

---

UNIVERZITA PALACKÉHO V OLOMOUCI  
FAKULTA PŘÍRODOVĚDECKÁ  
KATEDRA GEOINFORMATIKY

PETR PANEC

BAKÁLŘSKÁ PRÁCE

**OVĚŘOVÁNÍ MOŽNOSTÍ UŽIVATELSKÉHO  
ROZŠIŘOVÁNÍ FUNKCIONALITY APLIKACÍ  
ArcGIS**

(The Testing of a User's Approach to the  
Customization of ArcGIS Applications)

Vedoucí práce: ing. Zdena DOBEŠOVÁ

OLOMOUC 2006

---

## **ABSTRAKT**

Předkládaná práce popisuje a porovnává možné postupy vedoucí k rozšiřování funkcionality aplikací ArcGIS 9 Desktop. Autorův uživatelský pohled je postaven na absolvování kurzů vedených autorizovaným instruktorem ESRI a praktických zkušenostech. Praktická část přináší příklady použití jednotlivých přístupů v Útvaru rozvoje hlavního města Prahy. Na základě takto získaných zkušeností jsou pak diskutovány výhody a omezení jednotlivých přístupů. Na základě nich autor navrhuje postup volby adekvátního prostředku.

## **KLÍČOVÁ SLOVA**

ArcGIS, Python, ModelBuilder, Customizace

---

Prohlašuji, že jsem tuto práci vypracoval samostatně pod vedením Ing. Zdeny Dobešové. Všechny použité materiály a zdroje jsou citovány s ohledem na vědeckou etiku, autorská práva a zákony na ochranu duševního vlastnictví. ....V Olomouci dne 25.května 2006.

---

---

Děkuji vedoucí práce Ing. Zdeně Dobešové za podněty a připomínky. Za poskytnuté materiály, konzultace a školení děkuji firmě ARCDATA Praha s.r.o.

V neposlední řadě bych rád poděkoval vedení Útvaru rozvoje hlavního města Prahy za důvěru a podporu, bez které by nebylo možné práci dokončit v plném rozsahu.

---

---

## Obsah

|                                                        |    |
|--------------------------------------------------------|----|
| 1. Úvod                                                | 7  |
| 2. Cíl práce                                           | 8  |
| 3. Použité metody a postupy                            | 9  |
| 4. Customlzing v ArcGIS                                | 11 |
| 4.1. Uživatelské přizpůsobování funkcionality          |    |
| 4.2. Základy ArcGIS 9.x Desktop                        |    |
| 4.3. Možnosti rozšiřování funkcionality v rámci ArcGIS |    |
| 5. Převzetí funkcionality                              | 16 |
| 5.1. Zdrojové oblasti                                  |    |
| 5.2. Implementace a formáty                            |    |
| 6. Model Builder                                       | 20 |
| 6.1. Proč použít Model Builde                          |    |
| 6.2. Prostředí a ovládán                               |    |
| 6.3. Možnosti použití                                  |    |
| 7. Skriptování v ArcGIS                                | 25 |
| 7.2 Proč Python?                                       |    |
| 7.3. Geoprocessor                                      |    |
| 7.4. Vývojové prostředí                                |    |
| 7.5. Použití                                           |    |
| 8.Praktické ověřování                                  | 32 |
| 8.1. Organizace vybraná k ověřování                    |    |
| 8.2. Přejímání funkcionality                           |    |
| 8.3. Model Builder                                     |    |
| 8.4. Python                                            |    |
| 9. Poznatky a hodnocení                                | 39 |
| 9.1. Přejímání funkcionality                           |    |
| 9.2. ModelBuilder                                      |    |
| 9.3. Python                                            |    |
| 9.4. Volba adekvátního prostředku                      |    |
| 10. Diskuse                                            | 46 |
| 11. Závěr                                              | 47 |
| 12. Literatura                                         | 48 |
| 13. Resume                                             | 50 |

---

## 1. ÚVOD

Tématem uživatelského rozšiřování funkcionality ArcGIS jsem se rozhodl zabývat z několika důvodů. Prvním z nich je rozšíření ArcGIS, které představuje jeden z nejpoužívanějších desktopových řešení pro GIS. Přesto, a možná právě proto, jeho uživatelé narážejí na absenci specifických funkcí, které k práci ve svém oboru potřebují. Podobně jako u jiných systémů je i v ArcGIS celá řada možností jak systém dále rozšiřovat nákupem dalších dílčích aplikací - extenzí. Ovšem ani tato možnost nemůže zcela pokrýt specifické potřeby všech uživatelů, a proto jsou pro ně připraveny různé způsoby, jakými si mohou funkcionalitu vlastní instalace rozšiřovat.

---

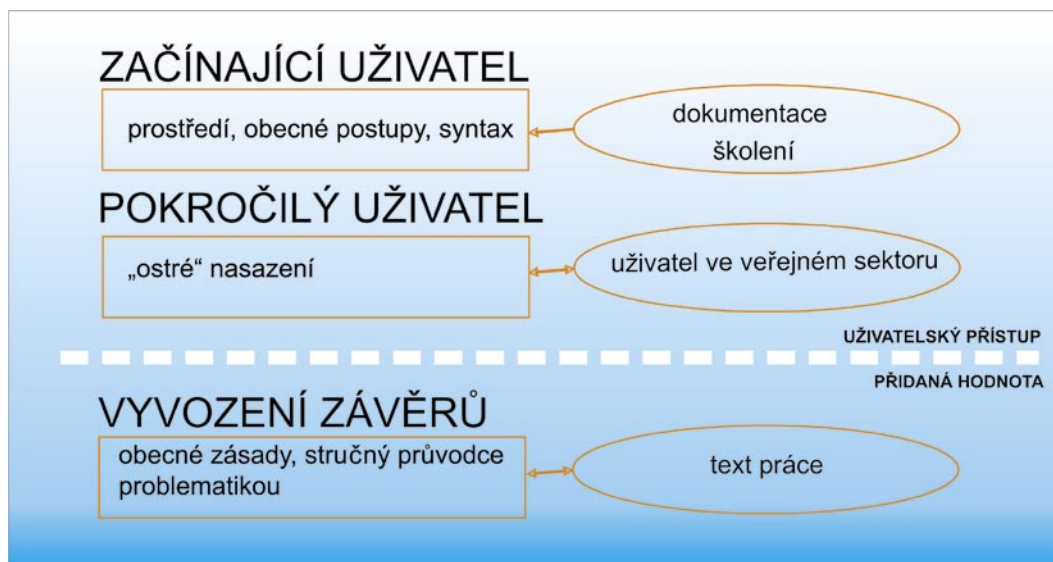
## 2. CÍL PRÁCE

Cílem práce je popsat možnosti uživatelského rozšiřování funkcionality. Snažím se připravit příklady, na kterých lze vyzkoušet možnosti uživatelského rozšiřování funkcionality a demonstrovat základní kroky nutné k této činnosti. Prvním z dílčích cílů práce je vymezení možností, jakými lze funkcionalitu rozšiřovat a popsat uživatelské přístupy, kterými toho dosáhnout. Pro tyto typové úlohy bude vypracována dokumentace, která umožní jejich další zopakování.

Na základě konkrétních požadavků uživatelů se pokouším vytvořit vlastní rozšíření funkčnosti a následně testuji výsledek. Dílčím cílem je otestovat náročnost takovýchto postupů i z hlediska časové zátěže a nároků na zdatnost uživatele. Zhodnocení složitosti implementace těchto postupů do každodenní praxe a možností využití technické podpory ze strany producenta, resp. jeho autorizovaného distributora, bude taktéž jedním z dalších výstupů práce.

### 3. POUŽITÉ METODY A POSTUPY

Z hlediska volby metod a postupu je zásadní především stanovení vlastností ArcGIS, které budou následně ověřovány. Ty jsou vymezeny v následující části práce. Postup zpracování práce pak reflektuje běžný uživatelský přístup k ověřování vlastností SW (viz Obr. 1.).



Obr.1. Aplikace uživatelského přístupu při zpracování problematiky

Základní popis potřebných postupů je již doplněn na základě praktických zkušeností získaných při praktickém ověřování. Závěrečné shrnutí a diskuse kriticky porovnává výhody a nevýhody jednotlivých přístupů.

U většiny absolventských prací na vysokých školách je zvykem pracovat s vědeckou literaturou. V tomto případě jsem však z větší míry odkázán na odbornou literaturu a materiály vydávané producentem daného SW. Ty, přes nepopiratelnou odbornou úroveň a výborné didaktické zpracování, mají svoje pevné místo v marketingové strategii firmy a je nutno k nim přistupovat s kritickým vědomím jejich omezené objektivity. Proto jsou i takto získané poznatky podrobeny v průběhu práce konfrontaci s praktickými zkušenostmi. Z formálního hlediska jsou v práci zachovány citace jednotlivých autorů.

Problematickou stránkou práce, která může vzbudit určité kontroverze, je použitá terminologie. Zkoumané SW prostředí je lokalizováno do českého prostředí, ale vzhledem k nutnosti pracovat s anglicky psanými zdroji

---

jsem zvolil jeho nelokalizovanou verzi. To se odráží i na používané terminologii. Je-li výklad nejednoznačný, snažím se uvést oba názvy - český i anglický. Zároveň dodržuji anglická pojmenování jednotlivých nástrojů, aplikací a modulů, které jsou však pro potřeby česky psaného textu skloňovány. Nedomnívám se, že by takovéto zacházení mohlo zásadním způsobem snížit použitelnost práce, protože většina současných uživatelů, kterým je tato práce určena, je na tento přístup zvyklá.

Seznámení s prostředím a základní funkcionalitou ArcGIS bylo dosaženo prostřednictvím tvorby mapových výstupů doplňujících seminární práce, které jsem vypracoval v rámci několika geografických předmětů. Další potřebné informace o pokročilé funkcionalitě jsem získal konzultacemi na odborné praxi, školeními a školícími materiály poskytnutými firmou ARCDATA Praha s.r.o. Tímto způsobem jsem měl zajištěn poměrně rychlý a mnohem snadnější průnik do dané problematiky.

Protože jedním ze základních cílů této práce je praktické ověřování, musel jsem najít organizaci, která používá ArcGIS a měla by zájem o rozšíření jeho funkcionality. Rozsah a hloubka praktického ověření jednotlivých přístupů tak spočívá především na požadavcích takovéto organizace.

---

## 4. CUSTOMIZING V ARCGIS

### 4.1. UŽIVATELSKÉ PŘIZPŮSOBOVÁNÍ FUNKCIONALITY

Už naprosto běžný textový editor jako Word umožňuje uživatelské přizpůsobení pomocí maker. Makro je akce nebo posloupnost akcí, jež lze použít pro automatizaci úloh. Značná část SW produktů dnešní doby představuje otevřené modulární systémy vyvinuté prostřednictvím objektově orientovaného programování. Takovéto systémy lze rozšiřovat právě pomocí modulů využívajících odpovídající třídy objektů nebo prostřednictvím skriptovacích jazyků.

