

VYUŽITÍ ORCHESTRACE SLUŽEB PRO ŘEŠENÍ ÚLOH V RÁMCI ISKR

Martin Prager¹, Vladimír Maršík²

¹Institut geoinformatiky, HGF, VŠB-TUO,
17. listopadu 15, 708 33 Ostrava Poruba,
martin.prager.hgf@vsb.cz

²T-MAPY spol. s r.o., Technologická 372/2,
708 00 Ostrava-Pustkovec
vladimir.marsik@tmapy.cz

Abstrakt. V rámci výzkumného projektu „Orchestrace služeb pro GeoWeb“ GA 205/07/0797 řešeného na Institutu geoinformatiky VŠB-TU Ostrava probíhá výzkum možností orchestrace webových služeb z oblasti GIS, ověření praktických možností dostupných jazyků pro popis a plánování obchodních procesů. Článek shrnuje teoretický základ a v současnosti dostupné technologie spolu s příkladem praktického využití v oblasti krizového řízení. Uvedený příklad demonstruje využití a schopnosti BPEL (Business Process Execution Language) jazyku, standardu určeného k orchestraci webových služeb. Konkrétně se jedná o kompozitní obchodní proces, který by měl podpořit rozhodování v případě potřeby evakuace obyvatel z ohrožených oblastí do náhradních ubytovacích prostorů. Tento proces spojuje ve funkční celek služby jako např.: tvorba obalových zón kolem ohrožujících jevů v území, vyhledávání zájmových objektů v těchto zónách, jejich překryvné operace, hledání nejkratších cest mezi objekty, vyhledání adresy atd. Jedná se o služby, které jsou na bázi „analytických úloh“ GIS, tak i o služby jiné. Byl navržen se snahou, aby co nejvíce logiky vykonával BPEL jazyk (např. paralelní zpracování, transformace dat, třídění dat ap.). Vzhledem k nedostatku praktických řešení v oblasti orchestrace, tento příklad pozitivně přispívá k jejímu rozvoji a celkově k pokroku servisně orientované architektury.

Klíčová slova: GIS, webové služby, orchestrace, BPEL, SOA, krizové řízení.

Abstract. Utilizing web service orchestration for solving tasks in ISKR area. Within the research project “GeoWeb services orchestration” GA 205/07/0797 which is being solved by Institute of geoinformatics VSB-Technical University of Ostrava is in progress the research of possibilities of web services orchestration in the GIS area, verification for practical possibilities of available languages designed for business processes description and planing. This paper summarizes the theoretical basis and at present available technologies together with practical example from the area of emergency management. Presented example is demonstrating application and capabilities of BPEL (Business Process Execution Language), the standard designated for web services orchestration. It is composite business process which should support decisions in case of need for citizen evacuation from dangerous to other temporary accommodation places. This process merges to one logical unit services like e.g.: creating buffer zones around dangerous features in the area, finding objects of interest in these zones, their overlay operations, searching the shortest way between objects, searching address etc. It's dealing with services build on “analytic tasks” GIS as well as with other services. It has been designed with tendency, that the BPEL should perform the most of logic (e.g.: parallel processing, data transformation, data sorting etc.). Due to absence of practical solutions in the orchestration area this example is positive supplying its expansion and globally service oriented architecture improvement.

Keywords: GIS, web services, orchestration, BPEL, SOA, emergency management.

1 Úvod

Webové služby se neodvratně stávají součástí většiny informačních systémů. S rostoucím počtem volně dostupných i komerčních služeb se nabízí možnosti jejich vzájemného propojování do funkčních celků. Pouhým statickým spojováním služeb nejsme schopni využít jejich potenciál, natož potenciál servisně orientované architektury (SOA), která přitahuje zájem všech oblastí IT průmyslu a rychle proniká do hlavních chodů aplikací zásadních pro plnění obchodních operací. Proto je zapotřebí začít služby řetězit dynamicky, tzn. spojovat je dle aktuálních potřeb, možností uživatele (stav připojení, finance, požadovaná přesnost výsledků, rychlost odezvy, ap.). V současnosti se mluví o dvou způsobech řetězení webových služeb, známých jako orchestrace a choreografie [PRADP].

Základním cílem projektu Informačního systému krizového řízení České Republiky ISKŘ je vybudovat informační systém zabezpečující automatizaci procesů plánování a následné realizace souboru opatření, činností a postupů při možných nevojenských mimořádných událostech a krizových situacích. V rámci tohoto projektu řeší firma T-Mapy problematiku GIS a práci s některými registry ISVS. Součástí je vytvoření „analytických úloh GIS“ které se využívají například v procesu přípravy krizových plánů.

