

Výuka vizuálního programování v geografických informačních systémech

Instruction of Visual Programming in Geographic Information Systems

Zdena Dobešová*

Abstrakt

Cílem příspěvku je prezentovat výuku vizuálního programování pro bakalářský studijní obor Geoinformatika a geografie na Univerzitě Palackého v Olomouci. Studenti mají výuku komponenty ModelBuilder pro ArcGIS ve druhém ročníku studia v rámci předmětu Programování 2. Výuce jsou věnovány tři výukové týdny. V příspěvku jsou prezentovány zkušenosti s výukou a srovnání s literaturou, která se zabývá výukou pro programátory-záčičníky a vizuálním programováním (Robins et al. 1996, Moody 2009).

Studenti se naučí sestavovat v grafickém prostředí komponenty ModelBuilder postupy zpracování prostorových dat, resp. definovat sled funkcí ve formě diagramu. Díky jednoduše ovladatelnému prostředí a grafickému znázornění jsou studenti schopni rychle sestavovat postupy ve formě diagramů. Po výuce vizuálního programování následuje v témže předmětu výuka v textovém programovacím jazyce Python. Náplní praktických úloh je opět návrh programů, které automaticky zpracují prostorová data pomocí GIS funkcí z programu ArcGIS Desktop.

Zvolený přístup, kdy studenti začínají s vizuálním programováním a to je následováno textovými programováním, je prospěšný pro získání dovedností ve tvorbě programů. Výhodou je zde možnost automatické konverze grafických diagramů z ModelBuilder do jazyka Python. Tyto zkonvertované diagramy lze použít jako výchozí fragmenty pro další rozpracování programů v jazyce Python.

Klíčová slova

dataflow diagram, grafická notace, vizuální programování, GIS, výuka

* Ing. Zdena Dobešová, Ph.D., e-mail: zdena.dobesova@upol.cz, Katedra geoinformatiky, Přírodovědecká fakulta, Univerzita Palackého, 17. listopadu 50, 771 46 Olomouc, ČR, tel. +420 58 563 4763

Abstract

The article describes the teaching of visual programming for the bachelor study branch Geoinformatics and Geography at Palacký University in Olomouc. Students take a part in subject “Programming 2” at the second study year where the lectures about ModelBuilder for ArcGIS are. The lectures take three weeks. The article presents the experiences with lecturing and compares it with the literature about lecturing for novice programmers and visual programming (Robins et al. 1996, Moody 2009).

The students train to design the workflow diagrams for processing spatial data in graphical component ModelBuilder. The intuitive interface and style of visual programming help to students to create very quickly fully functional diagrams. Textual programming in Python language follows the visual programming in this subject. Once again the tasks are programs that automatically processed spatial data by GIS function from software ArcGIS for Desktop.

The sequence of visual programming and textual programming is fruitful for the gathering of knowledge by students. Very helpful is the possibility of automatic conversion of diagrams to Python language. These program fragments could be used as start point for extension into functional Python programs.

Key words

dataflow diagram, graphic notation, visual programming, diagram, GIS, teaching

Literatura

- DOBEŠOVÁ Z. (2014a) E-learning for Visual Programming Language, ICETA 2014, Proceedings of 12th *International Conference on Emerging eLearning Technologies and Applications*, IEEE, elfa, Starý Smokovec, Slovensko, 2014, 103-108 s., ISBN 978-1-4799-7738-3.
- DOBESOVA Z. (2014b) Data Flow Diagrams in Geographic Information Systems: A Survey, SGEM 2014, *14th International Multidisciplinary Scientific GeoConference Volume I*, STEF92 Technology Ltd. Sofia, Bulgaria, 17-26 June 2014, 541-548 s. ISBN 978-671-7105-10-0, DOI: 10.5593/sgem2014B21
- ROBINS, A., ROUNTREE, J., ROUNTREE, N. Learning and Teaching Programming: A Review and Discussion, *Computer Science Education*, Vol.13, No.2, Swets & Zeitlinger, s. 137-172, 2003.
- MOODY, D. The “Physics” of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering*. 2009, roč. 35, č. 6, s. 756-779.

