

Spoločnosť pre otvorené informačné technológie

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



**Zborník príspevkov medzinárodnej konferencie
OSSConf 2012**

**2.–4. júla 2012
Žilina, Slovensko**

Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach

2.–4. júla 2012, Žilina, Slovensko

Vedecký a programový výbor:

prof. Ing. Miloš Šrámek, PhD., FMFI, Univ. Komenského, SOIT, Bratislava, predseda
prof. Oleg Černojarov, DrSc., Moscow Power Engineering Institute (RU)
doc. Ing. Karol Grondžák, PhD., Žilinská univerzita, SOIT, Žilina
doc. RNDr. Štefan Peško, PhD., Žilinská univerzita, SOIT, Žilina
doc. Ing. Jiří Rybička, Ph.D., MZLU Brno (CZ)
RNDr. Rudolf Blaško, PhD., Žilinská univerzita, SOIT, Žilina
RNDr. Ján Buša, CSc., FEI, Technická univerzita v Košiciach, SOIT, Košice
Mgr. Michal Kaukič, PhD., Žilinská univerzita, SOIT, Žilina
RNDr. Aleš Kozubík, PhD., Žilinská univerzita, SOIT, Žilina
Ing. Marek Mahut, Red Hat CZ, SOIT, Brno (CZ)
Ing. Pavel Stříž, Ph.D., UTB Zlín, Nakladatelství Martin Stříž, Bučovice (CZ)
RNDr. Ladislav Ševčovič, PhD., FEI, Technická univerzita v Košiciach, SOIT, Košice

Organizačný výbor:

Michal Kaukič, Žilinská univerzita, SOIT, Žilina, predseda
Rudolf Blaško, Žilinská univerzita, SOIT, Žilina
Slavko Fedorik, SOŠ elektrotechnická, Poprad
Lýdia Gábrišová, Žilinská univerzita, Žilina
Roman Hajtmanek, SOIT, Žilina
Aleš Kozubík, Žilinská univerzita, SOIT, Žilina
Tomáš Majer, Žilinská univerzita, SOIT, Žilina
Lubica Micháľková, Žilinská univerzita, Žilina
Peter Mráz, DRU, SOIT, Kremnica
Miloslav Ofúkaný, GeoCommunity, SOIT, Bratislava
Michal Páleník, Freemap Slovakia, SOIT, Bratislava
Martin Šechný, SPŠE, SOIT, Prešov
Peter Štrba, Gymnázium Mikuláša Galandu, SOIT, Turčianske Teplice

Editori: Michal Kaukič
Rudolf Blaško
Pavel Stříž
Ladislav Ševčovič

Vydavateľ: Spoločnosť pre otvorené informačné technológie, Bratislava

ISBN 978-80-970457-2-2

Copyright © 2012 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

Ing. Ján Bábel
Ing. Juraj Bednár
RNDr. Rudolf Blaško, PhD.
RNDr. Ján Buša, CSc.
doc. Ing. Karol Grondžák, PhD.
Ing. Lubor Homolka
Mgr. Michal Kaukič, PhD.
RNDr. Štefan Kovalík, PhD.
RNDr. Aleš Kozubík, PhD.
Ing. Pavol Lupták
doc. Ing. Penka Martincová, PhD.
Ing. Peter Palúch, PhD.
Ing. Libor Sarga
Ing. Pavel Stříž, Ph.D.
RNDr. Ladislav Ševčovič, PhD.
prof. Ing. Miloš Šrámek, PhD.
Ing. Eva Valentíková, PhD.
doc. Ing. Elena Zaitseva, PhD.

SPOLUORGANIZÁTOR KONFERENCIE



FAKULTA RIADENIA
A INFORMATIKY
ŽILINSKÁ UNIVERZITA

SPONZORI KONFERENCIE



PARTNERI KONFERENCIE



Obsah

Úvod	7
Cena SOIT a Liberixu za najlepšie študentské práce	9
Otvorený softvér vo vede a výskume	13
Ján Boháčik, Darryl N. Davis, Miroslav Benedikovič Risk Estimation of Cardiovascular Patients Using Weka	15
Matúš Chochlík Hardware Accelerated Polygonization of Dynamic Implicit Surfaces	21
Michal Chovanec Preemptívny multitasking pre mikrokontroléry s jadrom ARM Cortex M3 ..	27
Michal Kaukič Equal Aperture Angles Curve for some Convex Sets in the Plane	33
Miroslav Kvaššay Importance Measures of Multi-State Systems	39
Pavol Sokol, Peter Pisarčík HoneyNet postavený na báze virtualizácie na úrovni operačného systému ...	45
Otvorený softvér vo vzdelávaní a v IT riešeniach	51
Milan Frátrik Utilization of open hardware platforms in network applications	53
Tomasz Kanik, Marian Kováč NoSQL – non-relational database systems as the new generation of DBMS .	59

Ludovít Mikuš, Petr Ivaniga Používání OSS softvéru vo vyučování predmetu grafika v počítačových aplikáciách	65
Jan Růžička Využití geokettle pro zpracování výměnného formátu registru územní identifikace, adres a nemovitostí	71
T_EX a jeho priatelia	77
Rudolf Blaško Nebojme sa obrázkov v L ^A T _E X-u	79
Rudolf Blaško, Aleš Kozubík L ^A T _E X, teória grafov a trochu umenia	85
Aleš Kozubík Naučím vás kresliť alebo predstavenie balíčka TikZ	91
Zuzana Kozubíková, Aleš Kozubík Tabulkové výpočty v systéme L ^A T _E X	97
Jiří Rybička, Jan Přichystal Volba optimálního programu	103
Pavel Stříž Animujme výstupy z programu GeoGebra	109
Pavel Stříž Sweave: svázání L ^A T _E Xu s jazykem R	115
Abstrakty nerecenzovaných ukážok a prezentácií	121

Vážení čitatelia,

tento zborník je jedným z výstupov štvrtej 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 *Vikend 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.

Otvorený a slobodný softvér (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 chce 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 klasickom, akademickom ponímaní, ale najmä neformálne stretnutie ľudí, ktorých spája záujem o otvorený softvér. 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. Všetky príspevky, uvedené v zborníku, ako aj prezentácie, ktoré na konferencii odznejú, budú voľne prístupné na stránke podujatia, <http://ossconf.soit.sk>.

Záverečný deň konferencie bol venovaný hlavne učiteľom základných a stredných škôl. Po tretíkrát 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.

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 môže neskôr profitovať, či už profesionálne alebo v súkromí. Zároveň sa rozširuje jeho všeobecný rozhľad, lebo nie každý 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 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 mnohých z nás. 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.

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, 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.

CENA SOIT A LIBERIXU ZA NAJLEPŠIE ŠTUDENTSKÉ PRÁCE

Spoločnosť pre otvorené informačné technológie SOIT a Obecně prospěšná společnost zabývající se propagací a podporou svobodných informačních technologií Liberix vyhlásili v roku 2012 už 3. ročník soutěže o nejlepšíu bakalářsku a diplomovou práci súvisiacu s otvoreným softvérom a ďalšími otvorenými technológiami. Cieľom súťaže je 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.

SOIT (SK) a Liberix (CZ)

CENA SOIT A LIBERIXU ZA NAJLEPŠIE ŠTUDENTSKE PRÁCE

Ako každý rok, aj tento rok boli cieľovou skupinou Ceny slovenské a české univerzity so zameraním najmä na informačné technológie. Udeľovanie Ceny SOIT a Liberixu je každoročne súčasťou konferencie *Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach*. V tomto roku bolo do súťaže zaslaných 10 bakalárskych a diplomových prác, ktoré študovali a hodnotili nasledujúci posudzovatelia: *Juraj Bednár, Peter Fodrek, Michal Hučko, Aleš Kozubík, Peter Matkovski, Milan Moravčík, Jakub Ondrušek, Karol Pauchly, Miloš Šrámek, Rastislav Šrámek a Peter Štrba*.

Laureáti Ceny

Posudzovatelia sa rozhodli udeliť tri prvé ceny:

1. cena

Radek Lát, Masarykova univerzita v Brně:

Automatická kontrola nastavení DNSSEC na DNS serverech

Bakalárska práca, vedúci práce: Petr Matoušek

Abstrakt. Práce se zabývá návrhem a implementací nástroje pro kontrolu nastavení bezpečnostního rozšíření DNSSEC na DNS serverech. Cílem je provádět automatickou kontrolu podpisů, sledování používaných šifrovacích algoritmů a informovat o potencionálních nebo nalezených chybách. Práce vznikla ve spolupráci se sdružením CZ.NIC.

1. cena

Vratislav Podzimek, Fakulta informatiky Masarykovy univerzity v Brně:

Korelace příkonových metrik a reálného příkonu

Bakalárska práca, vedúci práce: Marek Grác

Abstrakt. Tato práce se věnuje oblasti správy napájení v operačních systémech GNU/Linux. Jsou zde představeny některé dostupné nástroje umožňující optimalizaci systému a aplikací za účelem snížení příkonu počítače. Dále jsou zde představeny postupy měření příkonu PC a získávání statistik pomocí nástroje Scomes, jimiž bylo získáno více než 2500 údajů. Z těchto údajů jsou v poslední části odvozeny závěry o korelaci statistik Scomes a skutečného příkonu systému.

1. cena

Lubomír Sedlář, Fakulta informatiky Masarykovy univerzity v Brně:

Detekce programovacích chyb v C/C++ programech pod OS GNU/Linux

Bakalárska práca, vedúci práce: Marek Grác

Abstrakt. Cílem této práce je prozkoumat dostupné svobodné nástroje pro statickou analýzu zdrojových kódů v C a C++ dostupné v operačním systému GNU/Linux a porovnat jejich výstupy nad aplikacemi z různých oblastí. Nalezené chyby byly klasifikovány a na jejich základě vytvořeno srovnání dostupných nástrojů.

Ostatné prihlášené práce

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

Peter Aust:

Pokročilé metódy optimalizácie rýchlosti webových aplikácií

Fakulta elektrotechniky a informatiky, STU v Bratislave,
vedúci práce: Zoltán Janík

Martin Líška:

Vyhľadávání v matematickém textu

Fakulta informatiky, Masarykova univerzita v Brně,
vedúci práce: Petr Sojka

Dominik Mačejovský:

Dotykové rozhraní správce stahování FatRat

Fakulta informatiky, Masarykova univerzita v Brně,
vedúci práce: Marek Grác

Martina Marčeková:

Interaktívna vektorová grafika v systémoch pre elektronické vzdelávanie

Fakulta elektrotechniky a informatiky, STU v Bratislave,
vedúci práce: Peter Pásztó

Martin Smolák:

Využitie operačného systému Android na riadenie mobilného robota

Fakulta elektrotechniky a informatiky, STU Bratislava,
vedúci práce: Peter Pásztó

Martin Veselý:

Měření spotřeby energie na různých verzích linuxové distribuce Fedora

Fakulta informatiky, Masarykova univerzita v Brně,
vedúci práce: Marek Grác

Ján Vorčák:

Nástroje pro analýzu kódu v Pythonu a vizualizace jejich výstupu

Fakulta informatiky, Masarykova univerzita v Brně,
vedúci práce: Marek Grác

OTVORENÝ SOFTVÉR VO VEDE A VÝSKUME

Otvorený softvér a otvorené dáta sú významnými pomocníkmi pri vedeckom výskume. Hlavne vďaka nim je zaručená reprodukovateľnosť a kontrolovateľnosť výsledkov. To výrazne skvalitňuje a urychľuje výskum. V mnohých prípadoch je otvorený softvér jedinou možnou alternatívou, pretože napr. komerčné systémy pre riešenie optimalizačných úloh sú veľmi finančne náročné. V zborníku je len niekoľko príspevkov, ale aj tie poukazujú na rôznorodosť problémov, na riešenie ktorých je vhodné použiť otvorený softvér a vývojárske nástroje.

Stretávame sa tu napr. so zaujímavým riešením „návnady“ určenej na odchyťovanie a analýzu útokov na sieťové servery. Používajú sa tu virtuálne počítače, vytvorené pomocou OpenVZ a na sieťovom smerovači je odchyťovač a analyzátor sieťovej prevádzky, založený na otvorenej aplikácii Xplico.

Veľmi potešiteľný je fakt, že do zborníka prispeli dvaja študenti spoluusporiadateľskej fakulty (FRI, Žilinská univerzita). Jeden z nich implementoval preemptívny multitasking (čo je základ prípadného rozšírenia na plnohodnotný operačný systém) pre mikrokontrolér s jadrom ARM Cortex M3. Na vývoj použil len otvorené vývojárske nástroje, prekladač gcc pod OS Linux a program stlink na nahratie kódu do mikrokontroléra.

Ďalší študent publikoval niektoré výsledky svojej diplomovej práce, ktorá sa zaoberala indexami dôležitosti viacstavových systémov (systémy, kde komponenty môžu mať viac stavov, nielen dva – funkčná a nefunkčná). Práca má i vedecký prínos, pretože študent navrhol modifikácie doteraz používaných indexov dôležitosti. Pre porovnanie vlastností rôznych indexov vytvoril vlastný otvorený softvér, ktorý zverejnil na obľubenom serveri OSS vývojárskej komunity, <http://www.github.com>. Použité nástroje – GNU kompilátor g++ a QT knižnica pre tvorbu GUI.

Zastúpená je i oblasť počítačovej grafiky a to príspevkom o hardvérovo akcelerovanej polygonizácii povrchov, definovaných implicitnou funkciou. Navrhnutý algoritmus vychádza z algoritmu „Marching tetrahedrons“, ale je optimalizovaný pre moderné GPU a využíva ich paralelnú architektúru a výpočtový výkon. Algoritmus je implementovaný v C++, používa OpenGL a GLSL, ako aj autorovu vlastnú knižnicu OGLplus (zdrojové kódy sú dostupné na <http://oglplus.org>).

Do oblasti výpočtovej geometrie a numerickej analýzy zasahuje príspevok o krivkách konštantných uhlov viditeľnosti pre niektoré konvexné množiny v rovine. Autor sa zaoberá hlavne jednoduchými prípadmi, keď je možné tieto krivky vyjadriť explicitne alebo parametricky pomocou softvéru pre symbolické výpočty. Používa sa široká škála otvorených nástrojov na báze programovacieho jazyka Python, napr. Sage, IPython, Matplotlib, Numpy, Scipy.

Z úplne inej oblasti je príspevok o odhadovaní rizika pri pacientoch s kardiovaskulárnymi chorobami. Autori vyvíjajú systém na podporu rozhodovania, ktorý by pomohol pacientom viesť produktívny život s minimalizáciou nákladov na liečbu. Rozhodovanie sa robí na základe dolovania údajov, zhromažďovaných v otvorenom softvérovom systéme WEKA. Autori vyvinuli viaceré modely, založené na technikách dolovania údajov a experimentálne ich porovnali.

Môžeme konštatovať, že napriek malému počtu príspevkov, konferencia i zborník presvedčivo ukazujú vhodnosť použitia otvoreného softvéru vo výskume a tešíme sa, že článkov v ďalších ročníkoch bude viac a budú čoraz kvalitnejšie.

Michal Kaukič



OSSConf 2012: 15–20

RISK ESTIMATION OF CARDIOVASCULAR PATIENTS USING WEKA

JÁN BOHÁČIK (UK, SK), DARRYL N. DAVIS (UK) AND MIROSLAV BENEDIKOVIČ (SK)

Abstract. Cardiovascular diseases remain the most prevalent cause of deaths worldwide and their prevention requires major life-style changes using limited health-care resources. Remote decision support for cardiovascular patients seems to allow them to lead a productive life and to minimize the costs of treatment. In this paper, risk estimation of cardiovascular patients on the basis of collected data used in our developing decision-making support system is described. The system makes use of some data mining techniques which are implemented in open source software tool Weka – Waikato Environment for Knowledge Analysis. The integration of Weka with our system, a description of used risk estimation models based on data mining techniques, and experimental results showing the performance of these models are also given.

Key words and phrases. Data mining, risk estimation, cardiology, telehealth.

ODHADOVANIE RISKU U KARDIOVASKULÁRNYCH PACIENTOV S NÁSTROJOM WEKA

Abstrakt. Kardiovaskulárne choroby pretrvávajú ako najdôležitejšia príčina smrti vo svete a ich prevencia si vyžaduje značné zmeny životného štýlu s využitím obmedzených zdrojov pre zdravotnú starostlivosť. Zdá sa, že vzdialená podpora rozhodovania umožňuje kardiovaskulárnym pacientom viesť produktívny život a minimalizovať náklady na liečbu. V tomto článku je popísané odhadovanie riziku u kardiovaskulárnych pacientov použité vo vyvíjanom systéme na podporu rozhodovania na základe zhromaždených údajov. Tento systém využíva niektoré techniky na dolovanie z údajov, ktoré sú implementované v open-source softvérovom nástroji Weka – Waikato Environment for Knowledge Analysis. Rozobraná je aj integrácia nástroja Weka vo vyvíjanom systéme, popis použitých modelov na odhadovanie riziku založených na technikách pre dolovanie z údajov a experimentálne výsledky ukazujúce výkonnosť jednotlivých modelov.

Kľúčové slová. Dolovanie z údajov, odhadovanie riziku, kardiológia, telezdravníctvo.

Introduction

Cardiovascular disease (CVD) is a major cause of disability and premature death throughout the world. Three areas of prevention can be distinguished [6]: a) prevention in the total population; b) prevention in high risk groups; and c) prevention after cardiovascular events. Prevention in high risk groups and prevention after cardiovascular events require major life style changes and medication using

limited health-care resources. As participants of the BraveHealth project, we are interested in continuous and remote monitoring and real time prevention of malignant events for people already diagnosed as subjects at risk. Patients in the project are required to use a wearable unit with sensors and other devices so that regular data can be obtained about them. This is used for decision support on several levels and with several techniques. The work reported in this paper is focused on developing decision support system on the remote server in the BraveHealth project and its data mining techniques for risk estimation of cardiovascular patients. Used open source software solutions are especially discussed.

The paper is organized as follows. Our developing decision-making support system, Weka and its integration with the system are discussed in Section 1. Data mining techniques used from Weka are described in Section 2. Section 3 contains collected cardiovascular data and carried experiments with it. Section 4 concludes this paper.

1. Developing decision support system and Weka

The developing decision support system (DDSS) is intended to provide support for both the clinician and the patient. There is a remote server that receives data about patients in real time. Alert rules [1] and data mining techniques [2,3] are applied on this data. For the purpose of processing, the data is described by attributes which are created according to the classes shown with UML [8] in Figure 1. The data is held in instances created according to the classes shown in Figure 2. The instances are used by data mining techniques and alert rules implemented in the DDSS. Class PilotInstances also allows us to transform the data so that it can be used in data mining tools such as NeticaTM [7] and Weka [5].

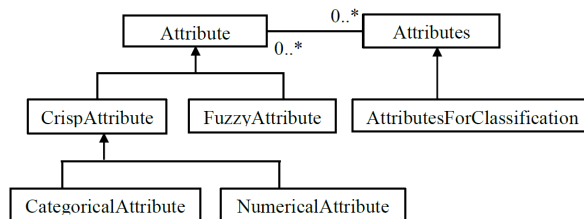


Figure 1. Class diagram for attributes

Weka (Waikato Environment for Knowledge Learning) is an open source data mining system implemented in Java. Releasing Weka as an open source software and implementing it in Java are two factors which ensure that it remains maintainable and modifiable irrespective of the commitment or health of any particular institution or company. Its aim is to provide a comprehensive collection of machine learning algorithms and data preprocessing tools to researchers and

practitioners alike. It can cope with preprocessing and data analysis, classification models, association models, and evaluation metrics. There are three modes of Weka operation: a) GUI, b) commandline and c) Java API [9]. Our developing system uses Java API of Weka for running some data mining techniques so that a part of risk estimation of cardiovascular patients is conducted.

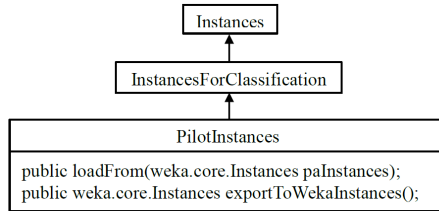


Figure 2. Class diagram for instances

2. Data mining techniques

Given a set of cardiovascular patients (instances) \mathbf{V} where each instance is described by attributes $\mathbf{A} = \{A_1; \dots; A_k; \dots; A_K\}$ and classified into a class $c_j \in \mathcal{C}$, the task is to build a risk estimation model that predicts risk $c_j \in \mathcal{C}$ of an (unseen) cardiovascular patient. Attributes in \mathbf{A} are categorical and numerical. Four principally different data mining techniques are described here: a Bayes network classifier (BNC), a decision tree classifier (DTC), a neural network classifier (MLP) and a nearest neighbor classifier (NNC) [2].

A BNC is based on a Bayesian network which represents a joint probability distribution over a set of categorical attributes. Since it is a distribution over a set of categorical attributes, numerical attributes in \mathbf{A} are discretized and transformed into categorical. It consists of $\langle G; \Theta \rangle$, a directed acyclic graph G consisting of nodes and arcs and conditional probability tables $\Theta = (\theta_{A_1}; \dots; \theta_{A_K})$. The nodes represent attributes in \mathbf{A} and attribute C whereas the arcs indicate direct dependencies. The Bayesian network allows the computation of the (joint) posterior probability distribution of any subset of unobserved assignments of values to attributes in \mathbf{A} , which makes it possible to use for determination of $c_j \in \mathcal{C}$.

A DTC consists of a decision tree which is generated on the basis of instances in \mathbf{V} . There are two types of nodes in the decision tree: a) the root and internal nodes (associated with an attribute $A_k \in \mathbf{A}$); b) leaf nodes (associated with a $c_j \in \mathcal{C}$). Basically, each non-leaf node has an outgoing branch for each possible value $a_{k,l} \in A_k$, $A_k \in \mathbf{A}$ is an attribute associated with the node. Numerical attributes $A_k \in \mathbf{A}$ are discretized. Value $c_j \in \mathcal{C}$ is determined for a new instance using a decision tree, beginning with the root, successive internal nodes are visited until a leaf node is reached. At the root node and at each internal node, a test is

applied. The outcome of the test determines the branch traversed, and the next node visited. Value $c_j \in C$ for the instance is simply c_j of the final leaf node.

A MLP is based on a neural network of interconnected neurons. A neuron takes positive and negative numerical values from other neurons and when the weighted sum of the stimuli is greater than a given threshold value, it activates itself. Its output value is usually a non-linear transformation of the sum of the numerical values. It can also be adapted by some continuous functions.

A NNC assumes cardiovascular patients correspond to points in space R^n . All known cardiovascular patients in \mathbf{V} are remembered when the classifier is being made. When $c_j \in C$ for a new cardiovascular patient is being determined, k -nearest known cardiovascular patients to the new one are found and they are used with a weight. Greater points are given to closer points so that accuracy of determining $c_j \in C$ can be increased.

3. Cardiovascular data and experiments

The following cardiovascular data derived from clinical data collected at two clinical sites (the Hull site of 498 instances and the Dundee site of 341 instances) [4] is used. Describing attributes for cardiovascular patients \mathbf{A} are defined as $\mathbf{A} = \{A_1; \dots; A_k; \dots; A_{17}\} = \{Age; Gender; Heart\ disease; Diabetes; Stroke; Side; Respiratory\ problem; Renal\ failure; ASA\ grade; Hypertension; ECG; Duration; Blood\ loss; Shunt; Patch; Coronary\ artery\ bypass\ surgery; Consultant\}$. *Age* (A_1) and *Gender* (A_2) represent the age and the gender of the patient. *Heart disease* (A_3), *Diabetes* (A_4), *Stroke* (A_5), *Renal failure* (A_8), *Hypertension* (A_{10}), *Shunt* (A_{14}), *Coronary artery bypass surgery* (A_{16}) respectively indicate if any heart disease, diabetes, a stroke, renal insufficiency, a high blood pressure, a shunt, or a coronary artery bypass surgery are present. *Side* (A_6) holds the side of surgery. *ASA grade* (A_9) is used to classify the patient according to the American Society of Anesthesiologists classification. *ECG* (A_{11}) describes electrocardiography, i. e. a transthoracic (across the thorax or chest) interpretation of the electrical activity of the heart over a period of time. *Duration* (A_{12}) is the duration of surgery in hours. *Blood loss* (A_{13}) represents the blood loss in surgery in milliliters. *Patch* (A_{15}) indicates which material is used for by-pass patching in the patient's surgery. *Consultant* (A_{17}) describes the particular consultant employed for the patient's treatment. Cardiovascular patients are classified into two possible categorical values low (c_1) and high (c_2) meaning risk levels. It is denoted by $C = \{c_1; c_2\}$. The values of C are generated according to the following model [4]: a cardiovascular patient is classified into high if her/his death or severe cardiovascular event appears within 30 days after an operation.

The main purpose of the experimental study is to compare data mining techniques described in the previous section, implemented in Weka [5], and used

in our developing decision making support system for risk estimation of cardiovascular patients. The performance of algorithms is measured with sensitivity = $\frac{tp}{(tp+fn)}$, specificity = $\frac{tn}{(tn+fp)}$, positive predictive value = $\frac{tp}{(tp+fp)}$, negative predictive value = $\frac{tn}{(tn+fn)}$ and accuracy = $\frac{(tp+tn)}{(tp+fp+fn+tn)}$. In the formulas, tp/fp/fn/tn is the number of true positives/false positives/false negatives/true negatives. “*C* is *low*” is considered to be negative and “*C* is *high*” is considered positive. Values tp, fp, fn, tn are computed during 10-fold cross-validation.

Model	SEN (%)	SPEC (%)	PPV (%)	NPV (%)	ACC (%)
BNC	7.94	97.48	35.71	85.70	84.03
DTC	4.76	98.60	37.50	85.42	84.51
MLP	15.08	89.62	20.43	85.66	78.43
NNC	15.08	90.18	21.35	85.73	78.90

Table 1. Experimental results with Weka

The results of our experiments are given in Table 1. BNC denotes a Bayesian network which is implemented in Weka as class BayesNet. DTC is a decision tree classifier implemented in Weka as class J48. MLP is a neural network classifier implemented in Weka as class MultilayerPerception. NNC is a nearest neighbor classifier implemented in Weka as class NNGe. SEN is sensitivity, SPEC is specificity, PPV is positive predictive value, NPV is negative predictive value, ACC is classification accuracy. From the point of view of risk estimation of cardiovascular patients, sensitivity and accuracy are most important indicators. Sensitivity measures if high risk patients are not considered low risk in treatment. The best sensitivity (15.08%) is given by MLP and NNC. Classification accuracy measures the proportion of true results (both tp and tn). The best classification accuracy is achieved by DTC (84.51%), however, the other techniques achieve similar results.

4. Conclusions

The use of data mining techniques implemented in open source software Weka for risk estimation of cardiovascular patients in our developing decision support system of BraveHealth were discussed in this paper. Java API of Weka is used after data is transformed into a particular form compatible with Weka in the decision support system. Four principally different data mining techniques are used for risk estimation: a Bayes network classifier, a decision tree classifier, a neural network classifier and a nearest neighbor classifier. Experimental results where these data mining techniques are employed on collected data are also provided. It is expected these data mining techniques will be used together with other data mining techniques and alert rules implemented in the developing decision support system for alert notifications in the BraveHealth system.

Acknowledgment. This work is supported by the European Commission's 7th Framework Program: BRAVEHEALTH FP7-ICT-2009-4, Objective ICT-2009.5.1: Personal Health Systems: a) Minimally invasive systems and ICT-enabled artificial organs: a1) Cardiovascular diseases.

References

- [1] BOHÁČIK, J. – DAVIS, D. N.: *Alert rules for remote monitoring of cardiovascular patients*, Journal of Information Technologies, Vol. 5, Iss. 1, 2012, 16–23, ISSN 1337-7469.
- [2] BOHÁČIK, J. – DAVIS, D. N.: *Data mining applied to cardiovascular data*, Journal of Information Technologies, Vol. 3, Iss. 2, 2010, 14–21, ISSN 1337-7469.
- [3] BOHÁČIK, J. – DAVIS, D. N.: *Estimation of cardiovascular patient risk with a Bayesian network*, Proc. of the TRANSCOM 2011: 9-th European conf. of young research and scientific workers, University of Žilina, Žilina, Slovakia, 2011, 37–40, ISBN 978-80-554-0372-4.
- [4] DAVIS, D. N. – NGUYEN, T. T.: *Generating and Verifying Risk Prediction Models Using Data Mining: A Case Study from Cardiovascular Medicine. Chapter of Data Mining and Medical Knowledge Management: Cases and Applications* (1st edition), IGI Global Inc., 2009.
- [5] HALL, M. – FRANK, E. – HOLMES, G. – PFAHRINGER, B. – REUTEMANN, P. – WITTEN, I. H.: *The WEKA data mining software: An update*, ACM SIGKDD Explorations Newsletter, Vol. 11, Iss. 1, 2009, 10–18, ISSN 1931-0145.
- [6] LIESHOUT, J. v. – WENSING, M., GROU, R.: *Prevention of cardiovascular diseases: The role of primary care in Europe* (1st edition), Electronic book retrieved from EPA Cardio (www.epa-cardio.eu), 2008, 129 p., ISBN 978-90-76316-25-3.
- [7] NORSYS SOFTWARE CORP.: Netica™ Application, <http://www.norsys.com>.
- [8] PAGE-JONES, M.: *Fundamentals of object-oriented design in UML* (1st edition), Addison-Wesley Professional, 1999, 480 p., ISBN-10: 020169946X.
- [9] WITTEN, I. H. – FRANK, E. – HALL, M. A.: *Data Mining: Practical Machine Learning Tools and Techniques* (3rd edition), Morgan Kaufmann, 2011, 629 p., ISBN 978-0-12-374856-0.

Contact addresses

Ing. Ján Boháčik, PhD., EUR ING, Department of Computer Science, Faculty of Science, University of Hull, Cottingham Road, Hull, HU6 7RX, United Kingdom and Department of Informatics, Faculty of Management Science and Informatics, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic,

E-mail address: J.Bohacik@hull.ac.uk, Jan.Bohacik@fri.uniza.sk

Dr. Darryl N. Davis, Department of Computer Science, Faculty of Science, University of Hull, Cottingham Road, Hull, HU6 7RX, United Kingdom

RNDr. Miroslav Benedikovič, Department of Informatics, Faculty of Management Science and Informatics, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic



OSSConf 2012: 21–26

HARDWARE ACCELERATED POLYGONIZATION OF DYNAMIC IMPLICIT SURFACES

MATÚŠ CHOCHLÍK (SK)

Abstract. This paper discusses an algorithm that approximates dynamically changing implicit surfaces with a polygonal representation. The algorithm is loosely based on the “Marching tetrahedrons” algorithm, but is designed to run on modern GPUs, taking advantage of their highly parallelized architecture and computational power.

Unlike algorithms that approximate static surfaces and store the representation for subsequent rendering, this algorithm is better suited for situations where the surface definition changes over time and its polygonal representation at a particular moment in time is rendered only once and then discarded.

Key words and phrases. GPU accelerated rendering, Polygonization, Implicit surfaces, Dynamic surfaces, OpenGL 3.

HARDVÉROVO AKCELEROVANÁ POLYGONIZÁCIA DYNAMICKÝCH POVRCHOV DEFINOVANÝCH IMPLICITNOU FUNKCIOU

Abstrakt. Tento článok opisuje algoritmus, ktorý aproximuje dynamicky sa meniace implicitne definované povrchy polygonálnou reprezentáciou. Tento algoritmus vychádza z algoritmu „Marching tetrahedrons“, ale je optimalizovaný pre moderné GPU (Graphic Processing Unit) a využíva ich paralelnú architektúru a výpočtový výkon.

Na rozdiel od algoritmov aproximujúcich a ukladajúcich statické povrchy pre ďalšie vykresľovanie, prezentovaný algoritmus je vhodnejší v situáciách, kedy sa definícia vykresľovaného povrchu mení dynamicky a jeho polygonálna reprezentácia v jednotlivých okamihoch je použitá len na jednorazové vykreslenie a následne je vymazaná.

Kľúčové slová. Vykresľovanie pomocou GPU, Polygonizácia, Implicitné povrchy, Dynamické povrchy, OpenGL 3.

Introduction

Rendering of surfaces defined implicitly by a scalar function or field in 3D space has been an increasingly important subfield of computer graphics. Such surfaces can be used to represent parts of organic objects, fluids, etc. in computer modelling and simulation or used to visualize 3D representations of the data obtained from medical scanning equipment like CT, MRI, sonographs and others.

An implicit surface is defined by a function $f : R^N \rightarrow R$ as the set of points in space \vec{P} satisfying the following equation:

$$f(\vec{P}) = 0 \tag{1}$$

Most of the polygonization algorithms which approximate the surface with a set of planar polygons [13] can be divided into two distinct categories:

- Cell partitioning (Volume-based): A finite portion of space is tessellated (divided into closed cells that cover all points inside). Polygons are fitted to the surface inside of the cells and finding the intersection of the surface with the edges between the vertices. The vertices are then used to assemble a polygon approximating the surface. One of the first examples of this technique is the *Marching cubes* algorithm [8], later improved by the *Marching tetrahedrons* and its variants [3], [12], [2].
- Surface tracking (Surface-based): A point is found on the surface and new points are incrementally added by examining the local properties (like the gradient or curvature) of the surface and assembled into polygons. The *Marching triangles* [5], [6] and similar algorithms [7], are examples of this approach.

The algorithm described in this paper is inspired by the *Marching tetrahedrons* algorithm [12], but adapted to be executed on the GPUs of modern graphic adapters, instead of regular CPUs. It is best suited for direct visualization of dynamically changing implicit surfaces without storing the polygonal approximation in a long-term.

1. Algorithm overview

Like the *Marching tetrahedrons* algorithm, the proposed algorithm is volume-based. One or more cubic volumes of three dimensional space containing the portion of the surface or the whole surface that is to be visualized is divided by a regular orthogonal grid into smaller uniform, closely-packed cubes. Each of the cubes is then divided into 6 irregular tetrahedrons. These cells are completely covering all points in the original space volume. The resolution of the cubic grid determines the precision of the approximation - the finer the division is, the more precise is the approximation. However, higher resolutions also obviously require more computations.

For every vertex in the cubic grid (i.e. for every cube corner) the value of the function defining the surface is calculated. Then the tetrahedrons are assembled and depending on one of the 16 possible configurations a new triangle or a quadrilateral that approximates the surface inside the tetrahedron is emitted. The possible configurations are determined by the combination of the signs of values of the implicit function (which defines the surface) at the four corners of the tetrahedron. Care has been taken that the winding direction of the generated

triangles is consistent across the surface, which allows to use OpenGL's *polygon culling* to improve performance by removing hidden surfaces before rasterization if the visualized surface is closed.

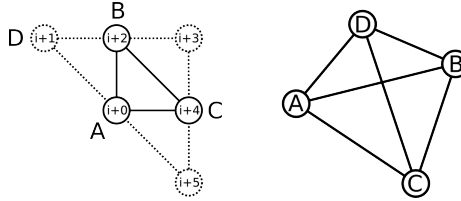


Figure 1. Representation of a single tetrahedral cell by the OpenGL's *triangle with adjacency* geometric primitive

2. Algorithm implementation

The algorithm is implemented in C++ using the *OpenGL* graphic system [11], the *OGLplus* library [9] and in GLSL (the OpenGL shading language) [4]. The implementation takes advantage of the properties of the OpenGL *graphic pipeline* [11], which provides an abstraction layer for modern graphic hardware.

OGLplus is a lightweight open-source library developed by the author of this paper, distributed under the Boost Software License (http://www.boost.org/LICENSE_1_0.txt). It implements a thin object-oriented facade over the OpenGL C API. It provides wrappers which automate resource and object management and make the use of OpenGL in C++ more robust, error-safe and convenient.

The algorithm uses the GPU's vertex processors to pre-calculate the function values at the grid vertices and the geometry processors to assemble the polygons that approximate the surface inside the individual tetrahedrons and to calculate the values of additional attributes at the polygon vertices. The resulting polygonal mesh can then be shaded by the fragment shader in an arbitrary way according to the application needs. Modern GPUs have several dozens of both the vertex and the geometry processors so the computations described above can be highly parallelized and are scalable as the number of these sub-processors increases.

2.1. Input data

The input data for the polygonization of the surface inside of a single cubic volume consists of the vertex coordinates (3D vectors) of the cubic grid that subdivides that volume and of the indices that specify the individual tetrahedrons. The *triangle with adjacency* geometric primitive type is used to represent a single tetrahedron. This primitive consists of 6 vertices, 4 of which are used as the tetrahedron corners as shown on figure 1, the remaining 2 are unused.

This input data is prepared by the application before the rendering starts and it is uploaded from the system memory into a *buffer object* in the graphic adapter's

dedicated memory. The positions are afterwards accessible by the *vertex shader* (described below) through the `Position` *input* variable.

The tetrahedron's edges on which the individual output polygon vertices are located in the individual configurations are defined by pairs of their adjacent vertices. The indices of the edge vertices are hardcoded into a table which is stored in a 1D unsigned integer texture and are accessible through the `Configurations` *sampler* variable. Alternatively the table lookup can be replaced by a sequence of branches and the individual configurations can be hardcoded into the geometry shader. See examples in [10] for complete source code of various versions of the algorithms since the limited scope of this paper does not permit to include them.

2.2. Vertex Shader

The vertex shader is executed once per each input vertex of the cubic grid in parallel on GPU's vertex processors. It consumes the `Position` of the input vertex, moves it by the `GridOffset` and calculates the function value and the normal vector at this position and stores them into the `vertValue` and the `vertNormal` output variables respectively.

2.3. Geometry Shader

The geometry shader is executed once for every *triangle with adjacency* input primitive representing a tetrahedron. The output is a *triangle strip* primitive with 3 or 4 vertices.

The `make_vertex` function emits a new vertex that lies on one of the edges of the tetrahedron specified by its adjacent vertices. It uses linear interpolation to find where approximately the surface intersects with the edge and calculates the position and additional attributes of the output vertex:

```
void make_vertex(int i0, int i1)
{
    float t = vertValue[i0]/(vertValue[i0] - vertValue[i1]);
    gl_Position = CameraMatrix * mix(
        gl_in[i0].gl_Position,
        gl_in[i1].gl_Position,
        t
    );
    geomNormal = mix(vertNormal[i0], vertNormal[i1], t);
    // other per-vertex calculations ...
    EmitVertex();
}
```

The `make_triangle` and `make_quad` functions emit a new triangle or a new quadrilateral primitive respectively. The individual pairs of vertices which specify

the edges on which output vertices are located are specified by their indices that are stored in `v0` and `v1`:

```

void make_triangle(const ivec4 v0, const ivec4 v1)
{
    make_vertex(v0.x, v1.x); make_vertex(v0.y, v1.y);
    make_vertex(v0.z, v1.z); EndPrimitive();
}

void make_quad(const ivec4 v0, const ivec4 v1)
{
    make_vertex(v0.x, v1.x); make_vertex(v0.y, v1.y);
    make_vertex(v0.z, v1.z); make_vertex(v0.w, v1.w);
    EndPrimitive();
}

```

`process_tetrahedron` first creates a boolean vector `i` representing the configuration of signs at the input tetrahedron's vertices. Then it calculates how many negative signs are in the configuration. If the number is 0 or 4 then no output polygon is generated. Otherwise, the index for the `Configuration` lookup table is found and a pair of indices of input vertices specifying the edges intersected (by the surface) in this configuration is retrieved. These are then used as the parameters for the `make_quad` or `make_triangle` functions. One of this functions is chosen based on the number of negative values in the combination. If the number is even, then a quadrilateral is generated, otherwise the result is a triangle.

```

void process_tetrahedron(int a, int b, int c, int d)
{
    ivec4 i=ivec4(vertSg[a],vertSg[b],vertSg[c],vertSg[d]);
    int si=int(dot(i, ivec4(1, 1, 1, 1))) % 4;
    if(si != 0)
    {
        int iv = int(dot(i, ivec4(16, 8, 4, 2)));
        ivec4 v0 = texelFetch(Configurations, iv+0, 0);
        ivec4 v1 = texelFetch(Configurations, iv+1, 0);
        if(si % 2 == 0) make_quad(v0, v1);
        else make_triangle(v0, v1);
    }
}

```

The main function of the shader processes a single input tetrahedron specified by the vertices 0, 2, 4 and 1 of the input primitive as depicted on Figure 1.

3. Conclusion

In this paper we described a polygonization algorithm that is executed by and takes advantage of the properties of modern GPUs. It is suitable for visualizing dynamically changing surfaces defined by an implicit function or a volumetric data-set in real or near-real time for 5×10^6 tetrahedral cells generating approximations consisting of roughly 1.5×10^5 triangles, even with older GPUs (for example nVidia 8500 GT). Complete examples using this algorithm including the source code and screenshots can be found in [10].

References

- [1] ARAUJO, B. R. – JORGE, J. A. P.: *Adaptive polygonization of implicit surfaces*, Computers & Graphics 29, 2005.
- [2] BLOOMENTHAL, J.: *Polygonization of Implicit Surfaces*, Xerox Corporation, 1987.
- [3] CHAN, S.L. – PURISIMA, E. O.: *A new tetrahedral tessellation scheme for isosurface generation*, Computers and Graphics, 22, 1998.
- [4] KESSENICH, J. (Ed.): *The OpenGL Shading Language: Language Version: 4.10, Document Revision: 6*, The Khronos Group Inc., 2010.
- [5] HILTON, A. – ILLINGWORTH, J.: *Marching triangles: Delaunay Implicit Surface Polygonization*.
- [6] HILTON, A. – STODDART, J. I. – WINDEATT, T.: *Marching triangles: range image fusion for complex object modeling* In: International Conference on Image Processing, 1996.
- [7] KARKANIS, T. – STEWART, A.J.: *Curvature-dependent triangulation of implicit surfaces*, IEEE Computer Graphics and Applications, 22, 2001.
- [8] LORENSEN, W. E. – CLINE, H. E.: *Marching cubes: a high resolution 3D surface construction algorithm*, Computer Graphics, 21, 1987.
- [9] The OGLplus project main page. <http://oglplus.org>
- [10] OGLplus online documentation. <http://kifri.fri.uniza.sk/~chochlik/oglplus/html>
- [11] SEGAL, M. – AKELEY, K.: *The OpenGL Graphic System: A Specification (Version 4.2. Core Profile)*, The Khronos Group Inc., 2011.
- [12] TREESE, G. M. – PRAGER, R. W. – Gee, A. H.: *Regularised marching tetrahedra: improved iso-surface extraction*, Computers & Graphics, 23, Elsevier, 1999.
- [13] ZHANG, H. – FENG, J.: *Polygonization of Implicit Surfaces*, State Key Lab of CAD&CG, Zhejiang University, 2007.

