Univerzita Palackého v Olomouci

Přírodovědecká fakulta

Katedra geoinformatiky

Marek BALUN

GIS NÁSTROJE PRO PODPORU EVAKUACE OBYVATEL PŘI ZATOPENÍ OBJEKTŮ V KRIZOVÝCH SITUACÍCH

Bakalářská práce

Vedoucí práce: Mgr. Rostislav Nétek

Olomouc 2013

Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci bakalářského studia oboru Geoinformatika a geografie vypracovala samostatně pod vedením Mgr. Rostislava Nétka.

Všechny použité materiály a zdroje jsou citovány s ohledem na vědeckou etiku, autorská práva a zákony na ochranu duševního vlastnictví.

Všechna poskytnutá i vytvořená digitální data nebudu bez souhlasu školy poskytovat.

V Olomouci 12. srpna 2013 Marek Balun

Děkuji vedoucímu práce Mgr. Rostislavu Nétkovi za odborné vedení, podněty, rady a připomínky při vypracování práce. Dále děkuji Ing. Janu Langrovi za odborné konzultace a rady. Za poskytnutá data děkuji Magistrátu města Olomouce.

Vložený originál **zadání** bakalářské/magisterské práce (s podpisy vedoucího katedry, vedoucího práce a razítkem katedry). Ve druhém výtisku práce je vevázána fotokopie zadání.

OBSAH

ÚVOD

Krizový management podobně jako jiné obory, které využívají prostorové informace adoptoval GIS nástroje. Byla to jistě jedna z největších a nejvýznamnějších změn, kterou v poslední době prodělal. Na rozdíl od mnoha jiných oborů, krizový management nevyužívá GIS nástroje pouze k prezentaci prostorových dat veřejnosti, ale často slouží jako nástroj pro podporu rozhodování v krizových situacích, ve kterých jsou ohroženy lidské životy a majetek. Dobré nebo špatné rozhodnutí může mít velký dopad na velikosti těchto ztrát na životech a na majetku. Operační střediska složek integrovaného záchranného systému jsou vybavena systémy, které umožňují řídit a mapovat pohyb vozidel v reálném čase, v případě nenadálé události ji umožňují rychle a přesně lokalizovat a navigovat k ní zásahová vozidla. Kromě přijímání okamžitých rozhodnutí a řízení zásahu v reálném čase jsou GIS nástroje využívány při dlouhodobém strategickém plánování, jako je detekce rizikových oblastí potenciálně ohrožených konkrétním hazardem např. stanovení záplavových oblastí podle teoretických průtoků a následně přijatá opatření – stavební uzávěry v těchto oblastech, výstavba protipovodňových hrází atd.

Tato práce se v rešerši zabývá krizovým managementem a GIS nástroji, zejména RIA technologiemi a také koncepcí Servisně orientované architektury. V praktické části je pak vyvíjena aplikace pro potřeby krizového managementu, konkrétně pro podporu evakuace obyvatel ze zatopených objektů.

1 CÍLE PRÁCE

Cílem praktické části je vytvořit aplikaci pro potřeby krizového managementu – pro podporu evakuace obyvatel ze zatopených objektů. V zadání práce jsou pouze obecné požadavky na aplikaci – aby vycházela z RIA koncepce a využívala servisně orientované architektury, ale bližší požadavky, jako je funkčnost, zájmové území, rozsah, konkrétní technologie atd. nejsou specifikovány. Ty byly více konkretizovány až během studentské praxe na Odboru ochrany Magistrátu města Olomouce, pro který bude aplikace vyvíjena, z čehož již vyplívá, že zájmovým územím bude město Olomouc.

Praktická část diplomové práce se bude skládat ze dvou částí, v první části bude nutné vybrat vhodná data a připravit je pro publikaci na serveru. Ve druhé části bude pak vyvíjena samotná aplikace.

Aby byla dodržena koncepce servisně orientované architektury, aplikace nebude obsahovat žádná vlastní data, ale bude je připojovat v podobě služeb poskytovaných vzdáleným serverem.

2 POUŽITÉ METODY A POSTUPY ZPRACOVÁNÍ

2.1 Použitá data

Základní data pro aplikaci poskytnul Magistrát města Olomouce, další byla z těchto dat vytvořena pomocí metod geoprocesingu. Jednotlivé datové sady byly buď ve formátu shapefile nebo v souborové geodatabázi. Pro aplikaci byly vybrány následující vrstvy:

- záplavové území pro teoretické průtoky Q_5, Q_{20}, Q_{100} řeky Moravy (polygonové vrstvy)
- zaplavené území během historických povodní z let 1997 a 2006 (polygonové vrstvy)
- záplavové území řeky Bystřice (polygonová vrstva)
- katastrální území (polygonová vrstva)
- vodní toky a vodní plochy (polygonová vrstva)
- zástavba (polygonová vrstva)
- komunikace (liniová vrstva)
- ulice (liniová vrstva)
- adresné body (bodová vrstva)

2.2 Použité programy

V praktické části byl využit následující software a technologie:

- ArcMap z rodiny produktů ArcGIS od společnosti esri, verze 10.1 – pro zpracování a úpravu datových vrstev
- ArcGIS for Server od společnosti Esri – pro publikaci datových vrstev
- Adobe Flash Builder verze 4.0 – bylo použito jako IDE pro vývoj aplikace
- Apache Flex verze 4.9 – technologie v níž byla aplikace vyvíjena
- ArcGIS API for Flex verze 3.1 – API pro Flex od společnosti Esri, které umožňuje ve Flexu tvořit aplikace využívající služeb ArcGIS for Server nebo ArcGIS Online

2.3 Postup zpracování

- Výběr technologie pro vývoj aplikace
- Získání dat
- Výběr vhodných datových vrstev pro aplikaci
- Zpracování a úprava dat
- Publikace dat na serveru
- Vývoj aplikace
- Testování aplikace

3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

3.1 Krizový management

3.1.1 Krize

Pojem krize v poslední době slýcháme velice často a to s různými přívlastky. Je jím označována celá řada nejrůznějších situací, které spolu na první pohled ani moc nesouvisí. V současnosti se nejvíce mluví o ekonomické krizi, ale máme i krize politické, vládní, sociální, společenské a tak dál až třeba po osobní krizi. Všechny tyto situace jsou charakteristické jistou nestabilitou daného systému – ekonomiky, politického systému, společnosti, jednotlivce… Roudný a Linhart definují krizi jako situaci, v níž je významným způsobem narušena rovnováha mezi základními charakteristikami systému (narušeno je poslání, filosofie, hodnoty, cíle, styl fungování systému) na jedné straně a postojem okolního prostředí systému na straně druhé. Podle zákona 240/200 Sb. A 110/1998 Sb. je krize stav, který nastává, když je vyhlášen stav nebezpečí, nouzový stav, stav ohrožení státu, nebo válečný stav. Krizi definují následující faktory (Roudný et Linhart 2004):

- Je vyvolána nějakou hrozbou
- Krizové stavy nejsou příliš časté a je obtížné je předvídat
- Krize mají vícerozměrné sociální dopady
- Pro postižené jsou velkou emocionální a mentální zátěží, projevují se zmatkem a napětím
- V krizových situacích jsou často k dispozici jen neurčité a neúplné informace na jejichž základě je potřeba přijímat rozhodnutí
- Krize jsou z principu zvladatelné, jejich řešení je naléhavé a vyžaduje rychlá rozhodnutí
- V krizových situacích často dochází ke střetu zájmů (tržních subjektů, států, mezinárodních organizací atd.)

3.1.2 Krizová situace

Krizová situace je odchylka od plánovaného nebo očekávaného chování nebo sled událostí, který ohrožuje nebo nepříznivě ovlivňuje lidi, majetek nebo životní prostředí. (Johnson 2000) Krizové situace bývají obvykle vyvolány mimořádnými, nečekanými a obtížně předvídatelnými událostmi. Krizové situace je možné dělit dle různých kritérií. Prvním kritériem je rozsah způsobených škod na majetku a případné ztráty na životech – viz následující tabulka.

Tab. 1 Typy krizových situací v závislosti na způsobených škodách a ztrátách na životech (Roudný et Linhart 2004)

Typ MU	Ztráty na lidských životech	Materiální ztráty řádově v Kč
ZÁVADA	Žádné	100
VADA	Žádné	1000
PORUCHA	Žádné	10 000
NEHODA	Jedinec	100 000
HAVÁRIE	Několik jedinců	1 mil.
ZÁVAŽNÁ HAVÁRIE	Desítky	10 mil.
POHROMA	Stovky	100 mil.
KATASTROFA	Tisíce	1 mld.
KATAKLYZMA	Statisíce	10 mld.
APOKALYPSA	Milióny	100 mld.

Dalším kritériem je geografický rozsah, dle něj můžeme mimořádné události dělit na lokální, regionální, celostátní a globální. Dle délky trvání je možné rozdělit na krátkodobé a dlouhodobé. Dále je možné krizové situace rozdělit dle typu ohrožení. Na přírodní, antropogenní a smíšení. Johnson dělí ohrožení na tyto obecné typy ohrožení:

Způsobené lidmi (Human caused) – zahrnují neplánované události nebo nehody, jež jsou výsledkem lidské činnosti. Například úniky chemických látek, úniky radiace, epidemie, exploze, dopravní nehody, požáry ve městech.

*Přírodní katastrofy (Natural Disaster*s) - zahrnují neplánované události, jež jsou výsledkem přírodních procesů a jevů jako například zemětřesení, tornádo, tsunami, blizardy, extrémní sucha, extrémní mrazy, přemnožení škodlivého hmyzu (nálety sarančat, kůrovcové roky).

Vnitřní nepokoje (Internal disturbances) – události nebo aktivity plánované skupinou jedinců se záměrem vyvolat zmatek, chaos. Např. vzpoury, demonstrace, útěky z vězení velkého měřítka, výtržnosti a nepokoje.

Nedostatek surovin a energií (Energy and material shortage) – způsobené například cenovou válkou, embargem, bojkotem.

Útok (Attack) – zahrnuje teroristické útoky velkého měřítka, útoky jadernými, chemickými a biologickými zbraněmi.

Bezpečnostní strategie České republiky, která je součástí Usnesení vlády ČR č. 1254/2003 rozděluje krizové situace následovně:

Krizové situace spojené s vnějším vojenským ohrožením státu – agrese nebo hrozba agresí cizí moci, zahrnuje situace od narušení vzdušného prostoru až po přímí útok vojenskými prostředky; vtažení nebo hrozba vtažení České republiky do válečného konfliktu; důsledky vyplívající z účasti na zahraničních mírových operacích

Krizové situace vnitřně bezpečnostního charakteru – zahrnuje stavy způsobené nekontrolovanou migrací obyvatelstva, vyhrocení etnických vztahů, náboženské nepokoje, vyhrocení sociální, ekonomické a politické situace, prudký nárůst kriminality a organizovaného zločinu.

Krizové situace spojené s ohrožením stability hospodářské a finanční soustavy státu – tyto situace jsou způsobeny výpadky ve fungování hospodářství státu, destabilizací měny, uvalením embarga na dovoz důležitých surovin atd.

Ostatní krizové situace spojené s ohrožením životů a zdraví obyvatelstva, ničením životního prostředí, majetkových a kulturních hodnot, ke kterým dochází v souvislosti ohrožením vnější nebo vnitřní bezpečnosti státu – sem spadají přírodní živelné pohromy jako požáry, povodně, vichřice atd. také sem patří antropogenní pohromy jako např. úniky nebezpečných látek, dopravní nehody, průmyslové havárie atd. (Konečný et al 2011)

3.1.3 Hrozba

Dle Antušáka a Kopeckého je hrozba libovolný subjekt, jenž svým působením (činností) může poškodit nebo zničit konkrétní chráněnou hodnotu nebo zájem jiného subjektu nebo jev či událost jako bezprostřední příčina poškození nebo zničení konkrétní chráněné hodnoty nebo zájmu.