Právě tato oblast byla zvolena pro praktické ověření možnosti využití orchestrace služeb.

2 Orchestrace

Standardní technologie jako např. WSDL (Web Service Description Language), SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery and Integration) pracující s webovými službami nám poskytují prostředky pro jejich jednotlivý popis, lokalizaci a spouštění. I když webová služba může poskytovat mnoho metod, každý WSDL soubor popisuje doslova atomické (na nízké úrovni) funkce. Co nám však tyto základní technologie neposkytují, jsou důležité detaily, které popisují chování služby jako součást větší, více komplexní spolupráce. Když se jedná o spolupráci, která je kolekcí aktivit (metod, služeb) navržených tak, aby úspěšně plnila daný business cíl, jedná se o tzv. business proces. A právě popis kolekcí aktivit, který tento business proces vytváří je nazýván orchestrace [THANV].

2.1 Vybrané jazyky a standardy související s orchestrací

- *BPEL (Business Process Execution Language)*

Na základě jazyků WSFL (Web Services Flow Language) a XLANG (Business modeling language for BizTalk) vzniká jazyk BPEL. V současnosti se jeví jako dominantní jazyk (více informací v samostatné kapitole).

- *OWL-S/(dříve DAML-S) (Ontology Web Language for Services) [DAML]*

Sekce sémantických webových služeb projektu DAML (DARPA Agent Markup Language) vyvíjí ontologii webových služeb postavenou na OWL (Web Ontology Language), stejně tak nástroje a agentové technologie umožňující automatizaci webových služeb v sémantickém webu. OWL-S dává poskytovatelům webových služeb sadu stavebních prvků značkovacího jazyka pro popis vlastností a schopností jejich služeb v jednoznačné, strojem čitelné formě.

- *XPDL (XML Process Definition Language) [WfMC]*

Jazyk standardizovaný organizací WfMC (Workflow Management Coalition). Momentálně dostupný ve verzi 2.0, zpětně kompatibilní s verzí 1.0 a předpokládá se, že bude používán jako souborový formát pro BPMN (Business Process Modeling Notation). XPDL a BPMN specifikace se zabývají stejným modelovacím problémem, ovšem každá jiným způsobem. XPDL poskytuje XML souborový formát, který se dá použít k uložení a výměně procesních modelů mezi nástroji. BPMN prostřednictvím grafického záznamu ulehčuje lidskou komunikaci mezi obchodními uživateli a technickými uživateli komplexního obchodního procesu. V současné době je k dispozici přes 70 nástrojů, které umožňují ukládání procesního modelu do jazyka XPDL.

- *WSCI (Web Service Choreography Interface) [W3C_WSCI]*

Jedná se o specifikaci od firem Sun, SAP, BEA a Intalio. WSCI je rozhraní popisující jazyk, založený na XML, který se používá k popisu toků zpráv vyměňovaných mezi webovými službami v procesním kontextu. WSCI popisuje dynamické rozhraní webové služby, účastníci se dané výměny zpráv, prostřednictvím opětovného použití operací definovaných v statickém rozhraní. WSCI pracuje ve spojení s jazykem WSDL (Web Service Description Language), může pracovat i s jiným jazykem popisujícím služby, který má stejnou charakteristiku jako WSDL.

- *Wf-XML (XML Based Protocol for Run-Time Integration of Process Engines)[WfMC]*

Standardní protokol od společnosti WfMC (Workflow Management Coalition). Je navržen a implementován jako rozšíření OASIS (Organization for the Advancement of Structured Information Standards) ASAP (Asynchronous Service Access Protocol) protokolu. ASAP poskytuje základní možnosti pro kontrolu a monitorování asynchronních webových služeb prostřednictvím SOAP (Simple Object Access Protocol). Kontrola zahrnuje operace jako: vytvoření instance služby, její nastavení, spuštění, zastavení, informace o vzniklých chybách, ukončení a výsledku služby. V případě monitorování se jedná o sledování aktuálního stavu dané služby a historii jejího spuštění. Speciálním případem asynchronních služeb jsou obchodní procesy. Tudiž Wf-XMP rozšiřuje funkcionalitu ASAP na úroveň procesních strojů (process engine).