Contact address

Ing. Matúš Chochlík, PhD., Department of Informatics, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic,
E-mail address: Matus.Chochlik@fri.uniza.sk



OSSConf 2012: 27–32

PREEMPTÍVNY MULTITASKING PRE MIKROKONTROLÉRY S JADROM ARM CORTEX M3

MICHAL CHOVANEC (SK)

Abstrakt. Článok popisuje realizáciu preemptívneho multitaskingu využitom v experimentálnom operačnom systéme pre mikrokontrolér s jadrom ARM Cortex M3. Vývoj je realizovaný za použitia OSS nástrojov, OS Linux, kompilátor gcc a samotný program na nahranie, stlink.

Kľúčové slová. Multitasking, operačný systém, ARM Cortex.

PREEMPTIVE MULTITASKING FOR ARM CORTEX M3 CORE MICROCONTROLLERS

Abstract. The article describes the implementation of preemptive multitasking in the experimental operating system for microcontrollers with ARM Cortex M3 core. The development is realized using OSS tools, Linux, gcc compiler and the program to load, stlink.

Keywords. Multitasking, Operating System, ARM Cortex.

Úvod

Zariadenia riadené monolitickým mikro počítačom sa stali súčasťou každodenného života. Nasadenie mikrokontrolérov už nie je len v jednoúčelových zariadeniach, ako sú napr. mikrovlonné rúry, či riadenie motora osobného auta, ale už bežne zasahujú do komplexnejších zariadení. Príkladom zo spotrebnej elektroniky sú napríklad mobilné telefóny, hudobné a DVD prehrávače. V priemysle je to napr. riadenie robotov, zber senzorických dát, či rôzne druhy regulátorov.

Akonáhle je požiadavka na viacúlohové využitie mikrokontroléra, dostáva použitie operačného systému s multitaskingom mimoriadnu prioritu pri návrhu. Operačné systémy určené pre PC často neprichádzajú do úvahy, zariadenie je limitované najmä RAM pamäťou. Pre experimentálne overenie jednoduchého systému schopného riadiť proces v reálnom čase, bol zvolený mikrokontrolér STM32F100 [1]. Je to dostupný mikrokontrolér, z rady STM32 patrí k menej výkonným a lacným modelom z rodiny ARM. Je ale možné použiť vyššie modely, napr. STM32F103, prípadne siahnuť po výrobkoch Texas Instruments a ich Stellaris mikrokontroléroch [2].

Použitú jadro CortexM3 predstavuje 32 bitovú architektúru z rodiny ARM. Jadro prebralo mnoho výhodných rysov risc architektúry. Medzi najdôležitejšie prvky zvyšujúce výkon patrí:

- množstvo pracovných registrov, 12 + 4,
- inštrukcie typu load store, minimalizujú prácu s pamäťou,
- nízky počet inštrukcií,
- trojstupňová pipeline,
- pokročilý prerušovací systém.

Vykonávanie inštrukcie prebieha v trojstupňovej pipeline – fetch, decode, execute. V kombinácii s nízkym počtom inštrukcií a load store architektúry je možné pomerne jednoducho predvídať aké prostriedky budú vyžadované nasledujúcou inštrukciou. To umožňuje vysoké zvýšenie výkonu, čiastočnú paralelizáciu vykonania inštrukcií a zmenšiť počet tranzistorov na čipe. Problematike architektúry ARM sa venuje [3].

1. Multitasking

Myšlienka OS je založená na silnej modularite, minimalizácií časových latencií a rešpektovania malej kapacity pamäte. Dôsledkom je koncepcia nano kernelu, kde samotný kernel len prepína úlohy a všetko ostatné je realizované formou knižníc, prípadne úloh. Ovládače hardvéru môžu byť realizované ako bežné volania funkcií, alebo formou posielania správ. Vtedy je ovládač založený ako samostatná úloha, čakajúca na správu. Niektoré situácie je vhodné riešiť čisto len použitím správ a riadiť proces udalosťami. Príkladom sú aplikácie, kde je dôraz kladený na spotrebu. V dobe neprítomnosti udalosti môže byť mikroprocesor v režime spánku. Názorná ukážka tejto myšlienky je uvedená v [4].

Preemptívny multitasking umožňuje prerušiť vykonávaný proces bez jeho aktívnej spolupráce a začať vykonávať ďalší proces. Plánovač cyklicky strieda procesy, napr. podľa priorít a umožňuje zdanlivý beh viacerých procesov súčasne. V prípade operačného systému použitom na mikrokontroléri sa skôr hovorí o vláknach. Celý program je umiestnený vo vnútornej flash pamäti. Pre náročnejšie aplikácie je výhodou umiestniť do flash len zavádzač spolu s jednoduchým monitorovacím programom a pripojiť externú RAM pamäť, do ktorej sa umiestni celý systém, vrátane požadovaných programov.

Samotná myšlienka realizácie multitaskingu na použitom mikrokontroléri je veľmi jednoduchá. Nevyhnutnou požiadavkou je prítomnosť časovača, ktorý umožňuje vyvolať prerušenie. Vo vyvíjanom operačnom systéme je podľa odporúčenia použitý sys tick timer. Ten periodicky vyvoláva prerušovaciu rutinu:

```
void SysTick_Handler(void) __attribute__(( naked ));
```

Následne je nutné uložiť kontext procesora, prepnúť na ďalšiu úlohu, obnoviť kontext a vystúpiť z prerušovacej rutiny. Kontext použitého procesora zahŕňa minimálne hodnoty registrov, programového čítača a stavové registre.

Spomenuté prepnutie kontextu znázorňuje nasledujúci kúsok programu:

```
void SysTick_Handler(void)
{
    save_context();

    //prečítanie hodnoty zásobníka
    uint *sp=(uint*)__get_MSP();

    //uloženie ukazovateľa zásobníka
    if (__current_task__!=SYSTEM_INIT)
        __task__[__current_task__].sp=(uint*)sp;
    else
        __current_task__=0;

    //nájdienie ďalšej pripravenej úlohy
    scheduler();

    //nastavenie zásobníka
    sp=__task__[__current_task__].sp;

    //obnova kontextu a návrat z prerušenia
    restore_context();
}
```

Funkcia `scheduler()` nastaví novú hodnotu pre premennú `current task`. Kľúčová myšlienka multitaskingu je existencia viacerých zásobníkov. Pre každú úlohu je prítomný jeden. Rutina ich cyklicky vymieňa a mení hodnotu fyzického ukazovateľa na zásobník. Prerušená úloha sa vďaka tomu môže vrátiť a pokračovať v pôvodnom stave. Samotný hardvér mikrokontroléra automaticky ukladá niektoré registre, to umožňuje minimalizovať čas potrebný na prepnutie kontextu. Uloženie a obnova kontextu je vďaka tomu veľmi jednoduchá:

```
#define save_context() __ASM volatile ("push {r4-r12}");
```

obnova:

```
#define restore_context()
__ASM volatile ("msr msp, %0\n\t" : : "r" (sp) );
__ASM volatile ("mvn lr,#6");
__ASM volatile ("pop {r4-r12}");
__ASM volatile ("bx lr");
```

Pre plánovanie ďalšej úlohy bol zvolený round robin algoritmus. Pre malý počet úloh je vyhovujúci. Výhodou je jednoduchosť spojená s rýchlosťou nájdienia ďalšej úlohy a prípadná možnosť implementácie prioritného plánovania. Nasledujúci

kúsok programu ukazuje možnú realizáciu plánovača. Podmienkou je aspoň jedna aktívna úloha, inak sa uvedený program zacyklí v nekonečnej slučke. Nakoľko je zmyslom operačného systém koordinovať aspoň jednu úlohu je táto podmienka vždy splnená.

```
void scheduler()
{
    do
    {
        __current_task__++;
        if (__current_task__>=TASK_MAX_COUNT)
            __current_task__=0;
    }
    while ( ( (__task__[__current_task__].flag&TF_CREATED)==0) );
}
```

Algoritmus jednoducho prechádza pole potenciálnych úloh a vyberie prvú existujúcu. Za povšimnutie stojí realizácia čítača bez funkcie modulo. Je to spôsobené nedostupnosťou inštrukcie delenia na niektorých mikroprocesoroch, kompilátor by potom musel použiť časovo náročnejšie riešenie. Podobné riešenia sú typické pre programovanie malých mikrokontrolérov.

Pre užívateľa systému je najdôležitejšia funkcia na vytvorenie novej úlohy. Parametrami je ukazovateľ na main funkciu úlohy, ukazovateľ na zásobník úlohy a jeho veľkosť. Vrátí id úlohy ak sa ju podarilo vytvoriť.

```
uint create_task(void *task_ptr, uint *s_ptr, uint stack_size)
{
    stack_size/=sizeof(uint);

    u32 task_id=TASK_MAX_COUNT;
    u32 res=TASK_MAX_COUNT;

    do
    {
        task_id--;
        CLI();
        if ((__task__[task_id].flag&TF_CREATED)==TF_NULL)
        {
            init_stack();
            __task__[task_id].flag=TF_CREATED;
            res=task_id;
        }
        SEI();
    }
    while ( (task_id!=0) && (res==TASK_MAX_COUNT) );

    return res;
}
```

```
}
```

Cyklus prejde zoznam voľných pozícií pre úlohy a prvú voľnú inicializuje pre úlohu. Nevyhnutné je vykonať porovnanie a rezerváciu atomicky. Z dôvodu prenositeľnosti a modularity systému je atomickosť riešená zákazom a povolením prerušenia (CLI() a SEI()).

2. Vývoj s oss nástrojmi

Kompilátor gcc pre použitý ARM je možné skompilovať pomocou dostupného skriptu `summon-arm-toolchain`. Najdôležitejší je ale nástroj `stlink`, slúžiaci na nahratie skompilovaného binárneho súboru do pamäte mikrokontroléra. Príkaz na nahratie pre mikrokontrolér `stm32f100` má nasledujúci tvar:

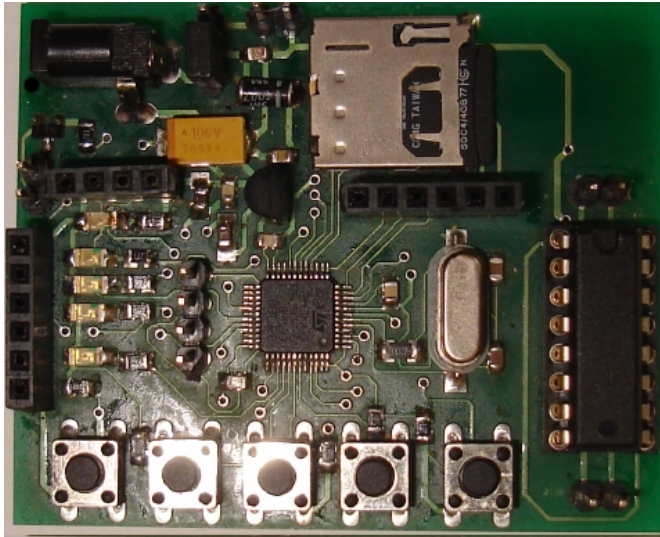
```
~/bin/stlink/flash/st-flash write main.bin 0x8000000
```

Samotné rozhranie je realizované cez USB s využitím dosky STM 32 Discovery [5]. Nahratie prebieha veľmi rýchlo a ihneď po dokončení začne mikrokontrolér vykonávať program. Existuje možnosť umiestniť program do internej RAM pamäte mikrokontroléra. Túto možnosť je vhodné použiť v štádiu ladenia programu, nakoľko flash pamäť má obmedzený počet prepísaní (výrobcovia udávajú 1000 až 10000). Pre kompiláciu programu je vhodné vytvoriť `makefile`, ktorý výrazne urýchli prácu. Výstupom gcc kompilátora je `elf` súbor. Ten je nutné pomocou `objdump` prekonvertovať na jednoduchý binárny súbor, bez akejkoľvek hlavičky. `Objdump` je možné využiť aj na výpis preloženého programu v assembleri. Pri vývoji je to neoceniteľná pomôcka, nakoľko kompilátor môže pri nevhodne nastavených optimalizáciách vytvoriť nefunkčný kód. Príkladom je nutnosť uviesť `volatile` pred každou deklaráciou globálnej premennej používanej v prerušeniach. Inak by mohlo dôjsť k zahodzeniu manipulácie s premennou počas optimalizácie, z dôvodu nezávislosti behu prerušenia od hlavnej slučky programu.

Odskúšanie funkčnosti systému je realizované na jednoduchej pokusnej doske. Tá okrem mikrokontroléra obsahuje ešte: vyvedené vstupno výstupné piny, 4 led a tlačítka, mikro SD kartu, mostík pre dva motory a napájaciu časť. Z vyvedených pinov je dostupná aj USART jednotka, čo umožňuje pripojiť terminálové rozhranie. Mostík pre motory umožňuje spolu s pripojením niekoľkých senzorov realizovať jednoduchého robota. SD karta ponúka možnosti implementácie súborových systémov. Dosku je možné napájať z adaptéra a vďaka `low drop` stabilizátoru aj z jedného `li-pol` článku. Doska neobsahuje žiadne špeciálne prvky a s trochou trpezlivosti je možné osadiť ju aj ručne.

3. Záver

Popísaná realizácia multitaskingu je principiálne použiteľná na akomkoľvek mikrokontroléri. Ukazuje možnosť využiť mikroprocesor na plný výkon a minimalizovať dobu aktívneho čakania. Vďaka jednoduchosti a otvorenému riešeniu ponúka



Obr. 1. Foto hotovej dosky

priestor pre experimentovanie. Rovnako umožňuje pochopiť fungovanie jadra systému a programovať ďalšie časti, napr. semaféry, systém správ alebo ovládače. Možnosti inovácií sú najmä vo voľbe plánovacieho algoritmu, kde sa ponúkajú aj praktické merania výkonnosti systému, času potrebného na prepnutie kontextu a náročnosti jadra na pamäť.

Literatúra

- [1] Výrobca rady stm32f: <http://www.st.com/internet/mcu/class/1734.jsp>.
- [2] Texas Instruments stellaris mikrokontroléry: <http://www.ti.com/ww/en/embedded/stellaris>.
- [3] ARM architektúra: <http://www.root.cz/clanky/mikroprocesory-s-architekturou-arm>.
- [4] Ukážka udalosťami riadeného programovania na mikrokontroléri: http://students.cs.byu.edu/~cs124ta/references/HowTos/HowTo_EventDriven.html.
- [5] Dostupná vývojová doska pre stm32: <http://www.st.com/internet/evalboard/product/250863.jsp>.
- [6] Dostupný operačný systém: https://github.com/michalnand/cortex_m3_os.

Kontaktná adresa

Bc. Michal Chovanec, študent, Fakulta riadenia a informatiky, Žilinská univerzita, Žilina,
E-mailová adresa: Michal.Chovanec@st.fri.uniza.sk



OSSConf 2012: 33–38

EQUAL APERTURE ANGLES CURVE FOR SOME CONVEX SETS IN THE PLANE

MICHAL KAUKIČ (SK)

Abstract. For given convex set K in the plane (not necessarily bounded), we can construct the curve C for which the visibility (aperture) angle of this set has the same, prescribed value. We give the implicit formula for C , discuss some issues concerning practical computations of C and bring several simple examples, when the equal visibility angle curves can be effectively computed explicitly. We conclude with some remarks about possible directions for further research in this area. Extensive use of Open Source software (Sage, Pylab, IPython, ...) is a key feature of this article.

Key words and phrases. Aperture angle, planar convex sets, equal aperture angle.

KRIVKA KONŠTANTNÝCH UHLOV VIDITELNOSTI PRE NIEKTORÉ KONVEXNÉ MNOŽINY V ROVINE

Abstrakt. Pre danú konvexnú množinu K v rovine (môže byť aj neohraničená) môžeme zostrojiť krivku C , pre ktorú je uhol viditeľnosti tejto množiny konštantný, teda nadobúda predpísanú hodnotu. V článku uvádzame implicitný vzorec na výpočet C , zamýšľame sa nad praktickými spôsobmi výpočtu tejto krivky a uvádzame niekoľko jednoduchých príkladov, kedy sa dajú krivky rovnakých uhlov viditeľnosti určiť explicitne. Nakoniec je naznačené, aké sú ďalšie možné smery výskumu v tejto oblasti. V článku je v podstatnej miere použitý otvorený softvér (Sage, Pylab, IPython, ...).

Kľúčové slová. Uhol viditeľnosti, rovinné konvexné množiny, konštantný uhol viditeľnosti.

1. Introductory remarks

Definition 1. Let Q be the convex set in the plane. The aperture (or visibility) angle $\varphi(X, Q)$ of a point $X \notin Q$ with respect to the set Q is the angle of the smallest cone with apex X that contains Q (see e.g., [1], [2]). In this paper we will assume that the set Q is closed, but not necessarily bounded.

For given angle α , $0 \leq \alpha < \pi$ and the convex planar set Q we will denote by $C(\alpha, Q)$ the set of all points X for which the aperture angle $\varphi(X, Q)$ has the value α .

Example 1. Given the wedge W with internal angle ϑ , for fixed angle α , three cases can occur:

1. for $\alpha < \vartheta$ the set $C(\alpha, W)$ is empty,

2. if $\alpha = \vartheta$ then $C(\alpha, W)$ is the wedge V (W rotated by angle π about its apex), see Figure 1,
3. for $\alpha > \vartheta$ the set $C(\alpha, W)$ is the angle V with value $2\alpha - \vartheta$, opposite to wedge W and having common bisector with angle ϑ , see Figure 2.

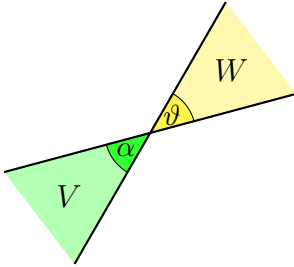


Figure 1. $C(\alpha, W) = V$ for $\alpha = \vartheta$

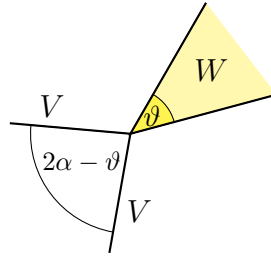


Figure 2. $C(\alpha, W) = V$ for $\alpha > \vartheta$

In the Example 1 the convex set W was unbounded and not strictly convex. The set $C(\alpha, W)$ was two-dimensional in one “singular” case. Next, we will consider unbounded, strictly convex set.

Example 2. Let us define the convex set Q as $Q = \{(x, y) \in \mathbb{R}^2 : y \geq x^2\}$, i.e., Q is the parabola $y = x^2$ together with its interior points.

In this case, it is not difficult to compute the set $C(\alpha, Q)$ explicitly. For arbitrary exterior point $A = (x_1, y_1) \notin Q$ we can construct exactly two tangent lines to Q passing through A . The slopes k_1, k_2 of that tangent lines are

$$k_1 = \tan \varphi_1 = 2 \left(x_1 + \sqrt{x_1^2 - y_1} \right) \quad k_2 = \tan \varphi_2 = 2 \left(x_1 - \sqrt{x_1^2 - y_1} \right), \quad (1)$$

where φ_1, φ_2 are the slope angles of two tangent lines. For given angle α we seek the set of all points A such that

$$\tan(\varphi_1 - \varphi_2) = \frac{\tan \varphi_1 - \tan \varphi_2}{1 + \tan \varphi_1 \tan \varphi_2} = \frac{k_1 - k_2}{1 + k_1 k_2} = \tan \alpha = K. \quad (2)$$

If $k_1 k_2 \neq -1$, we can substitute for k_1, k_2 from (1) and the solutions of resulting quadratic equation give us the explicit formula for the curve $C(\alpha, Q)$

$$y_p(x) = \frac{-K^2 - 2 \pm 2\sqrt{4K^2 x^2 + K^2 + 1}}{4K^2}, \quad (3)$$

where the plus sign is valid for angles $\alpha > \pi/2$ and the sign minus for sharp angles α . For $\alpha = \pi/4$ this formula is simply $y_p(x) = -(2\sqrt{4x^2 + 2} + 3)/4$ and this is the curve with asymptotes $y = \pm x$. For arbitrary $K \neq 0$, the asymptotes of curve given by formula (3) are $y = \pm x/K$.

For the singular case $k_1 k_2 = -1$, i.e., for $\alpha = \pi/2$ we obtain immediately $y_p(x) = -1/4$, which is the directrix of parabola $y = x^2$. From the formula (3)

we can see that for angle $\alpha < \pi/2$ the function $y_p(x)$ is concave and for $\alpha > \pi/2$ we obtain convex function $y_p(x)$.

On the Figure 3 we have plotted the graphs of the functions $y_p(x)$ for several angles α . From the bottom to top curve, the angles are: $\pi/4$, $\pi/3.5$, $\pi/3$, $\pi/2.5$, $\pi/2$, $\pi/1.5$, $\pi/1.2$.

Now, the set $C(\alpha, Q)$ is nonempty for all angles $0 \leq \alpha < \pi$. For compact (in \mathbb{R}^2 – closed and bounded) Q the set $C(\alpha, Q)$ (if nonempty) forms the closed curve, which we will call **constant visibility angle curve**. This curve have been investigated mainly for the special case of convex polygonal sets Q (see [6], [1], [2]). The papers cited are concerned with optimization problems (maximization) for visibility angle. In the paper [3] the authors solved three-dimensional problem of maximization of visibility angle for convex polyhedron, viewed from given line segment. Good sources of information about related problems are the Pirzadeh's Master thesis [4] and the article [5].

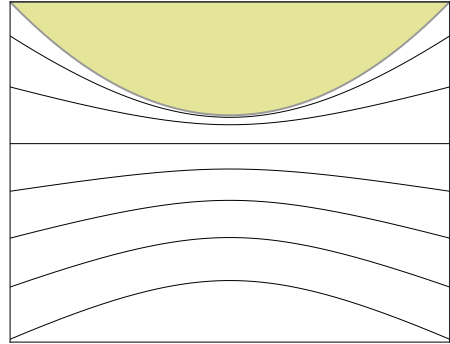


Figure 3. Functions $y_p(x)$ for parabolic region

2. Constant visibility angle curves for convex functions

We can, in principle, compute the visibility angle curve for arbitrary convex function. Let us assume that the function $y = f(x)$ is strictly convex on finite segment $I = [a, b]$. Further, we assume that $f(x)$ has continuous first and second derivative on I .

Given the (arbitrary, but fixed) point $\tilde{x} \in I$ and the angle α , the problem is to find another point $x \in I$ such that (cf. equation (2))

$$\frac{f'(\tilde{x}) - f'(x)}{1 + f'(\tilde{x})f'(x)} = \tan \alpha = K \text{ or } f'(x) = \frac{f'(\tilde{x}) - K}{1 + Kf'(\tilde{x})} = M. \quad (4)$$

For given \tilde{x} , the constant M is known, therefore we have to solve simple equation $F(x) = f'(x) - M = 0$. For some simple functions we can do that analytically but, in general, it is necessary to use one of suitable numerical methods. For example, using Newton method we get the iterational sequence:

$$\text{choose } x_0 \in I, \quad x_{n+1} = x_n - \frac{f'(x_n) - M}{f''(x_n)} \quad \text{for } n = 0, 1, \dots$$

We will not analyze the convergence assumptions here. It will be the subject of further research. In the next section we give two nontrivial examples of computing the constant visibility curve for smooth, bounded convex sets.

3. Constant visibility angle curves for sine and ellipse

Example 3. Let Q be the convex region, described by

$$Q = \{(x, y) \in \mathbb{R}^2 : 0 \leq x \leq \pi, -\sin x \leq y \leq 0\}.$$

We will take into account only visibility “from below”, i.e., from points with negative y coordinate.

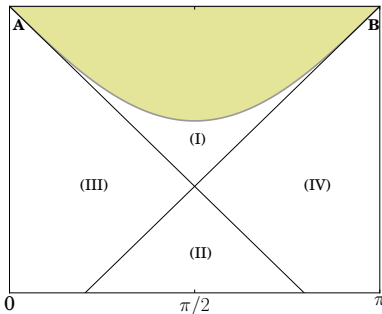


Figure 4. Regions for computing $C(\alpha, Q)$

On the Figure 4 we can see four main regions in the lower halplane. In each of them the computation of $C(\alpha, Q)$ is a bit different. From the symmetry of Q it follows that it is sufficient to investigate only parts of regions with $x \leq \pi/2$. In the region (I) there are only the points of curves $C(\alpha, Q)$ for $\alpha \geq \pi/2$. In the regions (III) resp. (IV) the visibility angles curve always contain points **A** resp. **B**. For the points in region (II) we have $\alpha \leq \pi/2$, the set Q is visible only as the segment AB , thus the curves $C(\alpha, Q)$ are circular arcs.

In this example, the curves $C(\alpha, Q)$ can be computed analytically, but to compute them “by hand” as in Example (2) is very tedious and error-prone. For symbolic computations, we utilized Open Source system Sage [8]. For numerical experiments the programming language Python [9], with user friendly shell IPython [12] and modules Numpy [10], Scipy [7] were used. Finally, nearly all graphics in this article was generated by Python module Matplotlib [11].

Programs used for computations and generation of graphics can be downloaded from <http://feelmath.info/images/constangle/constangle.zip>. For computation of points of the $C(\alpha, Q)$ curves (except circular arcs) we need:

1. take an arbitrary point ξ , $0 \leq \xi \leq \pi/2$ and the tangent line t_1 to Q in the point $T_1 = (\xi, -\sin \xi)$,
2. compute the associated tangent line t_2 to Q such that $\angle(t_1, t_2) = \alpha$,
3. the intersection point $T = (x(\xi), y(\xi))$ of tangent lines t_1, t_2 lies on the curve $C(\alpha, Q)$.

The behaviour of curve $C(\alpha, Q)$ depends on the angle α and it is very interesting. For α near (but less than) straight angle, those curves are convex. For certain obtuse angles close to $\pi/2$ the central part of curves (i.e., the part contained in the region (I)) have waveform shape. Minimal amplitudes of “wave

oscillations” we observed for the angle of about 1.9 radians. For obtuse angles, smaller than (approximately) 1.88 radians the above-mentioned central part is concave. On the figure 5 we can see the curves $C(\alpha, Q)$ for ten an-

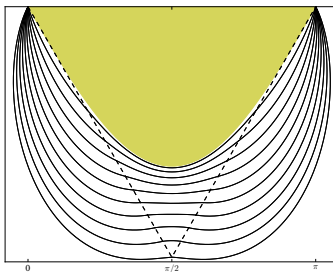


Figure 5. $C(\alpha, Q)$ for obtuse angles

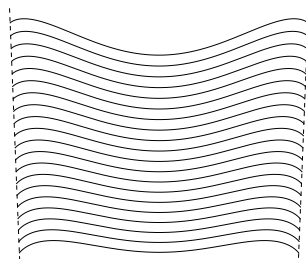


Figure 6. Wave-shaped central parts

gles $\alpha_k = \pi/(1 + 0.1k)$, $k = 1, 2, \dots, 10$. Figure 6 shows central waveform parts of twenty curves $C(\alpha, Q)$ for angles in interval from 1.92 to 1.95 radians (the picture is zoomed vertically). For sharp angles the curves $C(\alpha, Q)$ are not very interesting, for $\alpha \rightarrow 0$ they are close to big circular arcs with central angle 2α .

Example 4. We take the convex region Q , bounded by ellipse

$$x = a \cos \varphi, \quad y = b \sin \varphi, \quad 0 \leq \varphi < 2\pi, \quad a \geq b.$$

Here, using the Sage system, we can derive the formula for $C(\alpha, Q)$ in polar coordinates. Note, that the curve $C(\alpha, Q)$ for $\alpha = \pi/2$ is simply the circle with

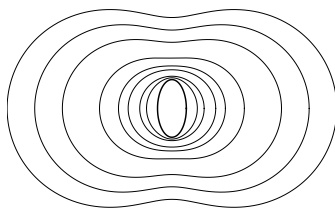


Figure 7. $C(\alpha, Q)$ for “thin” ellipses

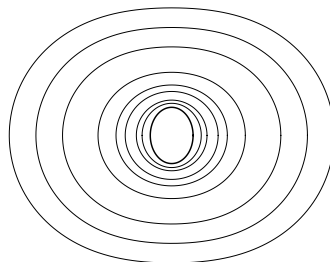


Figure 8. $C(\alpha, Q)$ for “thick” ellipses

radius $r = \sqrt{a^2 + b^2}$. Obviously, for angles close to straight angle, the constant view angle curves are convex. If the ratio b/a is greater than approximately 0.707 (hypothesis: maybe this value is $\sqrt{2}/2$), then the regions bounded by curves $C(\alpha, Q)$ are convex for all α . On the other side, if the ratio b/a is small, the curves $C(\alpha, Q)$ for small angles will approach the two circular arcs with central angle 2α .

Directions for further research

In this paper, we brought only simple examples, but already we have seen some interesting properties of curves of equal aperture angle. The broad research area is to find efficient numerical algorithms for computations of those curves. It would be interesting to approximate $C(\alpha, Q)$ with the corresponding curve for polygonal approximation of boundary of region Q . Here we can explore some results from [1] or [6] (e.g., the method of rotating wedges).

References

- [1] CHEONG, O., GUDMUNDSSON J.: *Aperture-Angle and Hausdorff-Approximation of Convex Figures*, *Discrete Comput. Geom.* (2008) 40, 414–429, DOI 10.1007/s00454-007-9039-5.
- [2] BOSE, P., HURTADO-DIAZ, F., OMAÑA-PULIDO, E., SNOEYINK, J., TOUSSAINT, G.T.: *Some Aperture-Angle Optimization Problems*, *Algorithmica* (2002) 33, 411–435, DOI: 10.1007/s00453-001-0112-9.
- [3] OMAÑA-PULIDO, E., TOUSSAINT, G.T.: *Aperture-Angle Optimization Problems in Three Dimensions*, *J. Math. Model. Algorithms* 1(4) (2002), 301–329.
- [4] PIRZADEH, H.: *Computational geometry with the rotating calipers*, Master thesis, McGill University, Montréal, 1999, available from <http://cgm.cs.mcgill.ca/~orm/thesis.ps.gz>.
- [5] TOUSSAINT, G.T.: *Solving geometric problems with the rotating calipers*, Proceedings of IEEE MELECON'83, Athens, Greece, May 1983, at <http://www-cgri.cs.mcgill.ca/~godfried/publications/calipers.ps.gz>.
- [6] TEICHMANN, M.: *Wedge placement optimization problems*, Master thesis, McGill University, Montréal, 1989, scan available from <http://digitool.library.mcgill.ca/thesisfile55636.pdf>.
- [7] JONES, E., OLIPHANT, T., PETERSON, P. and others: *SciPy: Open Source Scientific Tools for Python*, <http://www.scipy.org>.
- [8] STEIN, W. A. and others: *Sage Mathematics Software (Version 4.8)*, <http://www.sagemath.org>.
- [9] VAN ROSSUM, G., DRAKE, F. L., (eds): *Python Language Reference Manual*, <http://docs.python.org/reference/>.
- [10] NUMPY COMMUNITY: *Numpy 1.6 Reference Guide*, <http://docs.scipy.org/doc/>.
- [11] HUNTER, J. D.: *Matplotlib: A 2D graphics environment*, *Comput. Sci. Eng.*, 3 (9), 2007, 90–95.
- [12] PÉREZ, F., GRANGER, B. E.: *IPython: a System for Interactive Scientific Computing*, *Comput. Sci. Eng.*, 3 (9), 2007, 21–29, <http://ipython.org>.

Contact address

Mgr. Michal Kaukič, CSc., Department of Mathematical Methods, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic, *E-mail address:* Michal.Kaukic@fri.uniza.sk, <http://feelmath.info>



OSSConf 2012: 39–44

IMPORTANCE MEASURES OF MULTI-STATE SYSTEMS

MIROSLAV KVAŠŠAY (SK)

Abstract. Importance analysis allows identify the reliabilities within a system and quantify the criticality (importance) of the system components. Importance Measures estimate peculiarities of influence on a system particular system component. New definitions and algorithms for calculation of Multi-State System Importance Measures are proposed. The numerical experiments based on the Open Source software, which was implemented in C++ programming language, are principal part of the investigation of the Importance Measures.

Key words and phrases. Multi-State System, Importance Analysis in Reliability Engineering, Importance Measures.

INDEXY DÔLEŽITOSTI VIACSTAVOVÝCH SYSTÉMOV

Abstrakt. Cieľom analýzy dôležitosti je identifikovať slabiny systému a kvantifikovať vplyv komponentov na funkčnosť systému. Za týmto účelom boli navrhnuté indexy dôležitosti, ktoré slúžia na určenie vplyvu poruchy komponentu na funkčnosť systému. V článku sú prezentované nové definície vybraných ukazovateľov dôležitosti pre viacstavové systémy. Súčasťou výskumu indexov dôležitosti sú aj numerické experimenty. Tieto boli vykonané v OpenSource programe, implementovanom v jazyku C++.

Kľúčové slová. Viacstavový systém, analýza dôležitosti v teórii spoľahlivosti, indexy dôležitosti.

Introduction

The principal step in reliability analysis is a mathematical interpretation of the initial system. One of the basic techniques that is used in reliability engineering for the system representation is Structure Function (SF) $\phi(\mathbf{x})$:

$$\phi(x_1, x_2, \dots, x_n) = \phi(\mathbf{x}) : \{0, 1, \dots, M_1 - 1\} \times \{0, 1, \dots, M_2 - 1\} \times \dots \\ \dots \times \{0, 1, \dots, M_n - 1\} \rightarrow \{0, 1, \dots, M - 1\}, \quad (1)$$

where

- n is a number of system components,
- M_i is a number of states of the i -th component, for $i = 1, 2, \dots, n$,
- M is a number of states of the system,
- x_i is a state of the i -th component, for $i = 1, 2, \dots, n$ (0 – the component is failed; $(M_i - 1)$ – the component is fully functional).

SF defines the system state depending on components states. Components states are represented by state vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$. The SF (1) can be used for both Binary-State System (BSS) and Multi-State System (MSS). The number $M_i = M = 2$, for $i \in 1, 2, \dots, n$, in (1) for the BSS definition.

Besides SF the system component probabilities $p_{i,s}$ of the initial system are used in reliability analysis. This probability characterizes that the i -th system component has state s :

$$p_{i,s} = P(x_i = s).$$

1. Importance Measures

There are many tools for system reliability estimation. One of them is Importance Analysis that permits to investigate influence of the system component state change to the system state (reliability). Importance Measures (IM) indicate probabilistic properties of the system reliability change that causes by one of the system component state change. Consider some of the IM. The basic IM, which were developed for BSS, are derived from SF of BSS:

- Structural Measure (SM, $I_S(x_i)$) is a probabilistic characteristic of the system structure in point of view of reliability analysis. Therefore this measure calculation doesn't consider the probabilities of the system component states.
- Birnbaum's Measure (BM, $I_B(x_i)$) is the probability that the system is in such a state that component i is critical for the system

In the articles [3], [4] and [2] authors presented different generalizations for aforementioned IM. In the paper [3] author proposed the following definition of the SM of MSS, which is based on logical differential calculus:

$$I_S^Z(x_i = s | \phi = j) = \frac{\rho_{sij}}{|\mathcal{M}(x_i)|}, \quad (2)$$

where

- s is a given state of the i -th component; $s \in \{1, 2, \dots, M_i - 1\}$,
- j is a given state of the system; $j \in \{1, 2, \dots, M - 1\}$,
- ρ_{sij} is a number of nonzero elements of vector of direct partial logic derivative (DPLD) $\frac{\partial \phi(j \rightarrow j-1)}{\partial x_i(s \rightarrow s-1)}$,
- $\mathcal{M}(x_i)$ is a set of all state vectors without the i -th component,
- $|\mathcal{M}(x_i)|$ is a number of all state vectors without the i -th component:

$$|\mathcal{M}(x_i)| = M_1 \dots M_{i-1} M_{i+1} \dots M_n.$$

In the article [4] the following definition of the BM of the MSS have been considered:

$$I_B^1(x_i = s | \phi = j) = |P(\phi = j | x_i = s) - P(\phi = j | x_i = s - 1)|, \quad (3)$$

where $P(\phi = j|x_i = s)$ is a probability that the system is in state j when the i -th component is in state s .

This definition can also be used in the following form, which is not based on the system state but on the system availability:

$$I_B^{v1}(x_i = s|\phi \geq j) = |P(\phi \geq j|x_i = s) - P(\phi \geq j|x_i = s - 1)|. \quad (4)$$

In the paper [2] authors proposed another definition of BM of MSS:

$$I_B^{RC}(x_i|\phi \geq j) = \frac{\sum_{s=0}^{M_i-1} |P(\phi \geq j|x_i=s) - P(\phi \geq j)|}{M_i-1} \quad (5)$$

Next, we introduce average IM that are computed as:

$$\begin{aligned} \overline{I_S^Z}(x_i) &= \frac{\sum_{s=1}^{M_i-1} \sum_{j=1}^{M-1} I_S^Z(x_i=s|\phi=j)}{(M_i-1)(M-1)}, & \overline{I_B^{v1,=}}(x_i) &= \frac{\sum_{s=1}^{M_i-1} \sum_{j=1}^{M-1} I_B^{v1}(x_i=s|\phi=j)}{(M_i-1)(M-1)}, \\ \overline{I_B^{v1,\geq}}(x_i) &= \frac{\sum_{s=1}^{M_i-1} \sum_{j=1}^{M-1} I_B^{v1}(x_i=s|\phi \geq j)}{(M_i-1)(M-1)}, & \overline{I_B^{RC}}(x_i) &= \frac{\sum_{j=1}^{M-1} I_B^{RC}(x_i|\phi \geq j)}{(M-1)}. \end{aligned}$$

This measures indicate global impact of failure of the i -th component on the system performance or on the system availability. We made numerical experiments for these measures based on the Open Source software *rt_dt* (https://www.github.com/kvassay/rt_dt), which was implemented in C++ programming language.

2. Systems

Three systems are used for investigation of specifics of the SM (2) and BM (3), (4) and (5). Reliability block diagrams of these systems are in Figures 1 and 2.

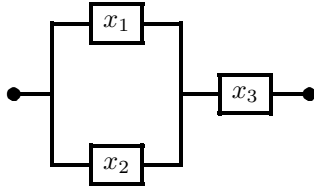


Figure 1. Systems A and B



Figure 2. System C

- **System A** consists of 3 components. Probabilities of the system component states are in table 1. Structure Function of the system is defined as $\phi(\mathbf{x}) = \min(\max(x_1, x_2), x_3)$.

	State		
Component	0	1	2
x_1	0.2	0.4	0.4
x_2	0.3	0.3	0.4
x_3	0.4	0.1	0.5

Table 1. Probabilities of states of components of the system A

- **System B** consists of 3 components with component state probabilities that are in table 2. The system SF is defined in table 3. This system is very similar to the system used in [4].

	State			
Component	0	1	2	3
x_1	0.2	0.8	–	–
x_2	0.3	0.7	–	–
x_3	0.1	0.3	0.4	0.2

Table 2. Probabilities of states of components of the system B

Components		x_3			
x_1	x_2	0	1	2	3
0	0	0	0	0	0
0	1	0	1	1	1
1	0	0	1	1	1
1	1	0	1	2	3

Table 3. Structure function of the system B

- **System C** consists of 2 components and has the SF $\phi(\mathbf{x}) = x_1x_2$. The system component state probabilities are in table 4.

	State			
Component	0	1	2	3
x_1	0.4	0.6	–	–
x_2	0.3	0.2	0.4	0.1

Table 4. Probabilities of states of components of the system C

3. Analysis of results of numerical experiments

Results of numerical experiments for system A, B and C are presented in Tables 5, 6 and 7. Detail analysis of these results has been performed in [1]. This analysis showed that different mathematical interpretations of IM for MSS cause unequal values of IM. But correction of the IM allows deciding this problem. New interpretation of the IM has been proposed and proved in [1]:

$$I_S(x_i = s | \phi = j) = \frac{\sum_{(\mathbf{x}, x_i) \in \mathcal{M}(x_i)} \mathbf{I}(\phi(\mathbf{x}|x_i=s)=j)(1-\mathbf{I}(\phi(\mathbf{x}|x_i=s-1)=j))}{|\mathcal{M}(x_i)|} = \frac{\rho\left(\frac{\partial \phi(j \setminus x_i)}{\partial x_i(s \rightarrow s-1)}\right)}{|\mathcal{M}(x_i)|},$$

$$I_S(x_i = s | \phi \geq j) = \frac{\sum_{(\mathbf{x}, x_i) \in \mathcal{M}(x_i)} (\mathbf{I}(\phi(\mathbf{x}|x_i=s) \geq j) - \mathbf{I}(\phi(\mathbf{x}|x_i=s-1) \geq j))}{|\mathcal{M}(x_i)|} = \frac{\rho\left(\frac{\partial\phi(\searrow j \searrow)}{\partial x_i(s \rightarrow s-1)}\right)}{|\mathcal{M}(x_i)|},$$

$$I_B(x_i = s | \phi = j) = \sum_{(\mathbf{x}, x_i) \in \mathcal{M}(x_i)} P(\phi(\mathbf{x}|x_i = s) = j)(1 - \mathbf{I}(\phi(\mathbf{x}|x_i = s - 1) = j)) = \\ = \mathbf{P}(x_i) \odot \frac{\partial\phi(j \searrow)}{\partial x_i(s \rightarrow s-1)},$$

$$I_B(x_i = s | \phi \geq j) = P(\phi \geq j | x_i = s) - P(\phi \geq j | x_i = s - 1) = \mathbf{P}(x_i) \odot \frac{\partial\phi(\searrow j \searrow)}{\partial x_i(s \rightarrow s-1)},$$

where

- (\mathbf{x}, x_i) is a state vector without the i -th component,
- $\mathbf{I}(\cdot)$ is an indication function that return 1 if an expression, which is the argument of this function, is true, else return 0,
- $\rho(\cdot)$ is a function that return number of nonzero elements of a vector, which is the argument of this function,
- $\frac{\partial\phi(j \searrow)}{\partial x_i(s \rightarrow s-1)}$ or $\frac{\partial\phi(\searrow j \searrow)}{\partial x_i(s \rightarrow s-1)}$ are vectors of the DPLD that identifies cases, in which change of component state from s to $(s - 1)$ results in change of system state or reliability from j to lower,
- $\mathbf{P}(x_i)$ is a vector of probabilities that correspond to state vectors (\mathbf{x}, x_i) , which are used in the calculation of the DPLD $\frac{\partial\phi(j \searrow)}{\partial x_i(s \rightarrow s-1)}$ or $\frac{\partial\phi(\searrow j \searrow)}{\partial x_i(s \rightarrow s-1)}$.