3.1.4 Riziko

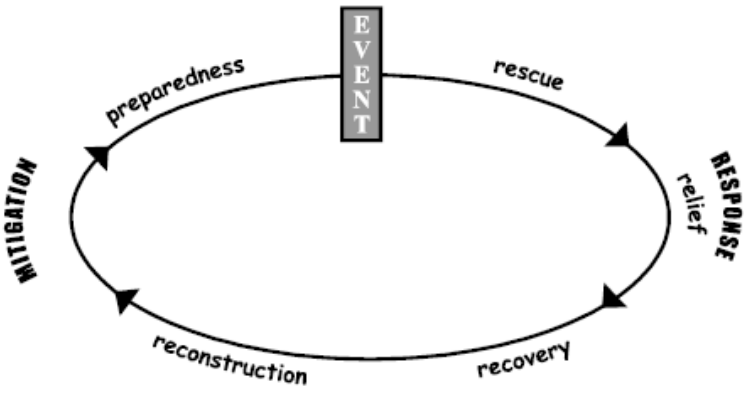
Riziko je pravděpodobnost výskytu ohrožení. Například riziko poškození infrastruktury při zemětřesení je vysoké nachází-li se v seismicky aktivní oblasti, naopak v seismicky neaktivních oblastech je riziko poškození infrastruktury nízké. (Johnson 2000) Antušák a Kopecký definují riziko jako veličinu spíše abstraktní, pravděpodobnostně kvantitativní, sekundárně (výpočtem, úvahou) odvozenou od hrozby; představujících možnost vzniku události s výsledkem odchylným od předpokládaného cíle a to s určitou objektivní matematickou nadějí či statistickou pravděpodobností. Jako kvantifikovanou nejistotu.

3.1.5 Krizový management

Krizový management nebo také krizové řízení je specifický druh managementu zaměřený zvládání krizových situací. Snaží se v krizových situacích minimalizovat ztráty na lidských životech a na majetku, poskytnout pomoc postiženým a danou situaci stabilizovat. Po překonání krizové situace je potom jeho úkolem odstraňováni vzniklých škod. Podle Antušáka a Kopeckého krizový management představuje ucelený soubor přístupů, názorů, zkušeností, doporučení a metod, které vedoucí pracovníci a krizoví manažeři užívají ke zvládnutí specifických činností při minimalizaci zdrojů příčin vzniku krizových situací, při přípravě na krizové situace, při bránění vzniku a eskalaci krizových situací, při redukci zdrojů krizových situací a při odstraňování následků působení negativních faktorů krizové situace.

3.1.6 Fáze krizového managementu

Předepsaný systém jak společnost reaguje na katastrofu je nejčastěji označován jako Emergency response cycle (ERC) – česky Cyklus krizového jevu. Tento cyklus se skládá ze čtyř základních fází, které jsou vzájemně propojené a navazují na sebe. Cyklus krizového jevu bývá často vizualizován v podobě kruhu. To velice dobře vystihuje danou problematiku. Cyklus začíná ve fázi prevence, ve které je snaha co nejvíce eliminovat riziko vzniku krizové situace. Další fází je příprava na zvládnutí potenciální krizové situace. V případě, že krizová situace skutečně nastane, následuje odezva. Po stabilizaci krizové situace následuje fáze obnovy. Když se společnost zotaví z následků katastrofy cyklus se vrací opět na začátek. (Cutter 2003)



Obr. 1 Emergency response cycle (Cutter 2003)

Prevence

Fáze prevence, anglicky Mitigation, se skládá z kroků které vedou snížení pravděpodobnosti vzniku katastrofy a také z kroků, které v případě vzniku katastrof

povedou ke zmenšení jejich dopadů. Například v případě snižování povodňového rizika mohou být čištěna a prohlubována říční koryta, stavěny protipovodňové hráze a suché nádrže. (Johnson 2000)

Připravenost

Fáze připravenost, anglicky preparedness, zahrnuje kroky v rozsahu, ve kterém již není první fáze prevence schopna předcházet katastrofě. V této fázi jsou na úrovní vlád, státních institucí a soukromých organizací vytvářeny konkrétní plány na záchrany lidský životů a minimalizaci škod způsobených katastrofou. Může zahrnovat budování hmotných rezerv, nácvik zvládání krizových situací, budování systémů včasného varování atd. opatření v této fázi mají rovněž za cíl zvýšit operativnost v následující fázi odezvy. (Johnson 2000)

Odezva

Odezva, anglicky response, následuje bezprostředně po katastrofě. Je koncipována jako pomoc postiženým, dále je charakterizována snahou stabilizovat situaci a snahou zabránit sekundárně vznikajícím škodám např. zabránit při povodních kontaminaci vody ze zaplavených chemických provozů, zabránit rabování majetku atd. Na konci této fáze jsou pak bilancovány škody a zahájeny obnovovací práce. (Johnson 2000)

Obnova

Fázi obnovy, anglicky recovery, lze rozdělit na krátkodobé obnovovací práce a na dlouhodobou obnovu. Krátkodobé obnovovací práce zahrnují obnovu životně důležité infrastruktury alespoň na minimální operační úroveň, dlouhodobé obnovovací práce pak následují často ještě řadu let, než jsou obnovy všechny škody způsobené katastrofou a oblast je uvedena do původního stavu. (Johnson 2000)

3.1.7 Krizový management v ČR

Krizové řízení v České republice upravuje zákon č. 118/2011 Sb. o krizovém řízení a o změně některých zákonů a je zajišťováno soustavou orgánů a organizací propojených vzájemnými vazbami. Tyto orgány mají čtyři hierarchické úrovně – celostátní, krajskou, na úrovni ORP a nejnižší je na úrovni obcí. Jejich složení závisí na tom, jestli se zrovna nachází ve stavu krizové připravenosti nebo v krizovém stavu. Na celostátní úrovni jsou tvořeny vládou, ministerstvy, Českou národní bankou a Bezpečnostní radou státu v době krizové připravenosti a Ústředním krizovým štábem v krizové situaci. Na krajské úrovni jsou tvořeny hejtmanem, Krajským úřadem a Bezpečnostní radou kraje v době krizové připravenosti a Krizovým štábem kraje v době krizových stavů. Na úrovni ORP a na úrovni obcí je složení obdobné – starosta, obecní úřad, krizový štáb v době krizové situace. (Konečný et al 2011) Dalším významným činitelem je Integrovaný záchranný

systém (IZS), který je tvořen Policií České republiky, Hasičským záchranným sborem a Zdravotnickou záchrannou službou. Dalšími složkami IZS vyčleněné síly ozbrojených sil, městská policie, havarijní a pohotovostní služby a další.

3.1.8 Geoinformační technologie používané IZS

Co se týče geoinformačních technologií je v současnosti nejlépe vybavený Hasičský záchranný sbor České republiky. Byl jedním z prvních, kdo začal GIT v rámci IZS využívat. První geoinformační systémy zde byly zavedeny již v roce 2000. HZS ČR je vybaven softwarem ArcGIS od společnosti Esri, GISelIZS od společnosti T-MAPY a některými dalšími aplikacemi od společností RCS Kladno a Profia. Systém využívá jeden centrální datový sklad, který se nachází v Lázních Bohdaneč a je společný pro všechna krajská ředitelství. (Konečný et al 2011)

Zdravotnická záchranná služba využívá nejvíce aplikaci GISelIZS od společnosti T-MAPY - celkem sedm krajských středisek. Tři krajská střediska využívají software od společnosti MEDIUMSOFT. Tyto systémy slouží k podpoře vnitřních organizačních procesů. Další aplikace, které jsou využívány ZZS jsou Navigace od firmy Kom Tes Chrudim, InfoMap od společnosti PJSOFT a Emergency Kontrol Center od firmy CPE. Sledování vozů záchranné služby v terénu je realizováno pomocí přijímačů GPS. Polohu vozidel lze sledovat v přímo v GIS např. v GISelIZS nebo NaviGate. (Konečný et al 2011)

Nejhorší situace, co se týče geoinformačních technologií, je u Policie ČR. Policie zatím využívá GIT velmi málo. V roce 2010 byl odstartován projekt v rámci kterého by měli být vytvořeny analytické a mapové služby, měli by také vzniknout WMS služby dostupné veřejnosti. (Konečný et al 2011)

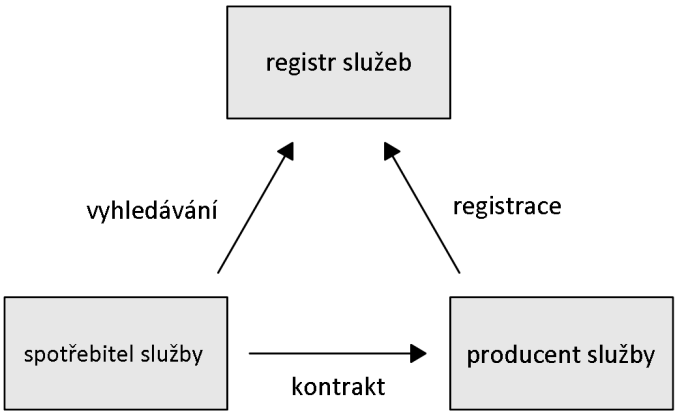
3.2 Servisně orientovaná architektura

3.2.1 SOA

Zkratka SOA znamená Service Oriented Architecture, česky servisně orientovaná architektura nebo také architektura orientovaná na služby. V současné době neexistuje žádná přesná nebo všeobecně přijímaná definice toho co to SOA je. Obecně lze říci, že se jedná přístup, obecnou koncepci při kompozici služeb nezávislou na implementaci a platformě. Objevují se i názory, že se jedná o paradigma. (Schreiner 2007) Nejběžnější implementací SOA jsou webové služby. V této práci se budeme dále zabývat servisně orientovanou architekturou jen z pohledu webových služeb.

Jak již napovídá název, základním stavebním kamenem SOA jsou služby. V obecné rovině je služba chápána jako nějaká posloupnost kroků, činností, která vede k zisku. V prostředí webových služeb se jedná o aplikaci, jež zapouzdřuje logiku. Základní model SOA je tvořen dvěma stranami – poskytovatelem služby (service provider) a

spotřebitelem služby (service consumer). Místem kde obě strany navážou kontakt je registr služeb. V registru se nachází popisy služeb, na jejichž základě si spotřebitel vyhledá konkrétní službu, poté s vybranou službou naváže komunikaci. Vztah spotřebitele služby a poskytovatele služby se nazývá kontrakt. (Schreiner 2007)



Obr. 2 Základní elementy SOA (upraveno dle Schreiner 2007)

3.2.2 Spolupráce služeb

V SOA jsou tři základní typy spolupráce služeb. Prvním z nich je kooperace – služba A spolupracuje se službou B, aby mohla uspokojit potřeby spotřebitele. Druhým typem spolupráce je agregace – služba A se spojí se službou B a vznikne služba C jakožto úplně nová služba. Třetím typem spolupráce je choreografie. Choreografie je spolupráce mnoha služeb napříč celým systémem za účelem dosažení byznys procesu. Byznys proces je nějaká posloupnost úkonů (workflow) která vede k hmotnému zisku. (Weiss et Rychlý 2007)

3.2.3 Struktura SOA

Servisně orientovaná architektura je z abstraktního pohledu tvořena třemi základními vrstvami. Vrstva využívá služeb nižší vrstvy a sama nabízí služby vyšší vrstvě. Vrstvy spolu komunikují přes rozhraní. Nejnižší vrstvou je vrstva komponent. Komponenty tvoří samotné služby a jsou zodpovědné za funkčnost služeb. Komponenty komunikují přes rozhraní.

Na vyšší úrovni je vrstva služeb. Služba je v nižší vrstvě tvořena komponentami, ke kterým přistupuje přes jejich rozraní. Úkol, který má služba zpracovat distribuuje svým komponentám. Služba má stejně jako komponenta rozhraní a popis.

Nejvyšší vrstva je vrstva byznys procesů. Byznys procesy v této vrstvě představují konkrétní aplikace.