- *ebXML (Electronic Business using eXtensible Markup Language) [OASIS_ebXML]*

ebXML je modulární sada specifikací (tzv. „all-in-one“ řešení), která umožňuje realizovat kompletní obchodní procedury (např. výměny zpráv, definování obchodních procesů, standardizovanou výměnu dat ap.) prostřednictvím Internetu. ebXML zahrnuje pět specifikací, které mohou být implementovány zvlášť i v tandemu:

- *Business Process,*
- *Collaboration Protocol Profile and Agreement (ISO 15000-1),*
- *Messaging Services - ebXML Messaging Services (ebMS) OASIS Standard (ISO 15000-2),*
- *Registries and Repositories - ebXML Registry Information Model OASIS Standard (ISO 15000-3) a ebXML Registry Services Specification OASIS Standard (ISO 15000-4),*
- *Core Components - ebXML Core Components Technical Specification (ISO 15000-5)*

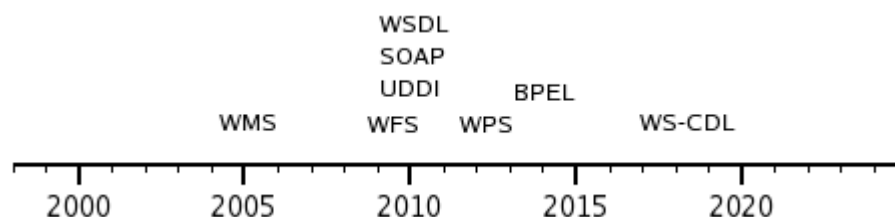
Na rozdíl od webových služeb (BPEL procesů), ke kterým může přistupovat v podstatě každý, je ebXML určený pro výměnu „well defined“ dokumentů mezi obchodními partnery za použití schváleného obchodního procesu [ebWS]. Dle [WSeb] je v současnosti ebXML výrazně méně akceptovaný nežli webové služby (pouhé 3% ve srovnání s webovými

službami).

3 Choreografie

Choreografie spolu s orchestrací nám poskytují otevřený, standardizovaný přístup pro spojování webových služeb. Choreografie nepopisuje žádné vnitřní akce, které se dějí uvnitř zúčastněných služeb. Zachycuje vztahy globálně (se všemi zúčastněnými službami je nakládáno rovnocenně), ve své povaze je více zaměřena na spolupráci, kde u každé zúčastněné strany v procesu je popsána její úloha [PRADP]. W3C popisuje choreografii jako „...*the external observable behaviour across multiple clients (which are generally Web Services but not exclusively so) in which external observable behaviour is defined as the presence or absence of messages that are exchanged between a Web Service and it's clients*“ [W3C_2006].

V případě choreografie není relevantní, jaký konkrétní jazyk byl využit při orchestraci [WSCDLFG]. V této oblasti vystupuje do popředí jazyk WS-CDL (Web Services Choreography Description Language). Jedná se o W3C specifikaci, určenou pro popis peer-to-peer spolupráce účastníků z globálního pohledu. Podle [WSCDLFG] není v současné době WS-CDL až tak potřebný. V budoucnosti, když bude fungovat nový business model, založený na servisně orientované architektuře, bude muset dojít ke zlepšení interoperability business procesů. Poté využití WS-CDL vypadá reálně. Kde se nachází na cestě k plně interoperabilním orchestrům momentálně geoinformatika? Odpověď poskytuje Obrázek 1.



Obrázek 1 - cesta k plně interoperabilním orchestrům [WSCDLFG]

4 Jazyk BPEL

Vzhledem k rozsáhlosti tohoto jazyka, je zde nastíněn pouze velmi stručně (více informací viz [OASIS_BPEL]).

Vyvinut firmami Microsoft a IBM, standardizován neprofitujícím konsorciem OASIS. Momentálně je dostupný ve verzi 2.0. Tento jazyk umožňuje specifikovat jak spustitelné, tak abstraktní procesy. Definiuje model a prostředky pro popis chování procesu, založeného na spolupráci mezi daným procesem a jeho partnery. Spolupráce mezi všemi partnery je zprostředkovávaná rozhraními webových služeb a struktura spojení na této úrovni je zapouzdřená do takzvaného *partnerLink*. BPEL proces definuje jak všechny tyto spolupráce koordinovat, aby bylo dosaženo daného business cíle, stejně tak okolnosti a logiku potřebnou k tomuto dosažení. Samozřejmě zde nechybí mechanismy pro práci s výjimkami a chybami [OASIS_BPEL].

Využívá několik XML specifikací: WSDL 1.1, XML Schema 1.0, XPath 1.0 a XSLT 1.0. WSDL zprávy a XML Schema poskytují datový model BPEL procesů. XPath spolu s XSLT ho rozšiřují o manipulaci s daty. Všechny externí zdroje a partneři jsou zde

reprezentovány jako WSDL služby. Do budoucna poskytuje rozšiřitelnost těchto standardů o novější verze, obzvláště XPath a s ním spojené standardy využívané při XML transformacích [OASIS_BPEL].