In [1] average IM have been also introduced. The average SM ($\overline{I_S^{\overline{=}}}(x_i)$ or $\overline{I_S^{\overline{>}}}(x_i)$) indicates an average relative frequency of boundary vectors of a given component in regard to a number of system states. The average BM ($\overline{I_B^{\overline{=}}}(x_i)$ or $\overline{I_B^{\overline{>}}}(x_i)$) defines an average probability that a given component is critical for the system with respect to a number of system states. These measures was also used in numerical experiments and their results are shown in tables 5, 6 and 7.

A	$\overline{I_S^{\overline{=}}}(x_i)$	$\overline{I_S^{\overline{>}}}(x_i)$	$\overline{I_S^{\overline{>}}}(x_i)$	$\overline{I_B^{\overline{=}}}(x_i)$	$\overline{I_B^{v1,=}}(x_i)$	$\overline{I_B^{\overline{>}}}(x_i)$	$\overline{I_B^{v1,\geq}}(x_i)$	$\overline{I_B^{RC}}(x_i)$
x_1	0.111	0.111	0.111	0.120	0.195	0.120	0.120	0.159
x_2	0.111	0.111	0.111	0.105	0.180	0.105	0.105	0.144
x_3	0.361	0.361	0.361	0.395	0.555	0.395	0.395	0.569

Table 5. Average SM and BM of components of the system A

4. Conclusion

In this paper different tools for IM calculation are compared. The principal step of the presented results depends on numerical experiments. These experiments were done in Open Source software *rt_dt* (https://www.github.com/kvassay/rt_dt) that was implemented in C++ programming language. The analysis of experiments allows to obtained

B	$\overline{I_S^=}(x_i)$	$\overline{I_S^Z}(x_i)$	$\overline{I_S^>}(x_i)$	$\overline{I_B^=}(x_i)$	$\overline{I_B^{v1,=}}(x_i)$	$\overline{I_B^>}(x_i)$	$\overline{I_B^{v1,>}}(x_i)$	$\overline{I_B^{RC}}(x_i)$
x_1	0.208	0.167	0.250	0.230	0.190	0.277	0.273	0.277
x_2	0.208	0.167	0.250	0.220	0.260	0.273	0.273	0.273
x_3	0.139	0.139	0.139	0.229	0.353	0.229	0.229	0.337

Table 6. Average SM and BM of components of the system B

C	$\overline{I_S^=}(x_i)$	$\overline{I_S^Z}(x_i)$	$\overline{I_S^>}(x_i)$	$\overline{I_B^=}(x_i)$	$\overline{I_B^{v1,=}}(x_i)$	$\overline{I_B^>}(x_i)$	$\overline{I_B^{v1,>}}(x_i)$	$\overline{I_B^{RC}}(x_i)$
x_1	0.250	0.083	0.500	0.233	0.233	0.433	0.433	0.433
x_2	0.167	0.167	0.167	0.200	0.067	0.200	0.200	0.320

Table 7. Average SM and BM of components of the system C

new definitions of the IM and to formulate next objectives. The SM in [3] ignores cases in which deterioration of a component state can cause deterioration of the system in more than one state. The BM in [4] can be used only in the form that is based on a system availability (4). Finally, the BM in [2] has been derived from a formula for calculation of the BM of BSS and so its meaning is different from the original meaning of the BM.

References

- [1] KVAŠŠAY, M.: *Analýza dôležitosti prvkov viacstavových systémov v teórii spoľahlivosti*, Diplomová práca, Žilinská univerzita, Žilina, 2012.
- [2] RAMIREZ-MARQUEZ, J. E. – COIT, W. D.: *Composite Importance Measures for Multi-State Systems with Multi-State Components*, IEEE Transactions on Reliability, September 2005, Iss. 54, **3**, 517–529.
- [3] ZAITSEVA, E.: *Importance Analysis of Multi-State System by Tools of Differential Logical Calculus*, Reliability, Risk and Safety. Theory and Applications, London, CRC Press, 2010, Iss. 1, 1579–1584, ISBN 0-415-55509-4.
- [4] ZAITSEVA, E.: *Importance Analysis of a Multi-State System Based on Multiple-Valued Logic Methods*, Recent Advances in System Reliability: Signatures, Multi-state Systems and Statistical Inference, London, Springer, 2012, 113–134, ISBN 1-4471-2206-2.

Contact address

Bc. Miroslav Kvaššay, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic,
E-mail address: MiroslavKvassay@zoznam.sk



OSSConf 2012: 45–50

HONEYNET POSTAVENÝ NA BÁZE VIRTUALIZÁCIE NA ÚROVNI OPERAČNÉHO SYSTÉMU

PAVOL SOKOL (SK) A PETER PISARČÍK (SK)

Abstrakt. Sieťová forénzna analýza predstavuje jedno zo základných odvetví digitálnej forénznej analýzy. Hlavný zámer sieťovej forénznej analýzy sa sústreďuje na monitorovanie a analýzu sieťovej prevádzky za účelom identifikácie sieťových prienikov, ako aj za účelom získavania právnych dôkazov. V článku predstavujeme špecializovaný počítačový systém prispôbený na zber forénznych údajov – honeynet. Tento systém je tvorený základnými stavebnými prvkami – honeypotmi, ktorých úlohou je identifikovať konkrétnu činnosť útočníka v procese prieniku na sieťové servery, rovnako však zaznamenať postup a metódy pri infiltrácii. V článku sa venujeme výhodám, ktoré predstavuje honeynet využívajúci technológiu virtualizácie na úrovni operačného systému a rozoberáme jednotlivé špecifiká takéhoto systému.

Kľúčové slová. Sieťová forénzna analýza, honeynet, honeypot, virtualizácia, OpenVZ.

HONEYNET BASED ON OPERATING-SYSTEM LEVEL VIRTUALIZATION

Abstract. Network forensic analysis is one of the basic branches of the digital forensic analysis which deals with monitoring and analysing network traffic order to identify network intrusions and obtain legal evidence. In article we present the special system adapted to collect forensic data from honeynet. It consists of several honeypots designed to identify the attacker intrusions on a network server and record the procedures and methods for infiltration of these servers. Created virtual machines are honeypots with the network services. They are the object of forensic analysis. In this article we deal with the benefits of the honeynet based on operating system level virtualization and emphasise the specifics of such system.

Keywords. Network forensic, Honeynet, Honeypot, virtualization, OpenVZ.

1. Etymológia pojmov

Pod sieťovou forénznou analýzou rozumieme použitie kritických empirických metód pre zber, zlučovanie, identifikáciu, koreláciu, analýzu a dokumentáciu digitálnej evidencie z niekoľkých spracovaných a prenositeľných digitálnych zdrojov za účelom identifikácie plánovaného zámeru neoprávnených činností znamenajúcich narušenie, porušenie alebo skompromitovanie systémových komponentov. Rovnako pod pojmom sieťovej forénznej analýzy rozumieme poskytovanie informácií s cieľom zotavenia sa z vyššie popísaných činností [4].

Honeypotom rozumieme elektronický informačný zdroj, ktorého zameranie a hodnota spočíva v tom, aby bol sondovaný, napadnutý, resp. ohrozený [5]. Honeypoty môžeme deliť podľa niekoľkých aspektov. Podľa účelu ich použitia rozoznávame výskumné a produkčné honeypoty. **Výskumný honeypot** je určený výslovne pre akademické použitie, a neprináša žiadnu priamu hodnotu pre inštitúciu [3]. Na druhej strane, **produkčný honeypot** je využívaný v rámci inštitúcie na ochranu kybernetického prostredia inštitúcie.

Podľa miery interakcie poznáme nízkoúrovňové (low-interaction) a vysokoúrovňové honeypoty (high-interaction). **Nízko-úrovňový honeypot** simuluje len vybrané sieťové služby. Cieľom **vysoko-úrovňového honeypotu** je poskytnúť útočníkovi interakciu so skutočným operačným systémom, v ktorom nie je nič simulované alebo obmedzené.

Pojmom **Honeynet** definujeme špeciálnu sieťovú štruktúru (sieť honeypotov), ktorej cieľom je dosiahnuť **kontrolu nad dátami, zachytávanie dát a zbieranie dát**. V súčasnosti existujú základné tri generácie honeynetov, pričom najnovšia, III. generácia, je široko používaná výskumnými skupinami po celom svete a považuje sa dnes za najlepší model detekcie a identifikácie prieniku do sieťového systému.

Virtualizáciou rozumieme technológiu, ktorá umožňuje spúšťanie niekoľkých logicky oddelených virtuálnych strojov na jednom fyzickom stroji. Rozoznávame niekoľko typov virtualizácie, a to plnú virtualizáciu a emuláciu, paravirtualizáciu a virtualizáciu na úrovni operačného systému.

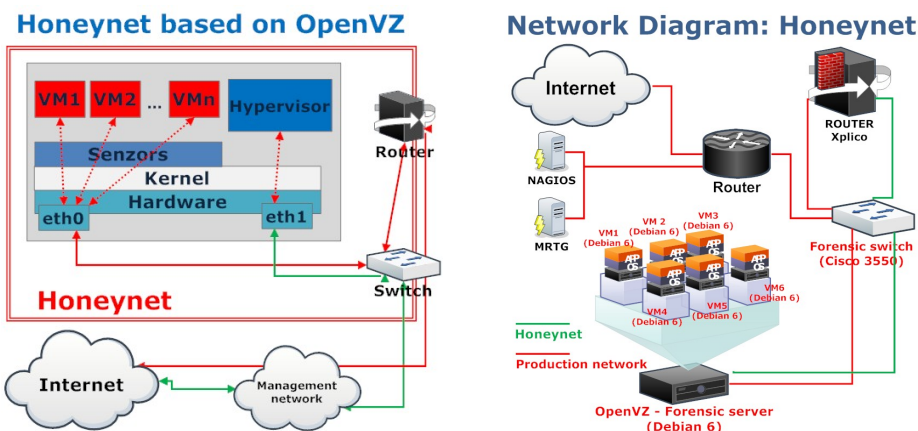
Virtualizácia, vzhľadom na možnosť tvorby nezávislých výpočtových jednotiek, je aktuálne takmer neoddeliteľnou súčasťou budovania honeynetov. Honeynet, ktorého honeypoty sú vytvorené za použitia technológie virtualizácie nazývame **virtuálnym honeynetom**.

2. Navrhovaný systém

2.1. Dizajn architektúry

Nami prezentovaný honeynet patrí do skupiny výskumných honeynetov a pozostáva z niekoľkých vysokoúrovňových honeypotov. Ako typ virtualizácie sme zvolili virtualizáciu na úrovni operačného systému. Z hľadiska účinnosti, ide o honeynet III. generácie. Základnú časť honeynetu tvorí jeden fyzický server, na ktorom bežia virtuálne stroje, ktoré predstavujú jednotlivé honeypoty a hypervízor, ktorý pre účely honeynetu plní funkciu riadiaceho a úložného miesta pre honeypoty.

Nami navrhovaný systém honeynetu, z hľadiska architektúry, nesie prvky paralelnej architektúry [6]. Podstatou tejto architektúry je to, že útočník môže útočiť na honeynet a produkčný systém súčasne. Oba systémy sú však od seba navzájom sieťovo oddelené. Informácie získané o útočníkovi a jeho aktivitách sú zaznamenávané do nezávislého a bezpečného systému, u ktorého sa predpokladá kvalitné



Obr. 1. Schéma navrhovaného honeynetu Obr. 2. Honeynet postavený na OpenVZ

zabezpečenie (v základe napr. firewallom). Nami navrhovaný honeynet sa z hľadiska architektúry líši od základnej paralelnej architektúry tým, že informácie sú ukladané priamo v honeynete bez použitia osobitného záznamového zabezpečného systému. Tiež dochádza k čiastočnému spojeniu honeynetu a produkčného systému, keďže prístup k týmto informáciám je možný len z podsiete produkčnej siete – manažovacej siete.

Okrem tohto fyzického servera súčasťou honeynetu je tiež sieťový prepínač a sieťový smerovač. Účelom **sieťového prepínača** je oddelene priviesť honeynet a manažment siete k správnym sieťovým rozhraniam na fyzickom serveri. Naproti tomu **sieťový smerovač** predstavuje dôležitú súčasť honeynetu, keďže okrem smerovania plní funkciu aj zberača údajov. Smerovač teda obsahuje špeciálny sniffer – odchytač a analyzátor sieťovej prevádzky – ktorého úlohu zastáva aplikácia s otvoreným zdrojovým kódom s názvom **Xplico**.

Sieťová prevádzka na fyzickom serveri je zabezpečovaná pomocou dvoch základných sieťových rozhraní, sieťových kariet – eth0 a eth1. V rámci sieťového rozhrania eth0 je, pre každý virtuálny stroj, vytvorené virtuálne sieťové rozhranie, ktorému podobne, ako rozhraniam eth0 a eth1 je pridelená verejná IPv4 adresa. Najdôležitejším miestom pri implementácii tohto systému sú tzv. **snímače (senzory)**, ktorých úlohou je bezpečne zaznamenávať určitý typ údajov. Nakoľko úlohou honeynetu je predstieranie reálneho produkčného systému, aj návrh senzorov musí spĺňať podmienku diskretnosti, čiže útočník nesmie za žiadnych okolností zistiť ich existenciu. Z tohto dôvodu sú tieto snímače umiestnené medzi jadrom operačného systému a virtuálnymi strojmi –honeypotmi. Honeypot o existencii týchto snímačov nevie a ani k nim nevie pristupovať. Ďalšou požiadavkou je, aby tieto snímače uchovávali získané dáta na bezpečnom mieste tak, aby nedošlo pri ich zbere a v čase analýzy k ich zmene, strate alebo zničeniu. Táto požiadavka je zabezpečená tým, že tieto údaje sú uložené na hypervízore

a virtuálne stroje k nim nemôžu pristupovať. K hypervízoru je možné pristupovať len zo špeciálnej manažovacej siete. Schému navrhovaného systému uvádzame na obr. 1.

2.2. Implementácia pomocou Open-source softvéru

V predchádzajúcej podkapitole príspevku sme predstavili základný návrh architektúry virtuálneho honeynetu. V tejto časti sa zameriame na detailnejšiu špecifikáciu navrhutej implementácie. K implementácii honeynetu založenom na báze virtualizácie na úrovni operačného systému sme použili výhradne systém s otvoreným zdrojovým kódom, a to z dôvodu bezplatnej licencie a možnosti modifikácie. Obrázok 2 zobrazuje implementáciu navrhovaného systému pomocou softvéru **OpenVZ**, ktorého popisu a využitiu v navrhovanom systéme sa budeme venovať v nasledujúcich odsekoch.

OpenVZ [7] umožňuje vytvoriť niekoľko bezpečných, viacúrovňovo izolovaných kontajnerov na úrovni Linuxového jadra na jednom fyzickom stroji, čo umožňuje lepšie využitie systémových prostriedkov. Každý kontajner je spustený a vykonávaný ako samostatný server s vlastnou IP adresou, vlastnými používateľmi, procesmi, súborovým systémom a pod. Tento softvér je vydaný pod GNU GPL licenciou, ktorá umožňuje prípadnú modifikáciu zdrojového kódu. Základnou vlastnosťou virtualizačnej vrstvy implementovanej v OpenVZ je skutočnosť, že kontajnery jedného fyzického servera sú od seba vzájomne úplne izolované. Každý kontajner sa javí a správa ako samostatný výpočtový systém, ktorý má všetky charakteristiky linuxového servera (napr. spúšťacie skripty). Dôležitou vlastnosťou tejto virtualizačnej vrstvy je aj možnosť behu softvéru vo vnútri kontajnera bez špecifických úprav. Napokon špecifickou vlastnosťou je oprávnenie správcu kontajnera (používateľ root) zmeniť akýkoľvek konfiguračný súbor a inštalovať ďalší softvér, resp. celkovo koordinovať beh kontajnera.

Pri implementácii hypervízora, ako aj virtuálnych strojov sme použili distribúciu GNU/Linuxu – Debian Squeeze, ktorá obsahuje jadro s už implementovaným OpenVZ. **Xplico** [8] predstavuje sieťový analytický forénzny nástroj, ktorého cieľom je extrahovať a analyzovať dáta získané zo sieťovej prevádzky. Xplico systém bol vyvinutý v niekoľkých programovacích jazykoch (C, Python, PHP a JavaScript) a je zložený zo štyroch makrokomponentov.

Bezpečnosť navrhovaného systému je založená na bezpečnosti virtualizačnej platformy OpenVZ a jadra operačného systému GNU/Linux.

3. OpenVZ a sieťová forénzna analýza

V súčasnej sieťovej forénznej analýze sa vyskytuje niekoľko oblastí problémov, ktorými je nevyhnutné sa zaoberať. V nasledujúcom texte prinášame stručný popis týchto problémov spoločne s ich vzťahom k virtualizácii na úrovni operačného systému a použitej implementácii OpenVZ.

3.1. Neodhalenie honeynetu a senzorov

Základnou požiadavkou kladenou na honeypot je, že musí odzrkadľovať reálny systém. Z tohto dôvodu nami navrhovaný honeynet je umiestnený popri prevádzkovej sieti. Okrem toho, každý honeypot je zvlášť koordinovaný jadrom OS simulujúc reálnu činnosť. Vzhľadom na použitie OpenVZ virtualizačnej technológie je možné ľahšie koordinovať činnosť všetkých honeypotov z jedného miesta – jadra OS a pridružených aplikácií bežiacich na hypervízore. Honeypoty nemajú žiadnu možnosť identifikovať sa ako honeypoty.

3.2. Zdroje dát

Počítačová sieť má niekoľko možných zdrojov dát, ktoré je potrebné v honeynete zaznamenávať pre ďalšie spracovanie (analýzu). Vzhľadom na použitie OpenVZ technológie, ktorá disponuje dvoma typmi virtuálnych sieťových rozhraní (venet, veth), je možné jednoduchšie zaznamenávanie sieťových dát, a to priamo na hypervízore.

3.3. Granularita dát

Ďalším problémom je rozhodnutie, do akých detailov je potrebné zaznamenávať údaje. Vzhľadom na zameranie nášho honeynetu, primárne dochádza k zaznamenávaniu udalostí súvisiacich s činnosťou prostredníctvom terminálu, ako tiež zaznamenávanie a analýza dátových segmentov protokolu aplikačnej vrstvy, HTTP. Zber sa vykonáva čiastočne na hypervízore a čiastočne systémom Xplico na smerovači.

3.4. Bezpečnosť systému a integrita dát

Kritické miesto sieťovej forenznej analýzy predstavuje bezpečnosť systému a integrita dát. Práve použitie OpenVZ technológie umožňuje zabezpečiť bezpečný, nedetekovateľný a efektívny zber údajov z kontajnerov, a to aj vďaka použitiu technológie inotify na hypervízore.

3.5. Otázky súkromia a možnosti použitia získaných dát ako dôkazov

Otázkami ochrany osobných údajov nie je potrebné sa zaoberať nakoľko v našom prípade ide o výskumný honeynet, ktorý nebude pracovať s osobnými údajmi.

3.6. Forenzna analýza za behu

Tradičná forenzna analýza využíva statickú analýzu, ktorej podstata spočíva vo vypnutí skúmaného stroja a následnom vytvorení bitovej kópie dát záznamových médií. Dôvodom pre tento typ analýzy je lepšia ochrana získaných dát [2].

Nakoľko však technológia OpenVZ, ako už bolo vyššie naznačené, umožňuje na úrovni hypervízora aktívny priamy prístup do súborových systémov bežiacich virtuálnych strojov, je možné s minimálnym úsilím realizovať forénznu analýzu počas behu všetkých kontajnerov.

4. Zhrnutie

Hlavným cieľom tohto príspevku bolo poukázať na možnosť použitia iných virtualizačných technológií, než ktoré sa v súčasnosti spravidla pri návrhu honey-netov používajú. Aktuálne sa využívajú dva typy virtualizácie, a to plná virtualizácia a paravirtualizácia. V príspevku sme poukázali na pozitívny prínos v prospech forénznej sieťovej analýzy pri použití virtualizácie na úrovni operačného systému, konkrétne implementácie OpenVZ. Honeynet postavený na virtualizačnej technológii produktu OpenVZ predstavuje lacný a efektívny systém pre zber dát a ich následnú analýzu. V príspevku sme zároveň poukázali na značné výhody virtualizácie na úrovni operačného systému oproti tradične používaným virtualizáciám, menovite priama kontrola procesov a IPC na úrovni hypervízora, prístup do živého súborového systému honeypotov, implementovanie senzorov priamo do modulov jadra OS, priamy prístup do pamäte kontajnerov.

Literatúra

- [1] ABBASI, F. – HARRIS, R.: *Experiences with a Generation III virtual Honeynet* Australasian Telecommunication Networks and Applications Conference (ATNAC), 1–6, Nov. 2009.
- [2] BROWN, C.: *Computer Evidence: Collection and Preservation* Hingham, MA: Charles River Media, 2005.
- [3] KARTHIK, S. – SAMUDRALA, B. – YANG, A.: *Design of Network Security Projects Using Honeypots*. Journal of Computing Sciences in Colleges, 20 (4), April 2005, 282–293.
- [4] PALMER, G.: *A road map for digital forensic research* Digital Forensic Research Workshop, Utica, New York, 2001.
- [5] SPITZNER, L.: *Honeypot: Definitions and values*. <http://www.spitzner.net>, May 2002.
- [6] YASINSAC, A. – MANZANO, Y.: *Honeytraps, a network forensic tool* Proceedings of the 6th World Multi-Conference on Systemics, Cybernetics, and Informatics (SCI 02), 2002.
- [7] OPENVZ: <http://wiki.openvz.org/>.
- [8] XPLICO: *Network Forensic Analysis Tool*, <http://www.xplico.org/>.

Kontaktné adresy

Mgr. Pavol Sokol, Ústav informatiky, Prírodovedecká fakulta, Univerzita Pavla Jozefa Šafárika, Jesenná 5, 040 01 Košice, Slovensko,
E-mailová adresa: pavol.sokol@upjs.sk, <http://ics.upjs.sk>

RNDr. PhDr. Peter Pisarcík, Ústav informatiky, Prírodovedecká fakulta, Univerzita Pavla Jozefa Šafárika, Jesenná 5, 040 01 Košice, Slovensko,
E-mailová adresa: peter.pisarcik@upjs.sk, <http://ics.upjs.sk>

OTVORENÝ SOFTVÉR VO VZDELÁVANÍ A V IT RIEŠENIACH

Otvorený softvér a otvorené údajové formáty tvoria dnes už neprehliadnuteľnú súčasť dnešného programového sveta, v ktorom zastávajú silnú opozíciu a alternatívu k proprietárnym programovým riešeniam. Vzhľadom na to je viac než žiaduce zahrnúť túto stránku sveta do vyučovacieho procesu. Nejedná sa len o inšpirujúce myšlienky altruizmu, ktorých je otvorený softvér nositeľom, ale aj o praktické dôsledky znižovania cien komerčných systémov, čím sa významne zužuje priestor na monopolistické správanie.

Šírka záberu otvoreného softvéru je vďaka filozofii vzniku ohromne rozsiahla. Pre mnohých je síce otvorený softvér synonymom nespoľahlivosti, ktoré je zosilnené mnohokrát nedostatočnou dokumentáciou, avšak ako vieme chybám sa v začiatkoch nevyhne ani proprietárny softvér. Dnes vďaka širokej tvorivej základni nadšencov postupným vývojom vznikajú produkty, ktoré sú štandardne používané aj profesionálnymi firmami. Tie tento softvér už bežne zahŕňajú do svojich produktov a dokonca ďalej ho ďalej rozvíjajú a rozširujú vlastnými prostriedkami. Vďaka tomuto pozitívnemu vývoju, dnes je už otvorený softvér mnohokrát jedinou možnosťou ako rýchlo a lacno vyriešiť požadovaný problém.

V zborníku je niekoľko príkladov ako sa dá otvorený softvér použiť na prácu.

V článku Milana Frátrika autor popisuje vytváranie bezpečných sieťových aplikácií na netradičných technických prostriedkoch, kde použil len vývojové prostriedky z produkcie otvoreného softvéru ako je gcc na operačnom systéme Linux. Vo svojom článku doktorandi Tomasz Kanik a Marian Kováč popísali súčasný rozvoj otvoreného softvéru v oblasti nerelačných databáz, ktorých potreba vyplýva z čoraz masívnejšieho objemu údajov, ktoré v súčasnosti treba ukladať a vyhľadávať. Ďalší článok popisuje metódy vyučovania predmetu „Grafika v počítačových aplikáciách“ výlučne pomocou otvorených programových prostriedkov tak, ako ich používajú autori Eudovít Mikuš a Petr Ivaniga. Jan Růžička sa v svojom článku venuje GeoKettle, otvorenému systému z rodiny nástrojov ETL, ktoré sú určené na čítanie a konverziu mnohých údajových formátov. Verím že tento výber tém v tejto oblasti konferenciu bude dostatočne reprezentovať rozvoj a využitie otvoreného softvéru.

Roman Hajtmanek



OSSConf 2012: 53–58

UTILIZATION OF OPEN HARDWARE PLATFORMS IN NETWORK APPLICATIONS

MILAN FRÁTRIK (SK)

Abstract. The aim of the article is to outline ARM architecture, its possible application and use of non-traditional technical means in network applications. The article can be divided into two parts. The first part introduces the PandaBoard platform as the representative of open hardware platform (OSHW¹). The second part outlines the applications focusing on network applications, mainly security and surveillance solutions. There will not be demonstrated data security in this article. The article focuses on physical security and its realization.

Key words and phrases. ARM, PandaBoard, Zoneminder, security, Linux.

VYUŽITIE OPEN HARDWARE PLATFORMIEM V SIEŤOVÝCH APLIKÁCIACH

Abstrakt. Cieľom tohto článku je priblížiť architektúru ARM a ukázať možné aplikácie a využitia netradičných technických prostriedkov v sieťových aplikáciách. Článok je možné rozdeliť do dvoch častí. V prvej časti je predstavená platforma PandaBoard, ako predstaviteľ Open-source hardware. V druhej časti sú načrtnuté aplikácie so zameraním na sieťové aplikácie, predovšetkým bezpečnosť. V tomto článku však nebude demonštrovaná dátová bezpečnosť, ale fyzická a prostriedky pre jej realizáciu.

Kľúčové slová. ARM, PandaBoard, Zoneminder, bezpečnosť, Linux.

Introduction

Nowadays the ARM architecture is not unfamiliar to most people. Many people have no idea that there is RISC processor in their cell phones or tablets. What makes the ARM architecture interesting is the fact, that in addition to low consumption it has fully satisfying power for many user applications. On the other hand, more often we meet with non-standard use of these processors e.g. server solutions [1]. Individual examples of usage will be demonstrated on development kit PandaBoard.

¹Open-source hardware

1. PandaBoard

Similarly as in the other „animal“ boards, the basis of the development board is SoC (System on a Chip). In this case, it is TI OMAP 4430 processor with two Cortex-A9 cores. Except for two main cores, it disposes with four Cortex-M3 cores, which serve for operating peripherals.

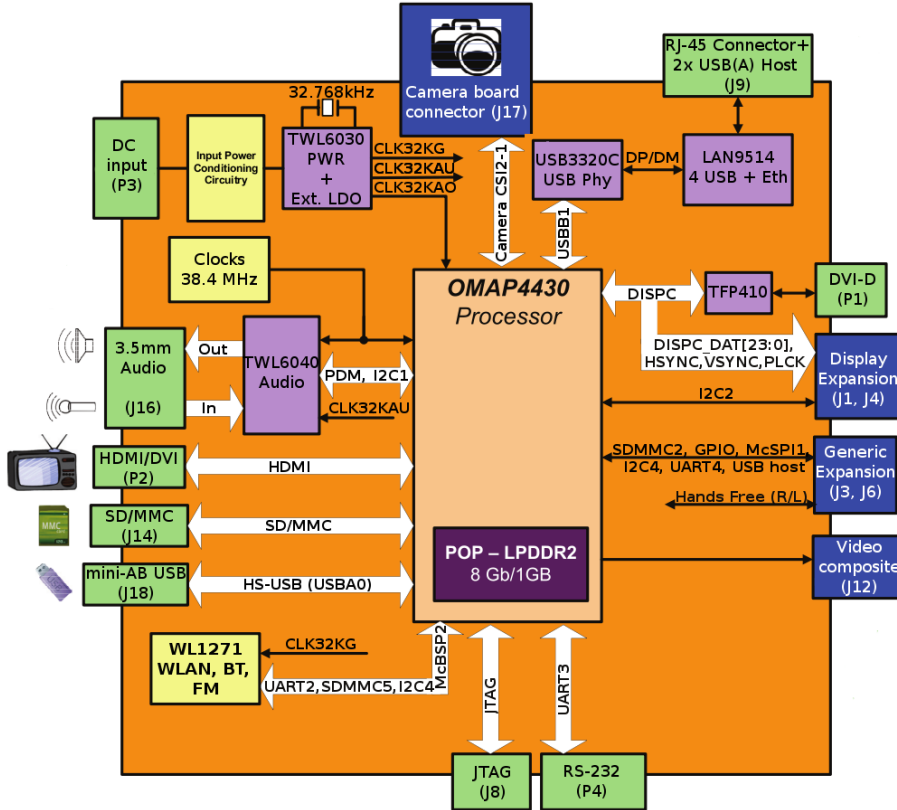


Figure 1. PandaBoard block diagram

The clock frequency of computing cores is 1 GHz. Total computing power can reach $2 \times 2.5 \text{ DMIPS/MHz} \times 1 \text{ GHz}$. There is also GPU Power VR SGX540 and DSP processor C64x on the chip. The 8 Gbit LP-DDR2 RAM is stacked to processor using the POP (package on package) method. Memory card SD/MMC serves for primary storage data and deployment of operating system, while by a little change it is possible to get booting from USB medium. On the board there are available 3 USB slots, one of them works in OTG mode. Graphic and audio output ensures HDMI and DVI output, respectively 3.5 mm jack I/O. In terms of connectivity, there are 10/100 Mbit Ethernet, WiFi 802.11 b/g/n. and

Bluetooth in 2.1 + EDR. Since this is a development kit, there are various pinouts. One of them is determined as the interface for LCD touchscreen, the second one is 24 bit interface for camera and the third connector with 40 outputs, which serves for access to MCU peripherals such GPIO, USART, SPI, I²C, etc. For the advanced users there is a standard JTAG connector which allows possibility to debug software directly on the chip. Block diagram of the system is shown on Figure 1.

2. Standard applications on non-standard platform

This article deals mainly with applications, which use connection to LAN network or internet. To run a full operating system, the memory management unit – MMU is needed. This unit provides the translation of virtual addresses into physical addresses, memory protection, bus arbitration and cache memory check. Open hardware goes hand in hand with open source software. Excellent example is the use of operating system Linux on this platform, specifically operating system Ubuntu 12.04 server edition.

2.1. Database and web server

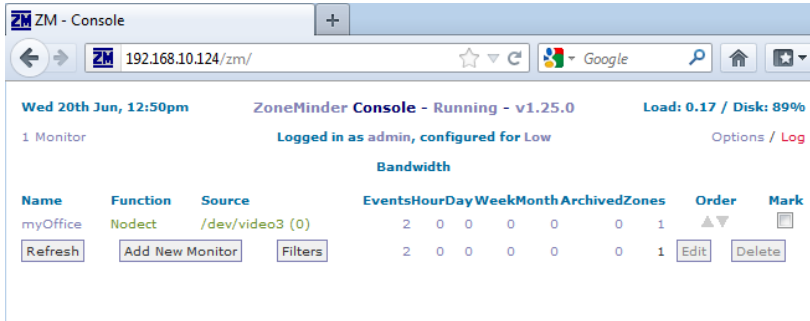
Perhaps the most common application on servers is HTTP server accompanied by a database server and support for scripting languages. The most common used solution is so called LAMP (software bundle). This package contains Apache – HTTP server, MySQL – database server and PHP as scripting language. Thanks to these components, PandaBoard is ready to host all the applications using HTTP server.

The article [3] is focused on measurement of response time and throughput of web server. The results show that this web server is fully sufficient for personal sites with lower number of visitors. Web server based on CPU with Cortex-A9 core has now the best ratio between processed requests and consumed energy [2].

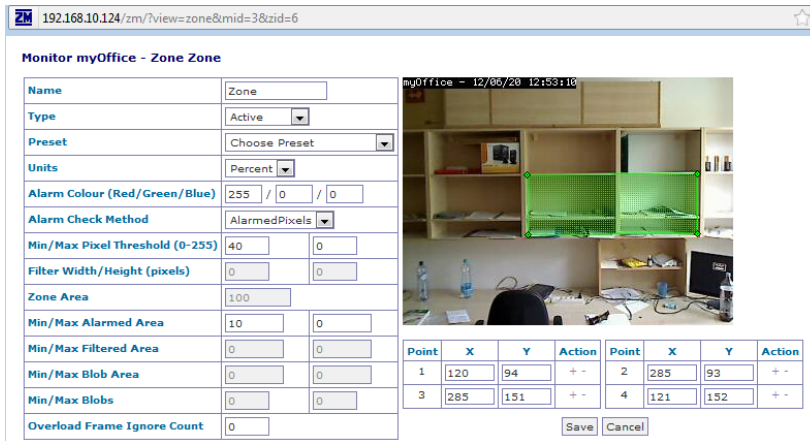
2.2. Multimedia and storage

Current trend of file storage is based mainly on cloud solutions and virtual storage. There is also an open source solution – ownCloud, which can be installed on PandaBoard. This suite provides location independent storage for data. Thus it requires only LAMP bundle, it is easily installable on this platform as mentioned before. The biggest advantage is, that the client application is available for all the platforms (Windows, Linux, Mac, Android). Another possibility is to use the device for streaming audio respectively video and to create home VoD² server.

²Video on Demand



(a) ZoneMinder console



(b) Zone definition

Figure 2. Console and zone definition for motion detection

2.3. Security

Perhaps the most interesting is to use this board for monitoring of selected areas. Devices for area monitoring are usually expensive. Fortunately, there is an open source solution named ZoneMinder. As mentioned at 2.1 PandaBoard can be used as the server. ZoneMinder disposes with a web interface, which is fully configurable. It is possible to configure all the cameras (support for multiple cameras) and the streaming directly from web interface (2a). Camera can be configured in various modes – monitor, modetect, nodetect, record. The software supports CCTV, IP and USB webcams. Evidently from the names of the modes, the software can also detect motion. In addition to detection in wide areas, it is possible to select only certain part of area and to detect motion especially there (2b). In case, the motion is detected, the system sends a warning via email or

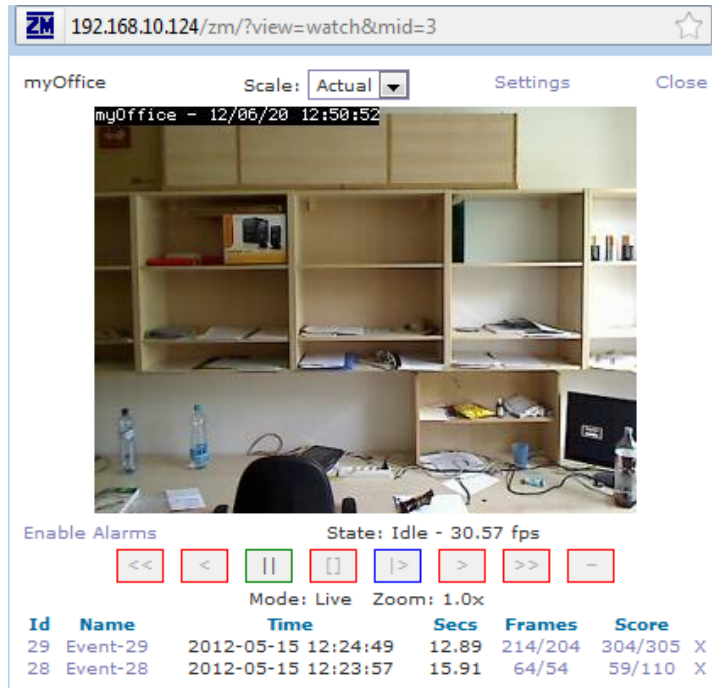


Figure 3. ZoneMinder camera view

sms message. The big advantage is the automatic recording of individual events, when the motion appeared. Recorded sequences can be stored on the cloud server and user gets information via cell phone, tablet or PC.

Obviously this solution is fully applicable on ARM architecture. Very important factor is the load of CPU and frames per second (FPS) which could be processed at given resolution. At Figure 4a average FPS for each tested resolution are displayed. At Figure 4b average load³ for tested resolutions is displayed. As seen from the graphs (4a, 4a) the user should decide between load and frames per second.

3. Conclusion

Based on testing, it is easy to find suitable video quality and server load ratio. The most of commercial system use resolution 640x480 px. The CPU load at this resolution is low, while FPS (~25 fps) is sufficient.

The aim of this article was to show non-traditional use of TI OMAP4430 processor in network applications. Such devices may serve as private web servers

³standard unix-like load for 15 minutes interval

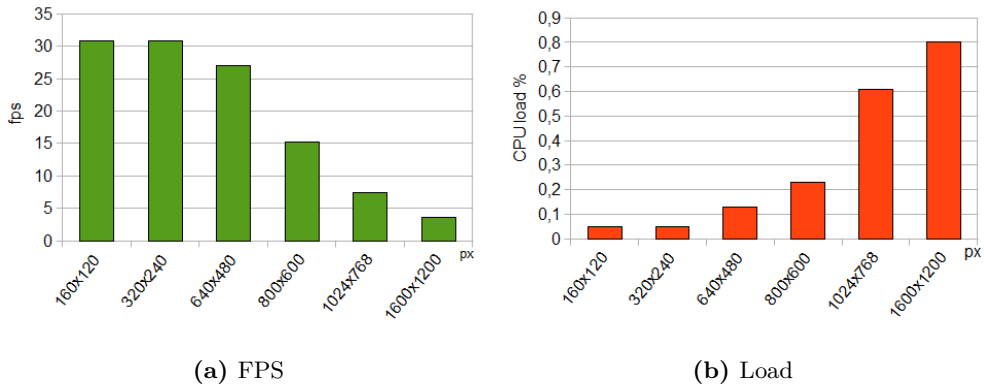


Figure 4. Pandaboard performance for ZoneMinder video streaming

or cloud servers, because not all the users would provide their data to the third party providers. Devices working on „lightweight architecture“ may also replace personal computers in many offices. Sufficient power, low energy consumption and last but not least broad connectivity create good conditions for broad range of users.

References

- [1] HP (2011): *HP Shapes the Future of Extreme Low-energy Server Technology*, <http://www.hp.com/hpinfo/newsroom/press/2011/111101xa.html>, [7. 6. 2012].
- [2] SVANFELDT-WINTER, O. – LAFOND, S. – LILIUS, J.: *Cost and Energy Reduction Evaluation for ARM Based Web Servers*, Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on, 12.-14. December 2011, 480–487.
- [3] FRÁTRIK, M.: *When ARM architecture meets internet applications*, ICTIC 2012 : Proceedings in information and communication technologies – International conference, Žilina, 19.–23. March 2012, 18–20, ISBN 978-80-554-0513-1.
- [4] TEXAS INSTRUMENTS: *OMAP4430 Multimedia Device*, Technical Reference Manual, 5483 p., March 2011.
- [5] FISHER, R.: *SDP4430 Software Development Platform*, User manual, http://pandaboard.org/sites/default/files/board_reference/pandaboard-a/panda-a-manual.pdf [7. 6. 2012].

Contact address

Ing. Milan Frátrik, Department of Information Networks, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic,
E-mail address: milan.fratrik@fri.uniza.sk



OSSConf 2012: 59–64

NOSQL – NON-RELATIONAL DATABASE SYSTEMS AS THE NEW GENERATION OF DBMS

TOMASZ KANIK (PL) AND MARIAN KOVÁČ (SK)

Abstract. The first database systems were based on the relational model, which rose in the 70's of last century and until now it is used in most of the implementations of various systems. However, increasing demands on the size of stored data and the speed of database systems led to the creation of non-relational database systems. The article describes the basic principles of the non-relational databases behavior and a brief history of their origin. There are listed basic problems that led to the creation of these systems, there is also an overview of basic types of non-relational databases, their brief description and current and future possibilities for their development and use in practice.

Key words and phrases. NoSQL, non-relational database.

NOSQL – NERELAČNY DATABAZOVÝ SYSTÉM AKO NOVÁ GENERÁCIA DBMS

Abstrakt. Prvé databázové systémy boli založené na relačnom modeli, ktorý vznikol v sedemdesiatych rokoch minulého storočia a až dodnes je nasadený vo väčšine implementácií rôznych systémov. Avšak zvyšujúce sa nároky na veľkosť uchovávaných dát a rýchlosť databázových systémov viedli k vytvoreniu nerelačných databázových systémov. Článok popisuje základné princípy fungovania systémov nerelačných databáz a stručnú históriu ich vzniku. Sú tu uvedené základné problémy ktoré viedli k vzniku týchto systémov, taktiež je uvedený aj prehľad základných typov nerelačných databáz, ich stručná charakteristika a súčasne aj budúce možnosti ich nasadenia a využitia v praxi.

Kľúčové slová. NoSQL, nerelačne databázy.

Introduction

The concept of non-relational database systems is very young and currently it is developing. We can say that abbreviation NoSQL exists in computer science just since 1998, but it gained greater importance only in the last three years. This stems from the fact that not all elements associated with NoSQL are known to mass programmers, IT architects and administrators. This article aims to show the problems associated with NoSQL. The topic is very extensive, so we will try that solutions shown would have the greatest possible practical value for future users.

NoSQL is a term that describes the systems of database management, that do not use traditional relational model. These data storage systems do not require

well-defined schemes (tables), moreover they do not usually use JOIN operations and also allow simple horizontal scaling. We can immediately note, that the term NoSQL does not describe the exact technology, i.e. a solution. It is rather a notion, which conceals all solutions, which are the opposite of the relational model.

Despite significant marketing success, most of the experts assert, that the term is not entirely accurate and they recommend that all solutions hiding behind it have been identified as NoREL (no relation – without relation, non-relational) or NoJOIN (no join – without joining). That’s why under the term NoSQL we will understand Not Only SQL.

History of NoSQL

The relational data model rules virtually since its inception, i.e. since 1973, when Edgar Frank Codd (awarded in 1981 by Turing prize, what is the equivalent of the Nobel Prize - “for a major and lasting contribution to the theory and practice of database systems, especially in the relational database model”) published his work. During this period, IBM[®] (for which Codd worked) or Oracle[®] began to create solutions that in the near future hit all database market. It should be noted that some companies (including the aforementioned Oracle[®]) have been building its position on the market by relational solutions.