Tento příspěvek byl podpořen projektem CZ.1.07/2.3.00/20.0170.

Úvod

Program ArcGIS for Desktop od firmy Esri je jedním z nejběžnějších geografických informačních systémů (GIS) používaných v ČR i ve světě. Pro bakalářský studijní obor Geoinformatika a geografie, resp. navazující magisterský obor Geoinformatika je to jeden ze základních GIS programů využívaných ve výuce studentů.

V GIS se pořizují, analyzují a vizualizují prostorová data. Jednotlivé operace lze provádět interaktivně za použití jednotlivých funkcí programu. V případě, že je třeba více kroků zpracování je lépe postup práce automatizovat pomocí programů. Postup se pak realizuje jako dávkový program. Programy je výhodné sestavit pro úlohy, které se provádí opakovaně pro různá prostorová data nebo pro stejná data z různých časových období. Zde se projeví výhoda opakovaného použití hotových programů.

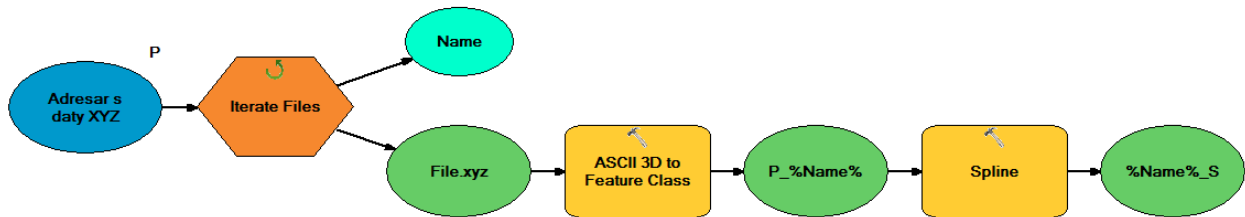
Tvorbu dávkových programů pro ArcGIS lze realizovat buď jako grafické modely v komponentě ModelBuilder nebo jako skripty v jazyce Python, který využívá knihovnu ArcPy. Grafický editor ModelBuilder je přímo součástí programu ArcGIS. ModelBuilder se díky grafickému navrhování řadí mezi vizuální programovací jazyky. Druhá možnost zápisu postupu zpracování dat je skript v jazyce Python, kde je možné volat funkce programu ArcGIS jako metody. Programové volání metod je velice dobře dokumentováno nápovědou programu ArcGIS s ukázkami kódu.

Popis komponenty ModelBuilder a jeho výuka ModelBuilder

Ve druhém ročníku ve čtvrtém semestru se studenti učí v rámci předmětu Programování 2 tvorbu modelů v komponentě ModelBuilder a skriptů v Python. Po předchozích třech semestrech zvládají na velice dobré úrovni různé GIS analýzy dat (překryvné analýzy, generování digitálních povrchů a práci s nimi atd.), tj. interaktivní práci v programu ArcGIS. V tento okamžik je vhodně zařazena výuka ModelBuilder. Výuce ModelBuilder jsou věnovány tři týdny výuky s dotací tři hodiny týdně.

Sestavování postupů – modelů se děje formou vytváření digramů, kde se definuje sled jednotlivých kroků zpracování. Data jsou reprezentována v modelu modrými a zelenými ovály a operace jsou reprezentovány žlutými obdélníky. Sestavování modelu se děje metodou táhni a pusť. Jednotlivé grafické prvky se propojují orientovanými šipkami naznačujícími směr zpracování. Pro jednotlivé GIS operace je nutné nastavovat příslušné parametry operace v dialogovém okně, které se otevře po dvojkliku na grafickém prvku. Diagramy lze postupně

testovat na funkčnost po částech tak, jak jsou sestavovány. Grafické prostředí je velice intuitivní a dobře studenty zvládnutelné. Ukázka modelu je na obr. 1.