3.2.4 Základní principy SOA

Volné vazby – služby o sobě navzájem uchovávají minimum informací, je plně postačující když o sobě služby navzájem ví, tento princip umožňuje snadné udržování a vývoj systému – kterákoliv služba může být kdykoliv nahrazena jinou se stejným rozhraním

Kontrakt služeb – služby mají dohodu o komunikaci – služby znají navzájem svoje popisy a mohou přistupovat ke svým rozhraním

Samospráva – služby jsou autonomní jednotky – mají kontrolu nad svou logikou

Abstrakce – služby zapouzdřují svou logiku, jiné služby nemohou přistupovat k jejich „vnitřnostem“, mohou s nimi komunikovat pouze přes definované rozhraní, jednotlivé služby tak pracují jako černé skříňky

Znovupoužitelnost – je snaha aby měla SOA co nejnižší počet služeb, ale zároveň aby jednu službu mohlo využívat co nejvíce aplikací

Kompozice – služby lze komponovat do větších celků, spojováním služeb vznikají nové služby

Bezstavovost – služby uchovávají minimum informace, pro vnějšího pozorovatele se jeví jako bezestavové

Zjistitelnost – služby se navenek prezentují svým popisem a rozhraním a je možné je vyhledat v registru nebo pomocí nějakého vyhledávacího mechanismu (Erl 2009)

3.3 Implementace SOA

3.3.1 WSDL

Web Service Description Language (verze 1.1 Web Service Definitions Language) je jazyk sloužící pro popis služeb. WSDL byl vyvinut společnostmi Microsoft a IBM a standardizovalo ho konsorcium W3C. Jazyk WSDL je k dispozici ve dvou verzích – WSDL 1.1 a WSDL 2.0. Verze 1.1 má status W3C Note, naproti tomu verze 2.0 má status W3C Recommendation a je plně podporována konsorciem W3C. (Schreiner 2007)

Jazyk WSDL je založen na XML dokumentech, používá se pro popis služeb a jejich rozhraní. Samotný dokument WSDL je rozdělen na dvě části – abstraktní popis a konkrétní popis. Abstraktní popis definuje rozhraní webové služby, ale nespecifikuje technologii, která bude použita k přenosu. Tímto je zabezpečena integrita bez ohledu

platformu. Abstraktní popis obsahuje tři prvky – interface (ve verzi 1.1 portType), operation a message. Interface slouží k abstraktnímu popisu rozhraní webové služby. Rozhraní je složeno z jednotlivých operací (operation), které služba poskytuje. Operace ke komunikaci mezi sebou využívají zpráv (message). (Erl 2009)

Aby byl popis služby kompletní je nutné ještě kromě abstraktní definice rozhraní definovat fyzický přenosový protokol. Ten je definován v konkrétním popisu jazyka WSDL. Konkrétní popis se skládá opět ze tří elementů – binding, endpoint (ve verzi 1.1 port) a service. Binding definuje konkrétní přenosový protokol např. SOAP. Endpoint odkazuje na URL na které je služba přístupná.

3.3.2 SOAP

Simple Object Access Protocol je protokol využívaný pro přenos zpráv mezi službami. Byl vyvinut Microsoftem a standardizován konsorciem W3C, které ho udržuje. V současnosti je aktuální verze SOAP 1.2, která má status W3C Recommendation. SOAP byl vyvinut, aby nahradil starší komunikační protokol RPC. (Erl 2009)

Zpráva SOAP je tvořena XML dokumentem, skládá z obálky (envelope), hlavičky (header) a vlastního těla dokumentu (body). Do dokumentu XML je možno vložit přílohy např. obrázky. SOAP používá pro přenos nejčastěji protokoly z rodiny TCP/IP jako HTTP, HTTPS nebo UDP.

3.3.3 REST

REST je zkratka Representational State Transfer. REST je rozhraní, které umožňuje přístup ke zdrojům dat přes protokol HTTP. Každý zdroj je jednoznačně identifikován svou URL a je reprezentován nejčastěji XML souborem, ale může jej reprezentovat i JSON, HTML, PDF, SVG atd. Pro odběr REST stačí pouze načíst stránku s příslušnou URL. REST vyvinul v rámci své disertační práce Roy Fielding.

REST má podobnou funkcionalitu jako SOAP, ale jeho použití je mnohem jednodušší, na druhou stranu neposkytuje tak široké možnosti jako SOAP. REST je vhodnější pro menší a jednodušší aplikace, zatímco SOAP se spíše uplatní v robustnějších projektech. (Velte et al 2011)

3.3.4 UDDI

Universal Description, Discovery and Integration je veřejný registr webových služeb, který umožňuje vyhledávat služby dle nejrůznějších kritérií. V podstatě se jedná taky o službu. UDDI na rozdíl od WSDL a SOAP nebyl nikdy standartizován.

3.4 RIA

RIA je zkratkou pro Rich Internet Applications, což by se dalo přeložit jako „Bohaté internetové aplikace“. Slovo bohaté zde vyjadřuje vysokou (vyšší) míru interaktivity oproti běžným HTML stránkám. RIA se snaží docílit toho, aby měl vývojář webových aplikací stejné možnosti jako při vývoji desktopových – snaží se implementovat typické vlastnosti desktopových aplikací jako je komplexní grafické rozhraní, přívětivé uživatelské chování (odstínění modelu žádost/odpověď), drag & drop, klávesové zkratky, kontextová nápověda, ... do webových aplikací.

RIA není jedna technologie, ale jedná se o více různých konkurenčních technologií od různých vývojářů. RIA aplikace můžeme dělit na ty jež jsou založené na JavaScriptu a na ty, jež jsou založeny na plug-inech.

Hlavním zástupcem RIA postavených na JavaScriptu je AJAX. JavaScriptu je odvozen od jazyka ECMAScript. Pro běh postačuje pouze webový prohlížeč. V dnešní již mají prakticky všechny prohlížeče vestavěn JavaScriptu engine který interpretuje zdrojový kód napsaný v JavaScriptu. Nevýhodou je, že aplikace postavené na JavaScriptu mohou být v různých prohlížečích zobrazeny odlišně. (Johannson 2010)

Hlavními reprezentanty technologií založených na plug-inech jsou Apache Flex a Microsoft Silverlight. Pro svůj běh vyžadují mít nainstalovaný plug-in v prohlížeči. Pro obě tyto prostředí jsou poskytována IDE s knihovnami, kompilátory debuggovacími nástroji. Výhodou těchto technologií je, že se zobrazují stejně ve všech prohlížečích ve kterých jsou podporovány. Apache Flex vyžaduje pro běh ve webovém prohlížeči nainstalovaný Adobe Flash Player nebo v případě desktopové nebo mobilní aplikace Adobe AIR.

V poslední době se RIA rozšířila i na mobilní platformy Android, iOS a Windows Phone.

3.4.1 AJAX

AJAX není technologie, více na místě by asi bylo označení technika, jelikož je založen na spolupráci více různých technologií. Zkratka AJAX původně vznikla spojením asynchronní JavaScript + XML, poprvé tento termín použil Jesse James Garrett v článku „AJAX: A new approach to web application“ a označoval jím aplikace vzniklé spojením právě JavaScriptu a XML. Dnes se ale AJAX spíše chápe jako obecné označení pro různé technologie a techniky, které umožní komunikaci prohlížeče se serverem bez nutnosti načítání celé stránky, ale pouze té části která se proti předcházející stránce změnila.

Pro rozvržení struktury stránky je využíváno HTML případně XHTML a pro nastavení stylů CSS stejně jako u klasických statických stránek. Dynamika stránky je ošetřena pomocí JavaScriptu a o výměnu dat mezi klientem a serverem se stará rozhraní XMLHhttpRequest (XHR), které zde funguje jako jakýsi prostředník. Pro výměnu dat se nejčastěji používá formát XML, ale lze využít např. i JSON. Velkou nevýhodou AJAXu je nefungující šipka krok zpět (undo). (Ašenbryl 2008)

3.4.2 Adobe/Apache Flex

Flex má mezi RIA aplikacemi dominantní postavení, v současnosti má největší podíl na trhu. Flex byl vyvinut společností Macromedia, po akvizici Macromedií jej získala společnost Adobe. Ta jej vyvíjela do verze 4.6, pak předala zdrojové kódy Apache Software Foundation, ta projekt umístila do Apache inkubátoru a uvolnila paritní verzi 4.8. V prosinci 2012 byl Flex vyňat z inkubátoru a byl zařazen do skupiny TLP (Top-Level-Project), následně poté byla vydána verze 4.9, která je momentálně aktuální. (Apache Flex 2013)

Flex patří mezi RIA technologie postavené na plug-inech, využívá dvě běhová prostředí Adobe AIR a Adobe Flash Player. Adobe AIR je využíván pro desktopové a mobilní aplikace a platforma Flash běžící na Adobe Flash Playeru pro webové aplikace.

Vlastní Flex – Flex SDK je tvořen balíkem, který obsahuje jazyky MXML a ActionScript, jejich kompilátory, knihovny předpřipravených komponent, debugger a další nástroje. (Borek 2011) Flex je postaven na dvou jazycích – MXML a ActionScript. Jazyk MXML je značkovací jazyk odvozený z jazyka XML, slouží k popisu uživatelského rozhraní. Tento jazyk má vyšší sémantiku než XML – obsahuje elementy pro definici jednotlivých komponent jako např. element pro definici tlačítka - `<s:Button>`. ActionScript je objektově orientovaný jazyk, který zodpovídá za dynamiku aplikace. ActionScript je odvozen ze standartu ECMAScript. (Borek 2011) Technologie Flex je postavená na platformě Flash, která rovněž využívá jazyk ActionScript, který se kompiluje do binárního souboru s příponou `.swf`. Zatímco Flex je určen spíše pro programátory, Flash cílí spíše na počítačové animátory a grafiky. (Johannson 2010)

Jako vývojové prostředí Flex primárně využívá Adobe Flash Builder postavený na platformě Eclipse, k dispozici jsou také open source alternativy - Flash Develop a IntelliJ IDEA, ale je možné použít i jakýkoliv textový editor. Adobe Flash Builder je komerční produkt, je dostupný ve verzích Flash Builder Standart a Flash Builder Premium. (Borek 2011) Samotné SDK a obě běhová prostředí jsou ke stažení zdarma.

3.4.3 Microsoft Silverlight

Microsoft Silverlight představuje asi nejvážnější pokus konkurovat Flexu. Jak už název napovídá Silverlight pochází z dílny Microsoftu. V současnosti je aktuální verze Silverlight 5. Je podporován ve vybraných prohlížečích ve Windows a Mac OS X. (Johannson 2010) Linux není oficiálně podporován - jen v podobě open source projektu Moonlight, jehož vývoj je již ale ukončen.

Silverlight využívá pro strukturování aplikace jazyk XAML, což je proprietární jazyk Microsoftu založený na XML. Dynamika stránek je zajišťována jazyky z rodiny .NET jako je VisualBasic.Net nebo C#, ale lze použít i JavaScript. Pro běh aplikace vyžaduje mít nainstalovaný Silverlight run-time environment.

Zvláštností Silverlightu je, že využívá dvě vývojová prostředí současně. Pro návrh grafického uživatelského rozhraní využívá Microsoft Expression Blend a pro

programování aplikace Microsoft Visual Studio. Je sice pravda, že v obou prostředích jde jak psát kód, tak i v nějakém návrhářském módu upravovat grafické rozhraní, takže teoreticky dostačuje k vytvoření aplikace pouze jedno prostředí, v praxi by to ale asi nebylo příliš efektivní a ergonomické. (Lacko 2010)

3.4.4 JavaFX

JavaFX je založená na platformě Java, která byla od devadesátých let vyvíjena Sun Microsystems. Po akvizici Sun Microsystems je platforma vyvíjena Oracle Corporation. JavaFX byla poprvé publikována v roce 2007. Je založená na 3F (Form Follows Function) vyvinutým zaměstnancem Sun Microsystems Christopherem Oliverem. JavaFX podporuje všechny běžné platformy jako Windows, Mac OS X, Linux a Solaris. Podporuje také mobilní platformy – JavaFX Mobile od verze 1.1. (Moritz 2008)

Velkou výhodou JavaFX je, že může využívat všech existujících tříd napsaných pro platformu Java. Aplikace jsou vyvíjeny ve skriptovacím jazyku JavaFX Script. Tento jazyk je odvozen od jazyka Java a je mu syntaxí velmi podobný. Soubor se zdrojovým kódem má příponu `.fx`, po kompilaci do binární podoby má příponu `.class` a jako běhové prostředí využívá Java Virtual Machine. Finální aplikace je pak distribuována ve formě souboru s příponou `.jar`. Tento soubor může být spouštěn jako desktopová aplikace, applet v prostředí webového prohlížeče, mobilní aplikace nebo na herní konzoli. Jako IDE pro JavaFX lze využít Eclipse nebo Netbeans. (Choleva 2011)

3.4.5 Open Laszlo

Open Laszlo je otevřená technologie vyvíjená společností Laszlo Systems. Název Laszlo byl zvolen na počest maďarského umělce Laszla Moholy-Nagy. (OpenLaszlo 2013)

Aplikace Open Laszlo jsou psány v jazyku LZX, který by se dal přirovnat ke směsi JavaScriptu, XML a xPath. Zatím poslední aktualizací Open Laszla je verze 4.9.0 vydaná v říjnu roku 2010, z čehož by se dalo usuzovat, že projekt je již neaktivní. (OpenLaszlo 2013)