Z hlediska uživatele je možných několik postupů při rozšiřování funkcionality.

- 1) Vložením makra nebo připojení modulu jiného autora:
  - a) Vytvořené autory programu a připravené pro uživatele k použití.
  - b) Vytvořené jinými autory a šířené například pomocí internetu.
- 2) Tvorba vlastního makra:
  - a) Pomocí nástroje integrovaného v daném SW produktu, který zaznamená souslednost kroků (např. Photoshop).
  - b) Pomocí programového kódu tvořeného v adekvátním vývojovém prostředí (skriptovací jazyky).
- 3) Tvorba modulů:
  - a) Pomocí integrovaného vývojového prostředí (VBA – VisualBasic for Application).
  - b) Pomocí jakéhokoliv vývojového prostředí.

Vztáhneme-li předchozí strukturu na prostředí ArcGIS, pak máme definovány základní případy, které budou následně popsány (včetně příkladů) a bude ověřena jejich použitelnost. Zde je vhodné přistoupit na terminologii používanou firmou ESRI a seznámit se s některými specifiky aplikace ArcGIS, která je oproti textovému editoru přeci jen o něco složitější a komplikovanější.

---

## 4.2. ZÁKLADY ARCGIS 9.X DESKTOP

Z pohledu autorů tohoto systému se v podstatě jedná o výkonného desktopového klienta v rámci kompletního řešení geografického informačního systému ArcGIS 9. ArcGIS Desktop je distribuován ve třech licenčních úrovních ArcView™, ArcEditor™ a ArcInfo™. Všechny aplikace tohoto systému byly vyvinuty v objektovém programovacím prostředí a jednotlivé komponenty představují skupinu patentovaných COM objektů, která nese název ArcObjects™. Ta je společná nejen pro tyto tři produkty, ale například i pro ArcReader™ nebo pro libovolné desktop klienty vytvořené z ArcObjects™ za pomoci vývojářského balíku ArcEngine™ či síťové klienty vytvořené za podpory ArcGIS Serveru™.

### A. Licenční úrovně

Všechny tři licenční úrovně ArcGIS Desktop mají stejné jádro základních funkcí pro prezentaci, editaci a analýzy prostorových dat. Chovají se v zásadě stejně i z pohledu rozšiřování funkcionality. Jednotlivé licenční úrovně obsahují dvě základní aplikace ArcMap™ a ArcCatalog™ (v kombinaci s licencí na 3D Analyst™ je to pak navíc ArcScene™ a ArcGlobe™).

### B. Funkcionalita a geoprocessing

Obě základní aplikace ArcMap™ a ArcCatalog™ mají rovnocenný přístup ke geoprocessingové funkcionalitě, která je centralizována v ArcToolboxu. Ten, jak z překladu vyplývá, představuje kontejner nástrojů. Z pohledu uživatele je ArcToolbox ukotvitelné okno, které je možné umístit v rámci jakékoliv aplikace ArcGIS Desktop. Pro větší přehlednost i s ohledem na problematiku vývoje systému je ArcToolBox dále hierarchicky členěn na Toolboxy a ToolSety. Nejmenšími stavebními jednotkami funkcionality jsou jednotlivé nástroje ve formě: systémový nástroj (Tool), skript (Script), model (Model). Základními toolboxy jsou: Analýzy (Analysis Tools), Kartografie (Cartography Tools), Konverze (Conversion Tools), Coverage, Správa dat (Data Management Tools), Geokódování (Geocoding Tools), Lineární referencování (Linear Referencing Tools), Prostorová statistika (Spatial Statistics Tools).

---

Počet nástrojů v licenčních úrovních a extenzích (Lindeman; Page; 2004):

- ArcView (102)
- ArcEditor (104)
- ArcInfo (251)
- + Spatial Analyst extension (158)
- + 3D Analyst extension (45)
- + Geostatistical Analyst extension (1)

Další část funkcionality představují jednotlivé moduly zastoupené v podobě Toolbars. Ty jsou určeny především k činnostem vázaným na interakci uživatele s grafickým prostředím. Klasickým příkladem takových činností jsou různé druhy editace a tvorba dat. Toolbars představují také jeden z prostředků customizace v podobě extenzí a modulů.

### C. Formáty

Aby základní informace o daném produktu byly kompletní, doplňuji ještě výčet datových typů podporovaných systémovými nástroji: rastrové soubory, CAD soubory (\*.dxf, \*.dgn), shapefiles (\*.shp), coverage, geodatabáze, tabulky, lyr (lyr) a nastavení prostředí aplikace.

Není cílem práce popisovat daný SW. Pochopení základních principů práce v dané aplikaci se předpokládá. Další informace lze získat z dokumentace obsažené v instalaci a ze školicích materiálů vydávaných firmou ESRI. Je možné obrátit se také na technickou podporu zajišťovanou producentem na [www.esri.support.com](http://www.esri.support.com) nebo v českém jazyce na stránkách českého distributora [www.arcdata.cz](http://www.arcdata.cz).

### 4.3. MOŽNOSTI ROZŠIŘOVÁNÍ FUNKCIONALITY V RÁMCI ArcGIS

V následujících odstavcích se pokusím popsat aplikaci výše uvedených obecných přístupů customizace do prostředí ArcGIS. Z pohledu rozšiřování funkcionality lze vyzdvihnout, že ESRI ve vztahu k uživatelům rozlišuje dvě úrovně. Tyto dvě úrovně jsou vůči sobě vymezeny především na základě struktury marketingové komunikace. Pro komunikaci s klientelou

---

z řad vývojářů byl v nedávné době zřízen nový komunikační kanál - ESRI Development Network (EDN) přinášející některé nadstandardní služby, přístup k dokumentaci apod. Na základě prostudování nabídky EDN i standardního ESRI support jsem se pokusil o popsání obou úrovní a jejich přístupu k dané problematice.

#### A. „Uživatel“ nebo „Vývojář“

Uživatelskou úroveň můžeme vymezit pohledem uživatele, který využívá k rozšiřování funkcionality maximálně skriptovací nástroje. Programátorské zkušenosti takového uživatele odpovídají základům programování a algoritmizace. Snahou uživatele je většinou ušetřit si práci při komplikovaných a často opakovaných činnostech. Jeho snahy vedou k jejich automatizaci či dávkovému zpracování na základě využití stávajících funkcí v požadovaném sledu. Cílem takového uživatele může být i potřeba umožnit použití jím dosažených postupů a kombinací funkcí jiným uživatelům pro zefektivnění práce celého kolektivu.

Vývojářskou úroveň lze definovat pohledem specializovaných organizací, pracovišť či pracovníků zaměřených na přizpůsobení aplikace ArcGIS. Jejich cílem může být integrace ArcGIS do jiných IS nebo vývoj samostatných GIS aplikací na stejném základě COM objektu ArcObjects. Klasickým prostředkem je některé ze standardních vývojových prostředí (C++ apod).

Z definice cílů práce vyplývá, že se budu v dalších částech práce snažit věnovat problematice pouze na úrovni „uživatelského“ pohledu.

#### B. Prostředky rozšiřování funkcionality

Jednotlivé prostředky sloužící k rozšiřování funkcionality můžeme členit podle dvou hledisek:

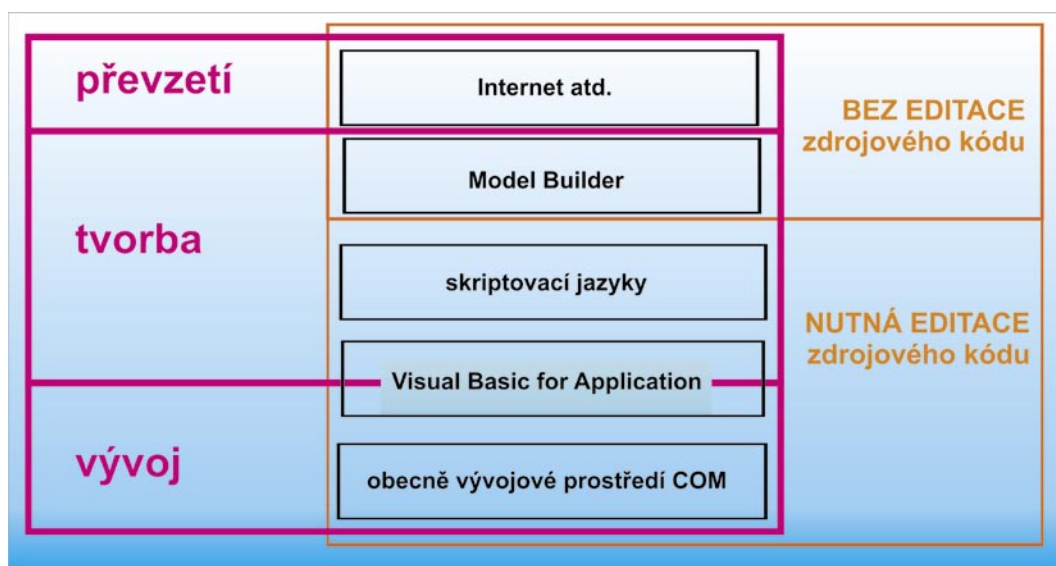
1. Dle nutnosti editovat programový kód (Lindemann; Page 2004):
  - bez nutnosti programování
  - s programováním

2. Na základě mentální (a tím i časové) náročnosti:

- převzetí
- tvorba
- vývoj (mimo vymezený rámec práce)

Právě převzetí funkcionality je způsobem rozšíření funkcionality, který není v rámci materiálů producenta adekvátně zpracován. Tento způsob považuji za základní a nejčastěji aplikovaný úvod do problematiky.

Jednotlivé prostředky k rozšíření funkcionality, které ArcGIS nabízí, jsou uvedeny v následujícím schématu. Jelikož se práce zaměřuje na ověření uživatelských přístupů, rozhodl jsem se podrobit bližšímu zkoumání pouze první tři prostředky z uvedeného přehledu.



Obr.2. Přehled prostředků použitelných pro rozšíření funkcionality jednotlivé prostředky (černá), začlenění dle uživatelské náročnosti (fialová) dle nutnosti editace programového kódu (oranžová)

---

## 5. PŘEVZETÍ FUNKCIONALITY

Tato cesta k rozšíření funkcionality je důležitým prostředkem nejen ke snadnému získání nové funkcionality, ale je také základním mechanismem jak takovou funkcionalitu distribuovat. Tyto postupy tedy musí zvládnout i ten, kdo se chce věnovat pouze tvorbě nebo vývoji vlastních funkcí. Je opět možné vymyslet několik různých členění tohoto způsobu. Já jsem se rozhodl uvést a postupovat podle dvou z nich.