It is clear that the relational model has its shortcomings, such as:

- Complex systems contain tables with hundreds of columns, but only a small part of them is essential for the specified record.
- The process of database normalization leads to creation of many relations, which makes the data structure look like a tree and it significantly adversely affects the performance of database system.
- Changing the database schema (e.g. add a new table, column or relation) can be very complex operation. This especially concerns large databases (billions of rows), where re-indexing can be very time-consuming operation.

Therefore relational model met the expectations of data storage and access speed to data and there was not a need to find another solution (though the relational model has its mistakes). Gradually coming period of global Internet, Web 2.0 and social networks caused that even server farms (containing several thousand servers) became inadequate in performance often. We can therefore wonder why relational database systems do not have sufficient performance. The answer is simple: at the time of the relational model creation and its first implementations were quite different demands on created systems.

It should be borne in mind that the amount of stored data (especially of Internet stocks) grows continually. Most experts say that every year the amount of stored data doubles. Here we can find an analogy with the Moore’s law, which

says that the number of transistors in a semiconductor chips doubles roughly in the same time intervals.

The dynamic increase of the data amount is a phenomenon that has impact on many users. But what exactly means “large amount” of data? In 2000, the largest table of one major Internet service provider contained more than 200 million records. Today, we talk about such amount in daily data gain. The “big” database can now be considered one that gains over 10 million records daily. If in the table should be stored data from the last five years (which in case of the most important data is not excessively long), it may happen that the whole table will contains more than 18 billion records (10 mil. records per day * 365 days * 5 years). Real example: a likely increase in the amount of monthly auctions in the auction service Aukro[®] is 85 million auctions. In the auction system probably exists a table (or other data structure) to which millions of records each month are added. Regardless of the adopted assumptions and calculations, we can talk about large system.

Data models used in NoSQL solutions

Nosql is very diverse environment. This follows from the fundamentals of this system. Therefore, we can not talk about united models. However, we can distinguish several dominant models in which several solutions have been implemented. Below we show some of the most popular models.

Key-value databases – have a very simple design. Simplified we can say that it is a table that contains two text columns: key and value. In more complex models the value can be a type of number, string, table or list. This model may appear as a simple, but it has one important property – it is very fast (in writing as well as in reading data). The most widely used databases of this type are MemcacheDB [8], Amazon DynamoDB [7], Redis [5].

Column databases – the approach of data observation is changed, i.e. instead of the traditional information storing in rows, they are written in columns. This approach improves performance especially in data centers and analysis systems that require large amounts of data aggregation. In columns are stored data of the equal type and therefore it is possible to use more efficient algorithms for data compression. The most widely used databases of this type are Google Bigtable, Cassandra [3], HBase [9].

Document databases – there are available several such solutions on the market for years (e.g. Lotus Notes), but the real progress occurred in the last 3 years. Instead of rows we use term “documents” in these databases. Documents contain a key – value pairs, which are called objects. This approach is very flexible and allows natural representation of real data in information systems. The most widely used databases of this type are CouchDB [11], MongoDB [4], Amazon SimpleDB [10], Riak [6].

XML databases – along with the development of various database systems there was a need to exchange information among them. Language XML (and others such like, for example YAML) is perfect for this purpose and it essentially became a standard of network services or configuration files. Therefore, creation of database solutions storing data in XML is a natural way of development. Such solutions are for example eXist [12], Attic Apache [13], Oracle Berkeley DB XML [14], MarkLogic Server [15]. In addition, lookup mechanisms were introduced such as XQuery (XML Query Language) and XPath. These technologies have become interesting for large market players, which led to the creation of a new column for data deposition in the XML form in many DBMS. In many cases, extensions allowing the creation of SQL queries on XML documents were added into SQL. All this led to decrease interest in solutions based on XML only.

Object-oriented databases – in the last decade in programming, object methodology is used primarily, which is convenient in almost all aspects. One of the few elements that interferes with programmers is so-called object-relation mapping (ORM). This stems from the fact that it is not always easy to carry the relational model to object model. The solution to this problem was introducing framework object-relational mapping in the programming. Systems such as Hibernate [16] (and others like it, made for all popular programming languages) have become industry standards. These products are catch, but they have not become solution to the ORM problem explicitly, they rather try to hide it (cause of the problem is not solved). Thus arose solutions, where objects and even entire object structures are written into the database instead of rows. This solution has many advantages, but it is still not extensively used. The reason is low efficiency of these solutions mainly, insufficient optimization of queries and inability to exchange data among different languages (objects created by language Java[®] can be often read only by applications written in the same language).

As we can see, a variety of models is quite large. On one hand it is positive, because we can find the ideal solution for our needs. On the other hand, the amount of existing data models and their implementations makes that the selection process may require significant time cost. Exactly that fact repels potential future users.

Further ways of NoSQL development

No matter what will happen with NoSQL solutions (significant progress or least practical use), there is a need to think about the further ways of development. If NoSQL solutions should consolidate their position, not only excellent technologies must be used, but also problems associated with the entire infrastructure must be solved. The most important ways for further development can include:

- **Sharing and propagation of knowledge among users such as administrators and programmers.** It should be noted that is not currently

available literature, which would adequately describe NoSQL. For example in Amazon[®] service (May 2012) in search of “NoSQL” term are found 48 titles (English only). In search of “SQL” term are found more than 7000 unique titles.

- **Providing support for solutions.** The current problem is to require companies to create enhanced technical support for their developed NoSQL solutions. Organization of trainings, certification, consulting for the solutions implementation and support in crisis situations are important elements of credibility-enhancing environment. This can be problematic because most of NoSQL solutions are Open Source, produced mostly by small firms (solution is produced by one or two programmers often).
- **The standardization of access interfaces to NoSQL databases.** In the standardization it is necessary to use models from relational databases, in which despite of various internal solutions it is possible to use the SQL language DBMS (at least at a basic level of compatibility). Therefore it is necessary to create a unified solution and promote it as a standard. Without this element the use of other NoSQL solutions will always be associated with having to learn the data model and language of their service, which may repel potential interested persons.
- **The involvement of big companies associated with the DBMS.** Into the NoSQL idea must involve companies from the “big triplet” (Oracle[®], Microsoft[®], IBM[®]) after some time. Recent NoSQL development is mainly driven by small firms which are suppliers of database solutions. It is hard to say how listed companies will behave.
- **Elimination of weaker solutions.** The mechanism of evolution works and therefore only the best solutions must remain on the market. Wide range of solutions is not always optimal (especially if it is necessary to choose among 50 technologies). To select the correct final solution, we need to know them, understand them, test them, etc.

Current and further areas of development

There is a group of subscribers, for whom managing huge amounts of data is a priority task. These firms are very important for the NoSQL development and they have a great impact on the choice of technologies, mainly because of their market position and financial capability. Their mode of action is to adapt the non-relational solutions to current needs. Nosql systems are largely created to solve specific problems. Nevertheless, we can distinguish several characteristic features:

- Processing and storing massive amount of information.
- They are the global internet systems with very complex infrastructure.

- It is not necessary to use known mechanisms from relational systems (joining, transaction, data schemes).
- There is a need for dynamic and cheap scaling with short access time.
- The high level of accessibility (working 24 hours a day, 365 days a year).

These properties determine solutions, where NoSQL has chance to be used, i.e. large Internet systems or mechanisms to support these systems. Therefore we can not in the near future expect that NoSQL solutions will be used in environments treating small amount of information or requiring solutions associated with relational databases. It may seem that NoSQL already found its place and that it will not expand to other areas. It may be true, but according to us, migration from relational solutions to more efficient NoSQL solutions will come in the future.

References

- [1] Manifesto for Agile Software Development: <http://www.agilemanifesto.org>.
- [2] SimplyHired[®]: <http://www.simplyhired.com/>.
- [3] Apache Cassandra: <http://cassandra.apache.org/>.
- [4] MongoDB: <http://www.mongodb.org/>.
- [5] Redis: <http://redis.io/>.
- [6] Riak: <http://basho.com/>.
- [7] Amazon[®] DynamoDB: <http://aws.amazon.com/dynamodb/>.
- [8] MemcacheDB: <http://memcached.org/>.
- [9] Apache HBase: <http://hbase.apache.org/>.
- [10] Amazon[®] SimpleDB: <http://aws.amazon.com/simpledb/>.
- [11] Apache CouchDB: <http://couchdb.apache.org/>.
- [12] eXist-db: <http://exist-db.org/>.
- [13] Apache Attic: <http://attic.apache.org/>.
- [14] Oracle[®] Berkeley DB: <http://www.oracle.com/>.
- [15] MarcLogic[®]: <http://www.marklogic.com/>.
- [16] Hibernate: <http://www.hibernate.org/>.

Contact addresses

Ing. Tomasz Kanik, Department of Mathematics Methods, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic,
E-mail address: tomasz.kanik@fri.uniza.sk, <http://frcatel.fri.uniza.sk/~kanik/>

Mgr. Marian Kováč, Department of Mathematics Methods, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic,
E-mail address: marian.kovac@fri.uniza.sk, <http://frcatel.fri.uniza.sk/~kovac/>



OSSConf 2012: 65–70

POUŽÍVANIE OSS SOFTVÉRU VO VYUČOVANÍ PREDMETU GRAFIKA V POČÍTAČOVÝCH APLIKÁCIÁCH

LUDOVÍT MIKUŠ (SK) A PETR IVANIGA (SK)

Abstrakt. Článok popisuje používanie otvoreného softvéru vo voliteľnom predmete „Grafika v počítačových aplikáciách“ na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline. Popísané sú metódy vyučovania, spôsoby poskytovania výučbových materiálov a formy hodnotenia znalostí a zručností študentov.

Kľúčové slová. LMS Moodle, Inkscape, GIMP, Google Sketchup, Blender.

THE USE OF OSS SOFTWARE IN TEACHING THE SUBJECT COMPUTER GRAPHICS IN APPLICATIONS

Abstract. In this paper, we describe the use of Open Source software in elective course “Computer Graphics in Applications” at the Faculty of Management Science and Informatics, University of Žilina. We discuss teaching methods, the means and tools for providing of teaching materials and the methods of assessment of students knowledge and skills.

Keywords. LMS Moodle, Inkscape, GIMP, Google Sketchup, Blender.

1. Úvod

Predmet „Grafika v počítačových aplikáciách“ vznikol na základe potreby zoznámiť študentov s grafikou na počítači nielen z pohľadu jej technickej realizácie ale aj z pohľadu estetického dojmu. Často sa totiž môžeme stretnúť s tým, že počítačové aplikácie vytvorené študentmi technických zameraní sú síce technicky a funkčne dobre pripravené ale ich grafický vzhľad je na veľmi nízkej úrovni.

Preto sme pripravili predmet, ktorého úlohou je zoznámiť študentov s úlohou farieb v grafike, vektorovou a rastrovou grafikou, 3D grafikou a softvérom, pomocou ktorého môžeme vytvárať rôzne grafické výstupy. Vzhľadom na vysoké finančné nároky mnohých komerčných softvérových systémov sme sa rozhodli v rámci predmetu používať OSS softvér alebo aspoň voľne dostupný softvér.

2. Prostredie používané pri vyučovaní

Pre vyučovanie predmetu Grafika v počítačových aplikáciách používame LMS Moodle (<http://moodle.org>), ktorý je súčasťou portálu e-vzdelávania Žilinskej univerzity v Žiline. Predmet je voliteľným predmetom ako pre študentov v Žiline tak aj pre študentov detašovaného pracoviska v Prievidzi. Tu vzniká komplikácia

s rozvrhom, pretože na predmet sa prihlasujú študenti rôznych ročníkov a rôznych zameraní a teda je prakticky nemožné nájsť časový priestor v rozvrhu tak, aby sa všetci mohli zúčastňovať vyučovania. Preto v predmete využívame výhody elektronického vzdelávania, kedy študenti nemusia byť fyzicky prítomní na každej prednáške alebo cvičení ale prostredníctvom portálu e-vzdelávania si nájdú všetky potrebné študijné materiály na jednom mieste. Aj preverovanie a hodnotenie vedomostí a zručností je organizované elektronicky cez LMS Moodle.

Ďalším zaujímavým prvkom v rámci predmetu je vyučovanie v Prievidzi výhradne prostredníctvom videokonferenčného systému. Táto forma vyučovania sa využíva ako pri prednáškach, tak aj pri cvičeniach. Výrazne sa tým znížia náklady na cestovanie a prvý priamy kontakt (F2F) majú čítelia so študentmi vlastne až pri skúške. Samozrejme, že je vhodné zladit' termíny prednášok a cvičení tak aby prebiehali súčasne v Prievidzi a aj v Žiline.

3. Metódy vyučovania

Predmet „Grafika v počítačových aplikáciách“ obsahuje desať tematických celkov, v rámci ktorých dostanú študenti študijné materiály v podobe multimedialných kurzov. Príklad takéhoto kurzu je na http://www.kis.fri.uniza.sk/~ludo/ekurzy/Grafika_PC/start.html. Témy sú napr:

- Grafika a počítač.
- Farba – fyzikálna podstata.
- Farba v multimedialných aplikáciách.
- Formáty grafických súborov.
- Grafický dizajn.
- Perspektíva.
- 3D grafika – základy.
- 3D grafika – krivky a plochy.

Úlohou študentov je na základe štúdia uvedených kurzov absolvovať testy z tematických celkov. Forma testovania je cez LMS Moodle, pričom ku každej téme je pripravená databáza minimálne 40-tich otázok. Z týchto otázok dostane študent do testu 10 náhodne vybraných. Testy majú dve verzie: cvičnú a ostrú. Cvičné testy slúžia na precvičenie si znalostí a test sa môže opakovať ľubovoľne veľa krát, pričom výsledky z neho nie sú započítavané do celkového počtu bodov. Ostrý test sa môže absolvovať len jeden krát. Obe verzie testov majú časové obmedzenie a to nielen na dĺžku trvania testu (4–5 min) ale aj na možnosť jeho absolvovania (dva týždne). Vzhľadom na to, že nemôže byť všetci študenti prítomní na cvičení (dôvody sme uviedli vyššie v článku) je prakticky nemožné zabrániť opisovaniu resp. inému druhu podvádžania pri testoch a to aj ostrých.

Na preverenie získaných zručností sa v predmete používajú zadania typu:

V programe <Inkscape, GIMP, Blender, ...> vytvorte <logo, banner, tričko, ...> ...

Úloha 1

V programe Inkscape vytvorte **logo** pre **Ústav celoživotného vzdelávania Žilinskej Univerzity**. Základnou podmienkou je aby písmená **ÚCVŽU** alebo text **Ústav celoživotného vzdelávania Žilinskej Univerzity** tvorili **maximálne 30%** plochy loga. Logo navrhnete tak aby bolo čitateľné aj na vizitke (môže to byť jeho zjednodušená verzia). Vaše práce budú hodnotiť vaši kolegovia z FEL (50%) a aj ja (50%). Ja budem hodnotiť: pomer textu a grafiky, farebnosť, čitateľnosť, originalitu, zapamätateľnosť. Oni budú hodnotiť subjektívne pôsobenie loga.

Logo pri odovzdávaní pomenujte svojim priezviskom.

Poz.: skôr ako začnete pozrite si kurz o tvorbe loga

 [Logotyp a jeho tvorba](#)

 [Logo sa neustále vyvíja](#)

 [Vaše práce - logo](#)

 [Logá, ktoré ste odovzdali](#)

 [Prémia za návrh - logo](#)

Obr. 1. Príklad zadania

Príklad takéhoto zadania je na obr. 1.

Tieto úlohy je potrebné odovzdať do určitého časového obdobia (dva týždne). Následne ich spracujeme do podoby hlasovacieho formulára (pomocou Formulára z Google dokumentov), ktorý sa poskytne študentom a ich úlohou je ohodnotiť svoje práce. Keďže je viac skupín študentov – obvykle nehodnotia svoje práce ale práce inej skupiny študentov. Tieto práce hodnotíme aj my a z oboch hodnotení vytvoríme celkové hodnotenie študentských prác s bodovým ocenením 1–10 bodov. Príklad hodnotiaceho formulára je na obr. 2. (http://www.kis.fri.uniza.sk/~ludo/GvPA_2011/01/viewform.htm)

Študenti sa môžu podieľať na zlepšení predmetu návrhom iných zadaní, či vytvorením vlastných návodov na prácu s aktuálne používaným programom. Za každú takúto iniciatívu sú následne odmenení bodmi, ktoré sa započítavajú do celkového výsledku študenta.

4. Softvér používaný pri vyučovaní

Vzhľadom na vysoké finančné nároky komerčných softvérových systémov sme sa rozhodli v čo najväčšej možnej miere používať otvorený softvér alebo aspoň softvér voľne dostupný. Toto je oceňované aj študentmi, pretože tento softvér môžu používať aj na svojich osobných počítačoch a nie sú nútení investovať peniaze do softvéru prípadne používať nelegálny softvér.

Logo ÚCV - FRI

Tu budete hodnotiť logo ÚCV vytvorené študentmi FRI. V hodnotení použite celú bodovú škálu od 0 po 10.

* Povinné

Meno a priezvisko *

1 *



0 1 2 3 4 5 6 7 8 9 10

Obr. 2. Príklad formulára na hodnotenie

Pri všetkých vybraných softvérových systémoch vychádzame z predpokladu, že študenti sú začiatočníci a teda úlohy, ktoré od nich požadujeme nie sú príliš náročné na spracovanie. Aj výklad na cvičeniach je postavený na uvedenom predpoklade.

Pre prácu s vektorovou grafikou sme vybrali program **Inkscape**, ktorý plne postačuje na tvorbu vektorových obrázkov a jeho natívny formát **svg** je použiteľný aj na web stránkach ako jeden z mála vektorových formátov. Pre overenie zručností z práce v tomto programe kreslia študenti v rámci zadania logo (fakulty, predmetu, UCV, ...), tričko (fakulty, školy, ...) a potlač na CD-ROM.

Príklad výstupu študentských prác je na:

http://www.kis.fri.uniza.sk/~ludo/GvPA_2011/02/viewform.htm

Pre prácu s rastrovou grafikou sme vybrali program **GIMP**, ktorý je alternatívou voči komerčným softvérovým systémom určeným na prácu s fotografiami. Jeho posledné verzie už stratili nevýhodu neprehľadného grafického rozhrania a tak sa jeho používanie stalo jednoduchším. Pre overenie zručností z práce v programe **GIMP** tvoria študenti koláže, fotomontáže a špeciálne úpravy fotografií.

Príklad výstupu študentských prác je na:

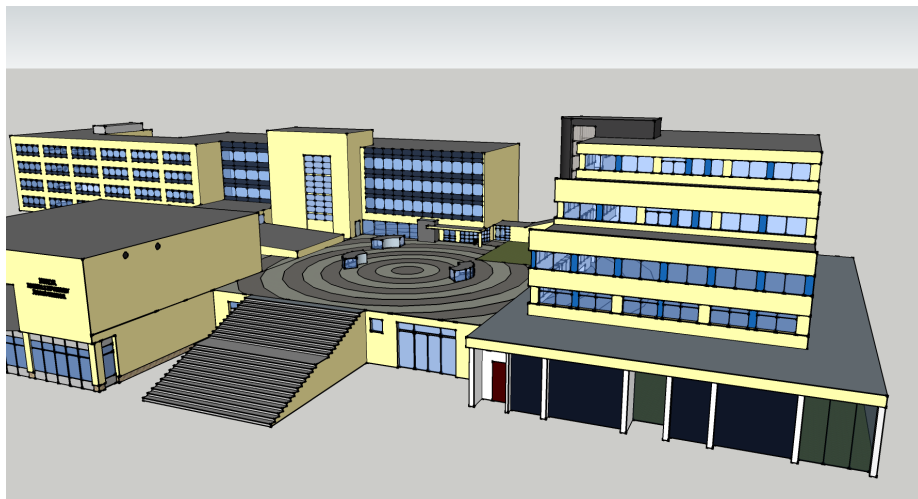
http://www.kis.fri.uniza.sk/~ludo/GvPA_2011/05/viewform.htm

Pre prácu s 3D grafikou sme vybrali programy **Google Sketchup** a **Blender**. **Google Sketchup** nie je síce otvoreným softvérom ale je voľne dostupný a je

požívateľsky prívetivý. V tomto programe dokazujú študenti svoje zručnosti vytvorením 3D modelu nábytku a 3D modelu niektorej budovy Žilinskej univerzity, máme napríklad k dispozícii aj budovu FRI, vytvorenú v rámci predmetu obr. 3.

Príklad výstupu študentských prác je na:

http://www.kis.fri.uniza.sk/~ludo/GvPA_2011/07/viewform.htm



Obr. 3. Budova FRI vytvorená v Google SketchUp

V programe Blender, ktorý predstavuje plnohodnotnú náhradu komerčných 3D modelovacích softvérových systémov dokazujú študenti svoje nadobudnuté zručnosti vytvorením 3D modelu vázy a vytvorením 3D zátišia s ovocím.

Príklad výstupu študentských prác je na:

http://www.kis.fri.uniza.sk/~ludo/GvPA_2011/09/viewform.htm

5. Záver

Predmet Grafika v počítačových aplikáciách, v ktorom sa používa prevažne OSS softvér sa vyučuje zatiaľ iba dva roky. Už teraz ale môžeme skonštatovať, že používanie OSS softvéru vo vyučovaní prináša svoje pozitíva najmä v tom, že študenti spoznajú alternatívny softvér ku drahým komerčným programom. Zároveň sú tým vedení ku používaniu legálneho softvéru. Niektoré názory študentov, ktoré uviedli do diskusného fóra v predmete sú na obr. 4.

V tematickom bloku konferencie OSSConf 2012 ukážeme konkrétnejšie výstupy študentov a zároveň si účastníci budú môcť vyskúšať prácu s vektorovým programom Inkscape.

Za hlavný prínos považujem oboznámenie sa s pracou vo viacerých grafických programoch, hlavne v tých 3D.

[Ukázať predošlý príspevok](#) | [Upraviť](#) | [Oddeliť](#) | [Odstrániť](#) | [Odpovedať](#)

Re: Klady

autor - Wednesday, 21 December 2011, 09:42

jednoznačne spoznanie nových programov a hlavne výučba cez internet

[Ukázať predošlý príspevok](#) | [Upraviť](#) | [Oddeliť](#) | [Odstrániť](#) | [Odpovedať](#)

Re: Klady

autor - Monday, 19 December 2011, 17:29

Za hlavný prínos pre mňa považujem niektoré teoretické vedomosti (hlavne v oblasti 3D grafiky), ktoré mi tento predmet dal, a ktoré mi dosť uľahčia tvorbu modelov alebo kompozíciu samotných scén.

[Ukázať predošlý príspevok](#) | [Upraviť](#) | [Oddeliť](#) | [Odstrániť](#) | [Odpovedať](#)

Re: Klady

autor - Monday, 19 December 2011, 09:30

Páči sa mi, cvičenia a prednášky neboli povinné.

[Ukázať predošlý príspevok](#) | [Upraviť](#) | [Oddeliť](#) | [Odstrániť](#) | [Odpovedať](#)

Re: Klady

autor - Monday, 12 December 2011, 10:16

Prínosom je skúsenosť s novými free programami i to, že študenti sú priebežne počas celého semestra nútení pracovať

Obr. 4. Niekoľko názorov z diskusného fóra

Literatúra

- [1] BLENDER: <http://www.blender.org>.
- [2] GIMP: <http://www.gimp.org>.
- [3] INKSCAPE: <http://inkscape.org>.
- [4] MOODLE: <http://moodle.org>.
- [5] SKETCHUP: <http://sketchup.google.com>.

Kontaktné adresy

Ing. Ludovít Mikuš, PhD., Katedra informačných sietí, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská republika,
E-mailová adresa: ludovit.mikus@fri.uniza.sk, <http://www.kis.fri.uniza.sk/>

Ing. Petr Ivaniga, PhD., Katedra informačných sietí, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská republika,
E-mailová adresa: petr.ivaniga@fri.uniza.sk, <http://www.kis.fri.uniza.sk/>



OSSConf 2012: 71–76

HOW TO USE GEOKETTLE TO HANDLE EXCHANGE FORMAT OF THE REGISTER OF TERRITORIAL IDENTIFICATION, ADDRESSES AND REAL ESTATE

JAN RŮŽIČKA (CZ)

Abstract. GeoKettle is a tool from the ETL tools family. ETL is an abbreviation for Extract Transform Load. ETL tools are able to read data from various formats, transform them with several modules and save to several formats (including database). What is important is the ability to do such operations in user friendly graphical environment. GeoKettle uses diagrams to describe a work-flow of a process. With a basic knowledge of XPath language any general user can simply handle various formats based on XML. The paper describes one of the possible ways how to extract geodata from an exchange format of the register of territorial identification, addresses and real estate (VFR). The VFR format is based on XML and uses GML namespace to describe geometry properties of features. From the VFR format can be extracted for example parcels or addresses. It can be done with GeoKettle modules and with another modules the extracted data can be loaded for example to PostGIS table. The paper describes one of the possible ways how to use GeoKettle to extract data from VFR format and save them to another more common format (for example ESRI Shapefile).

Key words and phrases. GeoKettle, RÚIAN, VFR, addresses, parcels.

VYUŽITÍ GEOKETTLE PRO ZPRACOVÁNÍ VÝMĚNNÉHO FORMÁTU REGISTRU ÚZEMNÍ IDENTIFIKACE, ADRES A NEMOVITOSTÍ

Abstrakt. GeoKettle je nástroj z rodiny nástrojů ETL. ETL znamená Extract Transform Load. Nástroje ETL jsou určeny k načtení dat z různých formátů, jejich transformaci s využitím řady modulů a jejich uložení do řady formátů (včetně relační databáze). To všechno nabízejí v uživatelsky příjemném grafickém prostředí. GeoKettle využívá diagramy k popisu procesu načtení, transformace a uložení. Běžný uživatel se základní znalostí jazyka XPath může velmi snadno tímto způsobem zpracovávat řadu formátů založených na jazyce XML. Příspěvek popisuje jeden z možných způsobů načtení geodat z výměnného formátu registru územní identifikace, adres a nemovitostí (VFR). Formát VFR je založen na XML a využívá jmenný prostor GML pro popis geometrických vlastností geoprvků. Z formátu VFR je možné získat např. parcely nebo adresní místa. Takovou operaci je možné realizovat s využitím modulů GeoKettle, výsledek může být uložen např. do tabulky systému PostgreSQL/PostGIS. Příspěvek popisuje jednu z možných cest jak využít GeoKettle k získání geodat z formátu VFR a uložit je do jiného běžnějšího formátu (např. ESRI Shapefile).

Klíčová slova. GeoKettle, RÚIAN, VFR, adresy, parcely.

Introduction

XML has been a common language for creating exchange formats. A new register of territorial identification, addresses and real estate that should be managed by Czech Office for Surveying, Mapping and Cadastre (COSMC) with cooperation of Czech Statistical Office will offer that data in exchange format based on XML as well. Users of geographical information systems (GIS) that are not familiar with XML language will not be able to use directly such format. The format is rather complex and contains several different types of information. The paper should show that it is not necessary to buy software that can handle such format, but that there are tools already available and free of charge.

1. ETL tools

ETL means Extract Transform Load. ETL tools allow to extract data from several sources, transform the data (for example change structure of attributes, transform coordinates or compute new attributes) and load data to another place (for example to PostgreSQL/PostGIS database).

There are several tools that can be used for ETL:

- HUMBOLDT Tools,
- GeoKettle,
- Spatial Data Integrator,
- XSLT,
- GDAL/OGR,
- SQL,
- ...

The HUMBOLDT Tools, GeoKettle and Spatial Data Integrator are focused mainly to ETL operations and have user friendly environment. General tools like languages XSLT or SQL or libraries GDAL/OGR can be used for ETL operations as well, but they are dedicated mainly for developers.

2. Exchange Format for Register of Territorial Identification, Addresses and Real Estate

The Register of Territorial Identification, Addresses and Real Estate is going to be a central register for all subject in public administration.

The Exchange Format for Register of Territorial Identification, Addresses and Real Estate is based on XML and GML (Geography Markup Language). It consist of several types of information such as municipality, cadastral area, parcel, building, street, address point.

There are available 19 individual types defined in this format and 2 shared types. GML is used for definition of the geometric attributes of the features.

The following example shows structure for one address point [1].

```
<vf:AdresniMisto gml:id="AD.12806471">
  <adi:Kod>12806471</adi:Kod>
  <adi:CisloDomovni>104</adi:CisloDomovni>
  <adi:Psc>25206</adi:Psc>
  <adi:StavebniObjekt>
    <soi:Kod>12717614</soi:Kod>
  </adi:StavebniObjekt>
  <adi:Ulice>
    <uli:Kod>434205</uli:Kod>
  </adi:Ulice>
  <adi:PlatiOd>2011-07-01T00:00:00</adi:PlatiOd>
  <adi:IdTransakce>0</adi:IdTransakce>
  <adi:GlobalniIdNavrhuZmeny>0</adi:GlobalniIdNavrhuZmeny>
  <adi:Geometrie>
    <adi:DefinicniBod>
      <adi:AdresniBod>
        <gml:Point srsDimension="2" gml:id="DAM.1937498"
          srsName="urn:ogc:def:crs:EPSG:2065">
          <gml:pos>747802.63 1062732.00</gml:pos>
        </gml:Point>
      </adi:AdresniBod>
    </adi:DefinicniBod>
  </adi:Geometrie>
</vf:AdresniMisto>
```

3. GeoKettle

The Geokettle [2] allows to specify all steps of a process in the graphical form of a diagram. The most simple diagram is shown at the Figure 1. There are only two items, the first item does an extract and a transformation of geodata and the second item does a save to ESRI Shapefile format.

An user specifies parameters for the extraction, transformation and load with the intuitive dialogues. Some of them are described at the following figures.

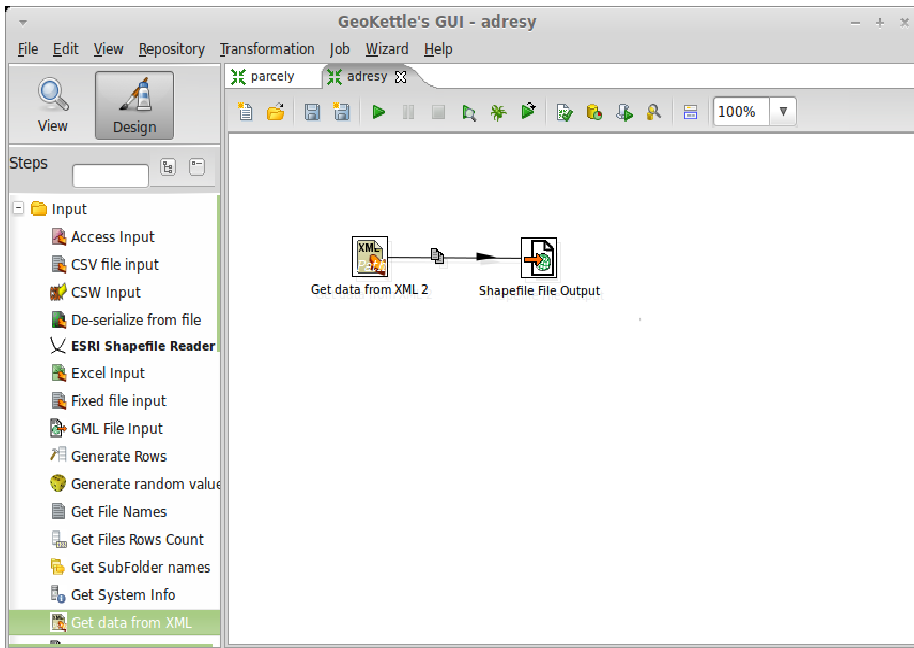


Figure 1. Diagram in GeoKettle GUI

4. Extract of Address Points From Exchange Format for Register of Territorial Identification, Addresses and Real Estate

There are only six simple steps for creating extract in GeoKettle.

- Construct simple diagram with two items (see Figure 1).
- Specify input XML file.
- Identify elements with address points (see Figure 2).
- Identify elements with coordinates.
- Set up transformation from gml:pos to WKT MULTIPOINT (see Figure 3).
- Set output to specified format.

The Figure 2 shows one of the possible ways how to identify elements with address points. An user can use GetXPath button to get all available paths that can be used for loop in XML file or the XPath can be written manually. In a case of address points of the specified exchange format XPath must be extended to exclude address points without geometry. If the extract contains also elements without geometry they can not be loaded into ESRI Shapefile, but another output (for example GML) can be specified.

Get XML Data

Step name: Get data from XML 2

Loop XPath: //vf:AdresniMisto[count(./adi:Geometrie) > '0'] Get XPath

Encoding: UTF-8

Namespace aware?

Ignore comments?

Validate XML?

Use token

Ignore empty file

Do not raise an error if no files

Limit: 0

Prune path to handle large files:

Include filename in output? Filename fieldname:

Rownum in output? Rownum fieldname:

Add files to result filename

Figure 2. Identify elements with address points

Final XPath expression can look like at the following example:

```
//vf:AdresniMisto [ count ( ./ adi:Geometrie ) > '0' ]
```

The Figure 3 shows setup of the output attributes stored in ESRI Shapefile including Geometry attribute. ESRI Shapefile load driver can read geometry only in WKT (Well Known Text) format according to OGC (Open Geospatial Consortium) so the coordinates from GML must be transformed with XPath to WKT format.

Final XPath expression can look like at the following Example:

```
//concat( 'MULTIPOINT_(( ', string(adi:Geometrie/
adi:DefinicniBod/adi:AdresniBod/gml:Point/gml:pos), '))' )
```

After these six steps the process can be executed. After the execution an user obtains results and report about the process run.

5. Conclusions

Extraction of data from Exchange format for Register of Territorial Identification, Addresses and Real Estate can be simply done in GeoKettle. I personally still prefer to do transformations of such files with XSLT and GDAL/OGR, but the

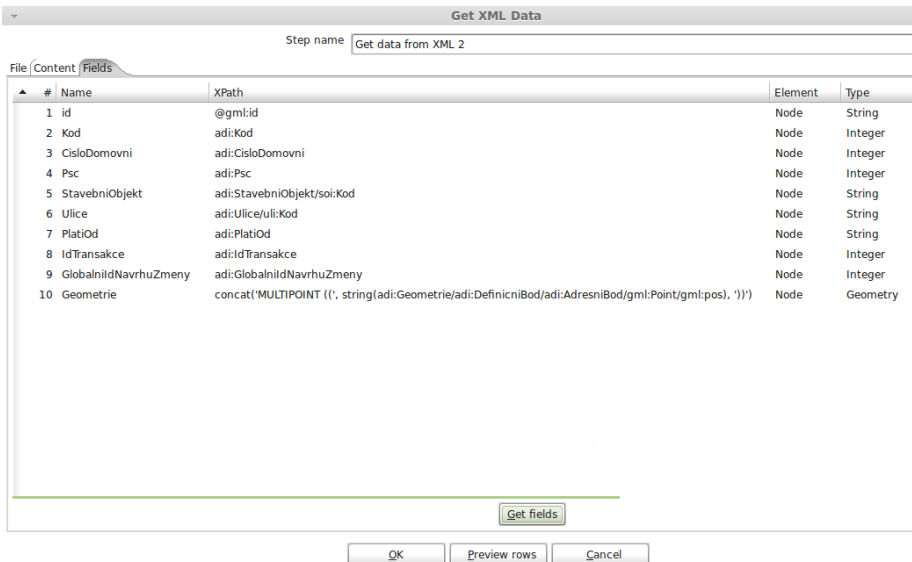


Figure 3. Setup transformation from gml:pos to WKT MULTIPOINT

GeoKettle option seems to be interesting and simple enough for many GIS users. What can be also important are the other possibilities in GeoKettle. The process can be executed on several input files in one run, so an user can extract data from several files and load result into one file. The diagram can be extended with other items, for example with coordinate transformation or split output into several files according to some conditions (for example identifier of a municipality).

Acknowledgment. This ‘research’ was supported by Antonín Bařinka from Regional Office of Zlín Region with his nice idea how to present possibilities of GeoKettle on a real example.

References

- [1] Czech Office for Surveying, Mapping and Cadastre: *Exchange format RUIAN (VFR)*, <http://www.cuzk.cz/Dokument.aspx?PRARESKOD=998&MENUID=10769&AKCE=DOC:10-VFR>.
- [2] Spatialytics.org: *GeoKettle*, <http://www.spatialytics.org/projects/geokettle/>.

Contact address

Ing. Jan Růžička, Ph.D., Institute of Geoinformatics, Faculty of Mining and Geology, VŠB-Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic,
E-mail address: jan.ruzicka@vsb.cz, <http://gis.vsb.cz/>

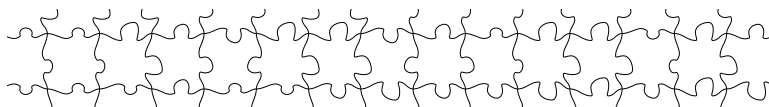
TEX A JEHO PRIATELIA

„Proč ten TeX?“, řekl-li by si návštěvník či návštěvnice OSSConf.

„Je to silný automatizační nástroj, bez kterého se nedá žít!“, by mohla znít odpověď organizátorů OSSConf. Podobně jako bez babylónu jazyků, jako jsou Python, PHP, MySQL, Java, Lua, ... Výrok podpoří či vyvrátí příspěvek Jiřího Rybičky a Jana Přichystala o volbě optimálního programu.

Letos užitečnost TeXu bude demonstrovat celá řada příspěvků. O kresbě pomocí balíčků `epic`, `eepic`, `curves` a `fp` (Rudolf Blaško), přes představení významného balíčku `tikz` (Aleš Kozubík) až po technicky náročnější kresbu v Teorii grafů (Rudolf Blaško, Aleš Kozubík).

U výpočtů zůstaneme, neb si bude možné vyslechnout přednášku o balíčku `sweave` (Pavol Král), balíčku `spreadtab` (Zuzana a Aleš Kozubíkovi) i výpočetním přístupem přes balíček `tikz`, např. na přípravu knižních obálek (Pavel Stříž).



Mezi praktickými problémy zůstaneme, a to u balíčků `pdfpages` na vyřazování PDF souborů a čárových kódů s jádrem v balíčku `pst-barcode` (Pavel Stříž).



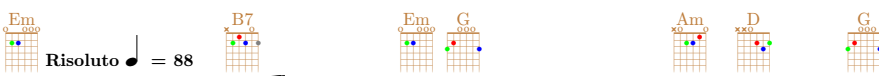
Podíváme se i na další oblasti TeXu. Především na WYSIWYG editor LyX (Pavol Král), webové prostředí TeXonWeb (Pavol Král) či nahlédneme na tvorbu animací u kreseb nakreslených v GeoGebře za pomoci TeXového světa, konkrétně balíčku `animate` (Pavel Stříž).

Bude to tedy jedna užitečnější oblast vedle druhé, ale čeká vás i několik překvapení. Neptejte se, nechejte se překvapit. ☺


Poodhalíme však roušku tajemství, že letos natukneme obzvláště vypečenou oblast, a tou budou experimenty v oblasti matematiky a umění (Math+Art). Za pozornost dáváme výstavku Vizualní objekty (Pavel Praks), přednášku Objímání paraboly (Michal Kaukič) a plamenně fraktálové chaosení pomocí programů `Apophysis` a `Incendia` v rámci příprav náhledových knih (Pavel Stříž).

Nebudeme mnoho smutnit, ale na všechny zajímavosti a užitečnosti \TeX ového světa se v roce 2012 nedostane. Například nepadne ani slovo k sazbě hudebnin, viz např. MusiX \TeX . Dovolíme si však v závěru úvodního slova \TeX ové sekce na tuto sazbu upozornit písní nejdůležitější.





Risoluto ♩ = 88



Nad Tatrou sa blýska, hromy di-vo bi-jú, nad Tatrou sa blýska, hromy di-vo bi-jú,
 To Slovensko na-še po-siaľ tvr-do spa-lo, to Slovensko na-še po-siaľ tvr-do spa-lo,

Typeset in the \TeX typography system (LuaLa)

Za \TeX ové přátele se na Vás těší,

Pavel Stříž
 Bučovice, 20. června 2012

NEBOJME SA OBRÁZKOV V \LaTeX -U

RUDOLF BLAŠKO (SK)

Abstrakt. Prostredie `picture` v \LaTeX -u ponúka možnosť ako priamo do textu vkladať pomocou jednoduchých príkazov vlastné interné obrázky bez nutnosti externých zásahov. Šikovný užívateľ dokáže v tomto prostredí nakresliť až neuveriteľné obrázky. V dnešnej dobe už existujú oveľa účinnejšie prostriedky na grafické ilustrácie výsledkov (napr. `TikZ`), ale to neznižuje kvalitu a význam prostredia `picture`.

Kľúčové slová. \LaTeX , `picture`, výpočty v pevnej rádovej čiarky, vizualizácia.

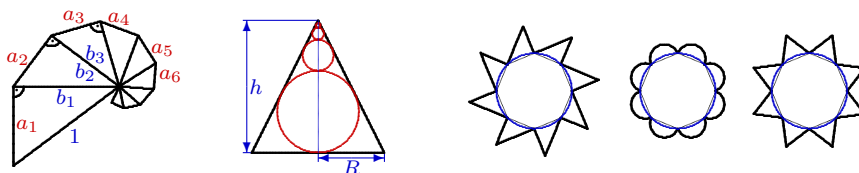
DO NOT BE AFRAID OF THE IMAGES IN \LaTeX

Abstract. The `picture` environment allows to program and include simple pictures into \LaTeX source without using external tools. Seasoned users are able to produce in this environment almost incredible graphics. Recently, there are more powerful tools for creating graphical illustrations of scientific results (e.g. `TikZ`), but the quality and simplicity of `picture` environment remains.

Keywords. \LaTeX , `picture`, Fixed Point Package, Visualisation.

Úvod

V osemdesiatych rokoch minulého storočia vytvoril *profesor Donald Knuth* typografický systém \TeX , a od samého počiatku má v sebe integrované prostredie `picture` na kreslenie obrázkov. Šikovný \LaTeX -ový maliar dokáže v tomto prostredí nakresliť pomerne zložité obrázky. Ale nič nie je zadarmo, tvorba zložitejších obrázkov nie je triviálna a predpokladá určité znalosti zo základov matematickej analýzy a z analytickej geometrie. Navyše tieto konštrukcie bývajú väčšinou dosť pracné a vyžadujú predprípravu z iných externých programov.