3.4.6 HTML5

HTML je také často zmiňováno v souvislosti s RIA aplikacemi. Předchozí verze 4.01 byla přijata jako standart v již roce 1999 a již dlouhodobě nedokáže vyhovět trendům současnosti a potřebám webových vývojářů, kteří jsou nuceni používat navíc další technologie, aby dosáhli interaktivity stránek. Nemá nativně vestavěnou podporu multimediálních dat, nepodporuje multi-threading JavaScriptových procedur, ukládání do cache paměti prohlížeče je velmi omezené. (Johannson 2010)

To by měla změnit nová verze HTML5 která obsahuje oproti svému předchůdci spoustu vylepšení. HTML5 podporuje multimediální data jako video a audio. Nově je zabudována přímo do HTML podpora vektorové grafiky – konkrétně formátu SVG. U dřívějších verzí bylo nutné, aby byla podpora vektorové grafiky ošetřena na straně prohlížeče. Podpora SVG nebyla mezi prohlížeči moc vysoká, např. Microsoft implementoval podporu SVG do svého prohlížeče Internet Explorer až od verze 9, do té doby podporoval pouze vlastní formát VML. (Taraldsvik 2011) Další novinkou je element canvas který je s SVG často zaměňován. Na rozdíl od vektorového SVG, které je součástí DOM a má definované vztahy mezi prvky, je canvas rastrový a nemá definované vztahy mezi objekty.(Taraldsvik 2011) Nově jsou implementováni i tzv. Web Workers, kteří umožňují JavaScriptu běh paralelních procesů v hlavním vláknu. Web Workers sice nepředstavují tak úplně multi-threading – ten podporuje běh více vláken, ale multi-threading simulují. Jednou z dalších významných změn oproti HTML 4.1 je nahrazení cookies konceptem nazvaným Web Storage. Cookies mají velké omezení v podobě „úniku“ informací mezi okny prohlížeče – má-li uživatel otevřeny stejné webové stránky ve dvou oknech a provede v jednom okně změnu, tato změna se projeví i v druhém, toto může být často nežádoucí. Web Storage využívá dva způsoby ukládání sessionStorage a localStorage. SessionStorage je pro krátkodobé ukládání dat, data se ukládají jen po dobu relace prohlížeče, po zavření okna se smažou. Data uložená v localStorage se nevymažou zavřením okna a jsou dostupná ve všech oknech a záložkách. (Johannson 2010) HTML5 nově také umožňuje tvorbu offline aplikací.

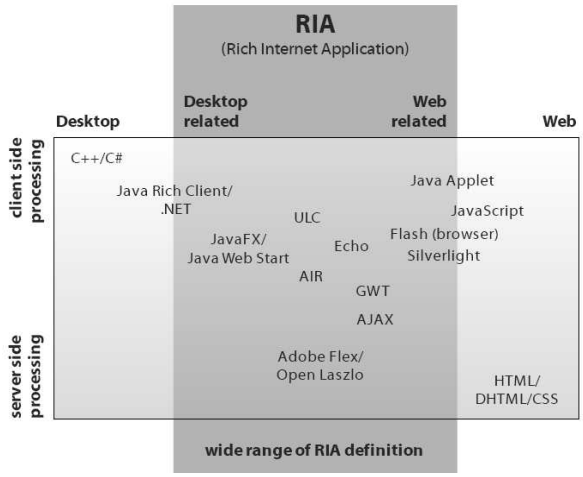
3.4.7 Srovnání RIA technologií

Prvním srovnávacím kritériem by mohlo být rozšíření technologie. Stoprocentní penetraci mají technologie založené na HTML a JavaScriptu které jsou nativně podporovány všemi prohlížeči. Mezi RIA technologiemi založenými na plug-inech má pravděpodobně největší rozšíření platforma Flash – Johannson udává penetraci téměř 97% všech zařízení. Naproti tomu konkurenční Silverlight jen kolem 55%. JavaFX má podle Cholevy penetraci asi 79% - pro svůj běh vyžaduje platformu Java.

Dalším kritériem je multiplatformnost. Zde opět nejlépe vychází AJAX a technologie postavené na HTML JavaScriptu, které prakticky nejsou platformou omezeny. Silverlight je omezen pouze na Windows a OS X, dříve byla Novellem vyvíjena i open source verze s názvem Moonlight, která byla určena pro Linux, ale její vývoj byl již ukončen. Flex aplikace využívající platformu Flash jsou podporovány na Windows, OS X, na Linuxu a Solaris. (Choleva 2011) Problémy mají s podporou na mobilních zařízeních od Applu s operačním systémem iOS, nově také nejsou podporovány na Androidu, mobilní verze Adobe Flash Playeru byla odstraněna z Google Play, s tím, že Android již Flash nebude dále podporovat. Do budoucna by mohli mít problémy i na Windows. V nejnovější verzi Windows 8 není v prostředí Metra také již Flash podporován a spekuluje se, že v dalších Windows by mohla být podpora ukončena již definitivně. Flex aplikace využívajících jako běhové prostředí Adobe AIR jsou podporovány na všech výše zmíněných

platformách i na iOS a na Androidu. JavaFX je široce akceptovaná a s podporou nemá problémy – vyskytuje se na většině nejběžnějších platforem – Windows, OS X, Linux, Solaris a na mobilních OS (JavaFX Mobile).

Dále by mohlo být hodnoceno vývojové prostředí. Pro vývoj HTML a JavaScriptu je možné použít libovolný textový editor, ale je k dispozici i celá řada specializovaných editorů a vývojových prostředí, jak komerčních tak nekomerčních. Za komerční můžeme jmenovat např. Adobe Dreamweaver a za nekomerční např. český PSPad. Pro tvorbu Flex aplikací lze také využít jakýkoli textový editor. Pokud budeme chtít specializované IDE, zvýrazňující syntaxi, máme na výběr mezi oficiálním vývojovým prostředím od Adobe – Flash Builder nebo open source Flash Develop a IntelliJ IDEA. Microsoft Silverlight vyžaduje Microsoft Visual Studio, to je k dispozici v komerční placené verzi nebo v bezplatné verzi Visual Studio Express for Web. Krom Visual Studia vyžaduje ještě Microsoft Expression Blend který je k dispozici pouze v placené verzi. JavaFX využívá stejné vývojové prostředí jako Java – Netbeans nebo Eclipse. Obě tato prostředí jsou open source.



Obr. 3 Srovnání webových a desktopových technologií (Moritz 2008)

3.5 Desktopové GIS nástroje

3.5.1 ArcGIS

ArcGIS je rodina produktů od americké společnosti Esri. Společnost Esri byla založena již v šedesátých letech minulého století a patřila k průkopníkům na poli GIS software. V současnosti patří k firmám s největším podílem na trhu v této oblasti.

Desktopové GIS nástroje od společnosti Esri - ArcMap v různých licencích patří k nejčastěji využívaným GIS nástrojům. V České republice je v oblasti krizového managementu využíván Hasičským záchranným sborem, Krajskými úřady a některými

ministerstvy – např. Ministerstvem průmyslu a obchodu – konkrétně Správou státních hmotných rezerv (Konečný et al 2011).

Aktuální verze ArcGIS je v současnosti 10.2. Desktopový nástroj nese označení ArcGIS for Desktop, skládá se ze dvou aplikací ArcMap a ArcCatalog. K dispozici je ve třech licencích Basic, Standart a nejvyšší Advanced. V předchozích verzích nesli tyto licence označení ArcView, ArcEditor a nejvyšší ArcInfo. K těmto verzím ještě lze dokoupit extenze obsahující specializované nástroje.

ArcGIS kromě vizualizace umožňuje i provádění složitých a komplexních analýz, umí pracovat jak s vektorovými tak s rastrovými daty (extenze Spatial Analyst), umožňuje tvorbu 3D modelů terénu. V oblasti povodňové problematiky jej lze např. využít k modelování povrchové odtoku a akumulace vody v terénních depresích.

Společnost Esri nabízí pro území Spojených států dataset Emergency Management Maps, mapující různé hazardy.

3.5.2 GISelIZS AE

GISelIZS AE je produkt české firmy T-MAPY z Hradce Králové. T-MAPY jsou přední českou firmou působící v oblasti GIS. GISelIZS je využíván Hasičským záchranným sborem a Zdravotnickou záchrannou službou.

Aplikace je založena bázi ArcObjects od firmy Esri, na platformě .NET 3.5 a umožňuje spolupráci s ArcGIS for Desktop. Skládá se z tří částí – IZS Administrátor, IZS operátor a IZS Search Admin. IZS Administrátor je extenze pro ArcGIS for Desktop, která umožňuje přípravu mapových projektů GIS administrátorem. IZS Operátor je GIS klient sloužící k vizualizaci mapových podkladů v operačním středisku. Nástroj IZS Search Admin slouží k přípravě uživatelsky definované vyhledávací databáze.

GISelIZS je určen pro operační střediska jednotlivých složek IZS. Jeho účelem je lokalizace místa nahlášené krizové události např. dopravní nehody nebo požáru. Sytém dokáže vyhledávat dle adresných bodů, podle kilometráže vodních toků, kilometráže pozemních komunikací, podle hektometrů na železnici a v pomístních názvech.

Vizualizace podkladové mapy je stejná pro všechny kraje, tematická data za jednotlivé kraje si v terénu sbírají GIS pracovníci HZS krajů. (GIS Portál HZS ČR 2013)

4 VÝVOJ APLIKACE

Prvním požadavkem na aplikaci je vizualizace zátopových oblastí řeky Moravy pro průtoky pětileté, dvacetileté a stoleté vody a budov a ulic, které budou při těchto průtocích zatopeny. Dalším požadavkem je podávání informací o počtu obyvatel, jež bude nutné z postižených oblastí evakuovat – na úrovni katastrálních území, ulic i jednotlivých adresných bodů, podávání informací o počtu zatopených objektů a jejich využití. Na magistrátu měli ještě jeden specifický požadavek – vypracovat jednoduchý plán pro přeparkování automobilů ze zatopených oblastí a vytipování vhodné lokality kde mohou být tyto automobily dočasně zaparkovány.

4.1 Výběr technologie a software

Jako technologie, ve které bude vyvíjena aplikace, byl zvolen Apache Flex. Bylo rozhodováno mezi technologiemi Apache Flex a Microsoft Silverlight. Ve prospěch Flexu mluvilo jeho dominantní postavení na trhu, to že se jedná o otevřenou technologii, a že pro jeho běh je vyžadován Adobe Flash Player, který je dnes nainstalován prakticky ve všech prohlížečích a který je také zárukou toho že se bude aplikace zobrazovat na všech prohlížečích stejně. Naopak v neprospěch Silverlightu hovořila podpora jen malého počtu vybraných prohlížečů (Internet Explorer, Mozilla Firefox, Google Chrome, Safari – zde se projevili i osobní preference, autor používá Operu) a nutnost instalace plug-inu Silverlight run-time environment, jehož přítomnost v prohlížečích rozhodně není samozřejmostí. Kvůli výše zmíněným důvodům by tedy vybrán Apache Flex, použita byla konkrétně verze 4.9. Flex SDK je volně ke stažení na stránkách flex.apache.org.

Pro publikaci dat byl vybrán ArcGIS for Server od společnosti Esri, hlavními důvody pro tento výběr bylo, že katedra geoinformatiky i Magistrát města Olomouce vlastní licenci na jeho užívání a v neposlední řadě také, že společnost Esri poskytuje API, které do Flex SDK přidává nové třídy objektů, jež umožňují připojit služby publikované serverovým řešením od Esri.

Pro přístup k datům bylo použito rozhraní REST. Pro vyvíjenou aplikaci je díky své jednoduchosti REST výhodnější než protokol SOAP.

Jak bylo zmíněno výše, pro propojení aplikace s daty byla použita ArcGIS API for Flex 3.1. API je vlastně balík obsahující předpřipravené třídy objektů a jejich metody, které umožňují tvořit mapové aplikace s využitím ArcGIS for Server. API je po zaregistrování u společnosti Esri zdarma ke stažení na portále ArcGIS resources.

Jako vývojové prostředí aplikaci byl použit Adobe Flash Builder 4. Pro úpravu dat před jejich publikováním byl použit ArcGIS for Desktop 10.1.

4.2 Zpracování dat

Data bylo nutné před použitím nejdříve upravit a také opravit. Data obsahovala poměrně velké množství topologických chyb – hlavně liniová vrstva komunikací obsahovala velké množství „přetahů“ a „nedotahů“, komunikace na sebe místy vůbec nenavazovaly nebo procházely skrz zástavbu. Bylo nutné opravovat i hodnoty i atributů, zejména pravopisné chyby v názvech ulic.