4.1 Struktura BPEL procesu [OASIS_BPEL]

```
<process...>
  <partnerLinks.../>
  <variables.../>
  <correlationSets.../>
  <faultHandlers.../>
  <compensationHandler.../>
  <eventHandlers.../>
  activities
</process>
```

Stručný popis jednotlivých elementů:

- *partnerLinks* - zde jsou definovány služby, které proces využívá a poskytuje,
- *variables* – specifikace proměnných,
- *correlationSets* – zabezpečuje doručování přicházejících zpráv odpovídajícím instancím procesů,
- *faultHandlers* – slouží ke zpracování chyb, které nastanou při běhu procesu,
- *compensationHandler* – poskytuje možnost zpětného zotavení z chyby ve specifikované oblasti,
- *eventHandlers* – dochází zde k zachycení událostí,
- *activities* – obsahují již samotný procesní tok

4.2 BPEL editory a stroje (engines)

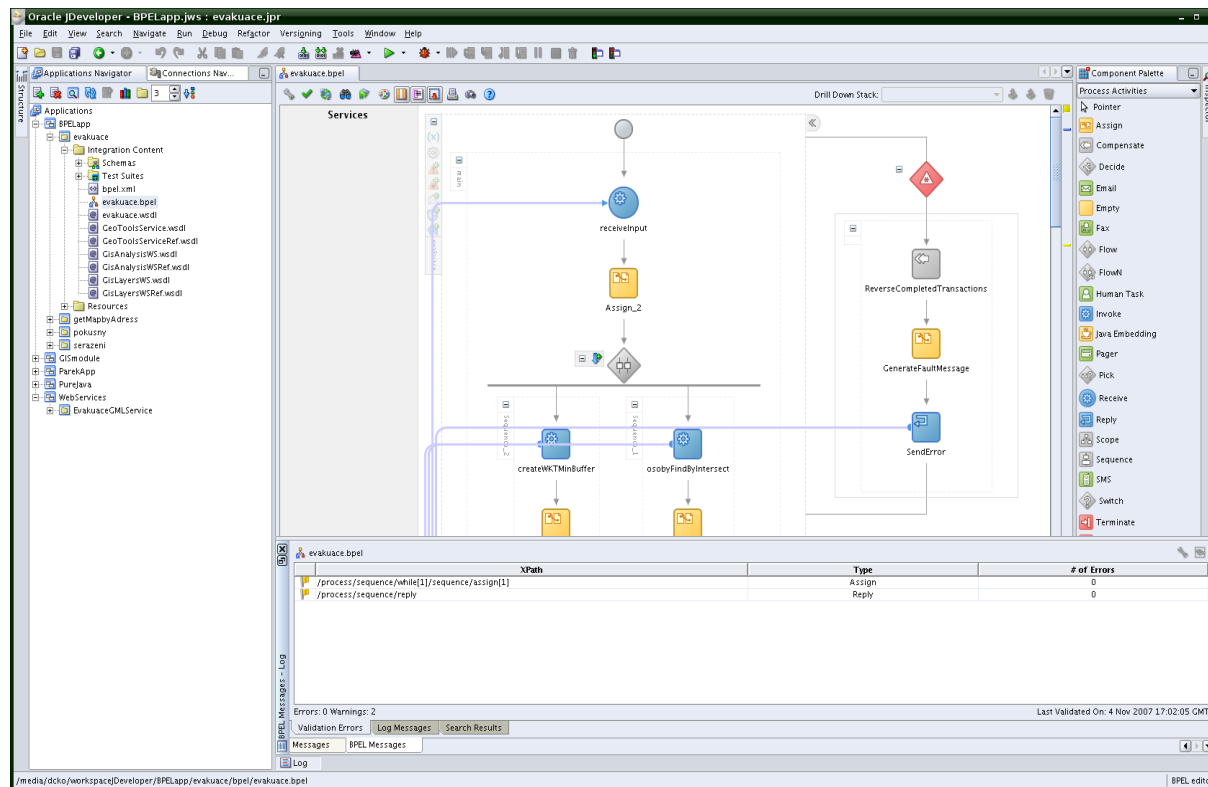
Jak je již z předcházejícího textu naprosto jasné, BPEL proces je XML dokument nejčastěji vygenerován různými grafickými návrhovými nástroji (editory). Tento návrh tvoří spíše business analytici než programátoři. Hotový proces je následně zveřejněn (zpřístupněn) pomocí nějakého konkrétního spouštěcího stroje (engine) a to buď jako rozhraní webové služby, nebo může reagovat na spouštěcí podmínky nastavené uvnitř procesu [OASIS_BPEL].

