverov). Naďalej prevádzkujeme autoatlas, ako i turistický atlas. Prevádzkujeme tiež vrstvu wikipédie, počasia, MHD, geocaching, cyklotrasy a lyžiarske trasy.

Ako vlastnú službu ponúkame galériu, možnosť lokácie objektov (napríklad aut) ako i lokalizované správy z terénu (aj formou SMS). Hlavnou novou službou je možnosť pridať nový bod záujmu (POI) priamo z prostredia portálu freemap.sk. Tento sa potom pridá aj do databázy OSM. Taktiež sme rapídne vylepšili vyhľadávanie trás, ktoré používa motor z gosmore.

HACKERSPACE A OPEN HARDWARE

RUSNAK, Pavol, (SK)

V súčasnosti sa aj v našich končinách rozmáha fenomén „hackerspace“ – miesta, kde sa ľudia so záujmom o vedu a techniku môžu stretávať a spolupracovať na rôznych projektoch. Jedná sa vlastne o modernizovanú podobu známych rádioklubov s rozšíreným záberom aktivít. Okrem elektroniky sa jeho členovia zaoberajú aj programovaním a algoritmi, informačnou bezpečnosťou, digitálnym umením či biológiou alebo chémiou.

Druhá časť prednášky sa bude venovať open source hardvéru, porovná ho s jeho starším bratrancom otvoreným softvérom a priblíži najznámejšie projekty Arduino, RepRap a USRP.

ČO JE NOVÉ V OPENSUSE?

RUSNÁK, Pavol, (SK)

Tato prednáška návštevníkom priblíži, čo nové prináša posledné vydanie openSUSE 11.4, ale taktiež prekročí rámec distribúcie a poobhliadne sa po zmenách v iných častiach tohoto open-source projektu.

PARALELNÍ VÝPOČTY S OPEN SOURCE GIS SOFTWARE

RŮŽIČKA, Jan, (CZ)

Paralelní výpočty urychlují časově náročné výpočty. Přestože se jedná o poměrně složitou problematiku, pokud se s ní začneme zabývat na vědecké úrovni, příspěvek se snaží ukázat, že je v mnoha případech možné jednoduché paralelní výpočty realizovat jen se základní znalostí programování. Ukázky možných paralelních výpočtů budou v prostředí GRASS GIS a s využitím nástrojů knihovny GDAL/OGR. Prezentován bude také přehled prostorových analýz nebo operací nad digitálním obrazovým záznamem, které je možné velmi snadno urychlit s využitím paralelního zpracování.

OPEN DOCUMENT FORMAT: POROVNANIE KVALITY GRAFICKÉHO VÝSTUPU KANCELÁRSKÝCH APLIKÁCIÍ

ŠRÁMEK, Miloš, (SK)

Vo svete kancelárskych aplikácií donedávna pevnou rukou vládli proprietárne štandardy spoločnosti Microsoft: doc, xls a ppt. Svet sa však mení a so všeobecným volaním po väčšej otvorenosti vznikli dva nové štandardy pre kancelárske aplikácie. Prvým bol Open Document Format (ODF, ISO 26300), ktorý s istým časovým odstupom nasledoval štandard Office Open XML (OOXML, ISO 27500).

Otvorený štandard ODF je známy najmä v súvislosti s kancelárskymi balíkmi OpenOffice a LibreOffice. ODF však podporujú aj mnohé ďalšie otvorené, ale aj komerčné aplikácie. V príspevku sa zameriame na porovnanie kvality zobrazovania textových dokumentov vo formáte ODF viacerými populárnymi balíkmi. Predstavíme distribuovanú službu Office-Shots, ktorá verejnosti na testovanie sprístupňuje viaceré kancelárske balíky a programy, čo môže slúžiť ako pomôcka pri ich výbere. Ďalej predstavíme jednoduchú metodiku na kvantitatívne porovnávanie renderovaných textov, ktorú spolu s výstupmi OfficeShots použijeme na porovnanie kvality zobrazovania textových dokumentov rôznymi programami.

FEDORA 15 A RED HAT ENTERPRISE LINUX 6

VOKÁL, Radek, (CZ)

Novinky v distribuci Fedora 15 z brněnské pobočky společnosti Red Hat:

- Gnome Shell – nový dekstop,
- systemd,
- ABRT a Retrace Server,
- SPICE a virt-manager,
- dynamický firewall,
- a další.

Druhá část bude věnována distribuci Red Hat Enterprise Linux 6, jakým způsobem vzniká a jak komunita přispívá k jejímu vývoji.

OTVORENÝ SOFTVÉR VO VZDELÁVANÍ, VÝSKUME A V IT RIEŠENIACH

Zborník príspevkov medzinárodnej konferencie OSSConf 2011

© Autori príspevkov

Prvé vydanie 2011

Počet strán 294

Elektronická sadzba programom pdfL^AT_EX

Vytlačili C-PRESS, Ing. Ivo ČABRA, Košice

ISBN 978-80-970457-1-5