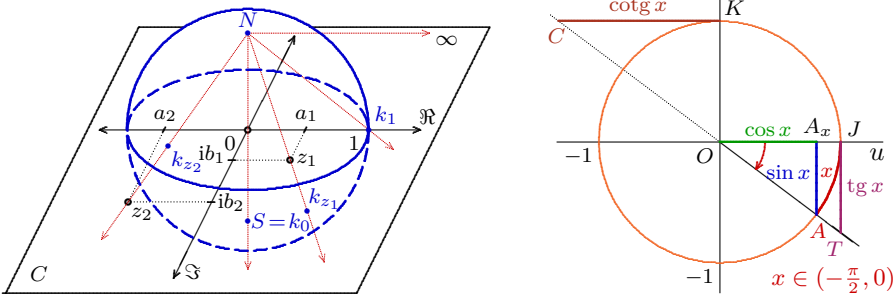
Obr. 1. Príklady nekonečných radov ilustrovaných v geometrii

Pre bežného užívateľa je vhodným prostriedkom na kreslenie balíčok `TikZ` so svojimi súčasťami [5]. `TikZ` je mladý projekt, vytvoril ho asi pred dvomi rokmi

Till Tantau z Lübecku. Názov je odvodený z nemeckého „TikZ ist kein Zeichenprogramm“ (TikZ nie je žiadny značkovací program, s *Zeichen* = (grafická) značka, znamenie, znak) a naznačuje jeho veľké možnosti. V spojení s balíčkom *fp* sa z neho stáva výkonný nástroj na kreslenie rôznorodých obrázkov.

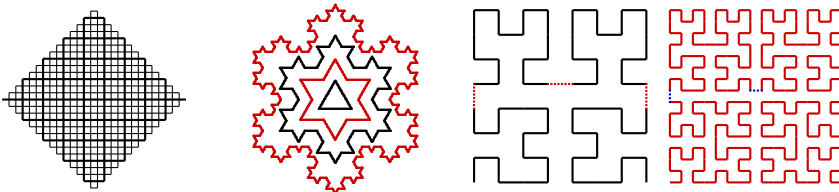
1. Prostredie *picture*

Cielom článku je poukázať na niektoré možnosti, ktoré poskytuje prostredie *picture*. V prvej časti príspevku sú obrázky vytvorené iba pomocou príkazov prostredia *picture*. V druhej časti je uvedené makro, ktoré (v spojení s balíčkom *fp*) k danej elipse vypočíta a nakreslí dotyčnicu tak, aby bola rovnobežná alebo kolmá na vopred určenú priamku.



Obr. 2. Stereografická projekcia komplexných čísel a definícia goniometrických funkcií

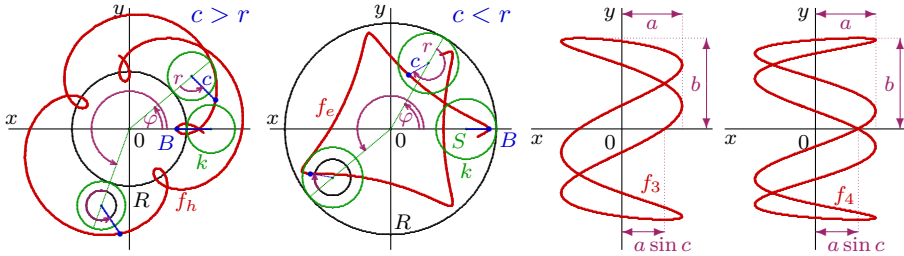
Väčšina obrázkov v tomto článku bola pôvodne vytvorená do vysokoškolskej učebnice *Matematická analýza 1*, ktorá v tlačenej podobe vyšla v roku 2009. Jej voľne dostupná, podstatne rozšírená, predchodkyňa je už vyše 10 rokov uložená na adrese <http://frcatel.fri.uniza.sk/~beerb/ma1/ma1.pdf>. Túto učebnicu som začal písať asi pred dvanástimi rokmi. V tom čase boli iba dve možnosti ako vytvoriť „pekné“ obrázky v *L^AT_EX*-u. Buď ich nakresliť v nejakom kresliacom programe a potom vložiť ako externé súbory, ale problém bol s popisným textom.



Obr. 3. Konštrukcia Peanovej krivky, van Kochovej snehovej vločky a Hilbertovej krivky

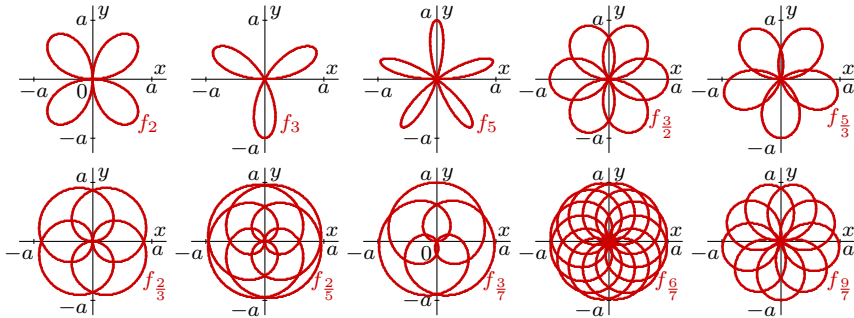
Tieto popisy ani zďaleka nedosahovali kvalitu okolitého textu a nivočiť *L^AT_EX*-ové dielo takýmito neestetickými zjavmi sa mi nezdalo vhodné. Tento problém sa dal čiastočne vyriešiť pomocou príkazu `\psfrag` (balíček *psfrag*). Toto riešenie

vyžadovalo vo vhodnom grafickom programe (napr. `xfig`, `tgif`) vytvoriť externý grafický súbor, v ňom urobiť relatívne odkazy na popisované miesta a na ich návestia sa odkazovať pomocou príkazu `\psfrag` v zdrojovom L^AT_EX-ovom súbore pri volaní externého grafického súboru, čo tiež nie je bezpracné. A navyše nutnou podmienkou bol celkový výstup do formátu `.ps` alebo `.eps`, t. j. preklad `TeX --> dvi --> eps`. Vo formáte `.dvi` popisy neboli viditeľné a formát `.pdf` ešte neexistoval. Toto riešenie som tiež nepovažoval za vhodné a zamietol som ho.



Obr. 4. Hypocykloida, epicykloida a Lissajousova krivka

Nakoniec som všetky obrázky nakreslil v prostredí `picture`. Všetky obrázky sú farebné (balíček `color` a neskôr `xcolor`). Problém bol s kreslením priamok a rôznych kriviek (príkaz `\line` má značne obmedzené možnosti). Tieto problémy riešia balíčky `curves` (príkazy `\curve`, `\arc`, `\bigcircle`, `\curvedashes`, ...) a `epic` (`\dottedline`, ...). Zložité krivky som kreslil pomocou príkazu `\curve`.



Obr. 5. Rovinná krivka s menom ruža

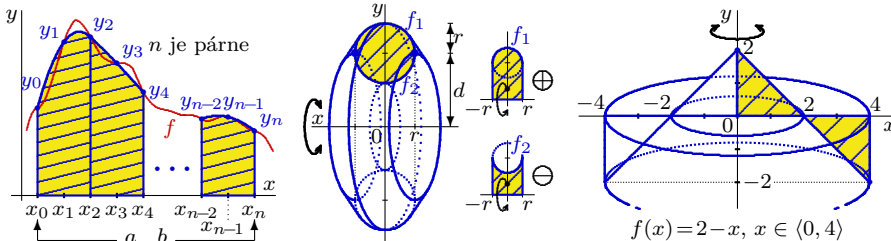
Najprv som si v nejakom programe vygeneroval x -ové aj y -ové súradnice (dostatočne blízky) bodov a následne ich vyexportoval do zdrojového L^AT_EX-ového súboru. Takto som robil prakticky všetky konštrukcie, v nejakom programe som si vypočítal súradnice bodov, polomery, resp. iné rozmery a následne som ich exportoval ako parametre kresliacich príkazov prostredia `picture`.

Obrázky uvádzam bez zdrojových kódov. Tieto spolu s použitými makrami sú uvedené na <http://frcatel.fri.uniza.sk/~beerb/latex/obr-makra.pdf> [3]. Ako dokazuje obr. 3, pri troche trpezlivosti sa v prostredí `picture` dajú nakresliť

aj fraktály. Na obr. 4 sú znázornené hypocykloida, epicykloida a Lissajousova krivka: $f_{h,e} : x = (R \mp r) \cos \varphi \pm c \cos \frac{(R \mp r)\varphi}{r}$, $y = (R \mp r) \sin \varphi - c \sin \frac{(R \mp r)\varphi}{r}$, $R > r > 0$, $c > 0$ $f_n : x = a \sin(nt+c)$, $y = b \sin t$, $a > 0$, $b > 0$, $c, t \in \mathbb{R}$, $n \in \mathbb{N}$.

Veľmi zaujímavá a variabilná je rovinná krivka ruža (rhodonea) ktorá je v polárnom systéme definovaná $f_c : \rho = a \sin c\varphi$, $a > 0$, $c > 0$, $\varphi \in \mathbb{R}$ (obr. 5).

Ako som už spomínal, obrázky sú farebné a do tlačenej podoby sa premenili na čiernebiele v preambule voľbou `\usepackage[monochrome]{xcolor}`. Problém je iba v tom, že všetky použité farby (vrátane bielej `white`) sa premenia na čiernu. Preto sa tento parameter nedá použiť pri farebne vyplňaných plochách. Vtedy je užitočná voľba `\usepackage[gray]{xcolor}`, pri ktorej sa farby prešikávajú na oddtíene šedej (analogicky ako pri robení čiernobielych fotografií z farebného negatívu). Ako dokazuje obr. 6, pri troche šikvosti a šťastia sa v prostredí `picture` dajú plochy vyplňať rôznymi farbami.



Obr. 6. Numerické integrovanie Simpsonovou metódou, objem anuloidu a objem rotačného telesa vzniknutého rotáciou trojuholníkov okolo osi y

2. Prostredie `picture` a balíček `fp`

Balíček `fp` umožňuje priame výpočty pri tvorbe dokumentu. Pri výpočtoch môžeme používať aritmetické operácie (+, -, ·, /), matematických funkcie (napr. `sin`, `cos`, `ln`, `√`) a taktiež konštanty π , e . Počítat môžeme s reálnymi číslami v rozsahu $\pm \underbrace{999 \dots 999}_{18 \text{ miest}}, \underbrace{999 \dots 999}_{18 \text{ miest}}$.

Pri výpočtoch (súradnic bodov, polomerov, uhlov ap.) budeme používať nasledujúce makrá. Makro `\rb{výraz}` upraví výraz na štyri desatinné miesta a zapíše ho do textu, druhé makro `\rbPr{\prikaz}{výraz}` pracuje podobne, ale výsledok priradí do príkazu `\prikaz` na ďalšie použitie:

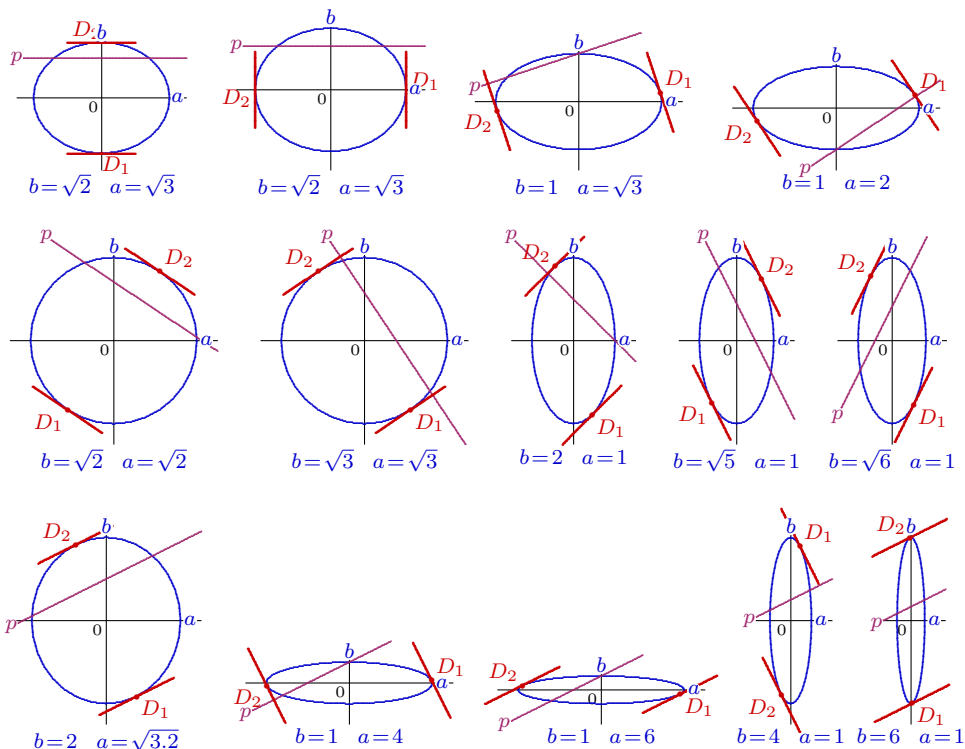
```
\newcommand\rb[2][4]{\FPeval{\rbV}{\clip(round(#2,#1))}\rbV}
\newcommand\rbPr[3][4]{\FPeval{\rbV}{\clip(round(#3,#1))}\FPset{#2}{\rbV}}
```

Pre ďalšie použitie je výhodné upraviť príkazy `\put` a `\multiput` tak, aby sme pozíciu zadávali ako parameter, ktorý sa pred umiestnením sám prepočíta:

```
\def\rbput[#1,#2]#3{\rbPr{\rbbAx}{#1}\rbPr{\rbbAy}{#2}\put(\rbbAx,\rbbAy){#3}}
\def\rbmultiput[#1,#2][#3,#4]#5#6{\rbPr{\rbbAx}{#1}\rbPr{\rbbAy}{#2}%
  \rbPr{\rbbBx}{#3}\rbPr{\rbbBy}{#4}\rbPr{\rbbPx}{#5}%
  \multiput(\rbbAx,\rbbAy)(\rbbBx,\rbbBy){\rbbPx}{#6}}
```

Ako príklad prepojenia prostredia `picture` a balíčka `fp` uvádzam na záver jeden z príkladov, ktorý ako učiteľ zadávam svojim študentom.

```
% E: x^2/A+y^2/B=1, p: y=sx/q+r, mierka=#1, rovnobezna/kolma=#2 [r resp. R/ine]
%   A=#3, B=#4, A>0, B>0,      s=#5, q=#6, r=#7, q<>0
%.....
\newcommand\prH[7][0.055mm]{\def\ZLE{{\cervena\bf BAD:}\
  mierka=[#1], rovnobezna/kolma=#2, $A=#3$, $B=#4$, $s=#5$,
  $q=#6$, $r=#7$\quad {\cervena\bf$\Rightarrow$ OK:}\quad
  $\frac{x^2}{A}+\frac{y^2}{B}=1$, $y=\frac{s}{q}x+r$, $A>0$, $B>0$, $q\ne 0$}%
\testujM[\rii]{-(#3)}\testujM[\riii]{-(#4)}\testujO[\rv]{#6}%
\ifthenelse{\rii=0 \or \riii=0 \or \rv=1}{\ZLE}{Nájdite dotyčnicu
k\testO{(#3)-(#4)}{u-kružnici}{~ellipse}~%
$\rbZlomokCelyHJ{1}{#3}{x^2}+\rbZlomokCelyHJ{1}{#4}{y^2}=1$
\ifthenelse{\equal{#2}{r}\or\equal{#2}{R}}{%
  rovnobežnú s~priamkou}{kolmú na~priamku}~%
$\rho:y=\rbZlomokCelyHJ{#5}{#6}{x}\testO{#5}{\rb{#7}}{\rbPMO{#7}}$}.
```



Obr. 7. Využitie makra `\prHobr`

Príklad je urobený ako makro `\prH` so siedmymi parametrami (mierka, rovnobežnosť, $A = a^2 > 0$, $B = b^2 > 0$ poloosi elipsy a $s, q \neq 0, r$ koeficienty priamky)

a zmenou jeho parametrov môžeme prakticky každému študentovi zadať iný príklad. K nemu patrí makro `\prHobr` (s rovnakými parametrami) na vykreslenie riešenia problému [3].¹ Po zadaní makra, napr. `\prH{r}{5}{1}{-2}{1}{-1}` sa vypíše:

„Nájdite dotyčnicu k elipse $\frac{x^2}{5} + y^2 = 1$ rovnobežnú s priamkou $p: y = -2x - 1$.“.

Po zadaní `\prH{k}{-5}{1}{-2}{1}{-1}` sa vypíše chybové hlásenie:

„**BAD:** mierka=[0.055mm], rovnobežna/kolma=k, $A = -5$, $B = 1$, $s = -2$, $q = 1$, $r = -1$ ⇒ **OK:** $\frac{x^2}{A} + \frac{y^2}{B} = 1$, $y = \frac{s}{q}x + r$, $A > 0$, $B > 0$, $q \neq 0$ “.

Na obrázku 7 je ilustrované praktické využitie tohto makra. V makre sa okrem iného upravujú zlomky a odmocniny, pokiaľ sú celými číslami [3]. Postupne je volané makro `\prHobr` s nasledujúcimi parametrami:

```
[0.045mm]{r}32021 [0.05mm]{k}32021 {k}31131 {k}4123{-1}
{r}22{-2}31 {k}33{-3}21 {k}14{-1}11 {r}15{-2}11 {r}16211
{r}{3.2}{2*2}121 {k}{16}1121 {r}{36}1121 {k}1{16}121 {r}1{36}121
```

Záver

Prostredie `picture` je výborný prostriedok ako kresliť obrázky priamo do textu. Jednoduché obrázky zvládne bez problémov aj \LaTeX -ový elév. Kresliť zložitejšie obrázky by som doporučoval aspoň stredne pokročilým čitateľom. Ale stále zostáva v zálohe slubne sa rozvíjajúci `TikZ`.

Literatúra

- [1] BALDA, M.: *Výpočty a diagramy v \LaTeX* , Zpravodaj Československého sdružení uživatelů \TeX , č. 2, ročník 14, CSTUG, 2004, Praha, ISSN 1213-8185.
- [2] BLAŠKO, R.: *\LaTeX a neobvyklé výpočty v pevnej rádovej čiarke*, Zborník 7. medzinárodnej konferencie Aplimat, 5.–8. februára 2008, Bratislava, ISBN 978-80-89313-04-4.
- [3] BLAŠKO, R.: *Súbor makier na kreslenie obrázkov*, <http://frcatel.fri.uniza.sk/~beerb/latex/obr-makra.pdf>.
- [4] KOZUBÍK, A.: *Naučím vás kresliť alebo predstavenie balíčka `TikZ`*, Otvorený softvér vo vzdelávaní, výskume a IT riešeníach, zborník medzinárodnej konferencie OSSConf 2012, Žilina, 2.–4. júla 2012, str. 91–96, ISBN 978-80-970457-2-2.
- [5] KOZUBÍK, A.: *Prezentačné materiály v triede `beamer`*, Otvorený softvér vo vzdelávaní, výskume a IT riešeníach, zborník medzinárodnej konferencie OSSConf 2011, Žilina, 6.–9. júla 2011, str. 249–258, ISBN 978-80-970457-1-5.

Kontaktná adresa

RNDr. Rudolf Blaško, PhD., Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská republika,
E-mailová adresa: beerb@frcatel.fri.uniza.sk, <http://frcatel.fri.uniza.sk/~beerb/>

¹K týmto makrám prináleží ešte jedno, ktoré zobrazí riešenie problému. Keďže nesúvisí so zameraním článku, neuvádzam ho.



OSSConf 2012: 85–90

\LaTeX , TEÓRIA GRAFOV A TROCHU UMENIA

RUDOLF BLAŠKO (SK) A ALEŠ KOZUBÍK (SK)

Abstrakt. TikZ je \LaTeX -ový balíček, ktorý umožňuje graficky ilustrovať prezentované výsledky (grafy funkcií, diagramy grafov, histogramy ap.). Ďalším účinným prostriedkom je súbor balíčkov fp, ktorý umožňuje vkladať do \LaTeX -ového textu priame výpočty aj s výsledkami. Spojenie týchto balíčkov dáva dosť účinný prostriedok ako na profesionálnej úrovni prezentovať v grafickej podobe výsledky svojho snaženia. V tejto práci sú naznačené možnosti ako vytvárať jednoduché makrá na kreslenie niektorých typov grafov (napr. úplný, bipartitný).

Kľúčové slová. \LaTeX , TikZ, teória grafov, výpočty v pevnej rádovej čiarky, vizualizácia.

\LaTeX , GRAPH THEORY AND A COUPLE OF ART

Abstract. TikZ is \LaTeX -package, which allows to illustrate the presented results (graphs of functions, charts, diagrams, histograms, etc.) graphically. Another effective tool is the set of fp packages that enables to enter direct calculations and their results in the text. The combination of these packages provides quite effective tools for professional graphical presentation of the results of our efforts. In the present paper we show the ways how to create simple macros to draw certain types of graphs (e.g. complete, bipartite etc.).

Keywords. \LaTeX , TikZ, Graph Theory, Fixed Point package, Visualisation.

Úvod

Typografický systém \LaTeX má integrované prostredie `picture` na kreslenie obrázkov. Zdatný užívateľ dokáže v tomto prostredí kresliť pomerne zložité obrázky [2]. Tvorba zložitejších konštrukcií je ale vykúpená značnou pracnosťou a predpokladá znalosti zo základov matematickej analýzy a z analytickej geometrie, čo v dnešnej dobe väčšine ľudí chýba. Pre bežného užívateľa je vhodným prostriedkom na kreslenie balíčkov TikZ so svojimi súčasťami [5].

Najčastejšie sa balíček TikZ používa na kreslenie rôznych grafov. A nie sú to len grafy funkcií, ale aj rôzne grafy vyskytujúce sa v bežnom živote (histogramy, stĺpcové, koláčové grafy ap.). V tomto príspevku ukážeme niekoľko možností ako využiť TikZ v teórii grafov, t. j. ako kresliť grafy zložené z vrcholov spojených orientovanými alebo neorientovanými hranami. Ako uvidíme neskôr, vhodným skladaním týchto grafov môžu vzniknúť elegantné „umelecké diela“.

Pri kreslení obrázkov (a nielen grafov) je vhodné TikZ doplniť o súbor balíčkov fp (Fixed Point package — výpočty v pevnej rádovej čiarky), ktorý dovoľuje používať matematické výpočty priamo v zdrojovom kóde \LaTeX -u a teda aj

v TikZ-e [1, 3]. Užitočný je taktiež balíček `ifthen`, ktorý umožňuje vetviť makrá do rôznych logických častí na základe ich argumentov.

Autori predpokladajú, že čitateľ má základné vedomosti o balíčkoch TikZ, fp a `ifthen`. Tento príspevok sa venuje tvorbe makier na vykresľovanie grafov a nie výkladu základných príkazov a funkcií uvedených balíčkov. Vzhľadom na limitovaný počet strán neuvádzame výpis týchto makier, iba ich stručný popis a použitie. Ich presné znenie je uvedené na adrese <http://frcatel.fri.uniza.sk/~beerb/latex/graf-makra.pdf> [4].

TikZ, fp a grafy

Balíček `fp` umožňuje priame výpočty pri tvorbe dokumentu. Pri výpočtoch môžeme používať aritmetické operácie (+, −, ·, /), matematické funkcie (napr. `sin`, `cos`, `ln`, `sgn`, $\sqrt{\cdot}$) a taktiež konštanty π , `e`. Počítat môžeme s reálnymi číslami v rozsahu $\pm \underbrace{999 \dots 999}_{18 \text{ miest}}, \underbrace{999 \dots 999}_{18 \text{ miest}}$.

Pri výpočtoch (súradnic bodov, polomerov, uhlov ap.) budeme používať nasledujúce makrá. Makro `\rb{výraz}` upraví výraz na štyri desatinné miesta a zapíše ho do textu, druhé makro `\rbPr{prikaz}{výraz}` pracuje podobne, ale výsledok priradí do príkazu `\prikaz` na ďalšie použitie:

```
\newcommand\rb[2][4]{\FPeval{\rbV}{\clip(round(#2,#1))}\rbV}
\newcommand\rbPr[3][4]{\FPeval{\rbV}{\clip(round(#3,#1))}\FPset{#2}{\rbV}}
```

Príkazy balíčka TikZ musia byť uvedené v prostredí `tikzpicture`, t. j. medzi `\begin{tikzpicture}` a `\end{tikzpicture}` alebo musia začínať príkazom `\tikz` (kvôli prehľadnosti ich v texte nebudeme pri príkladoch uvádzať). Vrcholy grafu môžeme nakresliť príkazom `\filldraw[blue](0,0)circle(8pt);` (modrý kruh ●). Na pohľad krajší výsledok ○ (bledomodrý kruh s modrým ohraničením) ponúka príkaz `\draw[blue,fill=blue!20](0,0)circle(6pt);`.

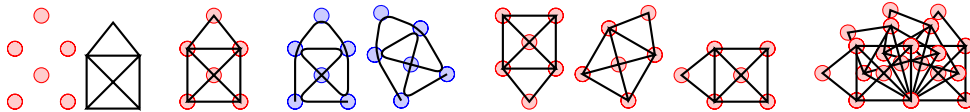
Použitím príkazov

```
\tikz{\draw[red](0,0)--(0,2);
\draw[blue,fill=blue!20](0,0)circle(8pt)(2,0)circle(8pt);}
```

dostaneme ○—○. Ak vymeníme poradie príkazov `\draw` dostaneme ○—○. Na obr. 1 je použité makro `\aaDom[n_par]`, ktoré vykresľuje detský domček¹. Nepovinné parametre `n_par` sa zadávajú pomocou `\def: \def\aaSkala{2}` (zmena škály), `\def\aaFarbaV{blue}` (farba vrcholov), `\def\aaFarbaH{green}` (farba hrán), `\def\aaOtoc{60}` (otočenie grafu), `\def\aaObtek{4pt}` (obtekanie vrcholov) [4].

Ak použijeme príkaz cyklu `\foreach \prem {zoznam}{príkazy}`, potom môžeme domček poskladať do zložitejších, až umeleckých konštrukcií (obr. 2, [4]).

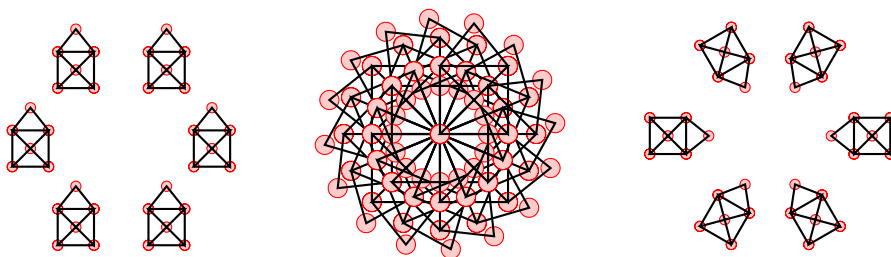
¹Nakresliť ho jedným ťahom bolo pre väčšinu z nás jedným z prvých, aj keď nevedomých, stretnutí s teóriou grafov.



Obr. 1. Detský domček, jeho kresba jedným ťahom, pootočenie a zloženie viacerých domčekov

Príkaz `\foreach` má povinný parameter `\prem`, ktorý postupne nadobúda všetky hodnoty z množiny `zoznam` a dosadzuje ich do príkazov `príkazy`. napríklad „kvietok“ (obr. 2 v strede) je vytvorený pomocou konštrukcie:

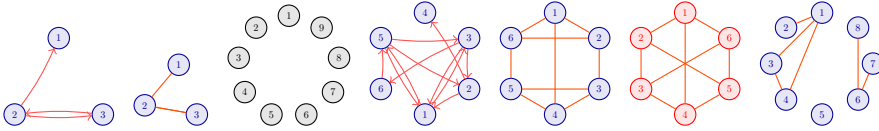
```
\tikz{\foreach \N in {1,...,16} {\rbPr{\NN}{360/16*(\N)}
\aaDom[\def\aaSkala{.5}\def\aaOtoc{\NN}]}}
```



Obr. 2. Použitie príkazu `\foreach` na kreslenie obrázkov

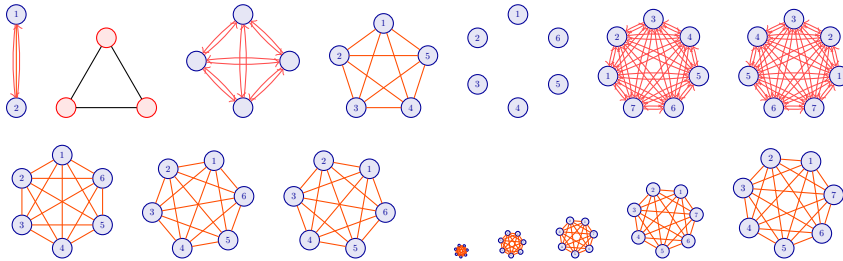
Nasledujúce makro vykreslí orientovaný alebo neorientovaný graf určený zoznamom hrán, vrcholy sa zobrazia do pravidelného n -uholníka:

```
\newcommand\CGraph[5] []{\def\aaPopis{+}\def\aaHcislo{1}\rbPr{\aaPolomer}{20}%
\def\aaFarbaV{blue!60!black}\def\aaFarbaH{red!80}\def\aaFarbaH0{red!70}%
\def\aaSkala{1}\rbPr{\aaMax}{32}\def\aaOtoc{0}\def\aaHrany{A}#1%
\rbPr{\aaahcislo}{\aaHcislo}\rbPr{\bbU}{\aaOtoc}\rbPr{\aaPocet}{#2}%
\def\aaOrient{#3}\def\aaCisla{#4}\rbPrCelaCast{\aaPocet}{abs(\aaPocet)}%
\rbPr{\aaPocet}{min(\aaPocet,\aaMax)}\rbPr{\aaPocet}{max(\aaPocet,1)}%
\ifthenelse{equal{\aaPopis}{+}}{\rbPr{\bbPopis}{1}}{\rbPr{\bbPopis}{-1}}%
\rbPr{\aaPocetN}{(\aaPocet)-1}\rbPr{\aaKrok}{360/(\aaPocet)}%
\begin{tikzpicture}[transform shape,scale=\aaSkala,rotate=-\bbU]
\foreach \number in {1,...,\aaPocet}{\rbPr{\ccU}{90-(\bbPopis)*360%
*(\aaahcislo)-1)/(\aaPocet)+(\bbPopis)*(\number-1)*(\aaKrok)}%
\ifthenelse{equal{\aaCisla}{A}}{%
\node[\aaFarbaV,fill=\aaFarbaV!10,draw,circle,minimum size=2.1em,rotate=\bbU]
(\number) at (\ccU:\aaPolomer em) {\textcolor{\aaFarbaV}{$\number$}};%
\node[\aaFarbaV,fill=\aaFarbaV!10,draw,circle,minimum size=2.1em]
(\number) at (\ccU:\aaPolomer em) {\phantom{88}};}%
\foreach \aaZ/\aaK in {#5}{%
\ifthenelse{equal{\aaOrient}{N}}{\path (\aaZ) edge[\aaFarbaH,-] (\aaK);}%
\path (\aaZ) edge[\aaFarbaH0,->,bend right=6] (\aaK);}%
\end{tikzpicture}}
```



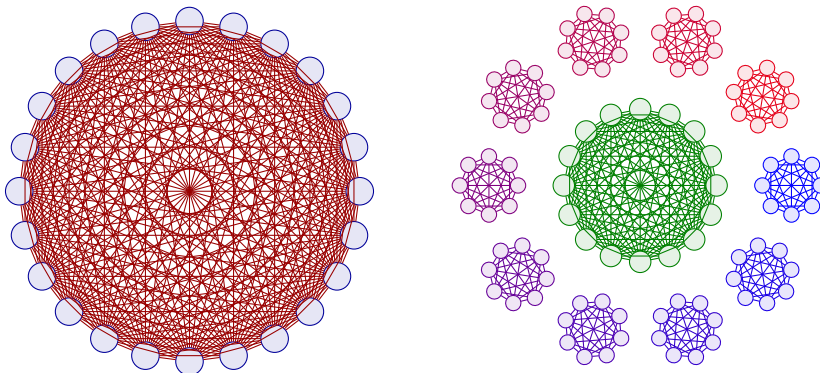
Obr. 3. Grafy zostrojené pomocou makra `\CGraph`

Hrany sa kreslia pomocou `\path (\aaZ) edge[\aaFarbaH,-] (\aaK);`, resp. `\path (\aaZ) edge[\aaFarbaH,->,bend right=6] (\aaK);`, kde `\aaZ` a `\aaK` sú začiatkové a koncové vrcholy. Makro má päť parametrov a používa sa v tvare `\CGraph[n_par]{n}{Or}{ukaz}{z_hran}` (obr. 3).

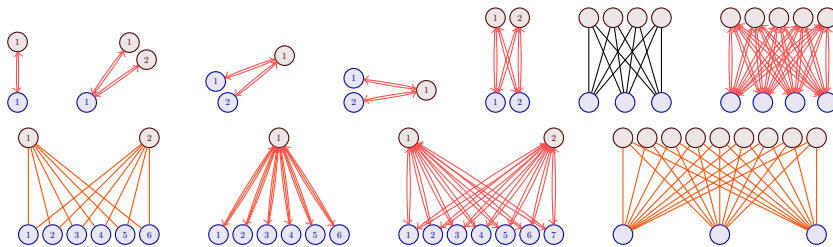


Obr. 4. Úplné grafy vytvorené pomocou `\CompleatGraph`

V parametri `n_par` môžeme definovať číslovanie vrcholov (+, resp. - zmysel) `\def\aaPopis{+}`, farbu vrcholov `\def\aaFarbaV{color}`, farbu neorientovaných a orientovaných hrán `\def\aaFarbaH{color}`, `\def\aaFarbaH0{color}`, polomer – vzdialenosť vrcholov od stredu `\def\aaPolomer{20}`, pootočenie grafu `\def\aaOtoc{0}` a škálu `\def\aaSkala{1}`.



Obr. 5. Aj graf môže tvoriť umelecké dielo

Obr. 6. Úplné bipartitné grafy vytvorené pomocou `\BiGraph`

Parameter `n` určuje počet vrcholov grafu (prirodzené číslo z intervalu $\langle 1, 32 \rangle$), parameter `Or` určuje orientovanosť grafu (0, resp. N) a parameter `ukaz` definuje, či sa bude alebo nebude zobrazovať číslovanie vrcholov (A, resp. N). Hrany grafu sa definujú v parametri `z_hran` ako zoznam dvojíc `cisloZ/cisloK` (číslo začiatočného/číslo koncového vrchola hrany), ktoré sú oddelené čiarkou. Graf na obr. 3 úplne vpravo je nakreslený príkazom:

```
\CGraph[\def\aaSkala{.35}\def\aaPolomer{5}]{8}{N}{A}{1/2,1/3,1/4,3/4,6/7,6/8}
```

Na záver tejto časti uvedieme príklady použitia niektorých nami vytvorených makier [4] na kreslenie rôznych úplných grafov. Úplný graf s n vrcholmi (zoraďenými do kruhu) nakreslí makro `\CompleatGraph[n_par]{n}{Or}{ukaz}` (obr. 4). Jeho vhodnou úpravou a kombinovaním môžeme vytvoriť rôzne elegantné tvary (obr. 5), ktoré zasahujú až do čipkárskeho umenia.

Makro `\BiGraph[n_par]{n1}{n2}{Or}{ukaz}` nakreslí úplný bipartitný graf s počtami vrcholov n_1 a n_2 (obr. 6, [4]).

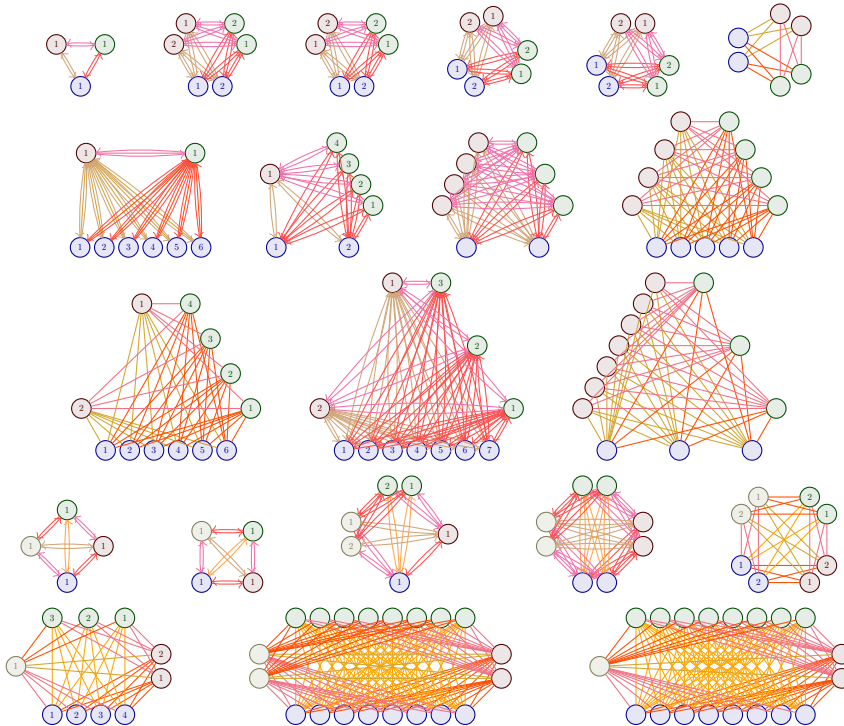
Makro `\TriGraph[n_par]{n1}{n2}{n3}{Or}{ukaz}` nakreslí úplný tripartitný graf s počtami vrcholov n_1 , n_2 a n_3 a analogický graf s počtami vrcholov n_1 , n_2 , n_3 a n_4 nakreslí makro `\TetraGraph[n_par]{n1}{n2}{n3}{n4}{Or}{ukaz}` (obr. 7, [4]).

Záver

Balíček `TikZ` je veľmi užitočný nástroj, pomocou ktorého aj nie veľmi pokročilí L^AT_EX-oví užívatelia dokážu kresliť pomerne zložité obrázky. Toto snaženie dokáže umocniť prepojenie s balíčkom `fp`, ktorý umožňuje v L^AT_EX-u priame výpočty.

Literatúra

- [1] BALDA, M.: *Výpočty a diagramy v L^AT_EXu*, Zpravodaj Československého sdružení uživatelů T_EXu, č. 2, ročník 14, CSTUG, 2004, Praha, ISSN 1213-8185.
- [2] BLÁŠKO, R.: *Nebojme sa obrázkov v L^AT_EX-u*, Otvorený softvér vo vzdelávaní, výskume a IT riešeniach, zborník medzinárodnej konferencie OSSConf 2012, Žilina, 2.–4. júla 2012, str. 79–84, ISBN 978-80-970457-2-2.



Obr. 7. Grafy vytvorené pomocou `\TriGraph` a `\TetraGraph`

- [3] BLAŠKO, R.: *LaTeX a neobvyklé výpočty v pevnej rádovej čiarky*, Zborník 7. medzinárodnej konferencie Aplimat, 5.–8. februára 2008, Bratislava, ISBN 978-80-89313-04-4.
- [4] BLAŠKO, R., KOZUBÍK, A.: *Súbor makier na kreslenie grafov*, <http://frcatel.fri.uniza.sk/~beerb/latex/graf-makra.pdf>.
- [5] KOZUBÍK, A.: *Naučím vás kresliť alebo predstavenie balíčka TikZ*, Otvorený softvér vo vzdelávaní, výskume a IT riešeníach, zborník medzinárodnej konferencie OSSConf 2012, Žilina, 2.–4. júla 2012, str. 91–96, ISBN 978-80-970457-2-2.

Kontaktné adresy

RNDr. Rudolf Blaško, PhD., Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská republika,
E-mailová adresa: beerb@frcatel.fri.uniza.sk, <http://frcatel.fri.uniza.sk/~beerb/>

RNDr. Aleš Kozubík, PhD., Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská republika,
E-mailová adresa: alesko@frcatel.fri.uniza.sk



OSSConf 2012: 91–96

NAUČÍM VÁS KRESLIŤ ALEBO PREDSTAVENIE BALÍČKA TikZ

ALEŠ KOZUBÍK (SK)

Abstrakt. Tento príspevok predstavuje balíček TikZ. Na obmedzenom priestore ilustruje elementárne nástroje tohto balíčka.

Kľúčové slová. L^AT_EX, TikZ, kresby.

I'LL TEACH YOU TO DRAW OR PRESENTATION OF THE TikZ PACKAGE

Abstract. This article brings the presentation of the TikZ package. On a limited space it introduces the very elementary tools of the package.

Keywords. L^AT_EX, TikZ, drawings.

Úvod

Ak sa už raz pustíme do sadzby dokumentov, skôr alebo neskôr nutne narazíme na potrebu vkladania grafických informácií do textu. Často sú to už hotové obrázky zhotovené pomocou špecializovaných grafických editorov alebo fotografie, ktoré jednoducho vkladáme na určené miesto v dokumente. No sú aj situácie, kedy žiadny zdroj potrebný grafický výstup neposkytuje a je teda na nás, aby sme si ho zhotovili takrečeno „svojpomocne“. V ďalšom texte sa predpokladá, že čitateľ má zvládnutú prácu so systémom T_EX, resp. L^AT_EX na úrovni príjemnej učebnice [4], podrobnej príručky [3] alebo aspoň zhutnených informácií z príspevkov [1] a [2].

Pri práci so systémom T_EX máme k dispozícii základnú sadu kresliacich nástrojov prostredia `picture`, ktorá umožňuje kreslenie jednoduchých aj zložitejších obrázkov, disponuje však pomerne obmedzenou sadou nástrojov a dosiahnutie požadovaného výsledku nemusí byť vždy jednoduché a užívateľsky pohodlné. Výhodou však je, že jeho výstupy dobre spolupracujú so všetkými drivermi. Ďalšou možnosťou je silný balík `pstricks`, je však špecializovaný na použitie postscriptovej grafiky a nie vždy spolupracuje s inými drivermi. Spomeňme aspoň vymenovaním niektoré ďalšie alternatívne balíčky ako sú `dratex`, `xypic`, `xfig`, alebo zo skupiny priateľov T_EXu program `metapost`.

My si v tomto príspevku v krátkosti predstavíme veľmi silný nástroj pre prácu s grafikou formátu `pgf` (Portable Graphics Format) v podobe balíčka TikZ (T ist kein Zeichenprogramm), ktorého autorom je Till Tantau. Manuál [6] tohto balíčka

je voľne dostupný napr. na adrese [8]. Nedajte sa odradiť jeho dĺžkou (takmer 730 strán) a majte vždy na pamäti, že manuál nie je váš nepriateľ.