### 5.1. ZDROJOVÉ OBLASTI

Jedno z možných členění je na čtyři nejčastější zdrojové oblasti, ze kterých lze rozšíření funkcionality ArcGIS převzít. Samozřejmě důležitým faktorem jsou i licenční podmínky a dostupnost:

#### A. Doplnkové součásti instalace (samples)

V případě instalace zahrnující balík DeveloperKit je najdeme na svém disku v `C:\Program Files\ESRI\DeveloperKit\samples`. Jinak lze nalézt DeveloperKit na instalačním CD nebo stránkách výrobce. Jednotlivé podadresáře obsahují zazipované soubory s dalšími rozšířeními funkcionality včetně zdrojových souborů (většinou i pro více vývojových prostředí). Řada z těchto utilit pro ArcGIS může vyřešit některý z uživatelských problémů, stejně jako zdrojové soubory mohou sloužit jako inspirace nebo podklad pro další vývoj vlastní funkcionality. Jedná se sice o zaplacenou přidanou funkcionalitu, ovšem podle mé zkušenosti ne vždy využívanou.

#### B. Funkce šířené uvnitř pracovní skupiny

U většiny organizací nebo řešitelských týmů dochází k dělbě práce, při které je potřeba umožnit ostatním uživatelům použití shodných nástrojů nebo postupů. V takovéto oblasti použití lze například šířit skripty a modely s absolutně nastavenými cestami k úložištím apod. Oblast pracovních skupin a kolektivů je také často prostředníkem v šíření potřebné funkcionality pocházející z některé z ostatních zdrojových oblastí mezi další uživatele.

---

### C. Volně šířená funkcionální (internet)

V dnešní době nevýčerpateľná zásobárna. Základním zdrojem je <http://support.esri.com> s databází skriptů vytvořených různými uživateli i vývojáři ESRI. Samozřejmostí v případě registrace uživatele na ESRI Support je možnost podílet se na rozvoji této databáze vkládáním vlastních skriptů a jejich aktualizací. (Při stahování skriptu je uživatel vyzván k přijetí licenčního ujednání). Stránky distributorů ESRI včetně českého, představují další hodnotný zdroj. Skripty, extenze a různé utility jsou dostupné i z jiných stránek než jen oficiálních zastoupení ESRI. Většinou to jsou sami autoři nebo jejich zaměstnavatelé, kteří na svých stránkách umožňují volné využití jimi vytvořené funkcionality. Příkladem může být extenze GeoWizzard ([www.ian-ko.com](http://www.ian-ko.com)).

## 5.2. IMPLEMENTACE A FORMÁTY

Nezávisle na širokých možnostech výše uváděných zdrojů je v podstatě velmi málo postupů jak tuto funkcionality převzít, respektive implementovat do konkrétní instalace ArcGIS Desktop. Tyto postupy jsou úzce spojeny s formátem v němž se k uživateli daná funkcionality dostane.

### A. Funkcionality v nativních formátech

Velká část funkcionality se uplatňuje prostřednictvím ArcToolboxu – jednotlivé toolboxy jsou pak uloženy ve formátu .tbx. Systémové, tedy needitovatelné toolboxy, jsou uloženy v případě standardní instalace v `c:\Program Files\ESRI\ArcGIS\ArcToolbox\Toolboxes\` personální, tedy editovatelné, jsou zpravidla umístovány do `c:\Documents and Settings\user\Data_aplikaci\ESRI\ArcToolbox\MyToolboxes\`. Toolbox v podobě souboru .tbx může být uložen kdekoliv v rámci adresářové struktury ArcCatalogu. Stejně tak lze toolbox učinit součástí personální geodatabáze. Jako její součást může být i distribuován. Distribuce modelů, jejichž tvorba je probírána v jiné části práce, není jinak než v rámci Toolboxu možná. Přidání do ArcToolboxu je otázkou několika vteřin (postup: kontextové menu `ArcToolbox>Add>Toolbox`).

---

## B. Dynamické knihovny, spustitelné soubory a aktivní ovládací prvky

Dynamické knihovny (\*.dll), spustitelné soubory (\*.exe) a aktivní ovládací prvky (\*.ocx) představují samostatné COM objekty. Jejich integrace do prostředí ArcGIS Desktop spočívá v jejich registraci. Některé nástroje jsou vybaveny instalčním souborem, který registraci zajistí sám. V ostatních případech jsou možné dvě varianty (v závislosti na požadovaném výsledku). Má-li být výsledek v podobě Toolbars, pak stačí vybrat z menu **Tools>Customize>Add from file**. Druhou možností je integrovat funkci jako nástroj do ArcToolboxu z kontextového menu personálního Toolboxu>Add Tool>Add from file.

## C. Zdrojový kód skriptu

V případě takového zdroje jsou možná tři následující řešení: spustit získaný skript z odpovídajícího skriptovacího prostředí či z příkazové řádky a třetí možností je spustit daný skript z ArcToolboxu.

První dvě možnosti jsou v podstatě rovnocenné, pokud skript obsahuje parametry, jsou zadávány textově ve stejném formátu.

Spuštění skriptu z ArcToolboxu vyžaduje nejdříve jeho přidání do některého z personálních toolboxů. Přidání samotného skriptu je podobně náročné jako přidání nového toolboxu a z pohledu uživatele představuje takový skript téměř nativní formát. Další nastavení v rámci Script Properties je již poněkud složitější. Výhodou tohoto postupu je ovšem snazší použitelnost pro méně zdatné uživatele a pro pokročilé uživatele zase možnost použití takto integrovaného skriptu jako komponentu při dalším rozšiřování funkcionality.

Z kontextového menu vybraného (samozřejmě personálního) Toolboxu:

Add>Script> [Name: <sup>1</sup>/Label: <sup>2</sup>/ Description: <sup>3</sup>/Stylesheet: <sup>4</sup>/Store relative path names (instead of absolute paths)<sup>5</sup> ] >Next>[ScriptFile: <sup>6</sup> / Show command window when executing skript: <sup>7</sup>]>Next>[Parametrs<sup>8</sup>]

1 jméno skriptu z pohledu aplikace (pozor na vyhrazené znaky a diakritiku)

2 jméno nástroje v Toolboxu (popisek - zde lze uplatnit diakritiku)

3 jednoduché vysvětlení činnosti skriptu

4 cesta k .xsl souboru představující upravený vzhled dialogu (editovatelné v

---

Notepadu)

5 toto nastavení ovlivní zápis cest mezi toolboxem a zdrojovým kódem a mezi toolboxem a .xsl

6 cesta k samotnému skriptu

7 sledování průběhu skriptu z příkazové řádky Windows

8 nastavení parametrů skriptu (nutné pochopení zdrojového kódu)

Display Name – pojmenování vstupu pro potřeby zapouzdřeného „inputboxu“ (jméno parametru může být jakékoliv včetně diakritiky a slouží pouze jako titulek v dialogu pro zadání parametrů, pořadí parametrů musí být zachováno)

Data Type – datový typ musí být zachován stejný jako ve skriptu

- s ohledem na skript musí být nastaveny i další vlastnosti parametru Parametr Properties:

Type (Required, Optional) - nutný nebo volitelný parametr

Direction (Output, Input) - vstup nebo výstup z procesů obsažených ve skriptu

MultiValue (Yes, No) – může nebo nemůže nabývat více hodnot

Default – defaultní hodnota

Environment – po dobu běhu skriptu změni hodnotu nastavení konkrétní vlastnosti prostředí

Domain – omezení rozsahu hodnot parametru

Dependency – závislost daného parametru na jiném vstupním parametru

Z pohledu metod integrace získané funkcionality a její distribuce je určitě uživatelsky nejméně náročné využívat nativních formátů \*.tbx a geodatbáze. Jejich jediným problémem je napojení na zdrojový kód skriptu. Cestu lze ale poměrně dobře opravit.

V závěru popisovaný proces přejímání zdrojového kódu nám logicky umožňuje mnohem efektivněji zasahovat do podoby získané funkcionality.

---

## 6. MODEL BUILDER

Z předchozího zařazení Model Builderu (MB) mezi prostředky rozšiřování funkcionality nevyžadující editaci programového kódu je patrné, že tento způsob plně využívá možnosti interakce uživatele s grafickým prostředím. To uživateli umožňuje velmi snadno zapojovat jednotlivé nástroje Toolboxu do složitějších procesů. Podobné interaktivní a intuitivní prostředí již bylo přístupné pro uživatele extenze Spatial Analyst u předchozích verzí ArcGIS a ArcView GIS. Modelovací nástroj s grafickým rozhraním je i jednou ze součástí IDRISI. Novinkou od verze ArcGIS 9.0 je přístupnost takového prostředí uživatelům ve všech licenčních úrovních. Rozdíly v licenční úrovni se však projeví v počtu použitelných nástrojů (viz 4.2.2. Funkcionality a geoprocessing) nebo na funkcionalitě modelu při změně licenční úrovně (aktuální v případě plovoucích licencí nebo distribuce modelu).

### 6.1. PROČ POUŽÍT MODEL BUILDER

K použití právě tohoto prostředku může mít uživatel celou řadu důvodů. Ty, dle mínění ESRI hlavní, uvádějí ve své knize Lindemann a Page (2004).

- 1) Rychlé analýzy a dotazy.
- 2) Grafická dokumentace postupu.
- 3) Možnost snadno měnit parametry ve známém grafickém rozhraní, opakovaně spouštět model, sledovat průchod modelem a porovnávat výsledky.
- 4) Komplexní zachycení postupu.

Ve snaze představit Model Builder jako nenahraditelný nástroj GIS dokonce vymezují tři základní typy modelů (typových situací), kdy lze tyto výhody uplatnit (upraveno dle Lindemann; Page, 2004).

- 1) Opakované úlohy – minimalizace základní práce, efektivní spuštění série často opakovaných nástrojů.
- 2) Modely pro určení vhodné lokality (Suitability models) – možnost měnit váhy a přidávat další rozhodovací faktory.
- 3) Procesní model – slouží k demonstraci průběhu určitého procesu.

---

To, zda jsou tato doporučení adekvátní, je řešeno na jiném místě práce. Protože jednou ze zásadních výhod, kterou od tohoto prostředí uživatel čeká, má být názornost a interakce s grafickým prostředím, soustředím se v následujících odstavcích na popis prostředí a ovládání tohoto prostředku.

## **6.2. PROSTŘEDÍ A OVLÁDÁNÍ**

### **A. Aktivace prostředí**

Celá řada modelů je již obsažena v systémových Toolboxech. Jejich status je však read-only a jejich editace je tedy nemožná. Nový model lze přiřadit pouze do nesystémového, tedy personálního Toolboxu (tvorba vlastního Toolboxu: `ArcToolBox>Add Toolbox>New Toolbox` viz předchozí oddíl práce). Nový model a tím i prostředí MB aktivujeme pomocí sekvence kliků `ToolBox>New>Model`. Editaci stávajícího nebo převzatého modelu zahájíme z kontextového menu modelu `Model>Edit`.