Další datové vrstvy byly vytvořeny pomocí geoprocesingu ze základních dat. Vrstvy sloužící pro vizualizaci zaplavených komunikací a budov při povodních různé intenzity byly extrahovány z vrstev komunikace a zástavba nástrojem Clip. Jako Clip Feature byly použity záplavové území. Stejným způsobem bylo postupováno i vrstvy ulic, adresných bodů a katastrálních území, tyto vrstvy ale nejsou v aplikaci viditelné – mají nastavenou průhlednost a v aplikaci jsou reprezentovány pouze atributovými tabulkami.

Údaje o počtu obyvatel původně obsahovala pouze vrstva s adresnými body, bylo nutné tuto informaci vložit i do vrstev zástavby, ulic a katastrálních území, aby bylo možné zjišťovat počet evakuovaných osob za jednotlivé ulice a katastrální území. Postup byl následující – ve vrstvách adresných bodů „oklipovaných“ dle jednotlivých záplavových území byl sumarizován počet obyvatel podle ulic a podle katastrálních území a následně exportován jako `.dbf` tabulka. Tato tabulka byla potom připojena (join) k cílové vrstvě a tato vrstva exportována.

Dále bylo třeba vytvořit vrstvu popisů ulic. Objekt `<esri:FeatureLayer>` použitý k připojení k připojení vrstev publikovaných serverem nepodporuje jak dynamické popisky (dynamic labeling), které se automaticky přizpůsobují měřítku mapy, tak ani statické popisky. Proto se používá náhradní řešení, kdy jsou popisky převedeny na polygonovou vrstvu a připojeny jako samostatná vrstva. Nejdříve jsou popisky konvertovány a uloženy do databáze, v dalším kroku jsou použity jako šablona k vytvoření polygonové vrstvy pomocí nástroje Feature Outline Mask, ve volbách tohoto nástroje je nutné vybrat ve volbě Mask Kind možnost Exact, aby výstupní polygon přesně kopíroval popisky a ve volbě Margin nastavit 0.

Pro další část aplikace zaměřenou na přeparkování osobních automobilů bylo třeba vytipovat vhodnou lokalitu pro parkování. Na tuto lokalitu bylo kladeno několik požadavků. Prvním požadavkem byla samozřejmě dostatečná rozloha lokality – ideální by bylo, kdyby se jednalo o jedinou lokalitu schopnou pojmout všechny automobily a ne vícero menších, u kterých by bylo problematické zabezpečení proti zlodějům a vandalům. Lokalita by se měla nacházet co nejblíže města, ideálně přímo v katastrálním území města mimo záplavové území – v případě, že by se lokalita nacházela mimo město, musel by být zajištěn odvoz pro občany, kteří zde přijeli zaparkovat. Dalším požadavkem je dobrá dopravní obslužnost a v neposlední řadě by bylo dobré, kdyby byl daný pozemek ve vlastnictví města – krizový štáb sice teoreticky může v případě ohrožení nařídit majitelům soukromých pozemků, aby je dali k dispozici, ale v praxi by s tím asi byly problémy např. v úvahu připadaly parkoviště u supermarketů, ale dá se předpokládat, že

tyto supermarkety budou chtít prodávat i v době povodní a takovýmto krokům by snažili bránit.

Jako vhodná lokalita bylo po poradě s vedením Odboru ochrany předem vytipováno Olomoucké letiště v Neředíně. Letiště se nachází v katastrálním území města, jeho vlastníkem je město Olomouc, dopravní spojení zajišťuje tramvajová linka. Bylo nutné pouze ověřit, jestli je letiště schopné pojmout všechny automobily z postižené oblasti – jako postižená oblast bylo pro úlohu zvoleno záplavové území Q_{100}, bylo tedy nutné zjistit počet automobilů evidovaných na občany bydlící v záplavové zóně Q_{100}. Nejprve byl vznesen požadavek na Odbor dopravy, jestli by nemohl poskytnout data s počty automobilů evidovaných na majitele bydlící v konkrétních postižených ulicích nebo alespoň počet registrovaných vozidel ve městě. Na dotaz přišla odpověď, že současný registr vozidel toto neumožňuje. Proto byla použita data z databáze Českého statistického úřadu. ČSÚ poskytuje data s evidovanými vozidly pouze na úrovni krajů a okresů. Počet osobních automobilů evidovaných v okrese Olomouc k 31. 12. 2011 je 88777. Z této hodnoty bylo vypočítáno množství automobilů registrovaných na občany bydlící v záplavové zóně Q_{100} – 21074 automobilů. Toto číslo je jen přibližné, ale tato přesnost je pro řešenou úlohu dostačující – lze předpokládat, že zdaleka ne všichni využijí možnost si zaparkovat automobil na letišti a evakuují se raději po vlastní ose; ne všechny automobily evidované v Olomouci jsou v Olomouci skutečně provozovány a naopak ne všechny automobily, které jezdí v Olomouci jsou zde evidovány. Jako referenční automobil byla zvolena Škoda Octavia, její rozměry jsou přibližně 4,5 x 2,5 (rozvor dveří) metrů. Délka byla navýšena ještě o půl metru na pět metrů, ať je ponechána rezerva. 21074 automobilů s těmito rozměry by zabralo plochu 26,34 hektarů, rozloha letiště je 58,61 hektarů (včetně budov) – je tedy více než dostačující.

Pro potřeby aplikace pak bylo potřeba vytvořit vrstvy komunikací, které budou označovat trasu přesunu. Tyto vrstvy se budou vizualizovat v závislosti na tom, které katastrální území zvolí uživatel. Tyto vrstvy byly vytvořeny z vrstvy komunikací. Směřují z vybraných sběrných bodů v jednotlivých katastrálních územích na letiště v Neředíně. Sběrné body byly umístěny na frekventované křižovatky hlavních komunikací v postižených částech katastrálních území. Samotné trasy přesunu pak byly voleny tak, aby využívali hlavních komunikací.

4.3 Zdrojový kód

V druhé polovině praktické části byla vyvíjena samotná aplikace. Jak již bylo napsáno aplikace byla vyvíjena v Apache Flex 4.9 s využitím ArcGIS API for Flex 3.1, jako vývojové prostředí bylo použito Adobe Flash Builder 4.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
         xmlns:s="library://ns.adobe.com/flex/spark"
```

```
xmlns:mx="library://ns.adobe.com/flex/mx"
xmlns:esri="http://www.esri.com/2008/ags"
pageTitle="Povodňový informační systém města Olomouce">
```

Aplikace je uvozena deklarací `<?xml version="1.0" encoding="UTF-8" ?>`, protože se jedná o platný dokument XML a je ohraničena kořenovými (root) elementy `<s:Application>`. Kořenový element má jako atributy `xmlns:fx`, `xmlns:s`, `xmlns:mx` a `xmlns:esri`, kterými jsou definovány jmenné prostory pro třídy objektů spark, esri, mx a fx. Zdrojový kód lze rozdělit na tři části. V první části je definována struktura aplikace, druhou část představují deklarace a třetí je tvořena skripty zodpovědnými za dynamiku aplikace.

První část, v níž je rozvržena struktura aplikace je tvořena jazykem MXML, ikdyž se zde vyskytují krátké fragmenty jazyka ActionScript (ve složených závorkách, sloužící k vázání dat). Je zde definovány tři kontejnery obsahující horní lištu s ovládacími a navigační prvky, vlastní mapové pole a legendu. Lištu tvoří kontejner `<s:controlBarContent>`, uvnitř kterého jsou uzavřeny všechny ovládací prvky. Ovládací prvky představují ikony – obrázky ve formátu png vložené pomocí elementu `<mx:Image>`, metodou `click` jsou propojeny se skripty, které zodpovídají za funkcionalitu. Kromě ikon je zde ještě objekt `<s:TextInput>` plnící funkci měřítka, které přijímá vstupy od uživatele a element `<s:Label>` ve kterém je uzavřen název webové stránky – aplikace. Mapové pole je představováno kontejnerem `<esri:Map>`. Mezi jeho atributy patří např. atribut `scale` pomocí kterého je nastaveno výchozí měřítko mapy. Uvnitř tohoto kontejneru jsou pak uzavřeny elementy `<esri:FeatureLayer>`, které slouží k připojování datových vrstev publikovaných pomocí ArcGIS for Server, k tomu slouží konkrétně atribut `url`. Atribut `symbol` odkazuje pomocí skriptu na deklarace vizuálních vlastností vrstev. Legenda je umístěna v kontejneru `<s:Panel>` a je tvořena jediným elementem `<esri:Legend>`.

Druhou část kódu představují deklarace uzavřené mezi elementy `<fx:Declarations>`. Tato část kódu je stejně jako předešlá tvořena jazykem MXML s krátkými vsuvkami jazyka ActionScript. Na rozdíl od předešlé části jsou zde deklarovány nevizuální vlastnosti a objekty. Slovo nevizuální může být docela zavádějící, protože jak již bylo řečeno o pár řádek výše, jsou zde deklarovány vizuální vlastnosti vrstev, jako je barva, průhlednost, ohraničení, šířka linie atd. Také jsou zde deklarována všechna popup okna. Obecně jsou těmi nevizuálními objekty myšleny objekty, jejichž instance jsou vytvořeny pouze v paměti počítače, tedy nemají grafickou reprezentaci v základním layoutu aplikace, ale mohou být vytvořeny konstruktory po provedení nějaké akce, ke které dá impuls uživatel např. otevření „vyskakovacího" (popup) okna a mohou být také odstraněny destruktory – zavření okna. Také jsou zde definovány vlastnosti objektů, které se mohou dynamicky měnit.

Vizuální vlastnosti vrstev jsou deklarovány elementy `<esri:SimpleFillSymbol>` pro polygonové vrstvy, `<esri:SimpleLineSymbol>` pro liniové vrstvy a

<esri:SimpleMarkerSymbol> pro bodové vrstvy. Tyto elementy mají atributy jako např. color pro nastavení barvy, alpha který nastavuje průhlednost, outline pro obvodové linie a width pro šířku linie. Hodnoty některých atributů jsou nastaveny „napevno“ - obsahují konkrétní hodnotu např. v atributu color je nastavena barva v hexadecimálním kódu, nebo odkazují skriptem na příslušný ovládací prvek, ve kterém si může uživatel sám zvolit hodnotu z určitého rozsahu např. color může odkazovat na objekt <mx:ColorPicker> ve kterém si uživatel vybere barvu z palety. Popup okna představují kontejnery <s:TitleWindow>. Uvnitř těchto oken jsou jednotlivé ovládací prvky umožňující uživateli výběr z nabídky předpřipravených možností, měnit vizuální vlastnosti vrstev a skrývat jednotlivé vrstvy. Mezi tyto prvky patří např. rolovací seznamy <s:DropDownList>, zaškrtávací políčka <s:CheckBox>, posuvníky <mx:HSlider>, palety barev <mx:ColorPicker> a číselníky <s:NumericSpepper>.

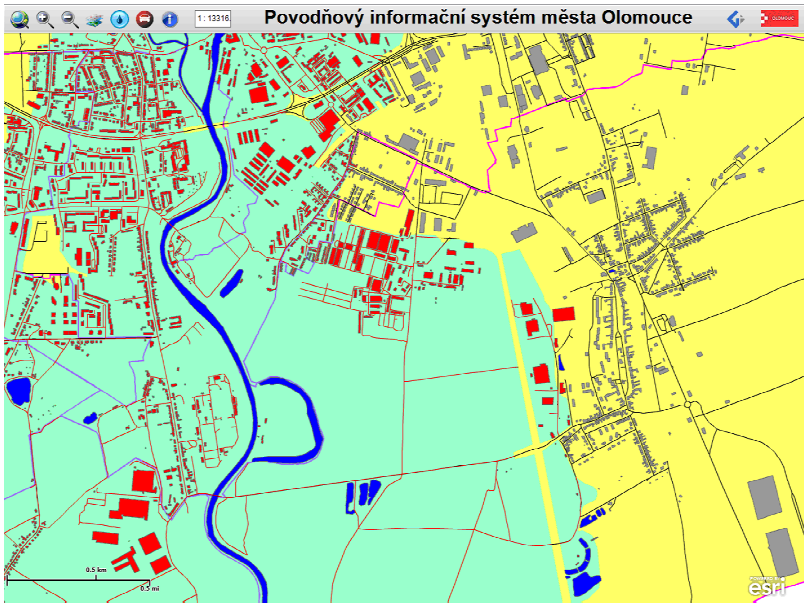
Třetí část kódu uzavřená mezi elementy <fx:Script> představuje skripty a je tvořená čistě jen jazykem ActionScript. V této části jsou deklarovány funkce spouštěné metodami objektů např. metodou click objektu <s:Buton>. Obecný tvar deklarace funkce v ActionScriptu je public function nazevFunkce():void {teloFunkce}. Klíčové slovo public je zde modifikátorem přístupu, který se stará o viditelnost funkce. Modifikátor public znamená, že lze k funkci přistupovat z libovolného souboru. Ještě lze použít modifikátory private, který omezuje přístup pouze na objekty dané tříd, protected jež je dostupný objektům dané třídy a jejich potomkům a internal, který je přístupný objektům uvnitř daného balíku. Je také možné si definovat vlastní modifikátor. (Borek 2011) Do kulatých závorek lze vložit argumenty funkce. Klíčové slovo void uvádí, že funkce nevrací návratovou hodnotu, pokud by vracela, změnilo by se na return. Těla funkcí obsahují rozhodovací struktury, které byly v aplikaci řešeny pomocí větvících cyklů if else nebo jeho zkráceným zápisem ve formě ternárního operátoru pomocí otazníku a dvojtečky. Pomocí skriptů bylo řešeno přepínání vrstev, změny vizuální vlastností vrstev jako barvy, průhlednosti, obrysové linie polygonů, zobrazování a skrývání popup oken, aktualizace měřítka. Část zdrojového kódu níže reprezentuje část skriptu sloužícího k přepínání vrstev.