Prv ako sa pustíme do experimentovania s balíčkom TikZ, treba si ešte povedať, že balíček dobre pracuje s výstupmi vo formátoch PostScript (`dvips`) a pdf (`pdflatex`, `dvipdfmx`), naopak dví prehliadače nemusia vždy výsledok zobrazovať korektne. Je preto potrebné pozeráť si vždy výstup `ps` alebo `pdf`.

1. Začíname

Ak chceme pre svoje spracovanie obrázkov využívať nástroje z balíčka TikZ, musíme v prvom rade v preambule štandardným spôsobom jeho použitie deklarováť, teda vložiť riadky:

```
\usepackage{tikz}
\usetikzlibrary{arrows,shapes,trees,...}
```

Kým prvý riadok je nevyhnutný pre použitie balíčka TikZ, druhý riadok je nepovinný a slúži pre použitie niektorých špeciálnych rozširujúcich knižníc balíčka.

Pre samotné zaradenie obrázkov do textu máme dve možnosti. Prvou je príkaz `\tikz ... ;`, slúžiaci pre jednotlivé príkazy a zobrazenie výsledku na riadku. Tak napríklad príkazom `\tikz \draw[->] (0pt,0pt) -- (20pt,6pt);` vložíme do riadku šípku \longrightarrow . V prípade, že výsledok má byť vytvorený postupnosťou viacerých príkazov, uzavrieme ich do skupinových zátvoriek.

Pre vloženie obrázkov väčšieho rozsahu používame prostredie `tikzpicture`, teda:

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

2. Základné stavebné kamene






Činnosť makier z balíčka TikZ sa opiera o dva základné kamene. Sú to „cesty“ (`path`) a „uzly“ (`node`). Cestu môžeme charakterizovať ako postupnosť segmentov priamok a kriviek. Uzly potom môžeme označiť za prvky, ktoré sú pridávané ku cestám po ich vykreslení. Priblížme si teraz tieto základné dva prvky detailnejšie.

2.1. Paths čiže cesty

Cesta zadávaná príkazom `\path` nám slúži na vcelku ľubovoľné spájanie dvoch alebo viacerých bodov, pričom týmto spojniciam je možné pridávať mnohé vlastnosti. V zásade je možné jednotlivé cesty kresliť (`draw`), vyplňať vzniklé uzavreté tvary (`fill`) alebo vystrihovať pre ich následné použitie v ďalšom kreslení (`clip`). Vykreslenie čiary teda zadávame príkazom `\path[draw]`, čo je možné skrátiť na `\draw`, ak chceme vyplniť vzniklý útvar (zatiaľ nešpecifikovanou výplňou), použijeme `\path[fill]` alebo skrátene len `\fill` a konečne pre vystrihnutie určitej

časti `\path[clip]` alebo len `\clip`. Výsledky jednotlivých príkazov a ich kombinácií ilustruje tabuľka 1.

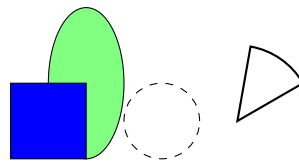
Tabuľka 1. Ukážky výsledkov základných príkazov pre kreslenie

	<code>\draw(1,1)--(1.5,1.5)--(2,1);</code>
	<code>\draw[line width=3pt](1,1)--(1.5,1.5)--(2,1)--cycle;</code>
	<code>\draw[fill=blue!30](1,1)--(1.5,1.5)--(2,1)--cycle;</code>
	<code>\fill[blue](1,1)--(1.5,1.5)--(2,1)--cycle;</code>
	<code>\draw[clip](1,1)--(1.5,1.5)--(2,1)--cycle;</code> <code>\fill[fill=red!50](1.5,1.6)circle(.5);</code>

Ako si možno všimnúť z tabuľky 1, jednotlivé príkazy majú rôzne možnosti voľby nepovinných argumentov. Ich význam v tu uvedených príkladoch je dostatočne intuitívny a kompletný zoznam možno nájsť v už spomenutom manuáli [6]. Všimnime si tiež, že každý príkaz je ukončený bodkočiarkou, čo je u príkazov balíčka TikZ povinné.

V balíčku TikZ sú preddefinované aj viaceré jednoduché geometrické tvary. Tak napríklad obdĺžnik vykreslíme príkazom `\draw(0,0)rectangle(2,1);`. Tu v prvej zátvorke uvádzame súradnice ľavého dolného rohu a v druhej súradnice pravého horného rohu. Vykreslenie kružnice so stredom v danom bode (a,b) a polomerom r docielime príkazom `\draw(a,b)circle(r);`. Konečne elipsu, ktorá je určená svojím stredom (s,t) a polosami (a,b) nakreslíme pomocou `\draw(s,t)ellipse(a and b);`. Použitie týchto príkazov a výsledok ilustruje nasledujúca ukážka.



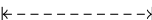



```
\begin{tikzpicture}
\draw[style=dashed](2,.5)circle(0.5);
\draw[fill=green!50](1,1)
ellipse(.5 and 1);
\draw[fill=blue](0,0)rectangle(1,1);
\draw[style=thick]
(3,.5)---+(30:1)arc(30:80:1)--cycle;
\end{tikzpicture}
```



V tejto ukážke si môžeme tiež všimnúť voliteľný parameter `style` príkazu `\draw`, ktorý upravuje vzhľad vykreslenej čiary na čiarkovanú, resp. tučnú. Posledný príkaz potom ilustruje možnosť použitia polárnych súradníc k vykresleniu časti oblúka `arc` kružnice, v tomto prípade v úseku od 30 po 80 stupňov.

Na záver tohto odstavca si ešte ukážeme niekoľko možností pre štýly kreslených čiar a možnosti vybavenia ich koncov šípkami rôznych tvarov. Pohľad do tabuľky 2 je určite dostatočne ilustratívny pre ich používanie.

Tabuľka 2. Ukážky rôznych štýlov kresliacich čiar

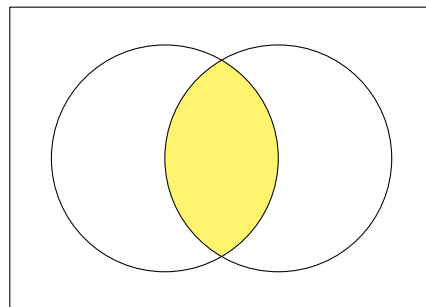
	<code>\draw[->] (0,0.5)--(2,0.5);</code>
	<code>\draw[thick,loosely dotted, >->>] (0,0.5)--(2,0.5);</code>
	<code>\draw[thin,densely dashed, <->] (0,0.5)--(2,0.5);</code>
	<code>\draw[double,>-<] (0,0.5)--(2,0.5);</code>
	<code>\draw[very thick,latex-latex] (0,0.5)--(2,0.5);</code>
	<code>\draw[stealth-latex reversed,dashed] (0,0.5)--(2,0.5);</code>

2.2. Vystrihovanie čiže clip

Voľný preklad anglického *clipping* do slovenčiny ako vystrihovanie nezodpovedá celkom presne tomu, čo sa pri použití príkazu `\clip` deje. Po použití príkazu `\clip` sú zo všetkých kresieb nasledujúcich po ňom „vystrihnuté“ len tie časti, ktoré ležia vo vnútri útvaru vykresleného pri použití `\clip`. Vysvetlíme si to na príklade v poslednom riadku tabuľky 1. Najskôr sme príkazom `\draw[clip]` „vystrihli“ trojuholník so zadanými vrcholmi. V ďalšom kroku sa už preto zobrazila len tá časť vyplneného kruhu, získaného pomocou `\fill`, ktorá leží vo vnútri tohto trojuholníka.

Trvalé vystrihnutie určitej oblasti z obrázku môže byť veľmi nepraktické. Preto máme k dispozícii prostredie `scope`, ktorým je možné obmedziť platnosť príkazu `\clip` iba v rámci tohto prostredia. Ukážeme si to na príklade zobrazenia prieniku dvoch množín. Tu najskôr do prostredia `scope` uzavrieme oblasť ohraničenú oblúkmi dvoch kružníc, ktorú následne vyfarbíme. Po ukončení prostredia `scope` pokračujeme vo vykreslení zvyšku oboch kružníc.

```
\begin{tikzpicture}
\draw (-2.8, 2) rectangle (2.8, -2);
\begin{scope}
\clip (-0.75, 0) circle (1.5);
\clip ( 0.75, 0) circle (1.5);
\fill[color=yellow!70] (-2,1.5)
rectangle (2,-1.5);
\end{scope}
\draw (-0.75, 0) circle (1.5);
\draw ( 0.75, 0) circle (1.5);
\end{tikzpicture}
```



2.3. Uzly čiže nodes

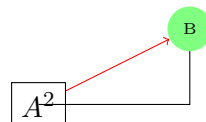
Uzly sú pridávané ku cestám po ich vykreslení. Pre ďalšie odkazovanie na tieto uzly pri tvorbe obrázka im môžeme priradiť aj mená. Jedným z praktických využití uzlov je pridávanie popiskov do obrázka. Tak napríklad lomenú čiaru, ktorej konce označíme písmenami môžeme vykresliť takto:

```
\begin{tikzpicture}
\draw (0,0) node {A}--(2,0)--(2,0.5) node {B};
\end{tikzpicture}
```



Ak by sme ešte chceli následne spojiť body A a B šípkou, využijeme možnosť uzly pomenovať. To prezentuje nasledujúca ukážka.

```
\begin{tikzpicture}
\draw (0,0) node[draw] (nodeA){ $A^2$ }--(2,0)--(2,1)
node[circle,fill=green!50] (nodeB) {\tiny B};
\draw[red,->] (nodeA)--(nodeB);
\end{tikzpicture}
```



Výhodnosť použitia pomenovaných uzlov môžeme pozorovať na červenej šípke, ktorá je viditeľná len po hranice objektov vykreslených v uzloch. Ak by sme sa spoliehali len na súradnice, prekryla by aj tieto objekty, čo by bolo v tomto prípade nežiadúce.

Každý uzol má okrem pomenovania a zadanej pozície dva parametre. Jedným je tvar a druhým textová značka. Tak sme v predchádzajúcej ukážke prideliť uzlu B tvar kruhu so zelenou výplňou a textovú značku B vysádzanú drobným písmom. Zároveň na obrázku pozorujeme, že umiestnenie písmena A je nevhodné a koliduje s vodorovnou čiarou, vychádzajúcou z tohto bodu. Tomu je možné zabrániť vhodným umiestnením textovej skupiny na vhodnú pozíciu relatívne ku polohe uzla. Nasledujúca ukážka ilustruje rozmiestnenie štyroch písmen na štyri základné pozície vzhľadom na polohu uzla vyznačeného bodkou.

```
\fill (0,2) circle (1pt) node[below] {$A$}
node[right] {$B$} node[above] {$C$}
node[left] {$D$};
```



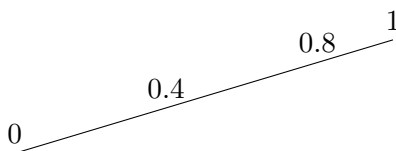
Polohu popisiek umiestnených do daného uzla je možné určiť aj kombinovaním štyroch základných polôh. Ilustruje to nasledujúca ukážka:

```
\fill (0,2) circle (1pt) node[below left] {$A$}
node[below right] {$B$} node[above right] {$C$}
node[above left] {$D$};
```



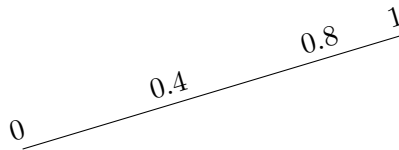
Na záver si ešte ukážeme, ako je možné umiestniť text do uzlov rozmiestnených pozdĺž vykreslenej cesty. K tomu slúži voliteľný parameter `pos` príkazu `\node`. Tento parameter nadobúda hodnoty medzi 0 a 1 a určuje tak relatívnu pozíciu uzlu medzi začiatkom a koncom vykreslenej cesty. Ukážka ilustruje rozmiestnenie štyroch uzlov pozdĺž cesty.

```
\draw (0,0)--(5,1.5)
node[pos=0,above]{0}
node[pos=.4,above]{0.4}
node[pos=0.8,above]{0.8}
node[pos=1,above]{1};
```



Ako vidíme z obrázka, poloha niektorých popisov nie je príliš estetická. To sa prejavuje u hodnôt 0.4 a 0.8, ktoré ako by smerom dopredu odstávali od vykreslenej priamky. Tento problém je potom možné odstrániť pridaním voliteľného parametra `sloped`, ktorý má za následok prispôsobenie sklonu písma ku sklonu vykreslenej cesty. Výsledok vidíme tu:

```
\draw (0,0)--(5,1.5)
  node[pos=0,above,sloped]{0}
  node[pos=.4,above,sloped]{0.4}
  node[pos=0.8,above,sloped]{0.8}
  node[pos=1,above,sloped]{1};
```



3. Záver

V príspevku sme si na obmedzenom priestore predstavili elementárne nástroje, ktoré môžu slúžiť pre potreby grafickej prezentácie obsahu. Akým silným nástrojom je balíček `TikZ` sa záujemcovia môžu presvedčiť napr. na stránke [7], ktorá obsahuje bohatú škálu príkladov, či už v podobe výsledných dokumentov formátu `pdf` alebo aj v podobe zdrojového kódu pre `LATEX`.

Literatúra

- [1] BLAŠKO, R.: *L^AT_EX nie je farba na maľovanie*, Zborník príspevkov medzinárodnej konferencie OSSConf 2010, 1.–4. júla 2010, Žilina, str. 43–52. ISBN 978-80-970457-0-8.
- [2] BLAŠKO, R.: *L^AT_EX nie je farba na maľovanie, ale na písanie*, Zborník príspevkov medzinárodnej konferencie OSSConf 2011, 6.–9. júla 2011, Žilina, str. 223–236. ISBN 978-80-970457-1-5.
- [3] KOPKA, H. – DALY, P.W.: *L^AT_EX – podrobný príručce*, Computer Press, Brno, 2004. ISBN 80-722-6973-9.
- [4] RYBIČKA, J.: *L^AT_EX pro začátečníky*, KONVOJ, Brno, 2003. ISBN 80-7302-049-1.
- [5] RYBIČKA, J. – ČAČKOVÁ, P. – PŘICHYSTAL, J.: *Průvodce tvorbou dokumentů*, NAKLADATELSTVÍ MARTIN STRÍŽ, Bučovice, 2011. ISBN 80-7302-049-1.
- [6] TANTAU, T.: *The TikZ and PGF Packages Manual for version 2.10*, <http://sourceforge.net/projects/pgf>.
- [7] <http://www.texample.net>, sample resources for T_EX users.
- [8] <http://www.ctan.org/tex-archive/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf>.

Kontaktná adresa

RNDr. Aleš Kozubík, PhD., Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská Republika,
E-mailová adresa: alesko@frcatel.fri.uniza.sk



OSSConf 2012: 97–102

TABUĽKOVÉ VÝPOČTY V SYSTÉME \LaTeX

ZUZANA KOZUBÍKOVÁ (SK) A ALEŠ KOZUBÍK (SK)

Abstrakt. V tomto príspevku predstavujeme balíček `spreadtab`, ktorý poskytuje nástroje na jednoduché tabuľkové výpočty v rámci \LaTeX -ového dokumentu.

Kľúčové slová. Tabuľkové výpočty, \LaTeX , `spreadtab`.

SPREADSHEET CALCULATIONS IN THE \LaTeX

Abstract. The article introduces the package `spreadtab`, that allows simple spreadsheet calculations in the \LaTeX documents.

Keywords. Spreadsheet calculations, \LaTeX , `spreadtab`.

Úvod

K významným súčasťam sadzby textu patrí aj spracovanie tabuliek. Táto činnosť je už sama o sebe pomerne náročná a navyše, pri sadzbe ekonomických textov (ale nielen pri nich), sa často stretáme s potrebou prepočítavať obsah tabuliek. Takéto výpočty sa obvykle realizujú v špecializovaných programoch – tabuľkových procesoroch a následne sa prenášajú do textu. To znamená spoliehať sa na exportné schopnosti týchto programov, alebo útrpné prenášanie údajov bunku po bunke z jednej tabuľky do druhej, s nemalým rizikom chýb a omylov.

Nedávno sa však v archíve CTAN objavil užitočný balíček `spreadtab`, ktorý dokáže tieto tabuľkové výpočty realizovať priamo v rámci systému \LaTeX . Jeho autorom je Christian Tellechea a nateraz posledná verzia balíčka bola realizovaná 13. mája 2012. Pre správnu činnosť balíčka `spreadtab` je potrebné updatovať balíček `xstring` na verziu 1.5 (uvolnená 28. 3. 2010) a mať nainštalovaný balíček `fp` pre výpočty v pevnej desatinnej čiarku.

V ďalšom texte predpokladáme, že čitateľ má zvládnutú prácu so systémom \TeX resp. \LaTeX aspoň na úrovni učebnice pre začiatočníkov [6], podrobnej príručky [5] alebo príspevkov [2] a [3] z minulých konferencií a je oboznámený s prácou s balíčkom `fp`, predstaveným v článku [1].

1. Vkládanie tabuľkových výpočtov

Pre používanie nástrojov je potrebné v preambule dokumentu načítať balíček `spreadtab` pomocou príkazu `\usepackage{spreadtab}`. Samotnú tabuľku, schopnú vykonávať tabuľkové kalkulácie je potom možné vložiť dvomi spôsobmi:

```
\begin{spreadtab}{<meno>}{<param>}          \spreadtab{<meno>}{<param>}
  samotný obsah tabuľky                       alebo1          samotný obsah tabuľky
\end{spreadtab}                               \endspreadtab
```

Tu `<meno>` znamená označenie ľubovoľného tabuľkového prostredia a `<param>` označuje jeho parametre.

Každú výpočtovú tabuľku môžeme považovať za dvojrozmerné pole, pričom každý jeho prvok – bunku tabuľky – môžeme jednoznačne identifikovať podľa riadku a stĺpca, v ktorých sa nachádza. To využívame pri odkazovaní sa na bunky vo výpočtoch.

1.1. Absolútne a relatívne odkazy

Absolútny odkaz na bunku je tvorený dvojicou `<stl><riad>`, kde `<stl>` je označenie stĺpca (písmenom, rozsah je tu limitovaný na písmená anglickej abecedy od `a` po `z`, teda 26 stĺpcov) a `<riad>` je označenie riadku (číslom).

Ako jednoduchý príklad si ilustrujme tabuľku, ktorá v poslednom stĺpci vypočíta súčin príslušného riadku a v poslednom riadku súčty stĺpcov.

```
\begin{spreadtab}{\tabular}{rrr|r}}
12 & 34 & 21& a1*b1*c1 \\
23 & 15 & 13& a2*b2*c2 \\
19 & 27 & 11& a3*b3*c3 \\ \hline
a1+a2+a3 & b1+b2+b3 & c1+c2+c3&a4+b4+c4
\end{spreadtab}
```

12	34	21		8568
23	15	13		4485
19	27	11		5643
54	76	45		175

Pri odkazovaní sa na jednotlivé bunky je často pohodlnejšie odkazy uvádzať relatívne, vzhľadom na polohu aktuálnej bunky, ktorej obsah sa určujeme. Takýto relatívny odkaz vytvárame dvojicou čísiel v tvare `[x,y]`, kde prvá hodnota `x` udáva vodorovný a `y` zvislý posun z pozície aktuálnej bunky. Tak napríklad odkaz `[-2,1]` odkazuje na bunku, ktorá sa nachádza o dva riadky vyššie ako je aktuálna pozícia a o jeden stĺpec vpravo od tejto pozície. Pripomeňme, že pre správnu interpretáciu znamienka mínus je pri použití balíčka `babel` s voľbami `czech` alebo `slovak` potrebné do dokumentu vložiť príkaz `\shorthandoff{-}`. Ako ilustratívny príklad si ukážeme tabuľku, v ktorej vygenerujeme niekoľko prvých členov Fibonacciho postupnosti.² Výsledok tentoraz zobrazíme v matematickom prostredí `matrix`, ktoré je definované v balíčku `amsmath`.

```

\spreadtab{\matrix{}}
1 & 1 & [-2,0]+[-1,0] \\
[2,-1]+[1,-1] & [1,-1]+[-1,0] & [-2,0]+[-1,0] \\
[2,-1]+[1,-1] & [1,-1]+[-1,0] & [-2,0]+[-1,0]
\endspreadtab$
```

1	1	2
3	5	8
13	21	34

¹Nové od verzie `Spreadtab v0.4b`.

²Pre nezásvätených pripomíname, že je to postupnosť, kde každý člen je súčtom dvoch predchádzajúcich členov.

Zo zdrojového kódu si možno všimnúť, že výhodou relatívnych odkazov je ich ľahké kopírovanie do iných buniek. (Druhý a tretí riadok sú prakticky rovnaké.)

1.2. Textové a zmiešané bunky

Nie je zriedkavosťou, že potrebujeme vytvoriť tabuľky, ktoré sú okrem číselných údajov tvorené aj bunkami, ktorých obsahom je text. Na začiatku bunky potom treba deklarovať, že s jej obsahom treba narábať ako s textom. V prostredí `spreadtab` je tento príznak textu implicitne nastavený na znak `@`. Tabuľka obsahujúca textové bunky teda môže vyzeráť napríklad takto:

```
\spreadtab{\tabular}{|l|*{4}{>{\$}c<{\$}|}}
\hline
@ hodnoty $x$ & 1 & 2 & 3 & -3\\
\hline
@ $f(x)=3x+1$ & 3*[0,-1]+1 & 3*[0,-1]+1
& 3*[0,-1]+1 & 3*[0,-1]+1 \\
\hline
\endspreadtab
```

hodnoty x	1	2	3	-3
$f(x) = 3x + 1$	4	7	10	-8

Pokiaľ chceme, môžeme si rozlíšenie textového obsahu bunky predefinovať. Príkazom `\renewcommand\STtextcell{’}` dosiahneme, aby sa textová bunka začínala znakom `’` (apostrof, ako je obvyklé u väčšiny tabuľkových procesorov). Tento príkaz umiestňujeme pred tabuľku, od ktorej chceme používať príslušný symbol na rozlíšenie textu.

V skutočnosti je každá bunka tabuľky tvorená dvomi poliami: textovým, ktoré je ignorované pri výpočtoch s balíčkom `fp`, a numerickým, ktoré obsahuje čísla a vzorce.

- Ak nie je nič deklarované, považuje sa celý obsah bunky za numerický a jej textové pole za prázdne.
- Ak sa pole bunky začína symbolom `@`, je bunka textová a numerické pole je prázdne a nedostupné.
- Ak pole bunky obsahuje značku `:=`, je nasledujúci argument, uzavretý v skupinových zátvorkách, považovaný za numerické pole a všetko ostatné za pole textové. Bunka má teda nasledujúcu štruktúru:

```
<textové pole>:=<numerické pole><zvyšok textového poľa>
```

Použitie znaku `:=`, je dôležité najmä v tvare `:=<{}>`, ktorý vytvára bunku s prázdny numerickým polom. Na rozdiel od textovej bunky s príznakom `@` je však toto numerické pole dostupné a je možné sa na jeho hodnotu odvolávať v odkazoch v rámci tabuľky. To je užitočné najmä pre príkaz `\STcopy`, ktorý si predstavíme v nasledujúcej sekcii.

Na záver tohto odstavca ukážka tabuľky so zmiešanými bunkami a výpočtom v nej.

```

\begin{spreadtab}{\tabular}{|c|c|}
\hline
číslo 1 : :={30} & číslo 2 : :={25} \\ \hline
súčet:\textbf{:=}{a1+b1} & priemer :
\textbf{:=}{(a1+b1)/2}} \\ \hline
\end{spreadtab}

```

číslo 1 : 30	číslo 2 : 25
súčet : 55	priemer : 27.5

2. Prenášanie vzorcov do viacerých buniek

Pre výpočty v tabuľkových procesoroch je typické opakovanie výpočtov podľa rovnakého vzorca vo viacerých bunkách. Ani v balíčku `spreadtab` takáto možnosť nechýba. Slúži k tomu príkaz `\STcopy` so syntaxou `\STcopy{>x,vy}<vzorec>`, kde x a y sú kladné konštanty. Hodnota x udáva, do koľkých buniek daného riadku smerom doprava sa vzorec nakopíruje a hodnota y udáva, do koľkých riadkov v danom stĺpci smerom nadol sa vzorec skopíruje.

Niektoré z vyššie uvedených ukážok by teda mohli s využitím kopírovania vyzerať takto:

```

$\spreadtab{\matrix{}}
1 & 1 & \STcopy{v}{[-2,0]+[-1,0]} \\
\STcopy{v}{[2,-1]+[1,-1]} & & \\
& \STcopy{v}{[1,-1]+[-1,0]} & \\
& & \\
\endspreadtab$

```

1	1	2
3	5	8
13	21	34

resp.

```

\spreadtab{\tabular}{|1|*{4}>{<{>}}|}}
\hline
@ hodnoty $x$ & 1 & 2&3&-3\\
\hline
@ $f(x)=3x+1$ & \STcopy{>}{3*[0,-1]+1}&&&\\
\hline
\endspreadtab

```

hodnoty x	1	2	3	-3
$f(x) = 3x + 1$	4	7	10	-8

3. Využitie pri sadzbe finančného textu

V tejto sekcii si ukážeme trochu pokročilejšie využitie prostredia `spreadtab` v kombinácii s výpočtami pomocou balíčka `fp` pri sadzbe textu z oblasti financií. Jedným z typických príkladov môže byť zvyšovanie efektívnej úrokovej miery pri zvyšovaní frekvencie úrokovania.³ Opäť je potrebné upozorniť na nutnosť inštrukcie `\catcode'\^=7` v prípade, že používame `babel` s voľbami `czech` alebo `slovak`. Predpokladajme, že základná ročná úroková miera je 10%. Túto hodnotu priradíme príkazom `\FPset{\AKi}{0.1}` do premennej `\AKi`, aby sme ju mohli ďalej používať vo výpočtoch. Hodnota n označuje počet úrokovacích období počas roka. Výslednú tabuľku potom získame takto:

³Podrobnejšie pozri [4]

```

\STautoround{8}
\FPset{\AKi}{0.1}
\begin{spreadtab}{\tabular}{|r|c|}}
\hline
@ $n$ & @ $(1+i)^n$ \\
\hline
1 & \STcopy{v}{(1+\AKi/[-1,0])^[-1,0]} \\
2 & \ 4 & \ 12 & \ 52 & \ 365 & \\
\hline
\end{spreadtab}

```

n	$(1+i)^n$
1	1.1
2	1.1025
4	1.10381289
12	1.10471307
52	1.10506479
365	1.10515578

Poznamenajme, že príkaz `\STautoround{x}` určuje najvyšší počet desatinných miest, na ktoré sa budú výsledky zaokrúhľovať.

Typickou výpočtovou schémou pri riadení financií je zostavenie umorovacieho plánu. Predpokladajme, že dlh vo výške $D = 20\,000$ je úročený úrokovou mierou 8% a má byť splatený konštantnými anuitami za obdobie 5 rokov. Úlohou je zostaviť umorovací plán. (Detaily pozri [4]). Najskôr si pripravíme výpočty pomocou balíčka `fp`. Vychádzame pri tom zo vzorcov napr. podľa [4].

```

\FPset{\AKi}{0.08}           % definovanie úrokovej miery
\FPset{\AKpocet}{5}         % definovanie počtu splátok
\FPset{\AKdlh}{20000}       % definovanie výšky dlhu
\FPadd\fa{\AKi}{1}          % výpočet úrokového faktoru 1+i
\FPpow\fa{\fa}{\AKpocet}    % úrokový faktor za n období (1+i)^n
\FPsub\fa{\fa}{1}
\FPdiv\AKsp{\fa}{\AKi}      % polehotný sporiteľ s=((1+i)^n-1)/i
\FPdiv\AKq{\AKdlh}{\AKsp}   % prvá splátka Q=D/s

```

Nasleduje už len práca s balíčkom `spreadtab`.

```

\STsetdecimalsep{,}         % definovanie oddeľovača desatinných miest
\renewcommand\STprintnum[1]{\numprint{#1}}
\nprouddigits2              % nastavenie pevného počtu 2 desatinných miest
\begin{center}
\begin{spreadtab}{\tabular}{|c|r|r|r|r|} \hline
@ Perióda & @\multicolumn{1}{c|}{D_{k-1}} \\
& @ \multicolumn{1}{c|}{Úroky} & @ \multicolumn{1}{c|}{Q_k} \\
& @\multicolumn{1}{c|}{Anuita} \\
\hline
@ $1$ & \AKdlh & \STcopy{v4}{[-1,0]*\AKi} & \AKq \\
& \STcopy{v4}{[-2,0]+[-1,0]} \\
@ $2$ & \STcopy{v3}{[0,-1]-[2,-1]} & \\
& \STcopy{v3}{[0,-1]*(1+\AKi)} & \\
@ $3$ & & \\
@ $4$ & & \\
@ $5$ & & \\
@ \textbf{Spolu} & \textbf{:=\sum(c2:c6)} & \textbf{:=\sum(d2:d6)} \\
& \textbf{:=\sum(e2:e6)} \\
\hline
\end{spreadtab}
\end{center}

```

A tu je výsledný umorovací plán. Symbol D_{k-1} označuje veľkosť dlhu na začiatku k -tej periódy, Q_k veľkosť splátky za danú periódu. Veľkosť anuity je potom súčtom splátky a úroku vypočítaného zo zostatku dlhu D_{k-1} .

Periódá	D_{k-1}	Úroky	Q_k	Anuita
1	20 000,00	1 600,00	3 409,13	5 009,13
2	16 590,87	1 327,27	3 681,86	5 009,13
3	12 909,01	1 032,72	3 976,41	5 009,13
4	8 932,60	714,61	4 294,52	5 009,13
5	4 638,08	371,05	4 638,08	5 009,13
Spolu		5 045,65	20 000,00	25 045,65

Záver

V príspevku sme si ukázali základné techniky pre tabuľkové výpočty s balíčkom `spreadtab`. Pomerne rýchlo sme dokázali zostaviť „samopočítaciu“ tabuľku umorovacieho plánu. Niektoré ďalšie techniky, ako je napr. práca s dátumom alebo náročnejšie formátovanie buniek, je možné nájsť v manuáli [7].

Podakovanie. Tento príspevok vznikol s podporou grantu VEGA 1/913/11.

Literatúra

- [1] BALDA, M.: *Výpočty a diagramy v L^AT_EXu*, Zpravodaj Československého sdružení uživatelů T_EXu, ročník 14 (2004), číslo 2, 54–110, ISSN 1211-6661.
- [2] BLAŠKO, R.: *L^AT_EX nie je farba na maľovanie*, Zborník príspevkov medzinárodnej konferencie OSSConf 2010, 1.–4. júla 2010, Žilina, 43–52, ISBN 978-80-970457-0-8.
- [3] BLAŠKO, R.: *L^AT_EX nie je farba na maľovanie, ale na písanie*, Zborník príspevkov medzinárodnej konferencie OSSConf 2011, 6.–9. júla 2011, Žilina, 223–236, ISBN 978-80-970457-1-5.
- [4] CIPRA, T.: *Finanční matematika v praxi*, EDICE HZ, Praha, 1993, ISBN 80-901495-1-0.
- [5] KOPKA, H. – DALY, P. W.: *L^AT_EX – podrobný průvodce*, Computer Press, Brno, 2004, ISBN 80-722-6973-9.
- [6] RYBIČKA, J.: *L^AT_EX pro začátečníky*, KONVOJ, Brno, 2003, ISBN 80-7302-049-1.
- [7] TELLECHEA, CH.: *Spreadtab. User's manual*, http://www.ctan.org/tex-archive/macros/latex/contrib/spreadtab/spreadtab_doc_en.pdf.

Kontaktné adresy

Ing. Zuzana Kozubíková, PhD., Katedra makro a mikroekonomiky, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská republika,
E-mailová adresa: zuko@frcatel.fri.uniza.sk

RNDr. Aleš Kozubík, PhD., Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská republika,
E-mailová adresa: alesko@frcatel.fri.uniza.sk



OSSConf 2012: 103–108

VOLBA OPTIMÁLNÍHO PROGRAMU

JIŘÍ RYBIČKA (CZ) A JAN PŘICHYSTAL (CZ)

Abstrakt. Volba optimálního programu pro zpracování dat je často ovlivňována subjektivními faktory souvisejícími především s předchozími zkušenostmi. Je však běžné, že tentýž problém je možné řešit více způsoby, a není nikde zajištěno, že právě volba daného programu je pro uvažované zpracování tou nejlepší. Druhou zcela přirozenou příčinou volby konkrétního programu je odpor k neznámému. Tím je často znevýhodňován svobodný software ve prospěch reklamou podporovaných placených variant. Volbu optimálního programu běžný uživatel nemůže zvládnout bez významné podpory – tou se snaží být projektovaný systém, který na základě zadaného typu zpracování dat (například typu dokumentu) a na základě osobních preferencí uživatele nabídne optimální program, v němž lze danou úlohu řešit, a to včetně návodu, jak daný program použít.

Klíčová slova. Zpracování dat, volba programu, alternativní řešení, svobodný software, systém pro volbu, požadavky uživatele.

OPTIMAL SOFTWARE SELECTION

Abstract. Optimal software selection for data processing is often influenced by subjective factors related to previous experience. It is usual the same problem can be solved different ways and nowhere is guaranteed certain software selection is the best for certain process. Second absolutely natural cause of certain software selection is resistance to unknown. This is the reason why the opensource software is often handicapped in favor to advertised paid variants. Regular user cannot manage optimal software selection without significant assistance. The designed system tries to do it based on defined type of processed data (e.g. document type) and based on user personal preference. Using it the user can solve given problem including instructions how to use given software.

Keywords. Data Processing, Program Selection, Alternative Solving, Free Software, Selection System, User Requirements.

1. Úvod

Pravděpodobně každý uživatel počítače si nejednou položil otázku: „V čem by bylo nejlepší tento dokument zpracovat?“ Nebo možná totéž v jiné variantě: „Proč tento program neumí funkci x , kterou bych právě teď tak potřeboval?“ Ale také se může v nejhorším stát, že uživatel ani k takovým otázkám nedospěje a říká si, že právě tak, jak to dělá teď, je to nejlepší.

V každém z uvedených případů se ale musíme zeptat, kdo tomu uživateli poradí ten nejlepší program, v němž půjde vše optimálním způsobem. Pokud nestačí

vlastní zkušenosti, pomohou zkušenosti kolegů, známých nebo zkušenosti vyčtené z literatury? Který z těchto zdrojů je dostatečně relevantní?

Jak je vidět, vynořuje se spousta otázek a zatím žádné odpovědi. Konkrétně ve výuce na všech stupních škol bychom mohli spatřovat nejrůznější přístupy k řešení běžných problémů. Dost často platí, že učitel učí to, co ovládá. V nepřehledném množství možností, zvláště u takových úloh, které mají masové rozšíření, jako je zpracování textů, grafických nebo tabulkových dat, si lze těžko vybírat a volba použitého programu je záležitostí obvykle zcela parciálních kritérií: tohle znám, v tomhle pracujeme na našem pracovišti, tenhle program je dostupný zdarma.

Co si můžeme představit pod optimální volbou programu? Čím se vlastně zabývá tento příspěvek?

Samotné slovo „optimální“ už v sobě skrývá jistou nejednoznačnost – pravděpodobně půjde o proces vícekriteriálního výběru (kritérií dokonce můžeme předpokládat opravdu hodně), kde výsledek je mimo jiné také hrou kompromisů. Pro účely tohoto textu však intuitivně vymezíme optimální programový systém jako ten, který zadanou úlohu vyřeší nejrychleji, nejspolehlivěji, s nejmenším úsilím uživatele a s nejmenšími nároky na technické vybavení stroje. Přitom souslovím „programový systém“ může být reprezentován jeden program, nebo i programů více, které si mezi sebou mohou předávat data v předem známých a dobře definovaných formátech.

Pro ilustraci si můžeme představit tento úkol: Zjistěte, jakou daň má zaplatit za uplynulý rok drobný investor, který prostřednictvím autorizovaného brokera obchoduje s akciemi na burze. Vstupem jsou pohyby akcií na osobním účtu investora, přičemž platí, že daň se počítá jen ze ziskových prodejů akcií držných méně než půl roku a je potřeba odečíst režijní náklady (poplatky za transakce).

2. Zpracování dat pro manažery

Manažer – student manažersko-ekonomického oboru přikládá tomuto označení až magické vlastnosti. Z nezkoušenosti z praktického života mnoho studentů nabývá dojmu, že manažer nemusí nic vědět, protože na všechno má „odborníky“, kterým (pravděpodobně telepaticky) sdělí, co by tak asi potřeboval, a tito odborníci to bez problémů všechno zvládnou, a to zcela optimálně, a když ne, tak je prostě vymění za jiné. V mnoha reakcích na výuku informatiky je tento postoj implicitně i explicitně vyjádřen.

Chceme-li tedy, aby takový student vyřešil nějaký konkrétní problém (například úlohu uvedenou výše), ale nezádáme, v jakém programu jej má zpracovat, dostává se do úzkých, i když je prokazatelné, že se všemi programy, které by mohl potřebovat, se už někde setkal. A to ještě vůbec v prvním přiblížení neuvažujeme optimalitu volby. Předmět „Zpracování dat pro manažery“ má mimo jiné za cíl poněkud rozšířit pohled na zpracování nejrůznějších dat počítačem tak,

aby výsledky mohly sloužit zejména jako podklady pro manažerské rozhodování. K tomu nezbytně problém výběru optimálního programového vybavení patří.

3. Svobodný software?

Pro mnohé uživatele je jedním ze zcela zásadních kritérií volby programového vybavení bezplatná dostupnost programu. Svobodný software bezpochyby tvoří velmi významnou součást nástrojů, které můžeme využít pro nejrůznější účely. Proč tedy ten otazník v názvu sekce?

Především je potřeba si uvědomit, nakolik je toto kritérium zásadní. Pokud vyžadujeme optimalitu volby programu pro zadaný úkol, nemůže kritérium bezplatné dostupnosti převážít kritéria technická zajišťující co nejrychlejší, nejefektivnější a nejbezpečnější vypracování požadovaných výsledků.

Z pohledu optimality je rovněž nepřijatelné stanovisko „Můj program P umí všechno!“. Programové systémy dneška jsou v mnoha případech nesmírně složité a skutečně oplývají tolika funkcemi, že lze tohoto dojmu nabýt. Dokonce to někdy i výrobce ve svých reklamních materiálech budoucím uživatelům slibuje. Z praktických zkušeností ale vyplývá, že většina uživatelů všechny funkce programů, s nimiž pracuje, nikdy neovládne, což má také svůj odraz v optimalitě volby a následného užití systému. Pokud například uživatel ve složitém programu využije 10 % jeho funkcí, mohl by tyto funkce použít v programu jiném, v němž tatáž skupina bude tvořit například 80 %. V čem bude rozdíl? Stejně funkce naprogramované jako součást většího a složitějšího systému budou pracovat pravděpodobně pomaleji, a to nejen z hlediska vnitřního naprogramování, ale také z hlediska rychlosti uživatelského ovládání, kdy výběr z více funkcí vždy bude obecně trvat déle.

4. Obecný model výběrového systému

V předchozích odstavcích jsme záměrně přinesli řadu argumentů pro nějaký spásný systém, který nějak zajistí optimální volbu toho pravého programu nebo programového systému pro řešení konkrétní uživatelské úlohy. Z těchto argumentů především vyplývají dvě zásadní věci:

1. Žádný uživatel jako jednotlivec není tak zkušený, aby pro každou úlohu zvolil optimální programový systém.
2. Do volby programu se zásadním způsobem promítají kritéria, která vedou k optimálnímu řešení.

Když jeden uživatel nemá dost zkušeností, může více uživatelů dohromady tento problém určitě vyřešit. Čím více zkušeností od většího počtu uživatelů, tím více informací pro optimální volbu dostaneme.

Stanovení kritérií volby programu by se mělo odvíjet především od charakteru řešené úlohy, nikoliv od osobních preferencí uživatele (osobní preference však zároveň nemůžeme zcela zavrhnout).

Předpokládejme určité vymezení optimálního výběru programu na oblast zpracování textů počítačem. Základní úlohou zde může být určitý typ dokumentu, výsledkem optimální volby pak návrh programu, který všechny požadované vlastnosti dokumentu co nejlépe dokáže realizovat.

Model optimálního výběru je znázorněn na obr. 1. V horní (vstupní) části je znázorněna dvojice zcela zásadních vstupních množin: D_i – množina parametrů (vlastností) i -tého typu dokumentu, E_d – množina definovaných vlastností tohoto typu dokumentu, která může být uživatelem U_i modifikována podle konkrétních potřeb na množinu E_{di} . Dále si uživatel může nadefinovat množinu osobních preferencí E_{ui} , čímž vznikne (sjednocením) úplná množina požadavků definujících úlohu.

Nyní přichází do hry druhá zásadní vstupní množina P týkající se vlastností programů. Pro možnost výběru kombinace programů, které si mezi sebou předávají data v souborových formátech F jsou stanoveny relace M_v a M_w definující schopnost určitý formát v daném programu číst, nebo formát v daném programu vypisovat.