### **B. Grafické rozhraní**

Celé okno Model Builderu se nám objeví až po aktivaci editace modelu. Základními ovládacími prvky jsou panel nástrojů a nástrojová lišta. Hlavní menu je velmi jednoduché a je určeno k ovládání prostředí MB. Sedmnácti tlačítková nástrojová lišta slouží především k interakci s elementy modelu, k pohybu v modelu a k jeho zobrazení. Ovládání je intuitivní (včetně označení tlačítek). Prvky modelu se zobrazují ve zbývajících třech částech grafického rozhraní, která je v angličtině nazývána „Display Window“. Toto okno slouží k samotné kompletaci modelu a je místem, kde jsou aplikovány všechny drag and drop akce uživatele.

### **C. Prvky modelu (Model Elements)**

Základními elementy modelu jsou procesy (Process). V případě velmi jednoduchého modelu může tvořit pouze jeden proces. Častěji se však setkáme s modely, které tvoří více procesů. Ty na sebe vzájemně navazují tak, že se překrývají ve vstupních a výstupních datech, které tvoří mezivýsledky. Složkami procesu jsou vstupní data (Input Data), nástroje či

---

funkce (Tool) a odvozená data (Derived Data). Pokud jsou odvozená data vstupními daty pro další procesy (jak je naznačeno výše u překrývání procesů), pak tvoří mezivýsledek, u nějž lze nastavit vlastnost Intermediate Data (toto nastavení zabezpečí jejich vymazání po skončení průchodu modelem). Stavy jednotlivých procesů jsou graficky rozlišeny do tří úrovní neschopen spuštění, schopen spuštění, proces proběhl. Poslední zmiňovaný stav se po změně jakéhokoliv nastavení vrátí do jednoho z předchozích. Mezivýsledky lze přepisovat, je-li tato možnost potvrzena i v nastavení prostředí celé aplikace.

#### D. Přidání prvků do modelu

Nástroje jsou nejsnáze přidávány metodou drag and drop z ArcToolboxu. Data jsou přidávána dvěma způsoby. Primárně opět metodou drag and drop z ArcCatalogu nebo ArcMapu. Druhým způsobem je vložení prázdné proměnné, která nahrazuje budoucí datový zdroj (**Create Variable...**). Všechny prvky také mohou být vkládány přes kopírování a vložení z jiného nebo stejného modelu. V případě kopírování procesu v rámci stejného modelu si obsažené nástroje pamatují původní nastavení včetně vstupní vrstvy.

#### E. Proměnné v modelu

Hlavní výhodou používání proměnných v modelu je možnost jeho zobecnění a přístupnost všech potřebných nastavení prostřednictvím grafického rozhraní podobného jiným nástrojům v systémových toolboxech.

Veškerá základní nastavení je možné provést přes dialogová okna jednotlivých nástrojů. Zároveň lze z některých nastavení nástroje vytvořit proměnnou. Z jakýchkoliv vstupních dat můžeme vytvořit parametr modelu. To, že daný prvek modelu bude parametrem nebo bude obsahovat proměnnou, je ovládáno z kontextového menu v editačním prostředí.

- nastavení parametru modelu: kontextové menu objektu
  - > **Model Parametr**
- nastavení proměnné u nástroje: kontextové menu nástroje
  - > **Make Variable > From Parametr** nebo **Make Variable > From Settings**

---

Vlastnosti jednotlivých parametrů a proměnných modelu jsou vzhledem k vazbě na konkrétní nástroje nastaveny. Proto odpadá složité nastavování vlastností proměnných. Problémem diskutovaným v závěru práce jsou omezení proměnných z parametru u některých nástrojů. Jedním z možných nastavení v rámci Vlastností modelu (Model Properties) je uložení relativních cest k datům i použitým nástrojům. Tato volba řeší při dodržení zažitých pravidel pro relativní cesty problémy při přenosu modelů mezi uživateli v podobě Toolboxu uloženého v geodatabázi.

#### F. Další možnosti prostředí Model Builderu

- 1) Grafické vlastnosti diagramu: zobrazení mřížky pro zarovnání, změna orientace schématu, změna barvy a tvaru prvků modelu a změna fontu názvu prvků.
- 2) Popisování a komentář modelu: volně plovoucí popisky, popisky prvků, pojítka mezi popiskem a prvkem včetně nastavení orientace popisku vzhledem ke směru průchodu modelem.
- 3) Export do grafiky: podporované formáty JPG, BMP, EMF.
- 4) Vzhled a tisk: možnost volit ohraničení modelu, titulek, velikost tiskové strany, počet stran a jejich číslování.
- 5) Možnost přizpůsobit dialogy nástrojů: změna viditelnosti, seskupování parametrů, změna grafického rozhraní formou linků a obrázků (editace .xsl v Notepadu).
- 6) Podpora editace dokumentace: možnost vlastní editace nápovědy pro jednotlivé nástroje.
- 7) Model Report: metainformační XML dokument s údaji o času spuštění, délce průběhu modelu, proměnných, procesech a hlášeníh. Jedná se o stejné informace jako ve stavové řádce při průchodu modelem (výhodou je možnost jejich archivace a následné analýzy). Informace o některých použitých nástrojích se ukládá automaticky i do metadat.

---

### 6.3. MOŽNOSTI POUŽITÍ

Na základě předchozích řádků je možné si vytvořit představu o použitelnosti modelů. Obecně mohou vzniknout dvě varianty modelů:

- 1) Pevné modely – modely postavené nad konkrétními vstupními daty a tvořící konkrétní výstupy na základě pevně nastavených parametrů.
- 2) Modely s proměnnými – takovéto modely mohou obsahovat konečný počet proměnných zahrnujících datové proměnné, proměnné parametry či nastavení prostředí.

Přínos použití prvního typu modelu vidím spíše v oblasti dokumentace procesů, což má v praxi velký význam. První typ modelů umožňuje opakované spuštění s inovovaným nastavením parametrů použitých nástrojů, ale v případě potřeby výměny vstupních dat jsme vázáni řadou těžko kontrolovaných podmínek (formát dat, jejich geometrie, struktura atributů apod.)

V modelech druhého typu spatřuji hlavní šance pro nasazení v praxi, které bude představovat opravdové rozšíření funkcionality, respektive zefektivnění jejího využívání. Paradoxně takovým způsobem může dojít i k omezení parametrů některého nástroje.

---

## 7. SKRIPTOVÁNÍ V ARCGIS

Protože celá řada geoprocessingových úloh je poměrně časově náročná, zvláště uvážíme-li jejich opakované spuštění nad velkou množinou vstupů, je pochopitelné, že při vývoji systému pro profesionální použití je žádoucí umožnit uživateli zautomatizování některých činností nebo jejich dávkové spuštění. Právě k tomu jsou určeny skriptovací nástroje podle autorů samotného SW: „skriptování je efektivní metoda automatizace geoprocessingových operací“ (Tucker, 2004). ArcGIS 9.x podporuje použití většiny dnešních oblíbených skriptovacích prostředí Python, VBScript, JScript a Perl. To je zajištěno COM strukturou aplikace. Oproti starším produktům ESRI tak odpadá nutnost učit se skriptovací jazyk určený pouze pro prostředí jedné aplikace (Avenue a AML). Stěžejním objektem z ArcObjects je pro skriptování „geoprocessor“, který řídí veškerou geoprocessingovou funkcionalitu přístupnou ze standardního prostředí ArcGIS přes ArcToolbox. Tento objekt spolupracuje s IGPDispatch rozhraním pro automatickou interpretaci textových instrukcí skriptovacích jazyků. Geoprocessor umožňuje využít celý rozsah geoprocessingové funkcionality. Ta může mít při použití všech extenzí a nejvyšší licenční úrovni rozsah větší než 400 dostupných funkcí (viz kapitola 4.2.2. Funkcionalita a geoprocessing). Kolik specifických funkcí můžeme vytvořit jejich kombinacemi je zřejmě pouze hypotetická otázka, každopádně to představuje velký prostor pro tvůrčí řešení celé řady problémů, se kterými se v praxi setkáváme.

„Skriptovací ArcObjects nenahrazují standardní ArcObjecty“ (Page; Batsukh, 2004). Limitujícím faktorem je „negrafičnost“ skriptování. Nelze jím měnit ani jinak využívat grafické uživatelské prostředí (GUI). Jedinou výjimku tvoří automaticky generovaný Inputbox, který je však vázaný na integraci skriptu do Toolboxu (viz kapitola 5.2.). Tím se nám však nabízí možnost využít vestavěné nástroje k editaci nápovědy. Prostředkem pro využití ArcObjects je programovací prostředí Visual Basic for Applications.

### 7.1. PROČ PSÁT SKRIPTY?

Na tuto otázku po smyslu používání skriptů odpovídají Lindeman a Page (2004) takto:

- 
- 1) Podobné výhody jako použití modelu.
  - 2) Podporuje dávkové zpracování.
  - 3) Použití logické kontroly nad výkonem nástrojů.
  - 4) Prostředí pro interakci s jinými aplikacemi.
  - 5) Samoobsažnost (skript představuje jeden soubor).
  - 6) Spustitelný kdykoliv.
  - 7) Příbuznost prostředí s AML a Avenue, které jsou uživatelům známé.
  - 8) Uživatel se již nemusí učit proprietární jazyky.

Otázka podobnosti či rozdílnosti výhod skriptů a modelů je jednou z hlavních otázek práce a je diskutována v jejím závěru. Dávkové zpracování a plánování úloh jsou stěžejní výhody při každodenním souboji s časem. Možnost interakce s jinými aplikacemi se nemusí omezovat pouze na předávání textových informací. Přesto je vždy otázkou znalost druhé aplikace a jejích COM objektů. Body 7) a 8) již byly částečně diskutovány.

## **7.2 PROČ PYTHON?**

Proč je z výše uvedených skriptovacích jazyků firmou ESRI preferován právě Python? Visual Basic Script i Java Script a jejich znalosti jsou mezi uživateli velmi frekventované a jejich osvojení je poměrně snadné. Co se týká osvojitelnosti Pythona, není v náročnosti pravděpodobně žádný zásadní rozdíl. JScript a VBScript jsou podobně jako programovací jazyky C a Visual Basic určeny k operacím v prostředí operačního systému Windows. Jako jednu z výhod Pythonu uvádějí Page a Batsukh (2004) jeho nezávislost na platformě operačního systému (schopnost operovat nad Windows, Linuxem i Unixem). Tato výhoda je diskutabilní ve vztahu k desktopové aplikaci ArcGIS, která je schopna provozu pouze pod Windows. Uvážíme-li, že producenti chtěli pravděpodobně zajistit jedno skriptovací prostředí pro celou rodinu produktů ArcGIS včetně síťových komponent (IMS, SDE), pak se jedná pochopitelně o výhodu. Ve svém výčtu výhod volby Pythonu jako skriptovacího prostředku ESRI uvádí (Page; Batsukh, 2004):

- 
- 1) Python je lehce naučitelný díky „čistému“ syntaxu a jednoduchým, jasným konceptům.
  - 2) Python podporuje objektově orientované programování se snadno pochopitelnými zásadami.
  - 3) Dokumentace skriptů je snadná díky čitelnosti kódu.
  - 4) Python snadno pracuje s komplikovanými datovými strukturami.
  - 5) Python je snadno integrovatelný do prostředí C++ a Fortran.
  - 6) Python může být bezešvě integrován do jazyku Java.
  - 7) Python je zdarma distribuovaný a má rozsáhlou uživatelskou základnu.