Obr. 1: Ukázka model v prostředí ModelBuilder pro ArcGIS

ModelBuilder je velice často používán i specialisty v praxi jako jsou ekologové, územní plánovači apod., pro jeho jednoduchost. Tito specialisté nejsou většinou programátoři a nemají znalosti textového programování. ModelBuilder je pro ně tedy jediná možnost, jak sestavit automatizovaný postup zpracování dat.

Z hlediska výuky je tento vizuální programovací jazyk dobrou přípravou i pro následující textové programování v Python. Jednak si tvůrce uvědomí logický sled operací, dokáže prakticky sestavit postup - algoritmus. Dalším přínosem z hlediska výuky programování je použití iterátorů, které reprezentují smyčky – opakování v ModelBuilderu. Grafický symbol pro iterátor je oranžový šestiúhelník (obr. 1). ModelBuilder disponuje dvanácti druhy iterátorů. Mezi základní patří iterátor For a While. Iterátor For dokáže iterovat v zadaném rozmezí číselných hodnot se zadaným krokem. To lze např. využít pro zadání různého počtu vstupních bodů pro interpolaci jedné buňky rastrového povrchu. Další iterátory nastavují opakované vyhledání vstupních dat v zadaném adresáři (geodatabázi, datasetu), které následně vstupují do zpracování. Pro výuku programování jsou z ModelBuilder přínosné i další obraty a to je sestavení modelu s parametrickým vstupem, použití vnitřních proměnných a možnost vložení vnořeného modelu. Vnořený model může být chápán jako podprogram.

Výuka ModelBuilder probíhá ve třech lekcích. Postupně jsou probírány jednoduché modely, kde je sestaven jen sled několika operací. Dále je probrán koncept parametrizace modelu, kdy lze při spuštění modelu zadat na vstupu libovolná vstupní data. Následuje použití vnitřních proměnných, které lze velice dobře využít k automatizovanému pojmenování výstupních dat podle názvu vstupních dat, tak aby bylo zřejmé, která data si odpovídají. Několik posledních úloh se týká použití různých druhů iterátorů, které se využívají pro automatizované vyhledání všech vstupních dat. Součástí výuky jsou i domácí úlohy. Ukázalo se jako nezbytné samostatné procvičování nabytých znalostí. Zejména je důležité procvičení techniky sestavování diagramů. Proto byl vytvořen koncept tří domácích úkolů, které jsou zadány v každém výukovém týdnu po realizaci výuky. Domácí úkoly obsahově na sebe

navazují a týkají se generování digitální povrchů. Druhá a třetí úloha částečně staví na výsledcích předchozích úloh. Pro šíření studijních materiálů a dat v předmětu „Programování 2“ je využit katedrální e-learningový portál na technologii Moodle. V tomto portále je použitý Assignment modul pro zadání a odevzdávání domácích úloh. Z hlediska vyučujícího je toto velice užitečný modul. Nastavení termínu odevzdání domácí úlohy má vliv na vypracování a včasné odevzdání domácích úloh (Dobešová 2014a). Motivací je i známkování úloh.

Výuka programování v Python pro ArcGIS

Ve druhé části semestru, která tvoří devět týdnů, je ve výuce probíráno programování (skriptování) v jazyce Python. Jako dobrým pomocníkem se ukázala možnost automatické konverze modelů z ModelBuilder do jazyka Python. Nejprve je tedy lepší sestavit model, ověřit jeho funkčnost na datech a následně jej exportovat do Python (obr. 2). Tyto exportované programy lze použít jako startovací fragmenty kódu pro další rozpracování již v Python. Konvertované programy lze použít i jako inspiraci pro správné nastavení parametrů volaných metod z ArcGIS. Nesprávné nastavení parametrů (jako konstanty nebo proměnné) je někdy důvodem nefunkčnosti programů. Na druhou stranu je někdy třeba exportovaný kód očistit od zbytečných řádků kódu. Textové programování je celkem obtížné pro programátory začátečníky, což dokumentuje i literatura (Robins et al. 1996). Studenti nemají často problém porozumět již napsanému hotovému kódu, ale mají problém sestavit nový programový kód.