V prvotním srovnání požadavků úlohy a dostupných vlastností programů může vzniknout několik výsledků v množině Ψ :

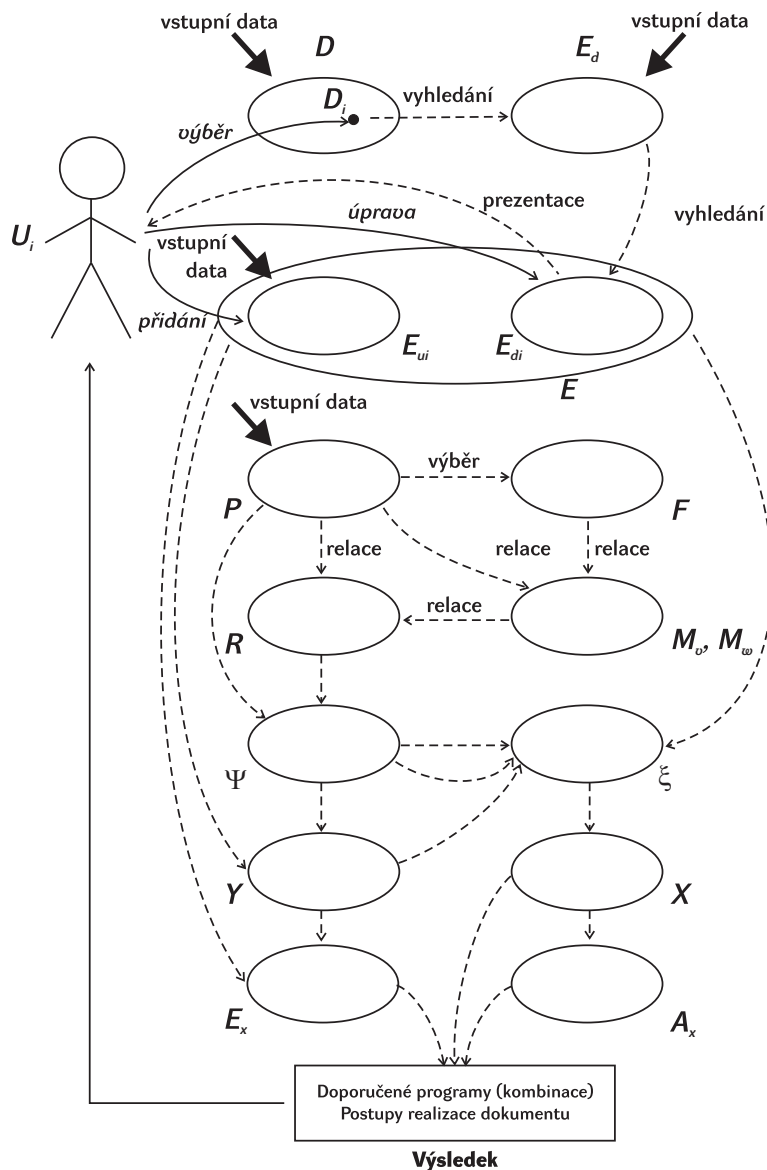
- Vybrán právě jeden program – nejjednodušší řešení.
- Vybráno více alternativních programů v množině ξ – uživatel může (nedeterministicky) vybrat libovolný z nabízených programů
- Nevybrán ani jeden program – fáze popsaná množinou Y , zde je vygenerována množina E_x těch požadavků, které nebyly splněny, aby uživatel mohl posoudit, nakolik jsou tyto požadavky závažné a mohl je pro opakovaný výběr modifikovat.

Úspěšný výběr vede k výslednému programu X , k němuž se přidají původní uživatelské požadavky tvořící výsledek činnosti celého systému. Tyto požadavky jsou společně s navázanými vlastnostmi vybraného programu doplněny o tzv. anotace – nejcennější část výsledků, které ke každé vlastnosti ve třech volitelných úrovních podrobnosti popisují, jak se daná funkce ve vybraném programu použije. Souhrn anotací tedy tvoří konkrétní návod, jak daný dokument ve vybraném programu realizovat.

Kompletně je model popsán a odvozen v [1]. Jádrem je odvozeno v článku [3].

5. Realizace ve webovém prostředí

Pro optimální přístup uživatelů k systému bylo započato s jeho webovou realizací. Uživatele předpokládáme ve dvou skupinách – čtenáři a přispěvatel. Vkládání



Obrázek 1. Schematické znázornění modelu výběru optimálního programu – podle [1, s. 17]

dat do systému by mělo být motivováno především potřebami konkrétních dokumentů a požadovanými vlastnostmi konkrétních programů. Příspěvatelé tedy budou mít možnost navrhovat vše, co zatím v systému není. Pro udržitelnost konzistence budou návrhy vkládány k definitivnímu využití až po kontrole moderátory, což jsou nejvíce privilegovaní uživatelé zaručující funkčnost systému.

Pro efektivní zadání požadavků se předpokládá jednak předdefinovaná skupina některých typických úloh (v první fázi se předpokládá oblast zpracování textů a předdefinované typy dokumentů) a jednak možnost ukládat osobní nastavení a preference.

6. Závěr

Myšlenka optimálního výběru programového systému pro řešení uživatelských úloh za podpory systému shromažďujícího zkušenosti a znalosti nemalé skupiny uživatelů by měla přispět k obohacení všech zúčastněných. Mohla by být cestou k propagaci dobrých programů a jejich vlastností zpřístupňujících nenáročnou formou využití i těm uživatelům, kteří nemají příliš mnoho času na detailní ovládnutí jim dosud neznámých funkcí. Zároveň lze očekávat i možnost výukového využití, kdy získané informace mohou sloužit jako podklad seznámení s programy a jejich funkcemi, ale také jako cvičné pole pro dobrou formulaci požadavků a ujasnění primární úlohy. Pokud bude alespoň částečně některý z uvedených cílů naplněn, bude možné konstatovat, že celé úsilí splnilo svůj účel.

Poděkování. Tento příspěvek vznikl v rámci řešení grantu FRVŠ č. 2406/2012 „Tvorba multimediálních studijních materiálů pro předmět Zpracování dat pro manažery“.

Reference

- [1] RYBIČKA, J., ČAČKOVÁ, P., PŘICHYSTAL, J.: *Principy efektivního zpracování dokumentů a využití v marketingové komunikaci*, Folia Universitatis Agriculturae et Silviculturae Mendelianae Brunensis IV, 2011, 9, ISSN 1803-2109, ISBN 978-80-7375-577-5.
- [2] RYBIČKA, J., PŘICHYSTAL, J., TALANDOVÁ, P.: *Uživatелеm definované požadavky a parametry systému*, <http://akela.mendelu.cz/~rybicka/prez/zpract/pozadav.pdf>.
- [3] RYBIČKA, J., TALANDOVÁ, P., PŘICHYSTAL, J.: *Systém optimalizující zpracování požadovaného dokumentu*, <https://akela.mendelu.cz/~rybicka/prez/zpract/model.pdf>.

Kontaktní adresy

doc. Ing. Jiří Rybička, Dr., Ústav informatiky, Provozně ekonomická fakulta, Mendelova univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika,
E-mailová adresa: rybicka@mendelu.cz, <http://akela.mendelu.cz/~rybicka>

Ing. Jan Přichystal, Ph.D., Ústav informatiky, Provozně ekonomická fakulta, Mendelova univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika,
E-mailová adresa: jprich@pef.mendelu.cz



OSSConf 2012: 109–114

ANIMUJME VÝSTUPY Z PROGRAMU GEOGEBRA

PAVEL STRÍŽ (CZ)

Abstrakt. V tomto příspěvku autor představuje exportní možnosti programu GeoGebra směřující k animacím. Vedle základních možností (dynamická webová stránka jako HTML, animovaný soubor GIF) jsou představeny i ty, které spadají pod $\text{T}_{\text{E}}\text{X}$ ový svět: export do třídy `article` nebo `beamer`, především však za asistence balíčku `animate`. V závěru článku autor upozorňuje i na možnosti přes balíčky `ocg` a `ocgtools`.

Klíčová slova. GeoGebra, $\text{T}_{\text{E}}\text{X}$, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, `beamer`, `animate`, `ocg`, `ocgtools`.

LET'S ANIMATE DRAWINGS FROM GEOGEBRA

Abstract. In this paper, the author introduces options how to animate drawings created in GeoGebra. Among basic options (dynamic worksheet in HTML, animated GIF file) he also introduces typical approach fitting in the $\text{T}_{\text{E}}\text{X}$ world: exports using the `article` and `beamer` classes with a special attention to options (timing, layering, grouping) via the `animate` package. In the last section, the paper also mentions a pseudo-animation opportunity via the `ocg` and `ocgtools` packages.

Keywords. GeoGebra, $\text{T}_{\text{E}}\text{X}$, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, `beamer`, `animate`, `ocg`, `ocgtools`.

1. Úvod

Uživatelé programu GeoGebra již tvoří nemalou komunitu (GeoGebra Forum, GeoGebraWiki, GeoGebraTube či GeoGebraChannel na YouTube), stručně řečeno je to program, který chyběl. Rozvoj je dynamický, jak dokládají podrobné tutoriály [1], včetně modulů pro Moodle, Mediawiki, WordPress, Joomla! a Drupal, jdoucí až do užití příkazového řádku, JavaScriptu či GeoGebraScriptu.

2. Exportní možnosti

V následujících odstavcích se podíváme jak na standardní exportní možnosti, tak ty, které směřují k animacím.

2.1. Export kresby

Standardní formát programu GeoGebra (v době psaní článku je poslední verze 4.0.33.0) je lidsky nečitelný (Soubor→Uložit; GGB), proto se obvykle snažíme získat použitelnější formu. Možnostmi exportu (Soubor→Export→Grafický náhled jako obrázek (png, eps)...) jsou formáty PNG, EMF, SVG, EPS a PDF.

Formát PNG je rastr, pro potřeby typografie formát nevýhodný. Pokud můžeme získat vektorovou podobu, uděláme to, té dáváme vždy přednost.

Formát EMF je lidsky nečitelný a vyžaduje další konverzi a zpracování.

Formát SVG je sice lidsky čitelný, ale u základní kresby s několika objekty získáváme soubor o několika desítkách KB. Ruční zásahy jsou nemožné, je potřeba automat či automatizační nástroj.

Formát EPS je těžko editovatelný a vyžaduje téměř vždy následnou konverzi do PDF. Obvyklé problémy jsou nutnost užít BoundingBox, hlídat si barevný prostor, velikost papíru a zajistit vložení užítých písem.

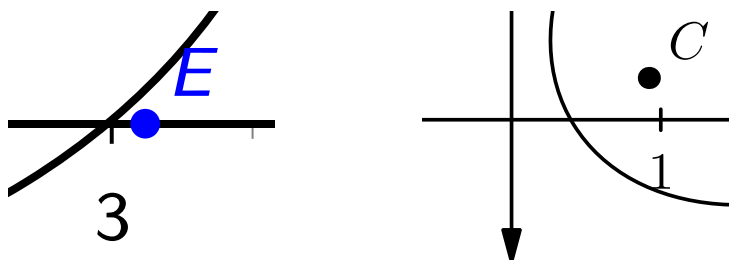
Formát PDF se z tohoto pohledu zdá nejpoužitelnější. Program GeoGebra používá FreeEP Graphics2D Driver [4], jak prozradily vlastnosti vzniklého PDF souboru. Editace je možná, např. programem PDFCreator, ale namáhavá. Typografům minimálně bude vadit spojovník místo znaménka mínus na osách.

Podívejme se proto na další exportní možnosti.

2.2. Další exportní možnosti

Pod Soubor→Export z menu GeoGebry najdeme tři možnosti exportu kresby do $\text{T}_\text{E}\text{X}$ ového světa: projekt PSTricks [5], balíček TikZ [6] a program Asymptote [7]. Na testování jsem použil jednoduchou kresbu (osy, kružnice zadaná dvěma body, elipsa zadaná třemi body).

Čtenář může ověřit zvednutím počtu desetinných míst, či jinými známými fin-tami, ale při základním nastavení a exportu do PSTricks nám body neprotínají křivky, detail na obr. 1 vlevo.



Neprotnutí všech bodů ležících na křivkách (vlevo; PSTricks),
uřezání hlaviček šipek a bez barvy (vpravo; Asymptote)

Obrázek 1. Závady alias exportní nedostatky

Při exportu do Asymptote ztrácíme barvy a díky nedokonalému ořezu ztrácíme špičky šipek na osách, detail na obr. 1 vpravo.

Nejlépe je na tom export do PGF/TikZ. Ovšem absence stylů znepříjemní jakoukoliv další manipulaci s kresbou. Např. `color=qqqqff` tam potkáme na každém druhém řádku zdrojového kódu. V této chvíli je to však nejmenší nepříjemnost. V tomto článku zůstaneme už jen u PGF/TikZ.

3. Animujme standardně

Možnosti máme tři: dynamický sešit HTML, animovaný GIF a přístup přes $\text{T}_\text{E}\text{X}$.

Aby se naše kresba stala animací, musíme použít posuvník. Je to nastavení proměnné v určitém rozsahu, kterou můžeme použít ve výpočtu souřadnic. Při otevření Nápovědy (ikonka vpravo od příkazového řádku ve spodní části) nalezneme funkci GeoGebra→AnimovanýGraf, ale je to de facto též posuvník.

Posuvník nalezneme pod ikonkami, případně z menu pod Nástroje→Aktivní prvky→Posuvník. Detaily viz <http://wiki.geogebra.org/cs/Animace> [2].

3.1. Dynamický sešit html

Máme-li v kresbě posuvník (můžeme jich mít na jednom sešitě víc), naše webová stránka bude interaktivní. Užíváme Soubor→Export→Dynamický pracovní list jako webová stránka (html)... Většina z nás přepne na Exportovat jako webovou stránku, poté Export. Vidíme však, že můžeme přímo exportovat na GeoGebraTube. Podobně lze užít Soubor→Sdílet.

Ukázek je na internetu nespočet. Nahlédněte například na http://www.fd.cvut.cz/department/k611/PEDAGOG/files/GM_Ulohy_index.html, popřípadě si projděte http://www.gymkrom.cz/web/ict/materialy/Dvacitka_GGB.pdf.

Kdo dává přednost humoru, nahlédněte například na „GeoGebra Animation – Fish“: <http://www.youtube.com/watch?v=SDqS6t87Sx8>.

3.2. Animovaný gif soubor

Méně používaná možnost je využít GIF soubor. Animuje se dle jednoho posuvníku a výstup je rastr. Bez možnosti interakce s uživatelem. Do $\text{T}_{\text{E}}\text{X}$ ového dokumentu lze načíst, pomocí balíčku `movie15`, případně přes jeho nástupce balíček `media9`. Balíček sám má mnoho výhod: lze načíst 3D modely (PRC, U3D), zvukové a filmové sekvence, otevřít si kanál rádia či video z YouTube.

Bohužel zobrazení je závislé na přehrávači a i poté se nám to v PDF může zobrazit v externím okně. Lepší je rozložit GIF na sérii obrázků a ty poté vrstvit.

Další nevýhodou je obří velikost takového souboru. Při jednoduché animaci jsem se dostal na téměř 1 MB. Soubor větší než průměrný vysázený dokument.

4. Animujeme přes $\text{T}_{\text{E}}\text{X}$ ový svět

Výhody jsou možnost nastavení třídy `article` (dokument) či `beamer` (prezentace); možnost zvolit si Plain $\text{T}_{\text{E}}\text{X}$ (jádro $\text{T}_{\text{E}}\text{X}$ u), $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ (nejpoužívanější formát, bude užít v tomto článku) či `CONTEXt`u (nejnovější formát). To vše díky `TikZ`u, protože výstup lze načíst do všech tří formátů. Navíc můžeme výpočty s vykreslením grafu funkcí předat programu `GnuPlot`.

4.1. Třída `beamer`

Základní možnost je nastavit ve formátu při exportu: `LaTeX (beamer class)`. Po Uložit jako získáváme $\text{T}_{\text{E}}\text{X}$ ový soubor, který stačí za $\text{T}_{\text{E}}\text{X}$ ovat. Získáváme PDF

o několika stranách/snímcích. Soubor můžeme používat přímo, případně si jej načíst zvlášť. Řadě uživatelů vadí navigační značky, ty lze vypnout příkazem:

```
\setbeamertemplate{navigation symbols}{}
```

Nevýhodou zůstává, že je těžká manipulace s jednotlivými snímky (příkaz `\onslide`), jako i se styly, které se musí nastavit až po exportu.

4.2. Balíček `animate` poprvé

Další možnost je si vyříznout jen výsek (parametry `viewport` a `clip`) z vygenerovaného souboru a animovat.

```
\begin{animateinline}[autoplay, poster=last, controls, loop]{1}
\multiframe{8}{istrana=1+1}{%
\includegraphics[page=\istrana, viewport=0.782cm 1.876cm
12.525cm 8.599cm, clip, scale=0.6]{sazbapgf.pdf}}%
\end{animateinline}
```

V tomto případě animujeme 8 snímků, levý dolní roh je 0,782 cm a 1,876 cm, pravý horní roh je určen souřadnicemi 12,525 cm a 8,599 cm. Tyto rozměry si lze snadno odměřit a získat programem `PS_View`.

Výhodou je, že máme kontrolu nad animací, která je ve vektoru. Nevýhoda je, že nemůžeme zasahovat do jednotlivých objektů. Také každý snímek je nezávislý, tedy objekty se opakují a pokaždé se sází znovu. To má přímý vliv na velikost uložených snímků.

4.3. Balíček `animate` podruhé

Ideální řešení je si nechat kresbu vykreslit do třídy `article`, kde si vrstvy dodefinujeme. Poté můžeme nastavit i takové perličky, že první snímek bude kreslicí plátno, osa x , poté se přidá osa y , pak body, kružnice, další body a elipsa. Diskuzi na toto téma lze nalézt na vlákně fóra <http://www.geogebra.org/forum/viewtopic.php?f=20&t=25986>.

Původní soubor obohatíme o balíček `animate`, kresbu obalíme do prostředí `animateinline`, na všechny snímky nastavíme jednotný `\clip`. Aby vše bylo dokonalé (po vykreslení kružnice a elipsy se popisky přesunou nad ně kvůli čitelnosti), nastavíme si automatický běh, vrstvy a skupiny, viz obr. 2 na straně 114.

```
\documentclass[10pt]{article}
\usepackage[utf8]{inputenc}
\usepackage{tikz}
\usetikzlibrary{arrows}
\pagestyle{empty}
\usepackage{animate}
\begin{document}
\definecolor{qqqqff}{rgb}{0,0,1}
\begin{animateinline}[
begin={\begin{tikzpicture}[line cap=round, line join=round,
```



```

>=triangle 45, x=1.0cm, y=1.0cm]
\clip(-2.63,-0.95) rectangle (8.7,5.47);}, % x: jednotné okno
end={\end{tikzpicture}}},
controls, autoplay, loop, timeline=casovani-utf8.tex, poster=none]{0.5}
\mbox{} % 0: prázdňá plocha
\newframe % 1: kružnice
\draw(0.62,2.12) circle (0.96cm);
\newframe % 2: elipsa
\draw [rotate around={31.67:(1.99,0.94)}] (1.99,0.94)
ellipse (1.85cm and 1.36cm);
\newframe\scriptsize % 3: body ke kružnici
\fill [color=qqqqff] (0.62,2.12) circle (1.5pt);
\draw[color=qqqqff] (0.75,2.35) node {$A$};
\fill [color=qqqqff] (1,3) circle (1.5pt);
\draw[color=qqqqff] (1.16,3.23) node {$B$};
\newframe\scriptsize % 4: body k elipse
\fill [color=qqqqff] (0.92,0.28) circle (1.5pt);
\draw[color=qqqqff] (1.09,0.49) node {$C$};
\fill [color=qqqqff] (3.06,1.6) circle (1.5pt);
\draw[color=qqqqff] (3.22,1.81) node {$D$};
\fill [color=qqqqff] (3,0) circle (1.5pt);
\draw[color=qqqqff] (3.16,0.23) node {$E$};
\newframe % 5: osa iks
\draw[->,color=black](-2.63,0) -- (8.7,0);
\foreach \x in {-2,-1,1,2,3,4,5,6,7,8}
\draw[shift={(\x,0)},color=black] (0pt,2pt) -- (0pt,-2pt)
node[below] {\footnotesize $\x$};
\newframe % 6: osa ypsilon
\draw[->,color=black](0,-0.95) -- (0,5.47);
\foreach \y in {,1,2,3,4,5}
\draw[shift={(0,\y)},color=black] (2pt,0pt) -- (-2pt,0pt)
node[left] {\footnotesize $\y$};
\draw[color=black] (0pt,-10pt) node[right] {\footnotesize $0$};
\end{animateinline}
\end{document}

```

Pomocný soubor `casovani-utf8.tex` s časováním a vrstvením vypadá takto:

```

::0 % prázdňá kreslicí plocha
::5x0 % osa x na všechny snímky
::6x0 % osa y na všechny snímky
::3 % body A a B, jedenkrát
::1x0;3x0 % kružnice, poté znovu body A a B
::4 % body C, D a E, jedenkrát
::2x0;4x0 % elipsa, poté znovu body C, D a E
% čárka znamená: v jedné vrstvě
% středník znamená: oddělení vrstev

```

4.4. Balíček ocg

Poslední možností, kterou v příspěvku zmíníme, je užít balíček `ocg` nebo `ocgtools`. Jedná se o zapínání a vypínání vrstev na úrovni PDF souboru. S vrstvami se mohli uživatelé GeoGebry také setkat, přístup jim nebude cizí (Vlastnosti→Pro pokročilé→Vrstva). Může se hodit víc než vlastní animace, zvláště u prezentací.

5. Závěr

Na co se uživatelé GeoGebry mohou těšit? Už nyní existuje GeoGebra 5.0 Demo, viz <http://www.geogebra.org/forum/viewtopic.php?f=52&t=19846> [3], která do budoucna slibuje tah ke 3D. Pak export do Asymptote bude mít velký význam.

Můžeme se těšit na OSSConf2012, zářijovou akci v Polsku (viz <http://www.geogebra.org/cms/pl/events>), ale též na příští ročník, tj. OSSConf2013, kde se očekává nová vlna přednášek z oblasti GeoGebry.

Obrázek 2. Kontrolovaná animace v $\text{T}_\text{E}\text{X}$ ovém světě

Reference

- [1] GeoGebra tutoriály: http://wiki.geogebra.org/en/Tutorial%3AMain_Page.
- [2] Animace přímo v programu GeoGebra: <http://wiki.geogebra.org/cs/Animace>.
- [3] GeoGebra 5.0 Demo: <http://www.geogebra.org/forum/viewtopic.php?f=52&t=19846>.
- [4] FreeEP Graphics2D Driver: <http://java.freehep.org/>.
- [5] Projekt PSTricks: <http://tug.org/PSTricks/main.cgi/>.
- [6] Balíček PGF/TikZ: <http://www.texample.net/tikz/examples/>.
- [7] Program Asymptote: <http://asymptote.sourceforge.net/>.

Kontaktní adresa

Ing. Pavel Stríž, Ph.D., Fakulta managementu a ekonomiky, Univerzita Tomáše Bati ve Zlíně,
T. G. Masaryka 5555, 760 01 Zlín, Česká republika,
E-mailová adresa: striz@fame.utb.cz



OSSConf 2012: 115–120

SWEAVE: SVÁZÁNÍ \LaTeX U S JAZYKEM R

PAVEL STRÍŽ (CZ)

Abstrakt. Autor si váží toho, že zrovna on vám můžeme představit program *Sweave* {čtete ten či ta svív}, který elegantním způsobem, s nastavením parametrů vzoru jazyka *noweb*, propojil \LaTeX s výpočetním prostředím R. Jako je Open Source Software \TeX přijímán komunitou typografů a jejich přátel, tak je R přijímán řadou matematiků a statistiků ve světě výpočetním, viz <http://www.r-project.org/>.

Sweave umožňuje a umí to, že výstupy R z RNW souboru dokáže vpasovat do \LaTeX dokumentu. Tím se dokument stává dynamickým a grafy a statistické analýzy jsou aktualizované z pohledu vstupních dat. Program si představíme, nainstalujeme a nastavíme pod Microsoft Windows, v Linuxu je to obdobné. Naučíme se s programem *Sweave* pracovat a ukážeme si typické příklady. Kvůli rozsahu je plná verze článku na webu OSSConf2012.

Klíčová slova. Výpočty, *Sweave*, R, \TeX , \LaTeX .

SWEAVE: MIXING \LaTeX WITH THE R LANGUAGE

Abstract. The article introduces professor Leisch's package *Sweave* from the R language and computing environment. The *Sweave* tool puts together \LaTeX and the R language in a simple way. The authors present their experiences with the installation and functionality testing together with setting the new user-defined batch files under Microsoft Windows.

The author also presents examples of chunks for basic R computing with variable handling and printing. Also table creation and graph production in R which includes graphic output into the \LaTeX file is presented. Enjoy *Sweaveing*! The full version of the article is available at the official OSSConf2012 website.

Keywords. Computing, *Sweave*, R, \TeX , \LaTeX .

Úvod

Otcem i matkou *Sweave* je profesor Friedrich Leisch [4]. Domovská stránka programu je <http://www.stat.uni-muenchen.de/~leisch/Sweave/>, kde je manuál, otázky a odpovědi (FAQ), články i několik ukázek. *Sweave* je součástí základního balíčku *utils* jazyka R.

R je oblíbený mezi statistiky a matematiky. Je to Open Source Software, programovatelný a s možností grid grafiky. R je založený na jazyku *S-PLUS*, blíže např. [3]. Závěrečná kapitola této knihy si všímá rozdílu mezi R a *S-PLUS*.

Pan Hlávka z MFF UK v Praze dodává [2], že oba jazyky jsou téměř shodné, ale nejsou kompatibilní, mohou se lišit jména některých příkazů a některé konstrukce mohou fungovat trochu jinak.

Instalace

Začněmež! Sedíme na počítači s Microsoft Windows XP, máme nainstalován \TeX Live 2011. Pokud sedíte na jiném operačním systému, máte výhodu, protože poinstalační Sweave zlobí právě pod Microsoft Windows, jak zmiňuje i referenční zdroj jazyka R [6].

Z oficiálních stránek <http://www.r-project.org/> si stáhneme binárky pro Microsoft Windows. Potřebujeme administrátorská práva. Nainstalujeme si.

Věc, kterou chceme nastavit je cesta (PATH) ke spustitelným souborům. Lze využít Start→Ovládací panely→Systém, karta Upřesnit, Proměnné prostředí, Systémové proměnné a PATH upravujeme. Děláme to z toho důvodu, abychom mohli R spouštět bez vypisování adresáře. Při například verzi 2.7.1 (září 2008) to znamená přidat za současné nastavení (středník na začátku, bez tečky na konci): ;C:\Program Files\R\R-2.7.1\bin\. Dále už jen OK. OK. OK.

Přes ikonku „R 2.7.1“ spouštíme RGui s oknem R Console. Kdo je v R nový, může vyzkoušet příkazy potvrzené klávesou Enter: `demo()`, `demo(graphics)`, ...

K hlubšímu studiu jazyka R doporučujeme:

http://zoonek2.free.fr/UNIX/48_R/all.html	{En, Fr}
http://bm2.genes.nig.ac.jp/RGM2/	{Galerie}
http://addictedtor.free.fr/graphiques/allgraph.php	{Galerie}

Poněvadž již víme, že Sweave je součástí základní instalace, tak vyzkoušejte: `help(Sweave)`, či `example(Sweave)`.

Poinstalační zásahy

Vše by mělo fungovat, s tím rozdílem, že bychom si přáli něco věcí vylepšit.

V dokumentaci se dočteme, že na tabulky lze užívat R balíček `xtable`. Nainstalujeme si jej. V grafickém R klikáme Packages→Set CRAN mirror... Vybereme si jeden z nich. OK. Pokračujeme Packages→Install packages(s)... , zvolíme `xtable`. OK. Kdo nerad kliká, necht' zapíše na jednom řádku přímo:

```
install.packages("xtable",
  contriburl=contrib.url("http://cran.r-project.org"))
```

Jak píše referenční příručka, bez chybějícího `\usepackage{Sweave}` v RNW se snaží Sweave přidat své nastavení, které pod Microsoft Windows zlobí (mezery, názvy delší než 8 znaků), resp. zlobí v \TeX u u aktivních znaků. Proto v RNW na balíček Sweave nezapomeneme. V opačném případě bychom nastavili cestu k `Sweave.sty`, nebo jej nakopírovali do lokálního adresáře.

Také si přejeme Sweave spouštět mimo R. Přes příkaz `sh sweave.sh` to pod Windows nepůjde samo o sobě.

Zkušený linuxový uživatel možná uvítá komplexní řešení¹ či jiný styl psaní o Sweave². Komentáře a zkušenosti se Sweave pod Linuxem a se stylem APA jsou v článku [7]. Jednoduše lze přes balíček R2HTML užít Sweave do HTML³.

Sweave dělá to, že vezme náš soubor (RNW) a generuje T_EXový soubor (TEX), a to tak, že vnitřní pasáže (chunk, tj. cvalíky) spouští v R a vrací nám výpočty, zdrojové kódy či grafy (PDF a zároveň EPS; T_EX načte jeden z nich dle přepínače). Běžné T_EX zdroje během generování nemění, jen slepě kopíruje.

První úkol zařídíme takto. Připravíme si soubor `mySweave.r`:

```
library(utils)
args <- commandArgs(TRUE)
Sweave(args)
# Stangle(args)
```

Tento soubor si načte R balíček `utils`. Vstupní parametry uloží do proměnné `args` a to vše prožene Sweave. Vstupní argument bude název zpracovávaného RNW souboru. Pokud smažeme `#` z posledního řádku, tak ovladač `Stangle` vám cvalíky vyextrahuje do speciálního R souboru.

Vedle toho si připravíme soubor `mySweave.bat`:

```
Rscript mySweave.r %1
R CMD pdflatex.exe %1.tex
R CMD pdflatex.exe %1.tex
```

Tento dávkový soubor nám `mySweave.r` prožene R. Zpracuje RNW na TEX. Pak by stačilo spustit běžné T_EX prostředí. Problém je, že cestu k `Sweave.sty` zná jen R. Proto doplníme R `CMD`, což nám zajistí přístup do standardního adresáře `C:/Program Files/R/R-2.7.1/share/texmf/`.

V příkazové řádce spouštíme, nebo v našem oblíbeném editoru nastavíme, dávkové spuštění `mySweave myfile`, kde `myfile` je myšlen `myfile.Rnw`.

Mějte tento článek, kde je řídicí soubor, který T_EXujete, a článkové soubory v podadresáři, tak `mySweave.bat` může vypadat takto (konkrétně podadresář Pavla Stríže `/PSsweave/` s článkem `PSsweave.Rnw`; argument `%1` je soubor řídicí):

```
cd PSsweave
Rscript mySweave.r PSsweave.Rnw
cd ..
R CMD lualatex.exe %1.tex
bibtex %1
R CMD lualatex.exe %1.tex
R CMD lualatex.exe %1.tex
```

¹<http://sites.google.com/site/gregorgorjanc/software/Sweave.sh>

²stat.epfl.ch/webdav/site/stat/shared/Regression/EPFL-Sweave-powerdot.pdf

³<http://biostat.mc.vanderbilt.edu/twiki/pub/Main/SweaveConvert/htmlWeave.pdf>

Na tomto místě dodáváme, že EXE je nutností, jinak by R hledal vnitřní balíčky, ne spustitelný soubor. Také za pozornost stojí, že `bibtex` stojí osamocen bez R CMD. Je to z toho důvodu, že nepotřebuje knihovny a balíčky od R.

Poslední věc, která nám může vadit je standardní nastavení pro výpis zdrojových kódů. Sweave, konkrétně jsou to poslední čtyři řádky v souboru `Sweave.sty`, pracuje s \TeX balíčkem `fancyvrb`, který užívá v základním nastavení.

Možnosti `fancyvrb` jsou poměrně bohaté a lze si upravit. Autor dává na zdrojové kódy přednost balíčku `listings`. Jeho zanesení a užití přes Sweave je běžná \TeX ová operativa. Záměrně zůstaneme u Sweave nastavení a prostředí `verbatim`.

Cvalíkujeme: užití chunks (cvalíků kódu)

Nahlédněte prosím na webové stránky OSSConf2012 pro plnou verzi článku.

Souhrnná ukázka

Tato ukázka je jedna z našich oblíbených R kódů pro studenty. Mají se najít dvě střední hodnoty vrcholů smíšených gaussových křivek nad zadanými daty.

Vstupní data jsou 1, 2, 2, 2, 1, 3, 2, 2, 2, 7, 8, 9, 9, 9, 9, 9, 10, 9, 9, 14, 8, 8, 9, 9, 9, 10, 11, 0, 2, 2, 4 a 5.

Na tento problém si v R doinstalujeme R balíček `mclust`, více viz [1], případně `help(Mclust)` zapsané v R po instalaci `mclust`. Je to balíček, který není licencován mimo studijní účely. Užijme k instalaci například kanadského zrcadla CRANu.

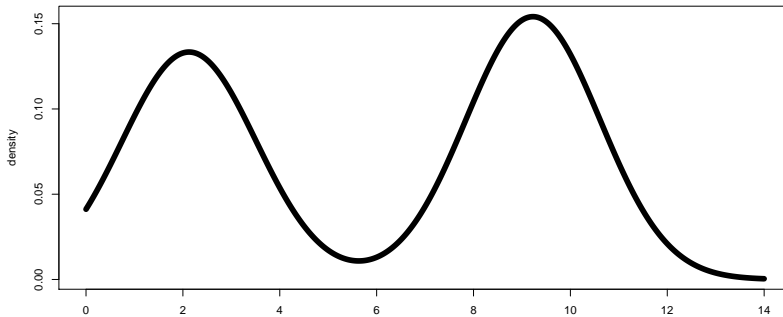
```
install.packages("mclust",
  contriburl=contrib.url("http://cran.stat.sfu.ca/"))
```

Řešení dle Bayesova informačního kritéria BIC (Bayes Information Criterion). První vrchol má souřadnici složky x 2.1273, druhý pak 9.2273.

Výpočet a výpis předchozího odstavce byl v `.Rnw` zapsán takto. Příkaz `rm()` vyčistí předchozí proměnné. Výstupní obrázek a jeho zdroj následují.

```
<<echo=FALSE,results=HIDE>>=
  rm(list=ls(all=TRUE))
  library(mclust)
  y<-c(1,2,2,2,1,3,2,2,2,7,8,9,9,9,10,9,9,14,
      8,8,9,9,9,10,11,0,2,2,4,5)
  yModel <- mclustModel(y, mclustBIC(y, G=2))
  yModel$parameters$mean
  @
```

Řešení dle BIC. První vrchol má souřadnici složky x `\Sexpr{round(yModel$parameters$mean[1], digits=4)}`, druhý pak `\Sexpr{round(yModel$parameters$mean[2], digits=4)}`.



Obrázek 1. Odhad hustoty součtem dvou gaussových křivek

```

\begin{figure}[hbt]
  \centering
  <<echo=FALSE, fig=TRUE, width=12>>=
    mclust1Dplot(y, parameters = yModel$parameters,
      z = yModel$z, what="density", identify = FALSE,
      ngrid = length(y)*15, lwd=8)
  @
  \caption{Odhad hustoty součtem dvou gaussových křivek}
  \label{hustota}
\end{figure}

```

Závěr

Myslíte-li to s programem Sweave vážně, tak si dejte pozor na to, že editujete soubor RNW a TEX je v tomto případě generován. Budete-li omylem editovat TEX soubor, tak si při stejného názvu RNW spuštěním Sweave TEX soubor přepíšete.

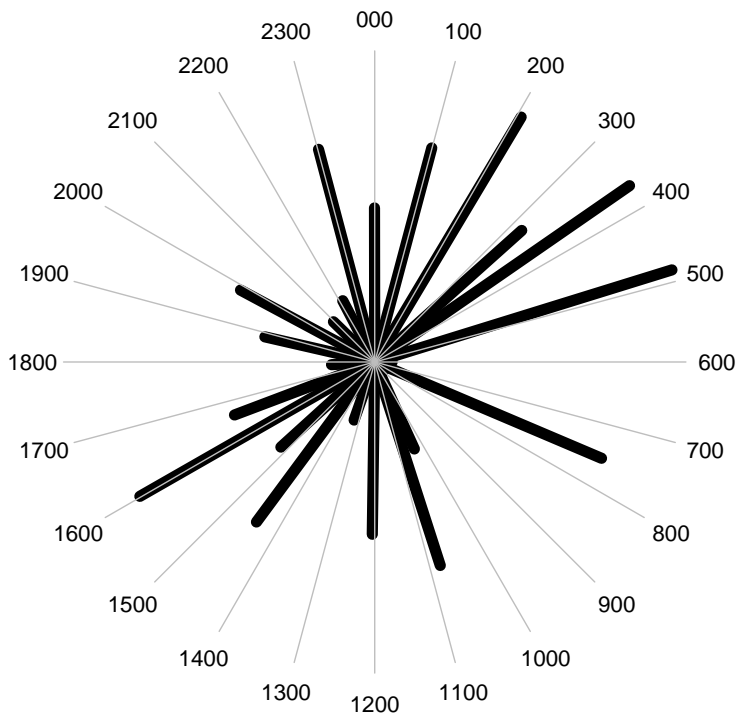
Profesor Leisch vytvořit také pomůcku vignettes [5], která rozšiřuje některé možnosti přípravy a prezentování dokumentace programů R a BioConductor.

V závěru Sweave manuálu profesor píše, že pracuje od nuly na balíčku S4weave, který by měl Sweave nahradit a překonat v nabízených možnostech. Máme se na co těšit, i když nekonečné potřeby smrtelníků bohatě pokrývá potenciál Sweave.

Zbytek zkoumání necháváme na vás. S vámi i se všemi cvalíky se loučí autor!

Reference

- [1] Fraley C., Raftery A. E.: *Mclust version 3 for R: Normal mixture modeling and model-based clustering*, Technical Report 504, University of Washington, Department of Statistics, September 2006.
- [2] Hlávka Z.: *Emailová odpověď velkého pražského eRkaře autorovi na vznesený vznešený dotaz*, 28. dubna 2008.



Obrázek 2. Grafický potenciál jazyka R je pro techniky neskutečný

- [3] Krause A., Olson M.: *The Basic of S-PLUS*, First edition, ISBN 0-387-26109-5.
- [4] Leisch F.: *Sweave*: Dynamic generation of statistical reports using literate data analysis, In Wolfgang Härdle and Bernd Rönz, editors, *Compstat 2002—Proceedings in Computational Statistics*, 575–580, Physica Verlag, Heidelberg, 2002, ISBN 3-7908-1517-9.
- [5] Leisch F.: *Sweave*, part II: Package vignettes, *R News*, 3(2): 21–24, October 2003.
- [6] The R Development Core Team: *R: A Language and Environment for Statistical Computing*, Reference Index. *Sweave*: Automatic Generation of Reports. The *utils* package, Version 2.7.1 (2008-06-23), ISBN 3-900051-07-0.
- [7] Zahn I.: *Learning To Sweave in APA Style*, The *PracTeX* Journal, (1), 2008, <http://tug.org/pracjourn/2008-1/zahn>.

Kontaktní adresa

Ing. Pavel Stríž, Ph.D., Fakulta managementu a ekonomiky, Univerzita Tomáše Bati ve Zlíně,
T. G. Masaryka 5555, 760 01 Zlín, Česká republika,
E-mailová adresa: striz@fame.utb.cz

ABSTRAKTY NERECENZOVANÝCH UKÁŽOK A PREZENTÁCIÍ

Podujatie *Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach* nie je len konferencia v klasickom, akademickom ponímaní, ale aj stretnutie ľudí, ktorých spája záujem o otvorený softvér. Popri recenzovaných príspevkoch bolo na podujatí prezentovaných mnoho ďalších nerecenzovaných príspevkov a ukážok využitia otvoreného softvéru a otvorených informačných technológií. Stručné abstrakty týchto ukážok a prezentácií sú uvedené v závere tejto časti.

Súčasťou konferencie sú rôzne sprievodné akcie: „Otvorená hardvérová a softvérová poradňa“ (skúsenejší používatelia poradia a pomôžu vyličiť, prípadne nainštalovať slobodný softvér), „Bezpečnostný audit“ (ukážky toho, ako sa presvedčiť, či máme systém a hardvér dostatočne zabezpečený a čo sa môže stať, ak nemáme), galéria obrazov „Matematika a umenie“, „Moderovaná diskusia o študijných programoch SŠ a VŠ v oblasti IT“, ale aj prednášky „OSS vo výučbe matematiky na Fakulte riadenia a informatiky ŽU“, „Výučba počítačovej grafiky na FRI ŽU“ a „Virtualizácia v počítačovej učebni v škole“.

Ďalšou neoddeliteľnou súčasťou konferencie sa stávajú odborné semináre pre učiteľov základných a stredných škôl. V tomto ročníku sú tri: „Blender – priestorové modelovanie“, „TEX-ujeme radi a veselo“ a „GeoGebra vo vyučovaní matematiky“.

GEOGEBRA VO VYUČOVANÍ MATEMATIKY WORKSHOP PRE UČITEĽOV

V posledných rokoch sa v školskej matematike objavujú nové vyučovacie metódy založené na využívaní digitálneho výučbového prostredia. Pre študentov základných a stredných škôl je digitálny svet prirodzenou súčasťou ich každodenného života. Nové hardvérové a softvérové vybavenie škôl prinášajú aj otázky ohľadom ich efektívneho využitia vo vyučovacom procese. Vyučovanie matematiky má z hľadiska didaktiky zvláštne postavenie a didaktika digitálneho vyučovania sa stáva jednou zo základných otázok.

Klasické a inovatívne vyučovacie metódy a formy (napríklad konštruktivizmus, projektové vyučovanie) majú v digitálnom prostredí nový rozmer, no vznikajú aj nové metódy priamo súvisiace s využívaním IKT (napríklad workshopová metóda, metóda peer instruction). Softvér GeoGebra má pre vyučovanie matematiky veľký význam hneď z viacerých pohľadov: je to open-source, teda je voľne dostupný učiteľom aj študentom, je to dynamický matematický softvér spájajúci geometriu, algebru a matematickú analýzu. GeoGebra bola vyvinutá pre účely vyučovania matematiky Markusom Hohenwarterom na Univerzite Florida Atlantic.

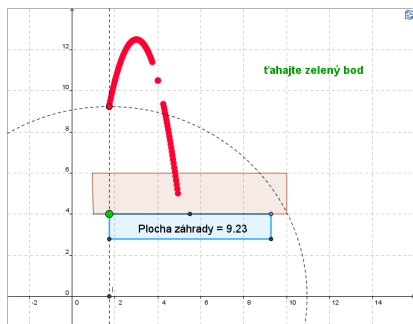
GeoGebra na internete

Hlavná stránka GeoGebry je na adrese <http://www.geogebra.org/> – na tejto stránke je možnosť nainštalovať si softvér, alebo spustiť priamo cez Webstart. Súčasťou stránky je pomocník, materiály na stiahnutie <http://www.geogebraTube.org/?lang=sk>, informácie o GeoGebra spoločenstve, ako aj informácie o konferenciách a workshopoch. Na stránke <http://geogebrawiki.wikispaces.com/> sa nachádzajú materiály roztriedené podľa typu školy a ročníka. Stránky slovenského GeoGebra inštitútu sú na <http://geogebra.ssgg.sk/> a Wiki o GeoGebre je na <http://wiki.geogebra.org/en>.

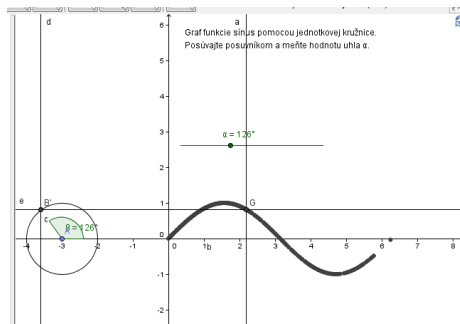
GeoGebra v školskej matematike – príklady

V školskej matematike môže učiteľ GeoGebru využiť:

- Ako motiváciu, motivačný príklad, motivačná ukážka na úvod vyučovacej hodiny. Ako príklad uvádzame motivačný príklad ako úvod do kvadratickej funkcie na stránke <http://elearn.ematik.fmph.uniba.sk/~babinska/kurz/kurzMatematika1.php?page=vlastnostiFcii> (obr. 1).
- Pri vysvetľovaní nového učiva, ako dynamickú vizualizáciu. Príkladom je vizualizácia grafu funkcie sínus <http://geogebra.ssgg.sk/Materialy/Slovak/SinusoidaAplet.html> (obr. 2).