Některé výše uvedené body stojí za bližší komentář. Tyto mnohdy obecně definované výhody lze jen těžko podrobit jednoduchému, ale přísnému testování s objektivními výsledky. Proto si dovolím jen několik poznámek k obecnému použití Pythonu jako skriptovacího prostředí ve vztahu k aplikaci ArcGIS.

Ad1) Syntax opravdu nepředstavuje žádnou zásadní překážku, kromě několika specifik, které si uživatel rychle osvojí. Problémem může být v určitých případech nejednoznačnost použití některých znaků ve vztahu k často využívaným SQL dotazům, zápisu cest, způsobu identifikace bloků (odpovědnost částečně padá i na vrub ESRI díky nedůslednostem při tvorbě dokumentace).

Ad2) Tento bod lze z uživatelského pohledu vysvětlit především velmi snadnou definicí proměnných, která je v mnohém blízká i dřívějším uživatelům Avenue. Python představuje v zásadě podobnou náročnost tvorby základních logických konstrukcí jako Avenue. Oproti konstrukcím ve VBA se dle mého názoru jedná opravdu o uživatelsky přátelštější koncept.

Ad3) Nejen dokumentace, ale také šíření a editace je díky čitelnému kódu bez nutnosti kompilace výhodou, která je ale vykoupena určitým zpomalením procesů při překladu instrukcí (možnost kompilace do souboru s koncovkou .pyc to částečně odstraňuje).

Ad4) Složitost datových struktur je ve vztahu k ArcGISu vázána především na datové struktury GDB a PGDB, které mají omezený počet objektů

---

a metod jejichž pomocí lze však v závislosti na nasazení vytvářet jak jednoduché základní struktury, tak obecně libovolně komplikované. Stejně důležitým prvkem je však schopnost pracovat s velkým množstvím záznamů v těchto databázích. Ovšem za předpokladu, že většina uživatelů při správě dat preferuje jednodušší strukturu i za cenu duplikace některých záznamů.

Ad 5) Možnost integrace do C++ je asi mnohem důležitější pro vývojáře. Většina nástrojů v systémových ToolBoxech je napsána právě v C++. Výhodou kompilace v C++ je právě větší rychlost při zpracování procesů a forma ochrany know-how (viz Ad 3).

Ad 7) To je samozřejmě výhoda, ale vzhledem k tomu, že veškeré potřebné komponenty včetně vlastního vývojového prostředí pod Windows jsou volitelnou součástí instalace ArcGIS, není tato výhoda pro uživatele nikterak zásadní.

Dle mé uživatelské zkušenosti je základní výhodou Pythonu to, že je firmou ESRI podporován jako základní skriptovací prostředí. Veškerá dokumentace k jednotlivým nástrojům zahrnutá v Helpu obsahuje syntax použití konkrétního nástroje a ukázkou vzorového kódu. Další součástí podpory jsou školení (v různých formách i cenách), publikace, přístupnost celé řady skriptů na stránkách technické podpory.

Ostatně preference jednoho z možných jazyků je právě jen důsledek nutnosti sjednotit dokumentaci (Tucker, 2004).

Součástí instalace ArcGIS je verze Python 2.1 a jeho nejnovější verze 2.4.3. Rozdíly ve verzích nedokážu posoudit, ale dle informací z uživatelských fór jsou v nových verzích zachovány všechny stávající vlastnosti a jsou dále rozvíjeny vlastnosti vztahující se k použití v internetovém prostředí. Další informace týkající se samotného Pythonu uživatel najde na [www.py.cz](http://www.py.cz) a [www.python.org](http://www.python.org).

### **7.3. GEOPROCESSOR**

Základním prostředníkem mezi zvoleným skriptovacím jazykem a funkcemi ArcGIS je Geoprocessor. Jeho funkcionalitu a význam se pokusím

---

osvětlit v několika dalších odstavcích.

Z podstaty objektových skriptovacích jazyků představuje skript nejvyšší modul v úrovni organizace tvořeného kódu. Pomocí příkazu `import` mohou být importovány další moduly, např. `os`, `sys` apod. Tímto způsobem je možné propojit danou funkci i s jinou aplikací než je ArcGIS. Pro využití geoprocessingové funkcionality jsou nezbytně potřebné dva moduly `win32com` a `IDispatch`. První z nich zajišťuje registraci COM objektů z pohledu operačního systému a druhý má zajišťovat interpretaci instrukcí skriptu. Samotný `geoprocessor` pak představuje jediný objekt, který využitím dynamického propojení COM objektů vytváří jednotný přístupový bod k jednotlivým nástrojům včetně uživatelsky vytvořených modelů. Vzhledem k tomu, že většina nástrojů má v sobě zapouzdřené grafické rozhraní, odpadá nutnost tvorby vlastních interface a vstupní formulář (`inputBox`) je vygenerována na základě nastavení `Skript Properties` (diskutováno v kapitole 5.2.).

Přesné schéma `geoprocessoru` se všemi metodami je součástí dokumentace obsažené v instalaci. Jiným zdrojem může být `Geoprocessor.pdf` dostupný na stránkách ESRI. Samotný objekt `geoprocessoru` najdete na svém disku při standardní instalaci jako `c:\Program Files\ESRI\ArcGIS\Bin\GpDispatch.dll`.

### A. Statické vlastnosti

Pomocí tří ze šestivlastností objektu `geoprocessoru` můžete ovlivňovat základní nastavení prostředí:

`OverwriteOutput` (povolení přepisování výsledků geoprocessingových operací).

`TemporaryMapLayers` (povolení ukládání pracovních vrstev).

`Toolbox` (asociuje `Toolbox`).

Zbývající tři jsou systémové:

`MessageCount` (kontrola číslování zpráv).

`ParamsCount` (kontrola číslování parametrů).

`MaxSeverity` (nastavení závažnosti chybových hlášení).

---

## B. Statické metody

Objekt obsahuje 48 statických metod (velmi přeneseně: „funkcí“), pomocí nichž můžeme řídit jednotlivé procesy, vytvářet a přistupovat k dalším objektům tvořících ArcGIS. Není smyslem práce popisovat všechny metody tohoto objektu, proto jsou v následujícím výčtu uvedeny po skupinách jen ty nejčastěji používané (více viz Příloha Geoprocessor).

- 1) Hlášení o průběhu skriptu: `AddMessage`, `AddError` atd.
- 2) Podmínky pro výkon nástrojů: `AddToolbox`, `Set Parametr` atd.
- 3) Kontrola podmínek a stavu: `RefreshCatalog`, `Exists`, `ValidateFieldName`.
- 4) Enumerace: `ListWorkspaces`, `ListRasters`.
- 5) Editace atributů: `InsertCursor`, `SearchCursor`, `UpdateCursor`.
- 6) Nápopěda: `Usage`.

## C. Dynamické vlastnosti a metody

Z hlediska geoprocessingových operací je snad nejčastěji používaná metoda goprocessoru:

Tool (jako parametr této metody může být zadán jakýkoliv nástroj z referencovaného toolboxu, který pak sám funguje jako metoda geoprocessoru). Dynamický chrakter má i vlastnost geoprocessoru `Enviroment`. Lze tedy měnit nastavení prostředí v průběhu skriptu v závislosti na používaných nástrojích.

## 7.4. VÝVOJOVÉ PROSTŘEDÍ

Vzhledem k textové interpretaci skriptovacích jazyků nepotřebuje ani Python vlastní vývojové prostředí a stačí jakýkoliv textový editor například Notepad. Ale protože je většina uživatelů zvyklá využívat vlastností integrovaného vývojového prostředí (Integrate Development Environment – IDE), je společně s instalací distribuováno i IDE. V případě standardní instalace ArcGIS je tímto IDE aplikace PythonWin usnadňující vývoj kódu v prostředí operačního systému Windows. Toto vývojové prostředí je také asociováno při volání editace skriptu z ArcToolboxu. Doplnkem pak může

---

být PythonIDLE 0.8, který je také součástí instalace distribuované společně s instalací ArcGIS a pochází od jednoho z autorů jazyka Python Guido van Rossuma.

Po spuštění PythonWin (např. ze standardního prostředí **Windows XP: Start > Programy > Python2.1 > PythonWin** nebo vyvolaném z kontextového menu skriptu integrovaného v Toolboxu příkazem **Edit**) se otevře interaktivní okno, které nejvíce poslouží k ladění skriptu. Samotné prostředí je opět velmi intuitivní a předchozí zkušenost s Avenue nebo AML umožňuje ihned začít pracovat. Náročnější je pouze použití nástrojů na odhalování chyb „debug“. Zde bych zájemce odkázal na některou z publikací, např. ArcGIS - Writing Geoprocessing Scripts (Tucker, 2004), která je ve formě PDF součástí dokumentace. Není v možnostech této práce ani jejího autora popsat všechny možnosti postupu, které toto pracovní prostředí skýtá.

## **7.5. Použití**

Vzhledem k velmi širokým možnostem využití skriptů je volba konkrétních aplikačních úloh určených k ověření použitelnosti ponechána čistě na potřebách uživatele. Obecně lze očekávat, že skripty se hodí všude, kde je potřeba dávkové zpracování.

---

## 8. PRAKTICKÉ OVĚŘOVÁNÍ

Ověřování použitelnosti v praxi se zakládá na čtyřměsíční zkušenosti s nasazením těchto přístupů v každodenní geoinformatické praxi. Proces tohoto ověřování je ovlivněn předmětem činnosti organizace, ve které ověřování probíhalo, proto je popsána níže. Zároveň v tak krátké době nelze ověřit všechny způsoby použití jednotlivých prostředků rozšiřování funkcionality. Není ani smyslem této práce popisovat každý pokus o rozšíření funkcionality včetně celé řady dílčích neúspěchů. Proto jsem se rozhodl popsat vždy jeden typický příklad a v souvislosti s ním uvést některé základní problémy, které jsou se zvoleným postupem spojeny. Závěrečné shrnutí pak přináší zobecnění dalších získaných poznatků včetně neúspěšných pokusů o aplikaci postupů customizace.

Tento přístup dostatečně pokrývá oblasti, které jsou k uživatelskému ověřování vhodné, jak je popisuje Patton (2005): grafické rozhraní, ergonomie, očekávaný výsledek, práce s dokumentací. Samozřejmě, že se v takovémto případě neočekává komplexní průchod veškerou funkcionalitou daného softwaru, ale důležité je, aby byly dobře vyřešeny často používané funkce.

Zvolené příklady nejlépe vystihují problematiku použití zvolených prostředků. Popis jejich použití je zaměřen spíše na problematiku implementace, časové náročnosti, efektivity apod. Dokumentace ke všem uváděným příkladům je součástí přiloženého CD společně se zdrojovými kódy resp. soubory, tak aby bylo možné jejich použití v rámci zkoumané aplikace..