```
public function zobrazitPovodne():void{
 if
(Q1997.visible==true||Q2006.visible==true||Q100.visible==true||Q20.visib
le==true||
 Q5.visible==true||bystrice.visible==true){
       Q1997.visible=false;
       Q2006.visible=false;
       Q100.visible=false;
       Q20.visible=false;
       Q5.visible=false;
```

```
       bystrice.visible=false; }
else if
(Q1997.visible==false&&Q2006.visible==false&&Q100.visible==false&&
 Q20.visible==false&&Q5.visible==false&&bystrice.visible==false){
  drop.selectedItem=="Povodeň 1997" ? Q1997.visible=true :
Q1997.visible=false;
      drop.selectedItem=="Povodeň 2006" ? Q2006.visible=true :
Q2006.visible=false;
      drop.selectedItem=="Q 100" ? Q100.visible=true :
Q100.visible=false;
      drop.selectedItem=="Q 20" ? Q20.visible=true : Q20.visible=false;

      drop.selectedItem=="Q 5" ? Q5.visible=true : Q5.visible=false;
      drop.selectedItem=="Bystřice" ? bystrice.visible=true :
bystrice.visible=false;

  }

 }
```

5 POPIS APLIKACE

Layout aplikace se skládá z mapového pole a horizontální lišty. Na liště jsou soustředěny základní ovládací a navigačními prvky. Další ovládací prvky jsou umístěny na popup oknech, jež se otvírají kliknutím na ikony na liště. Krom ikon je na liště ještě název aplikace, měřítko a loga katedry geoinformatiky Univerzity Palackého a Magistrátu města Olomouce, které fungují jako odkazy na webové stránky těchto institucí. Obrázky ikon byly převzaty z AcrGIS Viewer for Flex. V mapovém poli se pak ještě nachází grafické měřítko, logo společnosti Esri a pravém horním rohu je malé tlačítko se šipkou, které zobrazuje legendu.

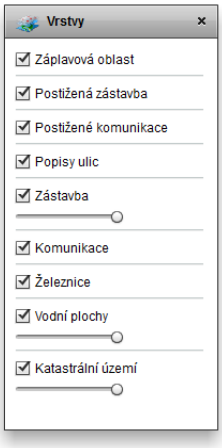


Obr. 4 Layout aplikace

 První ikona lupy a zeměkoule po kliknutí vrací úvodní obrazovku tj. výchozí měřítko 1 : 80 000 a výchozí polohu obrazovky.

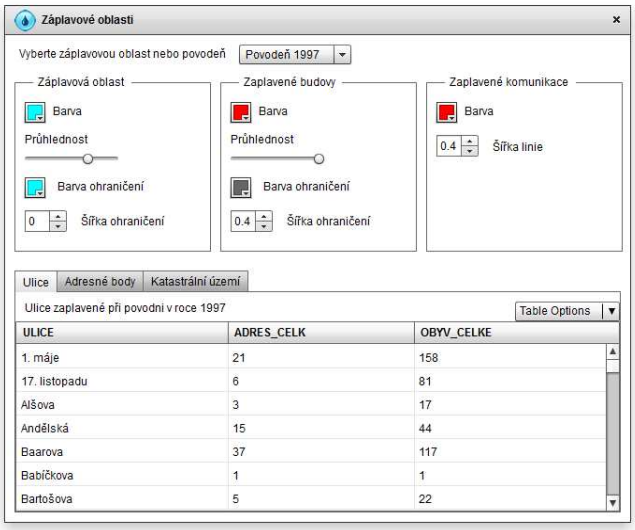
 Druhá a třetí ikona slouží k zoomování, první ikona lupy se znaménkem plus funguje jako zoom in, druhá se znaménkem mínus jako zoom out. Tyto ikony nahrazují prvek `zoomSlider`, který byl v aplikaci odstraněn, ale jinak je defaultně součástí layoutu.

 Tato ikona zobrazuje popup okno, které plní funkci jednoduchého správce vrstev. V tomto správci je pouze možnost zaškrtnout v checkboxu jestli se má vrstva zobrazit či nikoli. U polygonových vrstev lze nastavit průhlednost.



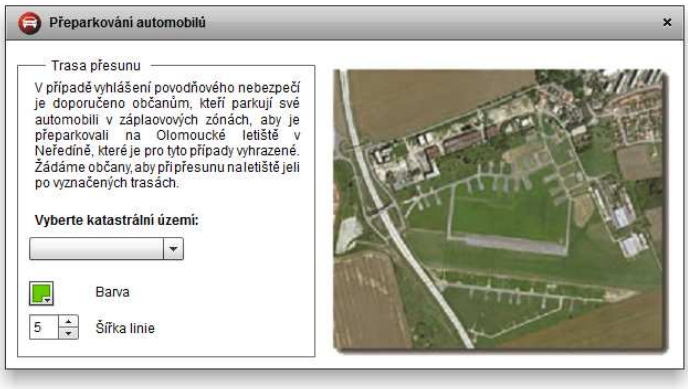
Obr. 5 Popup okno Vrstvy

 Tato ikona zobrazuje další popup okno s názvem Záplavové oblasti. V tomto okně si uživatel může zvolit, který průtok se má zobrazit – je možné vybrat mezi teoretickými průtoky Q_{100}, Q_{20}, Q_5, historickými povodněmi z let 1997 a 2006 na řece Moravě a stoletým průtokem na řece Bystřici. Po výběru průtoku se automaticky v mapovém poli červenou barvou vyznačí ulice a zástavba, které budou při tomto průtoku zatopeny. Okno dále obsahuje nástroje pro úpravu vizuálních vlastností vrstev záplavových oblastí, postižených ulic a budov a také obsahuje tabulky s přehledem zaplavených ulic, počtem adresných bodů v jednotlivých ulicích a obyvatel, které bude nutné evakuovat. Údaje v tabulkách jsou vybrané atributy vrstev ulic, adresných bodů a zaplavených katastrálních území – tyto vrstvy nejsou v aplikaci viditelné, mají nastavenou průhlednost pomocí vlastnosti `alpha` a slouží jen zobrazování informací z atributových tabulek. V tabulce se zobrazují pouze informace z nazoomovaného mapového výřezu. Je-li zoom pouze na jednu ulici, jsou v tabulce informace pouze o této ulici.



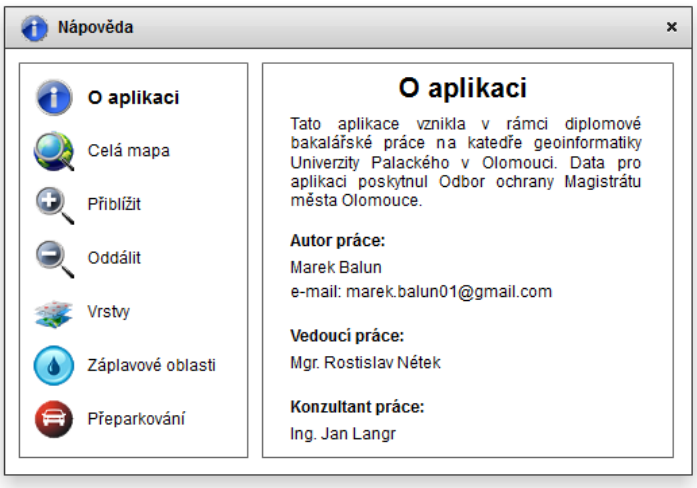
Obr. 6 Popup okno Záplavové území

Ikona automobilu otevírá okno s názvem přeparkování automobilů. V tomto okně uživatel vybere z nabídky katastrální území, ve kterém parkuje a v reakci na jeho výběr se na mapě ukáže trasa z příslušného katastrálního území na Olomoucké letiště v Neředíně. V okně je dále možné vybrat barvu a šířku linie znázorňující trasu.



Obr. 7 Popup okno Přeparkování automobilů

Po kliknutí na tuto ikonu se otevře okno nápovědy. Okno nápovědy je rozděleno na dva boxy. V levém boxu je seznam ikon, které se nacházejí na horní horizontální liště s popisy. Po kliknutí na ikonu se v pravém boxu objeví stručný popis a nápověda k danému ovládacímu prvku.

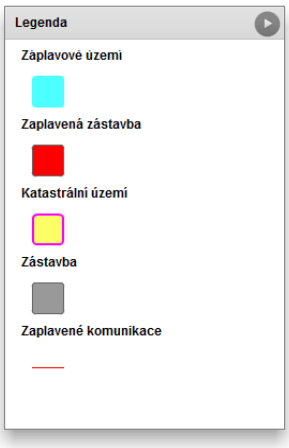


Obr. 8 Popup okno Nápověda

Na liště je ještě pole s číselným měřítkem. Pole s měřítkem nemá pouze informační funkci, ale přijímá i vstupy od uživatele. Měřítko je možné zadávat jak s jedničkou a dvojtečkou tak i bez ní.

Obr. 9 Horizontální lišta, v červeném rámečku je vyznačeno pole pro vkládání textu s měřítkem

V mapovém poli v pravém horním rohu se nachází šipka, která zobrazuje panel s legendou. Legenda má pouze informativní charakter – nelze na tomto panelu upravovat nebo měnit vizuální vlastnosti vrstev, ty se upravují na předešlých panelech. Legenda dynamicky reaguje na změny provedené v mapě. Při vypnutí vrstvy dotyčná vrstva zmizí i z legendy, při změně vizuálního stylu se symbol automaticky upraví i v legendě.



Obr. 10 Vysouvací panel s legendou

6 „ADMINISTRACE“ – MOŽNOST ÚPRAV

Aplikace neobsahuje žádné rozhraní pro administraci ani neumožňuje autorizovaný přístup, protože se zde nepředpokládají příliš časté úpravy jako doplňování nebo změna dat. Datové vrstvy jsou k aplikaci připojeny na „pevno“ a uživatel nemá možnost přidávat nové vrstvy nebo ty stávající odstraňovat, může je pouze skrývat. Webové technologie stárnou doslova přes noc a v době, kdy budou již zastaralá data kterých aplikace využívá bude také značně zastaralá i technologie, na které je tato aplikace postavená, proto je na zvážení jestli nebude potom lepší vytvořit novou.

V případě, že by bylo potřeba v aplikaci upravit nebo přidat data je nutné tuto úpravu provést přímo ve zdrojovém kódu. Soubor obsahující zdrojový kód se nachází ve složce src a má příponu `.mxml`. Pro přidání nové vrstvy musíme v kódu najít kontejner `<esri:Map>` a přidat do něj další objekt `<esri:FeatureLayer>`, který slouží k připojování vrstev publikovaných ArcGIS for Server. Samotnou službu připojíme přidáním její adresy do atributu `url` tohoto objektu.