V rámci psaní skriptů v Python jsou využitelné i další metody, které nejsou v grafickém prostředí ModelBuilder dostupné, tudíž lze pomocí Python lze napsat pokročilejší skripty. Zde už zkonvertované programy nepomůžou a musí se pokračovat dalším studiem a osvojováním metod geoprocessoru.

Vizuální programování a kognice

Naše Katedra geoinformatiky Univerzity Palackého se ve výzkumu zabývá kognitivními vlastnostmi vizuálních programovacích jazyků. V GIS programech je možnost tvorby programů grafickou cestou vcelku častá. Obdobnou komponentu jako je ModelBuilder v ArcGIS mají i programy ERDAS Imagine, IDRISI, AutoCAD Map 3D a nově i open source QGIS (Dobešová 2014b). Některé tyto komponenty jsou také vyučovány v dalších specializovaných předmětech, např. Workflow Designer pro AutoCAD Map 3D v navazujícím magisterském studiu.

```

# -*- coding: utf-8 -*-
# -----
# 0_IDW.py
# Created on: 2015-01-14 15:14:07.00000
# (generated by ArcGIS/ModelBuilder)
# Description:
# -----

# Import arcpy module
import arcpy

# Check out any necessary licenses
arcpy.CheckOutExtension("spatial")

# Local variables:
InputPoint = "SR"
IDW_Raster = "C:\\Seminar\\idw_raster"

# Process: IDW
arcpy.gp.Idw_sa(InputPoint, "ZV1", IDW_Raster, "16,4882595999986", "2", "VARIABLE 12", "")

```

Obr. 2 Exportovaný program v Python z komponenty ModelBuilder

Na správnou a efektivní kognici vizuálních programů má vliv základní grafická notace, kterou je daná komponenta vybavena. Jednotlivé grafické symboly se navzájem liší tvarem, barvou, textovým popisem, vnitřními ikonami atd. K hodnocení grafické notace lze použít teorii D. Moodyho „Physics of Notation“ (Moody 2009). Tato teorie hodnotí podle devíti principů, kdy klade důraz na sémiotickou čistotu notace, rozlišitelnost grafických symbolů, duální kódování, vizuální expresivitu atd. Cílem je, aby grafická notace vizuálního programovacího jazyka byla *kognitivně efektivní*, tzn. aby čtení, chápání, zpracování a zapamatování informací znázorněných v diagramu bylo co nejrychlejší a bezchybné.

Kognitivní efektivnost diagramů se projeví i při výuce studentů. Protože studenti začínají s tvorbou diagramů a teprve se učí sestavovat funkční diagramy, může mít kvalita notace vliv na správnost a rychlost sestavení funkčních diagramů. My jsme využili znalostí studentů s ModelBuilder a dalšími třemi komponentami pro VPL a otestovali jsme v roce 2014 různé diagramy v laboratoři pro sledování pohybu očí. Vyhodnocení správných a špatných odpovědí, vyhodnocení času odpovědí, trajektorií pohybu očí, délky fixací potvrzuje a doplňuje informace zjištěné hodnocením podle teorie „Physics of Notation“. Získané výsledky se snažíme promítnout i zpětně do výuky. Jedná se např. o návyk opakovaného automatického zarovnávání modelu již v průběhu vzniku modelu (pokud je tato funkce dostupná). Dalším dobrým návykem je zachování jediného směru orientace toku dat v diagramu. Klademe důraz i na dostatečný textový popis prvků podle principu duální notace. Vhodné je do diagramu vkládat i textové komentáře.

Plánujeme pokračovat v tomto výzkumu VPL pro GIS i v následujících rocích.