### 8.1. ORGANIZACE VYBRANÁ K OVĚŘOVÁNÍ

Posláním Útvaru rozvoje hlavního města Prahy je tvorba a správa Územního plánu hlavního města Prahy. Územní plánování a urbanismus je klasickou aplikační oblastí pro geoinformační technologie. Konkrétní oddělení Datových služeb, kde jsem ověřování prováděl, je jedním ze tří oddělení spadajících pod odbor Fondu Informací o území. Hlavní činností tohoto oddělení je správa tematických vrstev výkresové části Územního plánu. Správa datových sad představuje jak jejich tvorbu a aktualizaci,

---

tak i distribuci. Distribuce zahrnuje potřeby dalších oddělení v rámci organizace i požadavky organizací a subjektů podílejících se na tvorbě a schvalování územně plánovací dokumentace. Z pohledu datové struktury to představuje všechna prostorová data tématických vrstev, respektive výstupy a mapové kompozice z nich v analogové podobě i různých digitálních formátech.

Základním prostředím pro plnění těchto úkolů je na tomto pracovišti ArcGIS ve verzi 9.1 v licenčních úrovních Info a View. Značná část editace geometrie i část tvorby atributů základních vrstev probíhá v SW prostředí Kokeš. Výkresová dokumentace je prezentována formou mapové služby nad ArcIMS a T-mapServerem, pro mapové služby i vnitřní sdílení rastrové podoby výkresové dokumentace je také využíváno ArcSDE.

Poměrně složitý datový model vychází z principu množiny základních tématických vrstev, z nichž jsou odvozovány další tématické vrstvy pro potřeby konkrétních výkresů. To je právě úlohou instalací ArcGIS Info, zatímco na stanicích ArcGIS View probíhá následně vlastní konstrukce požadovaných mapových výstupů.

V době mého příchodu do organizace bylo možností uživatelského rozšiřování funkcionality využíváno velmi ojediněle, většinou formou přejímání funkcionality jiných autorů.

## **8.2. PŘEJÍMÁNÍ FUNKCIONALITY**

### **Příklad DSMapBook**

Tímto způsobem byl v minulosti do instalací ve verzi 9.0 implementován jeden z modulů obsažených v „samples“ DS MapBook z produkce ESRI, který velmi usnadňuje tvorbu a publikaci mapových výstupů skládajících se z více mapových listů. V závislosti na požadovaném měřítku produkuje toto oddělení mapové výstupy až o 42 listech z jedné mapové kompozice. Při přechodu na vyšší verzi 9.1 většina uživatelů tuto funkcionalitu opět hledala. Jedinou informací, kterou uživatelé dostali bylo umístění zapi-  
povaného adresáře DSMapBook na jejich počítači.

---

Samotná instalace nezpůsobila nikomu komplikace neboť je dobře podchycena .bat souborem, který zajistí registraci obou potřebných knihoven.

Hlavním úskalím bylo pro všechny uživatele nasazení funkce tohoto modulu v konkrétních projektech (.mxd). Velmi kvalitní dokumentace, která je součástí instalačního balíku v podobě dokumentu aplikace MS Word, zůstala většinou uživatelů nevyužita. Pravděpodobně je k tomu vedlo nejen její anglické znění, ale především velmi malá schopnost i ochota s dokumentací pracovat (oba problémy spolu velmi úzce souvisí). Logickým důsledkem tohoto „jazykového“ problému byla doba několika hodin mezi samotnou integrací do systému a integrací do projektu, vyplněná aplikováním metody pokus-omyl.

Nejrychlejší pracovník to zvládl bez využití dokumentace za více jak čtyři hodiny. Ostatní již pak dokázali potřebný čas zkrátit za využití jeho asistence. S využitím dokumentace lze celou operaci zvládnout na poprvé za zhruba jednu hodinu včetně poznání dalších funkcí. To je další výhoda kvalitní dokumentace. Uživatel hledající metodou pokus omyl jednu konkrétní funkci totiž výsledky ostatních funkcí hodnotí jako nesplnění svého požadavku a nevěnuje jim další pozornost.

### **8.3. MODEL BUILDER**

Očekával jsem, že možnosti Model Builderu s jeho intuitivním ovládáním budou využívány v mnohem větší míře i uživateli, kteří používají licenční verzi ArcView. V praxi však použití Model Builderu představovalo jednoho pracovníka, který byl s danou možností customizace nejen obeznámen (jako ostatní), ale úspěšně, leč sporadicky ji využíval (ve smyslu tvorby pevných modelů pro konkrétní složité úlohy).

#### **A. Dokumentace tvorby datových sad územního plánu (neparametrické modely)**

Zásadní zlom z pohledu využívání Model Builderu nastal při řešení otázky aktualizace a revize současných tématických vrstev pro potřeby návrhu změny územního plánu Z1000/00, která zásadním způsobem pozměňuje používaný datový model i zaběhlé pracovní procesy.

---

Při řešení tohoto problému se nám ve spolupráci s firmou T-Mapy podařilo standardizovat dokumentaci postupu při tvorbě a správě tématických vrstev nové verze Územního plánu. Základem je využití procesních modelů, tvořených právě v Model Builderu jako prostředku dokumentace všech operací provedených nad vstupními daty až po jejich finální verzi. Protože se jedná o poměrně komplikované procesy spojující velké množství operací, byla jedním z požadavků možnost jejich opětovného spuštění i s ohledem na jejich využití právě při další aktualizaci vzniklých datových vrstev.

Poměrně složitý datový model vychází z principu množiny základních tématických vrstev, z nichž jsou odvozovány další tématické vrstvy pro potřeby konkrétních výkresů. Protože dochází k neustálým změnám a zásahům do základních vrstev (složitý proces vnitřní kontroly), je hlavním problémem následná aktualizace odvozených vrstev. Ta by v případě klasických postupů znamenala velkou časovou náročnost a následně nesplnění časového limitu, který byl na dané práce přidělen.

Společným prostředím pro uložení dat i procesní modely je personální geodatabáze. Personální geodatabáze je členěna na konečný počet  $N$  datasetů. Tato geodatabáze obsahuje stejnojmenný toolbox, který je rozdělen na stejný počet  $N$  shodně pojmenovaných toolsetů (jinými slovy relace mezi Toolsety a Datasety je  $n:n$ ). Nezbytně nutné je dodržování konvencí pro číslování a pojmenovávání jednotlivých modelů v relaci  $n:m$  k pojmenování všech vstupních, pracovních i výstupních tříd prvků. Všechny modely si udržují relativní cesty ke vstupům i výstupům, čímž je zajištěna snadná přenositelnost databáze včetně přidané funkcionality a tedy i možnost distribuovat postup jejich tvorby.

Použité modely jsou neparametrické, většinou lineárního charakteru. Zásadně v nich nejsou používány vlastní skripty kvůli problémům s distribucí. Nástroje v nich obsažené odpovídají rozsahu licence ArcINFO bez dalších extenzí.

---

## B. Jednotné nastavení datasetů (parametrizovaný model)

Tento příklad vystihuje jak poměrně snadnými a uživatelsky nenáročnými kroky customizace můžeme výrazně zefektivnit některé činnosti. Protože v rámci pracoviště, kde bylo prováděno testování, často dochází k přenosům pracovních dat mezi různými zpracovateli, popřípadě jsou přejímány mezi různými projekty, je důležité udržovat integritu dat. V rámci geodatabáze je prvním krokem k tomu sjednocení používaného referenčního systému (Spatial Reference) a definičního oboru zobrazení (X/Y Domain).

Toho lze dosáhnout různými způsoby, ale pokud je činnost maximálně zautomatizována, dá se riziko špatného nastavení těchto vlastností datasetu vyloučit.

Vytvořil jsem jednoduchý model obsahující jeden jediný proces. Ten tvoří nástroj *Creat Future Dataset*. Za parametry modelu je zvoleno umístění a název datasetu, všechny ostatní parametry jsou pevné a koncový uživatel s nimi nepřijde do styku.

V našem případě, který je zdokumentovaný na přiloženém CD, je provedeno nastavení pro území hlavního města Prahy, ale stejným způsobem lze ošetřit jakékoliv jiné zájmové území. Na tomo příkladu byly také ověřeny možnosti editace nápovědy. Tímto způsobem připravený nástroj byl následně distribuován na všechny instalce ArcGIS v oddělení.

### 8.4. PYTHON

Jediným v práci popisovaným prostředkem, který nebyl vůbec používán, je právě skriptování. Jedinou výjimku tvořilo několik přejmutých skriptů, které byly neparametricky doeditovány pro konkrétní úlohy. To je v mnoha případech dostačující prostředek ke splnění potřebných úkolů. Ovšem při tomto postupu nejde předat danou činnost osobě bez znalosti základů programování.

V praxi se vyskytuje řada činností, jejichž zautomatizování do podoby bez nutných zásahů do zdrojového kódu je žádoucí. Ale časová náročnost vývoje takové funkcionality už je oproti ModelBuilderu o poznání složi-

---

tější a místo několika desítek minut se již pohybujeme i v řádu hodin až desítek hodin. Samozřejmostí by mělo být zpracování určité formy dokumentace pro takovouto funkcionalitu. Právě proto je nezbytně nutné velmi obezřetně vybírat činnosti, u nichž je četnost opakování adekvátní času vynaloženému na jejich automatizaci.

Jednou z takových činností je v případě tohoto pracoviště vydávání podkladů pro zpracování podrobné územně plánovací dokumentace. Výstupy jsou požadovány v různých datových formátech, ale z hlediska zjednodušení bylo rozhodnuto vydávat data ve dvou výměnných formátech – SHP pro GIS aplikace a DXF pro CAD aplikace.

Z hlediska naprogramování potřebné funkcionality se jedná o aplikaci dvou úloh – ořezu více datových vrstev „multiclipu“ a následného exportu do zvoleného formátu. První úloha je použitím nástroje `Clip_analysis` a druhá představuje volbu mezi dvěma nástroji: `FeatureclassToShapefile_conversion` / `ExportCAD_conversion`.

Z tohoto pohledu se zdá, že se jedná o velmi jednoduchý postup, při němž nemůžeme narazit na žádná úskalí.

První z řady problémů, tedy definování parametrů skriptu, jsem vyřešil volbou metody `arg` operačního systému - modulu `sys`. Ošetření použitelných datových typů apod. je ponecháno na nastavení parametrů při integraci v Toolboxu.

První parametr `inputFc` má charakter `multivalued` a je třeba tento dlouhý textový vstup rozdělit na seznam jednotlivých tříd prvků.

Atribut `OutDs` musí mít nastavenou vlastnost `Direction` na hodnotu `Input` (tedy výstupní dataset již musí existovat).

Nástroj `Clip` není určen pro anotační třídy prvků (`AnnotationFeatureclass`) ostatně stejně by nešly převést do `shp`. Proto je zvolen odlišný postup a prvky anotační třídy jsou převedeny na body a pak oříznuty. To si vynutilo vložení rozhodovací struktury.