```
<esri:Map>
<esri:FeatureLayer
      name="Katastrální území"
      id="katastralniUzemi"
      symbol="{katastrSymbol}"
      visible="{checkKatastr.selected}"
      url="http://server.example.cz:6080/data/0“ />
</esri:Map>
```

Takto připojená vrstva bude mít ale implicitně nastavené vlastnosti jako, barva průhlednost, styl atd. tak jak je nastavil administrátor serveru, který tuto službu publikoval. Pokud bychom chtěli nastavit vlastní symbologii, musíme do deklarací mezi tagy `<fx:Declarations>` přidat element `<esri:SimpleMarkerSymbol>`, `<esri:SimpleLineSymbol>` nebo `<esri:SimpleFillSymbol>` dle toho jestli se jedná o vrstvu bodovou, liniovou nebo polygonovou. Jednotlivé vlastnosti se nastavují jako atributy tohoto objektu. Barva vrstvy se nastavuje v atributu `color`, musí se zadat jako hexadecimální číslo ve formátu 0xff0000 (červená barva). Průhlednost se nastavuje atributem `alpha`, přičemž hodnota 0 je úplně průhledná – neviditelná vrstva a hodnota 1 neprůhledná vrstva. Atribut `width` je šířka u linie, hodnota se nastavuje v milimetrech. `Outline` je atributem pro obrysovou linii polygonové vrstvy, do tohoto atributu se na rozdíl od ostatních nezadávají konkrétní hodnoty, ale odkazuje skriptem na objekt `<esri:SimpleLineSymbol>`, ve kterém se styl obrysové linie definuje stejně jako styl liniové vrstvy. Zápis by mohl vypadat následovně `outline=“{obrysovaLinie}“`, `obrysovaLinie` je zde identifikátorem objektu `<esri:SimpleLineSymbol>`, který se

zapisuje do atributu `id`. Nakonec je potřeba navázat námi definované vlastnosti k připojené vrstvě. To by se udělalo podobně jako v předešlém příkladu – identifikátor objektu, ve kterém jsme vlastnosti definovali použijeme jako hodnotu atributu `symbol` u objektu `<esri:FeatureLayer>`, hodnota atributu musí být opět zapsaná ve složených závorkách – vše co se vyskytuje ve složených závorkách je interpretováno jako ActionScript výraz.

```
<fx:Declarations>
<esri:SimpleFillSymbol id="zatopenaZastavbaSymbol" color="0xff0000"
alpha="1" outline="{outlineZatopenaZastavba}" />
<esri:SimpleLineSymbol id="outlineZatopenaZastavba" color="0x666666"
alpha="1" width="0.4" />
</fx:Declarations>
```

Takto bychom sice nastavili vlastní vizuální vlastnosti, ale uživatel by je neměl možnost měnit. Pokud budeme chtít uživateli umožnit tyto vlastnosti libovolně měnit, mohli bychom příslušné atributy navázat na ovládací prvky pomocí skriptů, podobně jako v předešlých příkladech. Například na atribut `color` bychom mohli navázat objekt `<mx:ColorPicker>`, což je paleta s nabídkou barev. Syntaxe by opět vypadala podobně – `color="{paletaBarev.selectedColor}"` kde `paletaBarev` je identifikátor objektu `<mx:ColorPicker>` a `selectedColor` je atribut, který udává aktuálně vybranou barvu v paletě. Tedy jednoduše řečeno zápis říká, že hodnota atributu `color` se bude rovnat hodnotě atributu `selectedColor` v paletě. Ale vhodnějším řešením je ponechat atributům nastavené konkrétní defaultní hodnoty a změnu barvy vyřešit pomocí jednoduché funkce (procedury) umístěné mezi elementy `<fx:Script>`, vyhneme se tím případným problémům s poskakováním barev při načítání. Např. změnu barvy vrstvy bychom mohli ošetřit funkcí, kterou bude spouštět metoda `change` objektu `<mx:ColorPicker>`. Tělo funkce by pak mělo následující syntaxi `vrstva.color=ColorPicker.selectedColor`. Pro aktualizaci legendy bychom ještě přidali zápis `legenda.refresh()`, kde legenda `id` objektu `<esri:Legend>`. Příslušnou paletu barev bychom museli přidat do layoutu aplikace, nebo bychom mohli využít některé stávající – obsluhovala by potom současně více vrstev, které by měli vždy stejnou barvu.