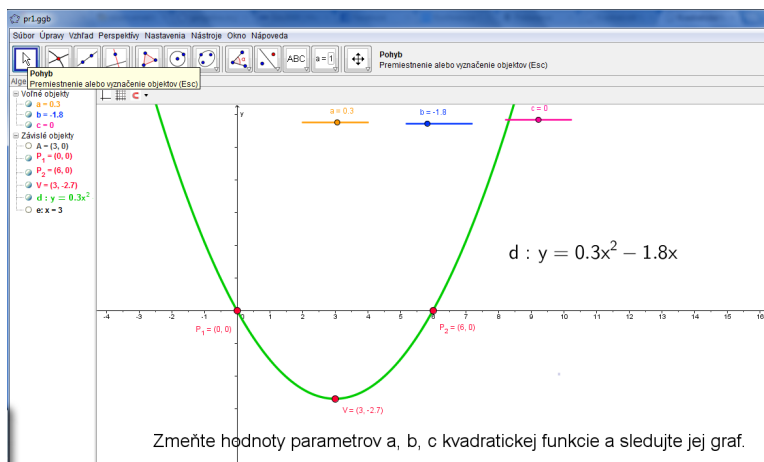


Obr. 1. Aplet vlastnosti funkcie.



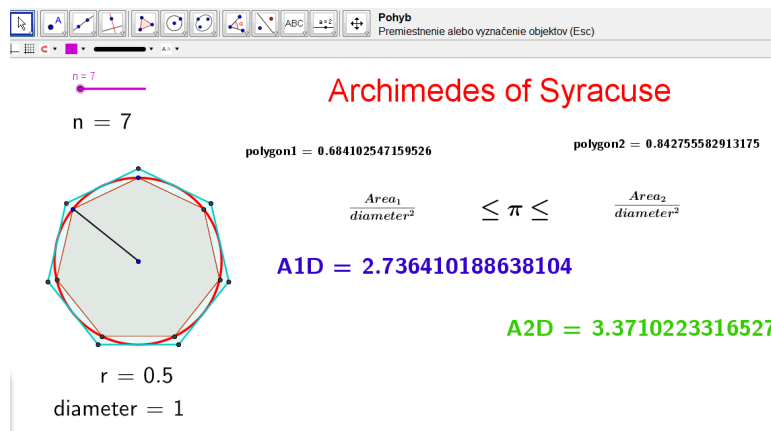
Obr. 2. Vizualizácia grafu funkcie sínus.

- Pri konštruktivistickom prístupe, ako riadené skúmanie. Ako príklad uvádzame úlohu, keď študenti majú objaviť vzťah medzi koeficientami kvadratickej funkcie a grafom funkcie (obr. 3). Druhý príklad je pre žiakov základnej školy, keď objavujú číslo π heuristickou metódou podobne ako Aristoteles (obr. 4).



Obr. 3. Skúmanie vlastností kvadratickej rovnice.

- Študenti môžu využiť GeoGebra aj v rámci projektového vyučovania. S možnosťami využitia GeoGebry sa zoznamujú aj študenti FMFI UK v rámci predmetov „Informačné a komunikačné technológie 5“ a „Didaktický softvér vo vyučovaní matematiky“. Kým prvom kurze sa oboznámi so softvérom ako takým, jeho základnými funkciami a možnosťami, v rámci druhého predmetu sa oboznámi s didaktikou využitia tohto softvéru. Zvýšiť atraktivnosť školského matematického vzdelávania je možné zaradením úloh z reálneho života a kontextových úloh, matematickým experimentovaním, miniskúmaním a pod.



Obr. 4. Aplet heuristickej metódy výpočtu čísla π .

Pre konštruktivistický prístup vo vyučovaní matematiky je veľmi vhodné digitálne prostredie GeoGebry. v rámci projektov pritom študenti vypracovali niekoľko vlastných námetov a metodík. Niektoré z nich sú sprístupnené na stránke <http://kniznica.sospreskoly.org/home/index.php?catid=15>. V budúcnosti chceme na tejto stránke uverejňovať aj záverečné práce učiteľov – absolventov kontinuálneho vzdelávania „Digitálne technológie vo vyučovaní matematiky základnej a strednej školy“ (informácie sú prístupné na <http://fmph.uniba.sk/index.php?id=2951#c5042>)

Program seminára GeoGebra vo vyučovaní matematiky

Súčasťou konferencie OSSConf 2012 je odborný seminár pre učiteľov základných a stredných škôl „GeoGebra vo vyučovaní matematiky“, v ktorom sa budeme venovať rôznym tematickým okruhom: **Motivácia vyučovania matematiky a fyziky pomocou softvéru GeoGebra** (doc. J. Gunčaga), **GeoGebra pre budúcich učiteľov matematiky na FMFI a v ďalšom vzdelávaní učiteľov** (M. Babinská), **Využitie Geogebra pri tvorbe pracovných listov** (V. Bejdová). Súčasťou seminára budú **prezentácie účastníkov (učiteľov)** a záverečný **workshop**.

Za organizátorov Lilla Koreňová

BLENDER, ZÁKLADY PRÁCE V 3D PROSTŘEDÍ

V súčasnej modernej dobe sa nástupom internetu (HTML5, WebGL), mobilných aplikácií, vizuálnych efektov a multimédií nezastaviteľne do popredia prediera 3D grafika a tvorba. Cieľom tématického bloku je predstaviť program Blender, užívateľské prostredie v ktorom sa pracuje a jeho základnú funkcionálnosť. Blender v súčasnosti nie je len nástroj na 3D modelovanie, ale komplexne rieši problematiku 3D tvorby a jej následného využitia napríklad vo filmovej tvorbe, webových a iných aplikáciách. Blender je nástroj na modelovanie, dajú sa v ňom simulovať niektoré procesy (napr. kvapaliny, oheň a dym, látky a pružné telesá), tvoriť realistické materiály a textúry, rendering a postprocesing. S výstupmi je možno ďalej pracovať a tvoriť animácie alebo hry, či tiež exportovať modely s príslušnými textúrami a použiť v ďalších aplikáciách.

História programu siaha až do roku 1988, kedy Ton Roosendaal, spoluzakladateľ holandského animačného ateliéru „NeoGeo“ začal prvé práce na programe, ktorý v súčasnosti poznáme ako Blender. v roku 2002 založil Ton neziskovú organizáciu „Blender Foundation“, ktorej základným cieľom bolo nájsť cestu pre rozvoj a podporu Blenderu. V júli 2002 Ton Roosendaal jednal s investormi o sprístupnení Blenderu ako Open source. Organizácia preto začala zháňať 100 000 eur na odkúpenie práv, zdrojového kódu a duševného vlastníctva. Túto sumu sa podarilo získať za sedem týždňov a v nedeľu 13. októbra 2002 sa stal Blender otvoreným programom. Na oslavu tejto významnej udalosti zorganizovala „Blender foundation“ prvú Blender konferenciu, ktorá sa odvtedy každoročne organizuje v Amsterdame v Holandsku.

Otvorenosť programu je jeho veľkou výhodou a umožňuje neustále program vylepšovať a to nielen zo strany oficiálnych vývojárov (Blender institute v Amsterdame v Holandsku). Významné sú príspevky, rozšírenia a vylepšenia zo strany nadšencov, pri ktorých sa často stalo, že po dopracovaní boli zavedené do oficiálnej verzie. Okrem toho vzniklo množstvo doplnkov, ktoré umožňujú a uľahčujú export a integráciu s inými aplikáciami, prípadne ďalšie použitie výstupov programu.

V tématickom bloku na konferencii si povieme krátko o histórii a evolúcii Blenderu. Ďalej sa zoznámime s užívateľským rozhraním a základmi modelovania. Ukážeme a skúsime si namodelovať jednoduchý objekt (scénu). V druhej časti dňa sa budeme venovať zložitejším modelovacím a renderovacím technikám (materiály, textúry, svetlo, tieň) a všetko toto si budú môcť účastníci vyskúšať.

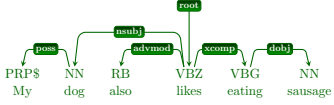
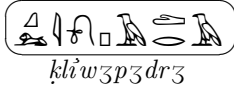
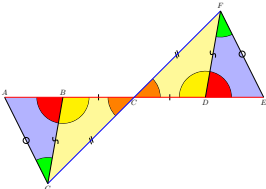
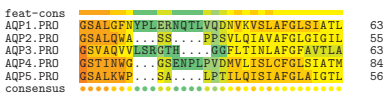
Za organizátorov Peter Tomašovič

TEXUJEME RÁDI A VESELE ANEB ZÁKLADNÍ A STŘEDNÍ ŠKOLA TEXU

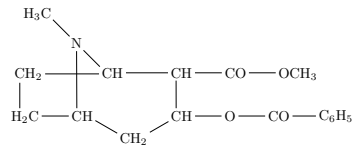
TEX uspokojuje řadu typografických potřeb s tahem na profesionální tvorbu dokumentů. Častější a opakovaná potřeba sazečů znamená větší šanci, že už to někdo řešil, vyřešil a zveřejnil. Tomuto rozšíření TEXu se v TEXovém světě říká balíček (angl. package). A že jich je! Dokonce můžeme říci, že jich je tolik, že pokryjí téměř vše, na co si sazečského vzpomeneme! Vy nevěříte, že?

Proklepli jsme předměty základních a středních škol a přikládáme důkaz jejich existence s odkazem na oficiální dokumentaci.

Nu, a pokud svůj předmět či oblíbenou řešenou oblast nevidíte, dejte nám vědět e-mailem či přijďte na seminář při OSSConf prodiskutovat osobně. Vyřešíme!

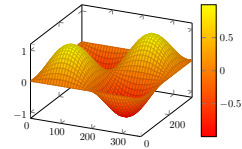
Předmět-zyk	Výběr z učiva	Ukázka
Český a slovenský jazyk	xcolor, tipa	meruňka – apri-cot [ˈæprɜːkɑːt] marhuľa – apri-cot [ˈeɪprɜːkɑːt]
Anglický jazyk	tikz-dependency	
Ruský jazyk	amsmaths	Добрый день, участники!
Staroegyptský jazyk	hieroglf	
Japonský a čínský jazyk	fontspec	今日は！ 你好。
Klingonský jazyk	luaotfload	᠘ᠠᠨᠠᠨᠠᠨᠠ ᠠᠨᠠᠨᠠᠨᠠ ᠠᠨᠠᠨᠠᠨᠠ ᠠᠨᠠᠨᠠᠨᠠ
Předmět-ie	Výběr z učiva	Ukázka
Geometrie	tkz-euclide	
Biologie	texshade	

Chemie

chemfig
mychemistry**Předmět-ika****Výběr z učiva****Ukázka**

Matematika

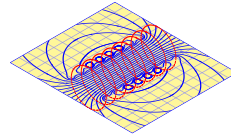
pgfplots, pst-plot



$$z(x, y) = \sin(x) \cdot \sin(y)$$

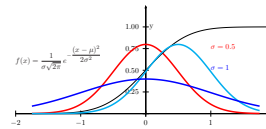
Fyzika

pst-magneticfield



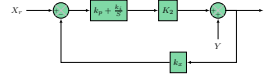
Statistika

pst-func

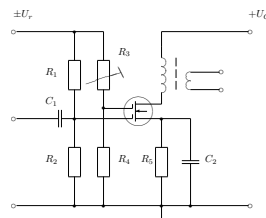


Informatika

bloques

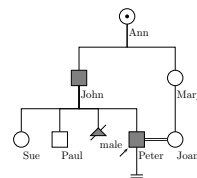


Elektrotechnika

eltex
circuit_macros**Předmět-pis****Výběr z učiva****Ukázka**

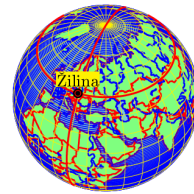
Přírodopis

pst-pdgr



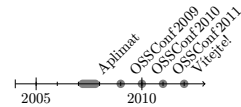
Zeměpis

pst-geo



Dějepis

chronology



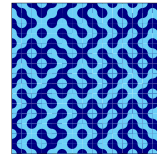
Předmět-ova

Výběr z učiva

Ukážka

Výtvarná výchova

pst-fill



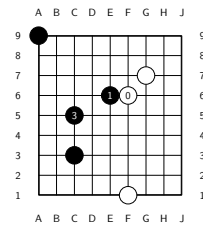
Hudební výchova

musixtex



Tělesná výchova a sport

psgo



MOTIVÁCIA VYUČOVANIA MATEMATIKY A FYZIKY POMOCOU SOFTVÉRU GEOGEBRA

GUNČAGA, Ján, (SK); JANČEK, Martin, (SK)

Otvorený softvér Geogebra je vhodný prostriedok ako ilustrovať výsledky získané pri matematických a fyzikálnych výpočtoch. Autori ukážu ako vhodne motivovať vyučovanie matematiky a fyziky pomocou Geogebry.

VYUŽITIE GEOGEBRY PRI TVORBE PRACOVNÝCH LISTOV

BEJDOVÁ, Veronika, (SK)

Súčasný trendy zavádzania digitálnych technológií do vyučovania sa okrem iného prejavujú nielen v podobe rôznych školení a ďalších vzdelávaní učiteľov z praxe, ale stávajú sa i súčasťou vzdelávania budúcich učiteľov na vysokých školách. Príkladom tohto tvrdenia je predmet Didaktický softvér vo vyučovaní matematiky, ktorý vznikol na pôde Fakulty matematiky, fyziky a informatiky Univerzity Komenského v Bratislave. Odporúča sa študentom prvého ročníka magisterského štúdia, patrí k skupine povinne voliteľných predmetov. Predchádza mu niekoľko povinných predmetov, v rámci ktorých sa študenti oboznamujú s rôznymi softvérmi a ich využitím.

APACHE JMETER A JEHO POUŽITIE PRI TESTOVANÍ MAPOVÝCH VYHLADÁVACÍCH SLUŽIEB

CHLEBEC, Stanislav, (SK)

V úvode bude predstavený program Apache JMeter. Je to OSS naprogramovaný v Jave a dá sa spustiť z bežného PC s akýmkoľvek operačným systémom – podmienkou je nainštalovanie Javy. Používa sa na meranie výkonnosti, záťažové testovanie a testovanie funkčnosti. V ďalšom sa budeme venovať použitiu programu. Program je možné použiť pri testovaní v rôznych situáciách. My sme ho použili pri testovaní mapových vyhľadávacích služieb, ktoré sú dostupné na webe pomocou HTTP/HTTPS protokolov. Testovanie sa riadilo požiadavkami smernice Európskej únie INSPIRE. Budeme demonštrovať možnosti programu na konkrétnych testovacích scenároch. Osobitne demonštrujeme:

- Scenáre: Príprava scenárov pre JMeter umožňuje automatizovať testovací proces.
- Komponent HTTP Request: Vieme zostaviť HTTP žiadosť GET a POST.
- Komponenty umožňujúce viacnásobne a simultánne odosielať žiadosti HTTP.
- Komponenty umožňujúce kontrolovať obsah vrátenej odpovede pomocou xpath a regulárnych výrazov.

- Komponenty umožňujúce úroveň vetvenia testovacieho scenáru.
- Komponenty umožňujúce spracovania výstupov pomocou skriptovacích jazykov napr. Javascript.
- Komponenty umožňujúce pripraviť vstupné dáta v CSV súbore.

GEOGEBRA PRE BUDÚCICH UČITELOV MATEMATIKY NA FMFI A V ĎALŠOM VZDELÁVANÍ UČITELOV

KOREŇOVÁ, Lilla, (SK); BABINSKÁ, Martina, (SK)

Otvorený softvér GeoGebra za posledných desať rokov svojej existencie prenikol do povedomia učiteľov matematiky na Slovensku a využíva sa vo vyučovaní na mnohých základných, stredných ale aj vysokých školách. V rámci povinne voliteľných predmetov sa s ním oboznamujú aj študenti FMFI UK (budúci učitelia matematiky) a učitelia z praxe sa budú môcť vzdelávať prostredníctvom kontinuálneho vzdelávania aj v oblasti didaktického využitia tohto softvéru. V príspevku prezentujeme obsah vzdelávania budúcich učiteľov a učiteľov z praxe ohľadne využitia softvéru GeoGebra. Uvádzame aj niekoľko námetov pre školskú matematiku.

STATISTICAL MODELING IN GRETL AND ITS USE IN TEACHING ECONOMETRICS

FALÁT, Lukáš (SK); ZÁBOVSKÁ, Katarína, (SK)

Nowadays, statistical models are very popular in various areas of our life. Econometrics is not an exception. Econometricians use statistical models to quantify various problems of economic field. Therefore econometrics is sometimes considered as a strategic tool for problem solving, decision-making or information management. No surprise that the selection of a good software is a key point to make good models in econometrics analysis. Gretl seems to be such a software. Paper discusses the main features of gretl, its advantages and properties. Later authors present the use of this software in teaching process of Econometrics.

OPEN SOURCE R AND ADVANCED METHODS OF PREDICTIVE MODELING

FALÁT, Lukáš (SK); PANČÍKOVÁ, Lucia, (SK)

Open Source currently offers many opportunities for students, teachers and researchers in finding unique solutions, creating new approaches. This article discusses the predictive modeling of economic and financial data in R environment and R Studio. It refers to the benefits, respectively disadvantages of this approach.

The authors deal with the chosen means of advanced statistical methods and machine learning methods in short time series modeling. They show the way to interesting solutions for the needs of education, science and practice.

ACRO_TE_X, DOTAZNÍKY A PHP SKRIPTY

ČANECKÝ, Peter, (SK)

Acro_TE_X ponúka široké možnosti ako vytvárať dotazníky, testy a následne ich vyhodnocovať. Získané výsledky (každej odpovede) je možné po skončení testu vyhodnotiť a následne exportovať do databázy alebo textového súboru, prípadne poslať zvolenému užívateľovi na e-mail.

LOG AND TRACE — THE APPLICATION DIARY

KANIK, Tomasz, (PL); KOVAČ, Marian, (SK)

Logging and tracing is the essential part of debugging of applications. The article deals with advanced Log and Trace system Log4j. There are listed basic logging and tracing operations troubles and main reasons for usage of the advanced logging systems. The rest of the article describes the Log4j system itself, its components, features, basic commands and their parameters with a description, the configuration file and examples of using this system.

AKO VZNIKÁ CYKLOATLAS

PÁLENÍK, Michal, (SK)

Komunita projektu www.freemap.sk prevádzkuje rôzne druhy máp (turistickú mapu, autoatlas, lyžiarsku mapu, cykloatlas, ...). Ukážeme si ako vzniká cykloatlas: vkladanie údajov (body, cesty), ich načítanie a renderovanie, export, atď. Predstavíme tiež projekt www.oma.sk, ktorý slovné popisuje jednotlivé (nielen) cyklotrasy Slovenska.

ČO NOVÉHO NA FREEMAPSLOVAKIA A V OPENSTREETMAP

PÁLENÍK, Michal, (SK)

Projekt www.OpenStreetMap.org je tu s nami už 8 rokov a projekt www.freemap.sk už 6 rokov. Počas tohto obdobia prešli obidva sesterské projekty búrlivým vývojom. Počas prednášky si predstavíme dôvody vzniku projektov, ich vývoj, zameranie a dopad, ale najmä novinky ktoré vznikli počas rokov 2011 a 2012.

JEDNODUCHÁ TVORBA DOKUMENTOV OBSAHUJÚCICH VÝSTUP Z R (RSTUDIO+SWEAVE+KNITR)

KRÁL, Pavol, (SK)

Dynamická tvorba dokumentov obsahujúcich výstupy zo štatistického systému R nám môže zjednodušiť prípravu článkov, prezentácií i testov. V príspevku predstavíme jednoduchý spôsob, ako generovať takéto dokumenty pomocou IDE RStudio a R balíčkov Sweave a knitr.

LYX

KRÁL, Pavol, (SK)

V príspevku predstavíme LyX, program na tvorbu dokumentov s využitím možností L^AT_EX-u bez potreby L^AT_EXsa naučiť. Ukážeme si nastavenie vlastností dokumentu, tvorbu štruktúry dokumentu a zoznamu literatúry, vloženie obsahu, matematických vzťahov, obrázkov, tabuliek, krížových odkazov a odkazov na literatúru.

T_EXONWEB A ALTERNATÍVY

KRÁL, Pavol, (SK)

V príspevku stručne predstavíme niektoré možnosti online tvorby dokumentov v L^AT_EX-u. Zameriame sa na T_EXonWeb, ScribT_EX, Verbosus, MonkeyT_EXa L^AT_EXlab.

PROČ SE OPEN SOURCE NEROZŠIŘUJE TAK RYCHLE, JAK SI MYSLÍME

OTT, Vlastimil, Liberix, (CZ)

Najít, pojmenovat příčiny malého rozšíření FOSS. Najít a pokusit se realizovat činnosti, které povedou k nápravě. Možné příčiny:

- open source firem je málo, ty stávající nevidí příležitosti, špatně se prezentují, neznají potřeby trhu,
- nízké povědomí o FOSS, předsudky, nedostačující řešení, malá nabídka,
- běžný uživatel nerozumí mnoha pojmům, bojí se, volí osvědčenou jistotu (=ne FOSS),
- co změnit, aby to bylo lepší – ve školství, ve firemním prostoru, ve veřejném prostoru.

Chci najít odpovědi, upozornit na to, že chyba je hlavně na straně nás, kteří se

snažíme ten soft propagovat. Motivovat nejen studenty k založení firem, prezentovat business možnosti open source. Pojmenovat rizika podnikání s FOSS, hledat východiska, najít partnery (akcelerátory, obchodní komory, velké firmy, asociace firem apod.).

JAK NA VLASTNÍ MRAK

HRUŠECKÝ, Michal, (CZ)

Stále více dat ukládáme různě po internetu. Můžeme ale těmto službám věřit, když jim svěřujeme citlivá data? Je to pohodlné, ale je to bezpečné? ownCloud nabízí pohodlí cloudu, ale zároveň umožňuje zachovat si kompletní kontrolu. Přednáška ukáže co ownCloud nabízí, jak ho rozběhnout a jak ho používat.

SVOBODNÝ ZDRAVOTNICKÝ SYSTÉM GNU HEALTH

JELÍNEK, Lukáš, (CZ)

Součástí velkého projektu svobodného softwaru GNU je i zdravotnický informační systém GNU Health. Během necelých čtyř let vývoje ušel velký kus cesty a stal se softwarem, který už rozhodně má co říct. Tento příspěvek ukazuje současné možnosti systému GNU Health, včetně zhodnocení kompatibility s prostředím středoevropského veřejného zdravotnictví.

PRŮVODCE TVORBOU DOKUMENTŮ

**RYBIČKA, Jiří, (CZ); ČAČKOVÁ, Petra, (CZ);
PŘICHYSTAL, Jan, (CZ)**

Představení publikace zabývající se zpracováním textů na počítači. Vznikla jako výsledek dlouhodobých zkušeností jak v oblasti výuky, tak i v praktických řešeních.

VIRTUALIZÁCIA NA DESKTOPOCH

PRÍVOZNÍK, Michal, (SK)

Virtualizácia ako moderný trend je skor doménou biznisu, než desktopového použitia. Avšak, nielen s najnovším hardverom je možné ju použiť a zároveň aj profitovať z nej. Prečo a ako by mala byť použitá? V prezentácii budú ukázané mnohé príklady a pokryté rôzne prípady použitia. Od passthrough PCI/USB zariadení, cez guest-agentov, nastavovanie QoS na virtuálnych NIC, až po migráciu. Taktiež je prezentovaných niekoľko nástrojov ako virt-manager a mnohé iné z virt-* rodiny, libvirt, virsh ako aj nástroj na integráciu do desktopového

prostredia (GNOME Boxes). S týmito nástrojmi používateľa už nie sú zafaznené správy a udržiavaním, ale môžu z virtualizácie spraviť užitočný nástroj.

ZKUŠENOSTI S OTEVŘENÝM SOFTWAREM PŘI VÝVOJI SYSTÉMU PRO STROJOVÉ ROZPOZNÁVÁNÍ KVALITY V PRŮMYSLOVÉ METALOGRAFII

PRAKS, Pavel, (CZ); KOTAS, Petr, (CZ); KRACÍK, Jan, (CZ)

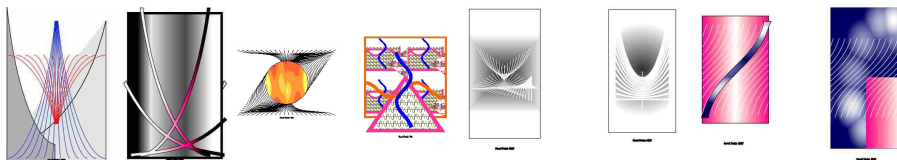
Cílem příspěvku je demonstrovat naše zkušenosti s vývojem systému pro strojové rozpoznávání kvality digitálních fotografií metalografických vzorků v průmyslové metalografii. Vyvíjený systém se skládá ze dvou částí. První část systému umožňuje automatickou segmentaci vzorku a automatické nalezení charakteristik středových objektů (vad). Druhá část systému automaticky analyzuje nalezené charakteristiky středových objektů a odhaduje na základě vyvinutého statistického modelu středovou kvalitu vzorku.

Při implementaci systému bylo s výhodou využito otevřených programových nástrojů, jmenovitě OpenCV a NumPy s kompilátorem do binárního kódu.

VIZUÁLNÍ OBJEKTY

PRAKS, Pavel, (CZ)

Spisovatelé či hudebníci využívají pro vyjadřování symboly – znaky abecedy. Obdobně je tomu i u Vizualních objektů, které používají pro vyjadřování mj. elementární matematické funkce.



VYŘAZOVÁNÍ A ÚPRAVY NAD PDF SOUBORY

STRÍŽ, Pavel, (CZ)

Mezi pravidelné problémy v nakladatelství obecně a v kancelářských předtiskových operacích obzvlášť patří seskládání a vyřazování vstupního PDF souboru. Ať už na brožurku, určené ke složení a na skobu, či na půlení několika výtisků na řezače, určené na vazbu brožovanou či vázanou. Někdy nám pomůže tiskový ovladač, ale není to pravidlem. Pomocníkem nám bude balíček `graphicx` a především `pdfpages`. Na přednášce si ukážeme škálování, výběry a výřezy stran (pomocníkem při odměřování nám bude program `PS_View`), dále spojování PDF souborů a jejich opatření záhlavím, zápatím či logem v některém rohu. Další typickou

situací je opatření stran vodoznakem, komentářem či jiným ochranným prvkem. Ukážeme si i zdánlivě náročnou operaci opatření stran ořezovými a bigovacími značkami (balíčky `zwpagelayout`, `crop` či přímo přes TikZ).

Pozpátku seřazené:	8	7	6	5	4	3	2	1
K rozpůlení na řezačku:	1 1	2 2	3 3	4 4	...			
K přehnutí vejpuřl na skobičku:	8 1	2 7	6 3	4 5				

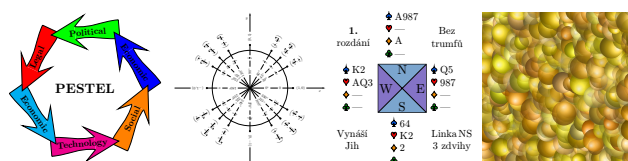
NA CO VŠE POUŽÍVÁME BALÍČEK TIKZ?

STŘÍŽ, Pavel, (CZ)

Na záludný to dotaz se dá odpovědět, že téměř na vše, co nám přijde grafického pod ruce. Od vykreslení vlastního designu šipek v diagramech, matematických pomůcek jako je jednotková kružnice, přes vlastní znaky karet v bridžových diagramech až po obálky knih s dvourozměrnou stratifikací objektů pseudo-náhodně umístovaných do pozadí. A je toho víc!

Těch menších i větších kreseb má autor za roky již několik desítek a postupně vzniká drobná kolekce (<http://striz8.fame.utb.cz/courses/dep/>), ze které si dovolíme vybrat a okomentovat i další vzorky.

Za ideálních okolností příchozím na konferenci OSSConf 2012 předáme i jeden vytištěný TikZový dárek.



O SAZBĚ ZDROJOVÝCH KÓDŮ

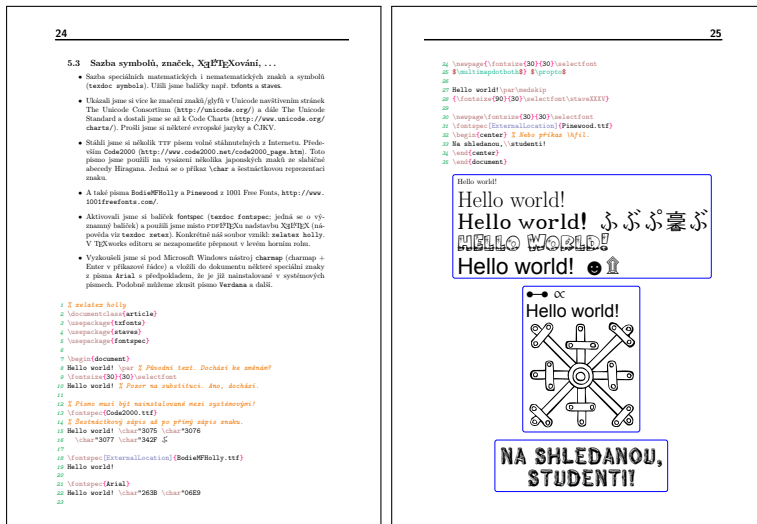
STŘÍŽ, Pavel, (CZ)

Při přípravě studijní pomůcky nazvané *Sazba v T_EXu a kresba v METAPOSTu*, ISBN 978-80-87106-51-8, což jsou sesbírané materiály a testy klasifikovaného zápočtu při výuce vysokoškolských studentů, autor narazil na problém, že současné nástroje sazby zdrojových kódů mu nevyhovují, a to ani kalibry balíčku listings či nástrojů jako je Pygments, WinHighlight, či tisk přes Gedit.

Ze zásadních to byla nekontrolovatelnost dělení řádků odstavců a stránkového zlomu, náročná manipulace křížových odkazů na jednotlivé řádky zdrojového kódu při zachování funkčnosti kódu, nemožnost v kódu či komentáři mít značkovou sadu při UTF-8 kódování, rychlé zapnutí/vypnutí číslování řádků přes více

kódů, dosáhnout nedělitelných mezer u jednoznakových předložek a spojek v komentářích či mít stejný barevný design pro kódy $\text{T}_{\text{E}}\text{X}$ u a $\text{M}_{\text{E}}\text{T}_{\text{A}}\text{P}_{\text{O}}\text{S}T$ u. Nejen kód vysázet (myšleno řešení úloh), ale i jej spustit a vysázet jeho výsledek (myšleno zadání typografických úloh). Nedejbuň chtít do sazby kódu dostat nějaký ten obrázek jako vykřičník nebo otazník, jak známe z materiálů pro kombinovanou formu výuky. Nehovoříce o automaticky tvořeném seznamu vysázených souborů, počtu jejich zdrojových řádků a velikosti na úrovni bajtů.

Až po mixáži nástrojů za přispění vlastních Bash kódů a využití balíčku TikZ na orámování, autor dosáhl chtěných výsledků. I poté však bylo co zlepšovat.



SAZBA ČÁROVÝCH KÓDŮ

STŘÍŽ, Pavel, (CZ)

V loňském sborníku (OSSConf2011) jsme si ukázali sazbu vybraných čárových kódů, praktického problému ve výrobní sféře a při manipulaci (poštovní, přepravní a letecké služby). Letos o tom pohovoříme a poukážeme na typické typografické problémy s čárovými kódy spojenými. Bude se jednat především o úpravu parametrů, které nejsou v dokumentaci balíčku pst-barcode zatím zaneseny. Dále pak rozmnožení jednoho čárového kódu na sadu na celou tiskovou stranu či předpřipravený formulář, přesné umístění na straně, tisk dávky čárových kódů či jejich umístění a ořez na potisk CD/DVD.

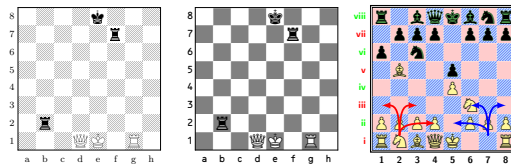


SAZBA ŠACHOVÝCH DIAGRAMŮ A EXOŠACHU

STRÍŽ, Pavel, (CZ)

Ačkoliv se to na první pohled nezdá, sazba šachu je náročná typografická oblast. Na přednášce bude zmíněn vývoj v $\text{T}_{\text{E}}\text{X}$ ovém světě až po současné výstřelky balíčků `chessboard` a `xskak`. Dokumentaci neobjevíme v $\text{T}_{\text{E}}\text{X}$ Live kolekci, ale až na `CTAN.ORG`, což je drobná bariéra, kterou překonáme. Zběžně prohlédneme dostupná šachová písma a problematiku jejich instalace za $\text{T}_{\text{E}}\text{X}$ ový svět.

Pokud by se nám i přesto zdálo, že je to procházka růžovým sadem, nahlédneme na problematiku sazby exošachu. Využijeme k tomu formátů $\text{XeL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ a $\text{LuaL}_{\text{A}}\text{T}_{\text{E}}\text{X}$, které umí přímou práci s písmovými formáty TTF/OTF.



ZE SVĚTA L^ATEXU: KNIHOVNY BIGNUM A BIGRAT

STRÍŽ, Pavel, (CZ)

Sám Donald E. Knuth, autor jádra $\text{T}_{\text{E}}\text{X}$ u, přiznal, že sice systém výpočetně pracuje na nejmenších možných fyzikálních tiskových možnostech, ale už to neplatí pro rozměry větší ani pro výpočty velkých čísel či kontrolu zaokrouhlování. Zde přichází na pomoc $\text{LuaT}_{\text{E}}\text{X}$, <http://www.luatex.org/>, jinými slovy, o tento druh problému se postará programovací jazyk Lua, <http://www.lua.org/>.

V našem případě přímo použijeme Lua knihovny `BigNumber` a `BigRational`, <http://oss.digirati.com.br/luabignum/>, jejich doinstalováním, přesněji nakopírováním do pracovního adresáře.

Užití je bohaté (výpočty v kombinatorice, finančnictví, fyzice a astronomii), ovšem my si můžeme uvést humornější ukázkou sazby pro výpočet 2^{5000} . Může to prosím někdo numericky ověřit?

$d=2^{5000}=141246703213942603683520966701614733366889617518454111681368808585711816984270751255808912631671152637335603208431366082764203838069979338335971185726639923431051777851865399011877999645131707069373498212631323752553111215372844035950900535954860733418453405575566736801565587405464699640499050849699472357900905617571376618228216434213181520991556677126498651782204174061830939239176861341383294018240225838692725596147005144243281075275629495339093813198966735633606329691023842454125835888656873133981287240980008838073668221804264432910894030789020219440578198488267339768238872279902157420307247570510423845868872596735891805818727796435753018518086641356012851302546726823009250218328018251907340245449863183265637987862198511046362985461949587281119139907228004385942880953958816554567625296086916885774828934449941362416588675326940332561103664556982622206834474219811081872404929503481991376740379825998791411879802717583854985751152994717434692411170702303981033786152327937102909926564448428955118303557331520208041579200900418119518804567055154683494461827317423276859892776076207095258783187664883683489650154749978641197654414333569280123441117657353363935578921493700434756820866595871776405929359288751429284355704708916487648311661569188620381299755690171892169733755224469032475078797830901321579940127337210694377283439922280274060798234786740434893458120198341101033812506720046609891160700284002100980452964039788704335302619337597862052192280371481132164147186514169090917191909376$

O PŘÍPRAVĚ KNIH LOGICKÝCH HER

STŘÍŽ, Pavel, (CZ)

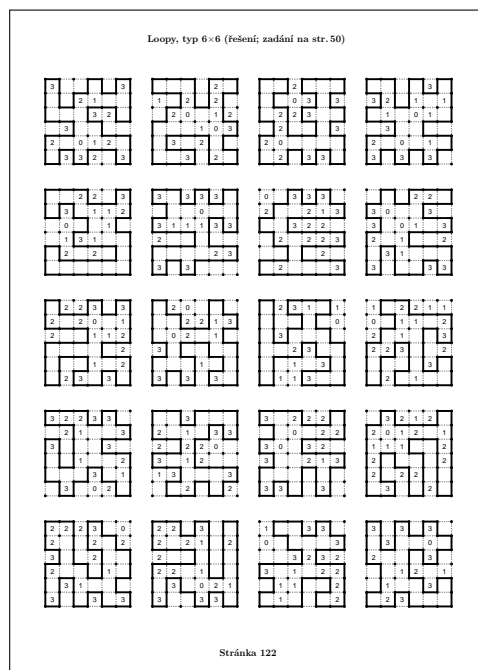
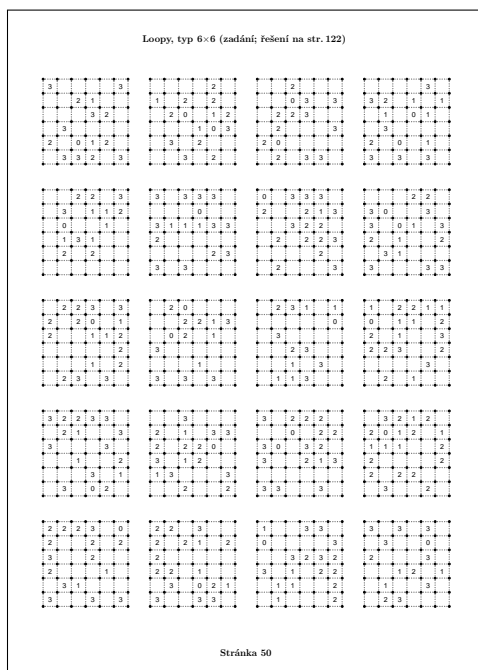
Při přípravě vánočních dárků (Vánoce 2011) pro neteřinky od sestry autor narazil na zajímavý balík v Linuxovém repozitáři, `sgt-puzzles`. Co ho zaujalo nejvíc bylo, že lze z příkazového řádku generovat zadání, řešení a nechat si některé hry vytisknout do PostScriptu, což lze snadno převést do PDF. Dokonce šlo výstupy zpracovávat z textového souboru. Na tom autor dále stavěl, především z pohledu seskládání her, sepsání jejich zadání a snímků obrazovek, taktiky, příprav osnov a křížových odkazů mezi zadáními a řešeními.

Takto vznikly dvě knihy. Jedna jako sbírka 23 her, pod názvem *Výběr 23 logických her*, ISBN 978-80-87106-52-5, kde jsem přidal mezi logické hry kolekce navíc bridž, bludiště a kakuro i se zápornými a nulovými hodnotami na základě úprav existujících programů napsaných v Pythonu. Za důležité je potřeba zmínit, že u většiny her této kolekce lze nastavit obtížnost, i toho bylo využito.

Druhá kniha byla více specializovaná s rozmanitějšími stupni obtížnosti a typů. Jedná se o knihu *Logické hry sudoku, loopy a kakuro*, ISBN 978-80-87106-53-2.

Během přednášky autor pohovoří hlavně o meznících příprav a využitých nástrojích. Mezi nejtěžší partie patřilo parsování bridžových pětikartových koncovek Richarda Pavlicka a kompletní vysázení zadání a řešení her kakuro.

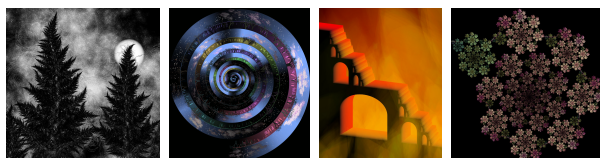
Za zajímavost lze zmínit, že volby `--save` a `--with-solutions` byly omezeny a nespolečněly, ale autor, Simon G. Tatham, byl vstřícný a dopracoval to.



PŘÍPRAVA NÁHLEDOVÝCH KNIH GALERIÍ: PRŮZKUM

STŘÍŽ, Pavel, (CZ)

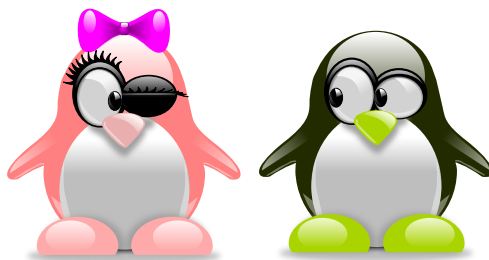
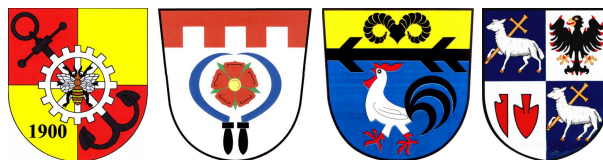
Přednáška poukáže z čeho se dá připravit náhledová kniha, především z oblastí matematiky a umění, ale možná dojde i na oblasti jiné. Na úvod přednášky budou poukázány zajímavé odkazy z <http://tessellations.org/> a programy Kali (kategorie Wallpaper, Frieze a Rosette), KXStitch a Context Free Art (ukázky).



Hlavním tématem budou fraktály, včetně relativně nového algoritmu Flame Fractal (volně přeloženo jako plamenný fraktál), především implementovaných do programů Apophysis+ApoMap a Incendia (ukázky). Navážeme průzkumem galerií, včetně prozkoumání dostupných gradientů pro program Apophysis a doplňkových aplikací u programu Incendia jako jsou Square Quilts a Hextiles.



Na závěr nahlédneme na další nápady z rešeršního průzkumu, jako by mohla být náhledová kniha textur, tangramů, vlajek států, státních znaků a erbů měst (ukázky). Za extrémní lze považovat myšlenky vydání herních schémat (Sokoban, Atomix, Pingus, Berušky, Pysolfc či Fish Fillets). Skončíme na začátku, tedy v Linuxu, a to u balíčků openclipart-svg a openclipart-png (poslední ukázka).



OTVORENÝ SOFTVÉR VO VZDELÁVANÍ, VÝSKUME A V IT RIEŠENIACH
Zborník príspevkov medzinárodnej konferencie OSSConf 2012

Prvé vydanie 2012

Elektronická sadzba programom pdfL^AT_EX Rudolf Blaško

Tlač C-PRESS, s. r. o., Košice

ISBN 978-80-970457-2-2