Exporty potřebují již seznam ořezaných tříd prvků uložených ve výstupním datasetu. `Geoprocessor` jej však stále vidí jako prázdný – meto-

---

da `RefreshCatalog`. Opětovné použití metody vytvoření enumeračního objektu, tentokrát však v podání geoprocessoru `ListFeatureClass`. Následuje použití dvou výše uvedených nástrojů z toolboxu `conversion`.

Celkem 40 řádek kódu i s komentářem (viz příloha) mi zabralo cca 12 hodin. Mnohem snazší je tvorba jednoúčelových skriptů pro potřeby dávkového zpracování (batch processingu) apod. I tak představuje použití Pythonu již mnohem větší časovou zátěž. Na tomto příkladě je vidět, že přestože se jedná o poměrně jednoduchou úlohu a přes usnadnění ošetření vstupních podmínek, se kód poměrně rozrostl.

---

## 9. POZNATKY A HODNOCENÍ

### 9.1. PŘEJÍMANÍ FUNKCIONALITY VYTVOŘENÉ JINÝM AUTOREM

Hlavní výhody:

- velmi rychlé a efektivní
- použitelné i jako základ pro vlastní tvorbu

Základní omezení:

- pouze obecné nástroje

Nevýhody:

- různá úroveň dokumentace
- zajištění kompaktibility s danou licenční úrovní

Komentář:

Není jeden nejlepší zdroj pro přejímaní funkcionality. Krokem, který poměrně často vede k alespoň částečnému vyřešení problému, je prohledání nejrozsáhlejší databáze volně šířené funkcionality na stránkách [www.esri.support.com](http://www.esri.support.com). Výhodou této databáze je, že obsahuje i součásti tzv symplex, čímž se v podstatě pokryjí dva možné zdroje najednou.

Přirozeným zdrojem je skupina spolupracovníků, která může být mnohdy nejrychlejší zdroj. Zároveň členové kolektivu často řeší podobné problémy nebo mají dobrou představu o řešeném úkolu. Ústní informace o omezeních a možnostech použití určité funkce může mít větší přínos než neformální několika řádkový komentář autora (obecným problémem je tedy nedostatečná dokumentace takovéto funkcionality).

Placené extenze představují sice velmi rychlé, většinou kvalitně zdokumentované a široce použitelné řešení, ale vždy je nutné uvážit, zda přínos odpovídá ceně.

Přebírání funkcionality je efektivní a urychluje postup, který je ovšem limitován uživatelskými schopnostmi. Kdy jeden uživatel danou funkcionalitu vyhledá, integruje, otestuje její použití a pak vystupuje v roli

---

„jeho distributora“ v rámci pracovního kolektivu. Jeho role tedy poté zahrnuje podobné funkce, které standardně obstarává „reálný distributor“. Připraví, popřípadě provede instalaci, prosadí kolektivní nasazení a dodává technickou podporu nebo dokonce lokalizuje prostředí získané funkcionality či její dokumentaci. Je samozřejmě již otázkou managementu daného pracoviště, zda podobný model přijme formálně nebo jej nechá fungovat v podobě, která se přirozeně vyvine.

## **9.2. POUŽÍVÁNÍ MODEL BUILDERU**

Hlavní výhody:

- intuitivní, plně grafické prostředí, ovládané drag and drop
- dávkové spuštění nástrojů
- dokumentace postupů
- ovládání parametrů přes známé graf. prostředí
- parametrizace a možnost využít zapouzdřené grafické rozhraní pro vstupní dialog

Základní omezení:

- absence rozhodovacích stromů a cyklů
- omezen na rozsah nástrojů v ArcToolBox

Nevýhody:

- tvorba značného množství mezivýsledků
- omezená použitelnost exportu do skriptu

### **A. Komentář**

V rámci výše popisovaného způsobu využití Model Builderu jako prostředku dokumentace všech operací nad daty bylo nezbytně nutné, aby si tvorbu procesních modelů osvojili všichni uživatelé zodpovědní za tvorbu dat.

Přes rozdílné zkušenosti s tímto prostředím se to povedlo všem zúčastněným. Velmi rychle přešli k jeho rutinnímu používání a mnohem větší problém znamenalo osvojení si pravidel pojmenovávání a číslování modelů, tříd prvků apod. Jednoduché osvojení je asi největší zbraní tohoto

---

prostředí.

Potvrdil se i přínos grafického ztvárnění procesů. Sestavení modelu pomáhá lépe chápat význam jednotlivých kroků u složitých postupů a umožňuje jejich snadné ladění. To samé platí i při konstrukci složitých atributových a prostorových dotazů. Pomáhá i možnost vkládat popisky.

Další výhodou, kterou ale Esri nepochopitelně nikde nepropaguje, je snadná přenositelnost v rámci toolboxu (žádné další soubory, cesty k nim apod.).

Jeho limity bych ovšem viděl v ostatních oblastech.

Tvorba modelů pro často opakované kombinace nástrojů „repetitive models“ je efektivní pouze v případě, že nám určitá část parametrů zůstává neměnná.

Základním omezením je možnost používat pouze nástroje z ArcToolboxu a absence rozhodovacích a řídicích mechanismů.

Implementace rozhodovacích struktur je možná pouze přes vložení skriptu, který vrátí logickou hodnotu, jež je podmínkou pro spuštění určité větve modelu.

Schopnost ModelBuilderu exportovat zdrojový kód Pythonu je také velmi omezená. Tuto schopnost bych přirovnal k tvorbě html z Wordu. Výhodou je kvalitní okomentování takto vytvořeného skriptu, nevýhodou pak téměř jistá nespolehlivost a značné množství proměnných (každý pracovní lyr model builderu vystupuje jako proměnná).

## B. Editační operace nad tabulkou atributů

Nástroje typu **AddField** a **DeleteField** se vykonají, ale jsou nezvratné. Tzn. při příkazu **Delete intermediate data** nedojde k navrácení jejich účinku a atributová složka dat není shodná jako před spuštěním modelu (to platí i pro nástroj **CalculateField**, který logicky doprovází použití nástroje **AddField**). Při dalším spuštění modelu tyto nástroje vykážou logickou chybu a model se na nich zastaví. Nástroj **CalculateField** nevyvolá chybu, ale provede se znovu, což může vést k nežádoucím výsledkům. To lze řešit tak, že vstupní data, nad kterými mají být vykonány nástroje tohoto typu,

---

na počátku modelu zkopírujeme a až následně aplikujeme tyto nástroje. Toto řešení zajistí, že příkaz **Delete intermediate data** smaže zkopírovaná data. Vstupní data jsou stále v původní atributové struktuře a jejich obsahu odpovídá nastavení nástrojů.

### C. Select Layer by Attribute

Tento nástroj vyžaduje jako vstup vrstvu (Layer), což lze velmi snadno vyřešit příkazem **MakeFeatureLayer**. Problém nastává v případech, kdy mají být aplikovány nad třídou prvků vzniklou až v průběhu modelu. Do vzniklé vrstvy se nepřenesou informace o attributech (do prvního spuštění modelu). To lze ošetřit buď spuštěním tohoto úseku modelu a až následnou aplikací selekcí s použitím SQL dotazů, nebo napsáním SQL dotazu bez použití QueryBuilder. Třetí, poněkud pracná, ale z hlediska správy dat nejčistší varianta, je důsledné používání domén pro nastavení rozsahu hodnot atributů. Právě toto třetí řešení je velmi vhodné při práci s nominálními atributy a řeší i zmiňovanou problematiku.

### D. Feature to Polygon

Tento nástroj neumožňuje zvolit parametr **Output Feature Class** za parametr modelu. Je to dáno pravděpodobně využitím objektu **Workspace** při konstrukci jména výstupní třídy prvků. Použitelnost tohoto nástroje je tedy v modelu omezená. Je těžké určit, zda se jedná o výjimku nebo o určité pravidlo. U jiných nástrojů jsem se s tím nesetkal.

## 9.3. PYTHON

Hlavní výhody:

- dávkové zpracování dat
- přístup k ArcGIS přes dobře zdokumentovaný objekt GPDispatch
- otevřený kód
- poměrně jednoduchý syntax, rychlé osvojení
- snadná deklarace proměnných
- možnost využít zapouzdřené grafické rozhraní pro vstupní dialog

---

### Základní omezení:

- není podporován přístup a interakce k GUI

### Nevýhody:

- komplikovaný debugging
- konflikty v přístupu ke geodatabázi

### A. Komentář

Mezi největší výhody nasazení skriptů oproti Model Builderu bych označil právě jejich možnost využít kontrolní a řídicí metody geoprocessoru. V podstatě se tak jedná o jedinou možnost jak v ArcGIS zajistit dávkové zpracování dat.

Nevýhodou je velká časová náročnost osvojení si potřebných dovedností. A osobně si myslím, že i pro člověka s dobrými základy programování je absolvování školení přinejmenším prospěšné. Celkově je vždy důležité uvážit, zda je opravdu nasazení skriptu nezbytně nutné. A dalším faktorem je, jak velký počet nezávislých proměnných se bude používat.

Jestli volit Python nebo jiný skriptovací jazyk nemůžu posoudit. Rozhodně jsem neměl zásadní problémy s pochopením jeho logiky nebo osvojením syntaxu. Pomocí ArcObjects přístupných v integrovaném vývojovém prostředí VBA je možné dosáhnout složitější funkcionality, ale na druhou stranu jsem se v praxi nesetkal s konkrétním požadavkem funkcionality, u kterého bych narážel na omezení schopnostmi skriptovacího jazyka.

### B. Konflikty přístupu k databázi

Zpočátku vedly veškeré nedokončené průběhy skriptu k zablokování databáze. Doporučení autorizovaného školitele je neustále restartovat ArcGIS i PythonWin při každém průchdu skriptem. Dle mé zkušenosti je velmi vhodné použití podmínky `try` a `except`, které i při nevykonání určitého bloku nechají provést následující instrukce skriptu až do jeho skončení. Tím dojde ke standardnímu ukončení i k ukončení přístupu k databázi. Tento postup problémy výrazně eliminuje.

---

## C. Debugger

Pro ladění programu je prostředí PythonWin vybaveno debuggerem. Ten je z mého osobního pohledu zbytečně složitý. Mnohem účinnější je pečlivé vkládání “ech” (`print` “odkaz na místo v programu, probíhající proces apod.”).

## D. Dokumentace

Přestože je celá dokumentace podřízena právě Pythonu, dochází u příkladů použití některých nástrojů k nejednoznačnostem v syntaxi. Jedná se především o nástroje, které vyžadují některý parametr zadat jako řetězec. Například nápověda k příkazu `FeatureClassToCoverage`<sup>1</sup> - v tomto případě jsou zcela opomenuty závorky a význam uvozovek je najasný, zvláště chcete-li místo konkrétních dat použít proměnnou.