```
<fx:Script> <![CDATA[
public function zatopenaZastavbaZmenaBarvy():void{
      zatopenaZastavbaSymbol.color=pickerZatopenaZastavba.selectedColor;
      legenda.refresh();
} ]]>
</fx:Script>
```

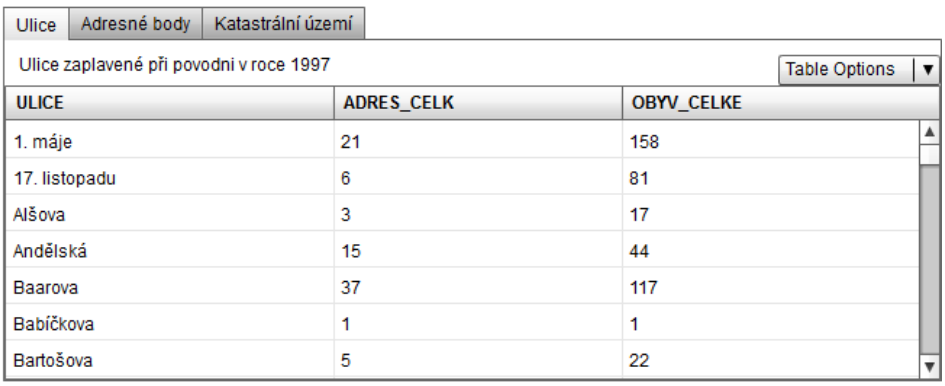
```
<mx:ColorPicker id="pickerZatopenaZastavba" top="20" left="10"
change="zatopenaZastavbaZmenaBarvy()" selectedColor="0xff0000" />
```

Uvedené syntaxe vázání dat nejsou jedinou možností, stejného výsledku lze docílit i jinými způsoby např. s využitím elementu `<fx:Binding>` a jeho atributů `source` a `destination`.

7 VÝSLEDKY

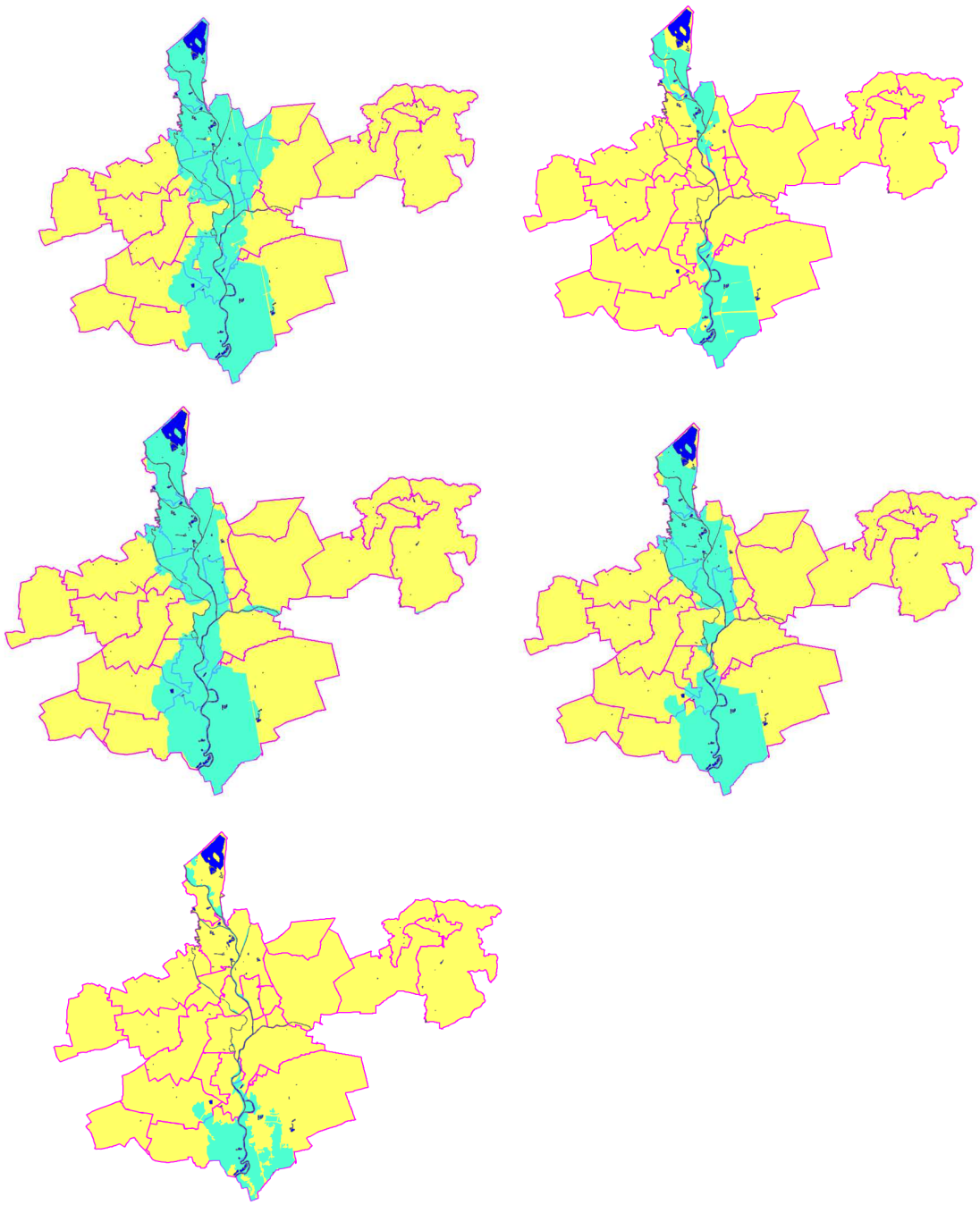
Tato práce má dva hlavní výstupy, prvním výstupem je aplikace, druhým výstupem je datová sada, které tato aplikace využívá. Datová sada byla vytvořena z podkladových dat poskytnutých Magistrátem města Olomouce pomocí metod geoprocesingu. Data jsou ve formátu shapefile. Tato datová sada obsahuje katastrální území města Olomouce, zástavbu, síť komunikací, ulice,adresné body, vodní plochy a záplavová území řeky Moravy pro Q100, Q20, Q5 a zatopená území během historických povodní z 1997 a 2006 – tyto vrstvy byly získány od Magistrátu, ale bylo je nutné upravit a opravit. Nově byly vytvořeny tematické vrstvy znázorňujících zaplavené budovy, zaplavené ulice, zaplavené adresné body a zaplavené části katastrálních území. Vrstvy ulic, katastrálních území a adresných bodů mají v aplikaci nastavenou průhlednost a vystupují pouze v podobě atributových tabulek. Data byla publikována katedrálním ArcGIS for Server a jsou dostupná v podobě REST služeb.

Hlavním výstupem je aplikace. Jedná se o webovou aplikaci postavenou na platformě Flash. Tato aplikace byla vyvíjena v rámci praxe pro Odbor ochrany Magistrátu Města Olomouce, který pro ni specifikoval své požadavky – vizualizace záplavových oblastí řeky Moravy, postižené infrastruktury, podávání informací o počtu osob je ž bude nutné evakuovat. Dalším specifickým požadavkem bylo vytipovat vhodnou lokalitu pro přeparkování osobních automobilů se zaplavených oblastí a v aplikaci potom vizualizovat trasy přesunu z postižených oblastí na tuto lokalitu. Oba tyto požadavky byly splněny. V aplikaci má uživatel možnost si vybrat jeden průtoků na řece Moravě. V závislosti na vybraném průtoku bude červenou barvou vyznačena všechna zástavba, která se nachází v záplavové oblasti pro daný průtok, kromě zástavby se vyznačí i zaplavené části ulic. V okně Záplavové oblasti se pak v tabulce zobrazí abecední seznam zaplavených ulic s počty adresných bodů v ulici a počtem obyvatel. Na další záložce je seznam adresných bodů, které se nacházejí ve zvolené záplavové oblasti. Na třetí záložce je seznam postižených katastrálních území s počty obyvatel jež bude nutné evakuovat a počtem zaplavených adresných bodů.



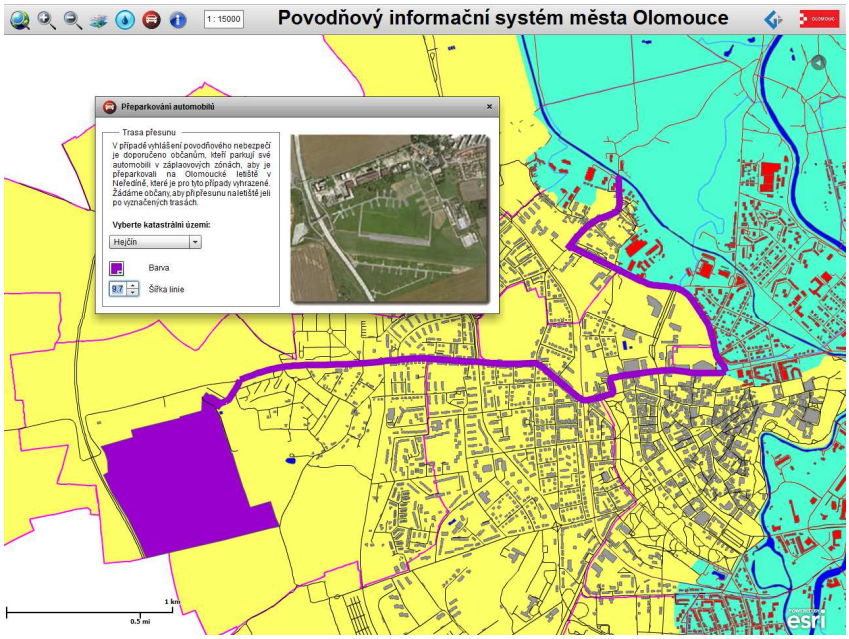
ULICE	ADRES_CELK	OBYV_CELKE
1. máje	21	158
17. listopadu	6	81
Alšova	3	17
Andělská	15	44
Baarova	37	117
Babíčkova	1	1
Bartošova	5	22

Obr. 11 Tabulky zaplavených Ulic s názvy ulic, počty adresných bodů a obyvatel



Obr. 12 Jednotlivé průtoky a povodně, mezi nimiž si může uživatel vybrat – povodeň z roku 1997, povodeň z roku 2006, Q_{100}, Q_{20}, Q_5 (zobrazeno bez zástavby a komunikací)

Pro další požadavek – přeparkování bylo po poradě s Ing. Janem Langrem z Odboru ochrany vytipováno Olomoucké letiště v Neředíně. V aplikaci jsou potom znázorněny trasy pro přesun vozidel. Uživatel si v okně Přeparkování automobilů vybere katastrální území, ve kterém parkuje svůj automobil, z vybraného katastrálního území se vyznačí trasa přesunu na letiště.



Obr. 13 Okno Přeparkování automobilů a vyznačená trasa přesunu (fialová barva) na letiště

Uživatel má možnost vrstvy skrývat a zobrazovat, u vybraných tematických vrstev si může nastavit vlastní symbologii – barvu, průhlednost, obrysovou linii a šířku linie. Uživatel nemá možnost připojovat vlastní vrstvy ani měnit pořadí vrstev. Jedním z požadavků na aplikaci byla jednoduchost - zejména jednoduché uživatelské rozhraní a k tomu by přemíra voleb a nastavení nepřispěla, navíc v aplikaci by změna pořadí vrstev neměla příliš opodstatnění.

Hotová aplikace je k dispozici jednak v podobě zdrojového kódu ve složce src – soubor s příponou .mxml nebo zkompilovaná do binárního souboru s příponou .swf. Binární soubor .swf může být přímo nasazen na server do provozu.

Výslednou aplikaci chce na svém webu zveřejnit Odbor Ochrany Magistrátu města Olomouce. Magistrát má vlastní ArcGIS for Server na kterém si pro aplikaci publikuje data, takže aplikace bude fungovat nezávisle na katedrálním serveru. Aplikace bude muset být upravena pro tento server – bude nutné změnit adresy služeb.

8 DISKUZE

Při řešení praktické části se vyskytlo několik více či méně závažných problémů. Asi největším problémem bylo „poskakování“ barev u datových vrstev při načítání. Po spuštění aplikace se načetly vrstvy s nesprávnými barvami např. záplavová oblast se načetla v černé barvě a ustálily se až po otevření popup oken s barevnými paletami pomocí kterých se nastavuje barva příslušných vrstev. Problém spočíval v tom, že atributy `color` u elementů `<esri:SimpleFillSymbol>` a `<esri:SimpleLineSymbol>` neobsahovaly hodnoty barev, ale pouze odkazovali skripty na palety barev - objekty `<mx:ColorPicker>`. Tyto palety se nacházejí na popup oknech, které jdou definovány pouze v oddíle deklarací. Po spuštění programu se vytvořily pouze instance těchto palet v paměti, ale aplikace s nimi dál nepracovala a nenastavila příslušným vrstvám barvy označené v těchto paletách. Správné barvy se nastavily až po tom co uživatel kliknul na liště na ikonu Záplavové oblasti – popup manager vytvořil instanci okna i v layoutu aplikace. Problém byl vyřešen úpravou kódu – do atributu `color` byly místo skriptů odkazujících na palety nastaveny hodnoty barev. Změna barev byla ošetřena procedurou v oddíle `<fx:Script>` kterou po kliknutí do palety spouští metoda `change`.

Dalším problém představovalo načtení ortofoto podkladu. Pro tento účel lze použít WMS služby nebo službu BaseMap nabízenou Esri. WMS službu se nepodařilo připojit, byly vyzkoušeny WMS z různých zdrojů např. ČÚZK, AOPK, HEIS ale služba nefungovala, přestože syntaxe byla správně dle dokumentace Esri na websitu ArcGIS Resources. K připojení byl použit element `<esri:WMSLayer>` v souladu s výše zmíněnou dokumentací. Chyba je zřejmě na straně ArcGIS API. Další možnost představuje připojení BaseMap což je sada podkladových rastrových map nabízená společností Esri. Mezi těmito mapami je k dispozici i podrobná letecká ortofotomapa. Problém u BaseMap je, že přesně nesedí na použité datové vrstvy v S-JTSK. Ortofotomapa je vůči použitým datům posunutá asi o 20 metrů.

Další problém může vyplívat přímo z omezení zvolené technologie. Flash platforma, na které je postavená tato aplikace není podporována dvěma nejrozšířenějšími mobilními platformami – iOS a Androidem. Uživatelům s těmito platformami tedy tato aplikace nebude fungovat. V případě iOS by bylo řešením použít Flex aplikaci využívající jako běhové prostředí Adobe AIR nebo technologii postavenou na JavaScriptu. Esri také nabízí ke stažení ArcGIS Runtime SDK for iOS , které umožňuje tvorbu aplikací využívajících služeb ArcGIS Online a ArcGIS for Server. Problém nefunkčnosti Flashe na iOS bych nepovažoval za tak závažný, na rozdíl od západních států jsou nás platformy od Applu stále velice málo rozšířené a naprostá většina uživatelů používá konkurenční Android. Právě v případě Androidu bych to spatřoval jako mnohem větší problém. Google ukončil podporu Flashe na Androidu, někdy od poloviny minulého již není možné na Google Play (dříve Android Market) stáhnout Adobe Flash Player pro Android. Google Flash Playeru odstranil s tím, že už dále nebude Flash v mobilních zařízeních podporovat. Řešením by zde mohla opět být technologie postavená na

JavaScriptu nebo Flex aplikace využívající jako běhové prostředí Adobe AIR. I pro Android Esri nabízí ke stažení ArcGIS Runtime SDK for Android.

Aplikace vyvinutá v rámci této práce není přizpůsobená pro mobilní prohlížeče, zejména pro malé displeje smartphonů. Předpokládá se použití PC případně tabletů s Windows 8 nebo Windows RT – na tabletech s Androidem a iOS není Flash podporován – viz výše. Jako největší slabinu aplikace spatřuji zdlouhavé načítání připojených dat, které může být v případě pomalého internetového připojení problematické.

9 ZÁVĚR

V rámci této práce byla vytvořena aplikaci pro potřeby krizového managementu. Konkrétně pro podporu evakuace obyvatel ze zatopených objektů. Aplikace je postavená na RIA koncepci a využívá servisně orientované architektury. Aplikace byla vytvořena v technologii Apache Flex s využitím ArcGIS API for Flex. Pro svůj běh využívá platformu Flash – jedná se tedy o webovou aplikaci. Aby byla dodržena koncepce servisně orientované architektury, aplikace neobsahuje žádná data ale připojuje je jako služby publikované prostřednictvím ArcGIS for server.

Výstupem z práce je tedy jednak samotná aplikace a také datová sada publikovaná na serveru v podobě služeb. Zájmovým územím je město Olomouc. Hlavním účelem aplikace je vizualizace záplavových oblastí řeky Moravy v Olomouci. V závislosti na zvoleném průtoku se v aplikaci vyznačí zaplavené budovy a ulice a v tabelární podobě se zobrazí jednotlivé zaplavené ulice a adresné body s počty obyvatel, které bude nutné evakuovat.

Aplikace byla vyvíjena v rámci praxe na Odboru ochrany Magistrátu města Olomouce, který pro ni také poskytnul základní data. Magistrát projevil o výslednou aplikaci zájem a chtěl by ji zveřejnit na svých webových stránkách.

POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

ANTUŠÁK, E., Z. KOPECKÝ: *Úvod do teorie krizového managementu I.,* 2. vydání. Praha, 2003. ISBN 80-245-0548-7.

AŠENBRYL, O.*: Tvorba internetových aplikací pomocí Rich Internet Application AJAX. Jihočeská univerzita v Českých Budějovicích, 2008.*

BOREK, B: *Adobe Flex: Kompletní průvodce tvorbou interaktivních aplikací*. Brno: Computer Press a.s., 2011. ISBN 978-80-251-2765-0.

CUTTER, S. L.: *GI Science, Disasters, and Emergency Management*. In: Transactions in GIS, Vol. 7, Issue 4, pp 439–446, 2003.

ERL, T.: *SOA: Servisně orientovaná architektura*. Brno: Computer Press a.s., 2009. ISBN 978-80-251-1886-3.

CHOLEVA, L.: *JavaFX platforma nejen pro internetové prezentace*. Praha, 2011. České Vysoké Učení Technické.

JOHANNSON, H.: *Rich Web Map Applications.* Chalmers University of Technology 2010.

JOHNSON, R.: *GIS Technology for Disasters and Emergency Management. Esri White Paper, 2000.*

KONEČNÝ, M. et al: *Dynamická geovizualizace v krizovém managementu*. Muni Press, Brno, 2011. Masarykova Univerzita.

LACKO, L.: *Silverlight: Výukový průvodce tvorbou interaktivních aplikací. Brno: Comapute Press a.s.,* 2010. ISBN 978-80-251-2716-2.

MORITZ, F.: *Rich Internet Applications (RIA): A convergence of user interface paradigms of web and desktop exemplified by JavaFX*. Zweibrücken, 2008. University of Applied Science.

ROUDNÝ, R., LINHART, P.: *Krizový management I.: Ochrana obyvatelstva, mimořádné události*. Pardubice, 2004. ISBN 80-7194-674-5.

SCHREINER, V.: *Implementace SOA pomocí moderních ICT principů*. Brno, 2007. Masarykova Univerzita.

TARALDSVIK, M.: *Exploring the future: is HTML5 solutions for GIS applications on the world wide web*. 2011. Norwegian University of Science and Technology.

VELTE, A. T., VELTE, T. J., ELSENPETER, R.: *Cloud computing: Praktický průvodce*. Brno: Computer Press a.s., 2011. ISBN 978-80-251-3333-0.

VOŽENÍLEK, V.: *Diplomové práce z geoinformatiky*. Vydavatelství Univerzity Palackého, Olomouc, 2002, UP, 31 s.

WEISS, P., RYCHLÝ, M.: *Architektura orientovaná na služby, návrh orientovaný na služby, webové služby*. Brno, 2007. Vysoké Učení Technické v Brně.

Apache Flex: [online]. [cit. 2013-08-12]. Dostupné z: http://flex.apache.org

GIS Portál HZS ČR: [online]. [cit. 2013-08-12]. Dostupné z: http://www.krizove-rizeni.cz/wordpress/?p=150

OpenLaszlo: [online]. [cit. 2013-08-12]. Dostupné z: http://weblog.openlaszlo.org

SUMMARY

The aim of this thesis is to create an application for the needs of the emergency management, specifically for the support of evacuation of inhabitants from flooded objects. This application is based on the RIA conception and it uses service-oriented architecture. The application was created in the Apache Flex technology using ArcGIS API for Flex. For its running it uses Flash platform, so this is a web application. To follow the concept of service-oriented architecture this application does not contain any data but it is connecting them like services published by ArcGIS for Server.

The output of this thesis is both application and data set published by the server as a service. The area of interest is Olomouc city. The main purpose of this application is a visualization of flood areas of Morava river in Olomouc. Depending on selected rate of flow the flooded buildings and streets will be displayed and in tabelar form will be displayed particular flooded streets and adress points with number of inhabitants that need to be evacuated.

This application was developed in practice on Department of Protection of Municipality of Olomouc, which also provided the basic data for this application.