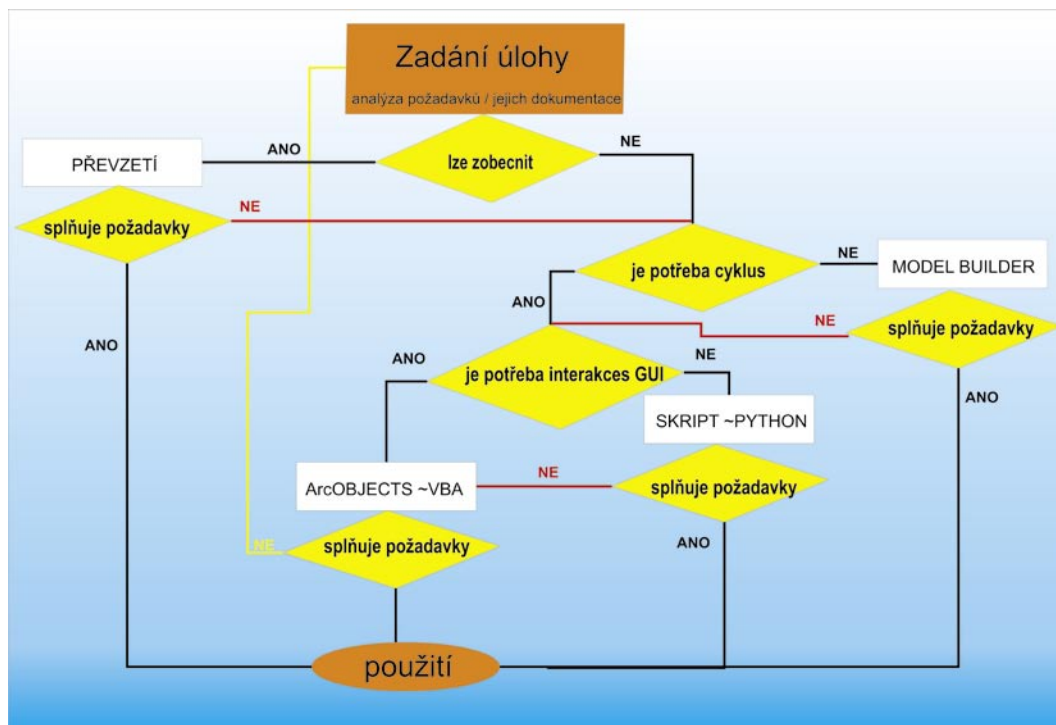
<sup>1</sup> `gp.featureclasstocoverage "harvestable.shp polygon; final_forest/polygon region; road.lyr arc", "parkland", "1.25", "double"`

## 9.4. VOLBA ADEKVÁTNÍHO PROSTŘEDKU

Získané zkušenosti jsem se pokusil aplikovat na obecný příklad řešeného problému. Vzniká tak algoritmus (rozhodovací strom) pro volbu vhodného prostředku k rozšíření funkcionality. Zohledněny jsou výhody a omezení jednotlivých přístupů. V případě možnosti zvolit dva nebo více prostředků je vždy volen ten, u kterého je předpokládána menší časová náročnost.

Dalším myšlenkovým postupem, který není v předkládaném schématu graficky ztvárněn, je kontrola správného použití zvoleného prostředku. Pokud výsledek naší snahy neodpovídá požadavkům, neznamená to nutně nedostatečnost zvoleného prostředku. Je tedy nejprve předpokládán několikanásobný průchod mezi použitím prostředku a porovnáním výsledků s požadavky, který zabezpečí odstranění chyb vzniklých nesprávným použitím zvoleného prostředku.

Dalším rozhodovacím prvkem, který by ale toto schéma zneřehlednil, je zjištění, zda převzatá funkcionální splňuje požadavky alespoň částečně a zda ji lze doeditovat. Pak by automaticky měla následovat spojnice s vývojovým prostředím, ve kterém byla přebíraná funkcionální vytvořena.



Obr. 3. Schéma volby adekvátního prostředí

Do tohoto algoritmu bylo zahrnuto i použití Visual Basicu, přestože v práci není podrobně zpracován. Zahrnut je především ze snahy o pokrytí celé problematiky a také je jedním z prostředků jak doeditovat částečně vyhovující funkcionální jiného autora. Do budoucna lze také očekávat vzrůstající programátorskou úroveň pokročilých uživatelů ArcGIS a tím také jeho častější používání.

---

## 10. DISKUSE

Funkcionalitu aplikace ArcGIS lze uživatelsky značně rozvíjet a existuje k tomu několik přístupů. Jejich hlavní výhody a nevýhody, resp. omezení, jsou popsány v předchozím oddílu práce.

Lze konstatovat, že všechny prostředky jsou použitelné, ale jejich výhody a omezení je předurčují k určitým oblastem použití. Základní otázky volby správného prostředku jsou zodpovězeny v podobě navrhnutého rozhodovacího stromu.

Cílem definovaným v zadání práce byla tvorby sady testovacích úloh, které měly možnosti těchto prostředků ověřit. Domnívám se, že rozhodnutí ověřit použitelnost v praxi a zdokumentovat praktické použití je dostatečným splněním tohoto cíle. Pro celou řadu uživatelů může být takovýto přístup cennější než jedno obecné zadání řešené z pohledu několika prostředků.

Základní výhodou celého prostředí aplikace ArcGIS je otevřenost a podpora takovýchto přístupů. Právě možnost vzájemně kombinovat jednotlivé přístupy prostřednictvím ArcToolboxu je z mého pohledu unikátní. Nemohu objektivně srovnávat možnosti ostatních GIS desktopových klientů na trhu, ale neznám komerční produkt s takto širokými možnostmi a podporou tvorby vlastní customizace.

Důležitou, ovšem v rámci práce jen částečně řešenou, otázkou je celkový management pracoviště a další provozní podmínky, které ovlivňují úspěšnou implementaci popisovaných přístupů mnohdy více než jejich technologická omezení a tvůrčí potenciál uživatelů.

---

## 11. ZÁVĚR

Vzhledem k rozsáhlosti tématu i problémům při volbě odpovídajícího přístupu jsem si vytvořil svůj uživatelský pohled na problematiku rozšiřování funkcionality. Popsal jsem možné prostředky a přístupy jak funkcionalitu uživatelsky rozšiřovat.

Ověřil jsem jejich použitelnost v praxi včetně problémů při jejich implementaci do pracovních postupů. Takto získané poznatky jsem zpětně zahrnul do popisu těchto prostředků a přístupů.

V diskusi pak představuji obecný rozhodovací model pro určení vhodného prostředku přizpůsobení. Přestože cíle možná byly poněkud širší a velmi obecně formulované, podařilo se mi dosáhnout určitého stupně poznání problematiky, aplikovat je v praxi a ze získaných poznatků vyabstrahovat stručné obecné závěry i doporučení.

---

## 12. LITERATURA

### Publikace:

Cameron, E. et al.: ArcGIS Desktop Developer Guide, ESRI, [PDF] publikováno 2005 [cit. 2006-04-25].

Lindemann, J.; Page, K.: What`s New in ArcGIS 9 - course lectures book, ESRI, 2004.

Lindemann, J.; Page, K.: What`s New in ArcGIS 9 - course excersises book, ESRI, 2004.

Page, K.; Batsukh, U.: Introducing to Geoprocessing Scripts using Python - course book, ESRI, 2004.

Page, K.; Batsukh, U.: Introducing to Geoprocessing Scripts using Python - excersise book, ESRI, 2004.

Minami, M.; Using ArcMap, ESRI, 2002.

Patton, R.; Testování softwaru, Computer Press, 2005.

Harlow, M.; et.al.: ArcGIS 9.1 Geoprocessing Comands Quick Reference Guide, ESRI, [PDF] publikováno 2004 [cit. 2006-04-25].

Švec, J.; Učebnice jazyka Python (aneb Létaující cirkus) release 2.2. (angl. originál: van Rossum, G; Drake, F.L.jr.; The Python Tutorial - An Introduction to Python) [PDF] publikováno 2002 [cit. 2006-03-28].

Tucker, C.; ArcGIS 9 - Writing the Geoprocessing Scripts with Python, ESRI, [PDF] publikováno 2004 [cit. 2006-05-02].

### Elektronické kurzy:

ESRI Virtual Campus (dostupné z <http://campus.esri.com/>)

Exploring the VBA Enviroment (virtual courses, 2005-06-10)

Exploring ArcObjects in VBA (online lead courses, 2005-11-10)

---

### Internetové zdroje:

Esri Development Network, [online], c.1995-2006, poslední revize 23.05.2006, [cit. 2006-05-24] <<http://edn.esri.com/>>

Esri Support Center, [online], c.1995-2006, poslední revize 18.05.2006, [cit. 2006-05-24] <<http://support.esri.com/>>

Arcdata Praha s.r.o., [online] c.1992-2006, poslední revize 18.05.2006, [cit. 2006-05-24] <[www.arcdata.cz](http://www.arcdata.cz)>

Python, [online] c. 1990-2006, poslední revize 18.05.2006, [cit. 2006-05-24] <[www.python.org](http://www.python.org)>

Python česky, [online] [cit. 2006-05-24]. <[www.py.cz](http://www.py.cz)>

GeoWizzard, [online] , poslední revize 07.05.2006, [cit. 2006-05-24] <[www.ian-ko.com](http://www.ian-ko.com)>

---

## 13. RESUMÉ

This diploma thesis describes and compares different approaches to the customization of the ArcGIS Desktop functionality.

First part of the thesis overwrites the methods and procedures. User's testing approach is selected logically as key method. The problematic resources and used terminology is also included to this part.

Next part is focused on ArcGIS Desktop application and sorts the different possibilities of customizing. It disputes the principles of licensing and accessing of the functionality in Arc Toolbox.

The using of distributed functionality is the first possibility to customize ArcGIS. The discussion about the resources and list of the potential distributors leads to problems of formats and their implementation. The implementation skills are necessary for next distribution of self-made functionality.

Model Builder looks simple for its user graphic interface and drag and drop instrumentation. It allows graphically connect the series of tool together. The implementation of the variables and parameters is more difficult, however it brings more efficiency.

The scripts are different way, but bring more control for the tools. The scripts are able to manage batch processing. Every tool of Arc Toolbox is accessible over GpDispatch COM, the object which is called geoprocessor. The methods and settings of this object are described and complete schema of geoprocessor is also included on CD.

The author's view of the problem is based on his participation in the courses provided by an ESRI authorized dealer and practical experience. The practical part of this thesis presents the approaches and their usefulness for the solution of typical problems of a GIS team in the City Development Department of Prague. Samples of customize scripts and models are documented on CD.

The discussion compares approaches and suggests a short guideline for their effective choice and usability. All approaches are functional a useful and can be great support for every day GIS computing.

---

## **SEZNAM PŘÍLOH:**

CD ROOM

Text práce [PDF]

Dokumentace uváděných příkladů [PDF]

Zdrojové soubory příkladů

Komwntované schéma geoprocessoru [PDF]

Učebnice jazyka Python (Švec, J.) [PDF]

Seznam všech příloh je na CD zpracován v podobě HTML