BPEL proces lze vytvořit v jakémkoliv textovém editoru, ale z praktických důvodů je tento postup značně neefektivní a vyžaduje hlubší znalosti XML. V současné době je dostupná spousta různých grafických editorů. Jedná se jak o komerční, tak i o volně dostupné a open-source produkty. Když zanedbáme jejich uživatelská prostředí, složitost, stabilitu ap., tak se většinou od sebe liší úrovní implementace BPEL a rozšiřitelností. Každá společnost většinou k danému editoru má i svůj vlastní stroj na kterém lze daný proces spouštět.

Vzhledem k trochu odlišné implementaci jazyka BPEL různými společnostmi a jeho možností rozšiřitelnosti o vlastní funkce dochází zde k nekompatibilitě těchto editorů a strojů. V případě nutnosti spouštět daný proces na jiném stroji než pro který byl určen, budeme v lepším případě potřebovat dobrou znalost obou produktů a jazyka BPEL. V horším případě to vůbec nebude možné.

Seznam některých produktů spolu se stručným popisem lze nalézt na následující adrese: <http://www.answers.com/topic/list-of-bpel-engines>, případně viz [PRADP].

Při tvorbě následujícího praktického příkladu byl používán implementovaný BPEL editor v prostředí *Oracle jDeveloper* (viz Obrázek 2). Spouštěn a testován byl nad *Oracle BPEL Process Manager*, který je součástí *Oracle* aplikačního serveru. Vše ve verzi 10g. Jedná se o volně dostupné produkty pro vývoj.



Obrázek 2 - prostředí Oracle jDeveloper

5 Praktický příklad

Cíl byl stanoven tak aby se ověřila možnost vytvoření netriviální typové analytické úlohy GIS ve formě komplexní služby vytvořené pomocí BPEL orchestrace z dílčích předem připravených služeb. Většina těchto služeb je publikována v podobě EJB Web services.

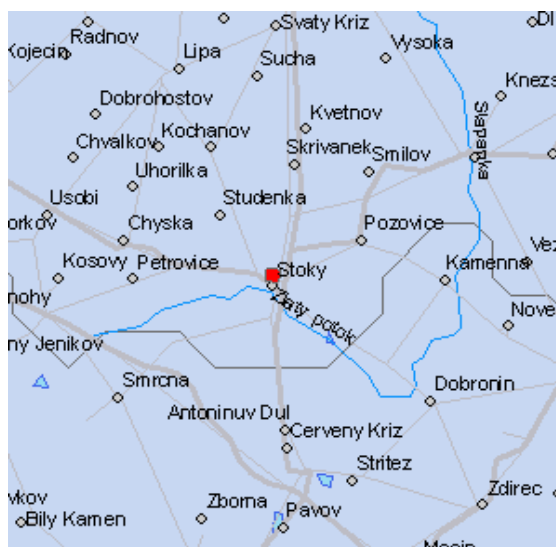
Bez odpovídající datové základny by podobný systém nemohl fungovat. Aplikace ISKŘ, včetně služeb použitých pro orchestraci využívá datový sklad, který vzniká v rámci HZS. Tento datový sklad zahrnuje velký rozsah dat pokrývající území ČR, sestavený z různých zdrojů (ČSÚ, ČUZK, Armáda ČR, CEDA) a doplňovaný daty pořizovanými v rámci HZS.

Tabulka 1 - seznam dílčích metod služeb

Název služby	Popis
Zobjekt objektFindById()	Vyhledá zájmový objekt podle hodnoty jeho identifikátoru
List<Zobjekt> objektFindByPoint()	Vyhledá zájmové objekty lokalizované souřadnicí
List<Zobjekt> objektFindByBuffer()	Vyhledá zájmové objekty uvnitř obálky vytvořené kolem zadané geometrie
List<Zobjekt> objektFindNN()	Vyhledá zadaný počet nejbližších zájmových objektů od místa určeného souřadnicí.
List<Zobjekt> objektFindByGeometryIntersect()	Vyhledá zájmové objekty které jsou v prostorovém vztahu „intersect“ se zadanou geometrií
Adresa adresaFindById()	Vyhledá adresu podle hodnoty jeho identifikátoru
List<Adresa> adresaFindByPoint()	Vyhledá adresu lokalizované souřadnicí
List<Adresa> adresaFindByBuffer()	Vyhledá geoprvky uvnitř obálky vytvořené kolem zadané geometrie
List<Adresa> adresaFindNN()	Vyhledá zadaný počet nejbližších geoprvků od místa určeného souřadnicí.
RsoPoctyOsob osobyFindById()	Vyhledá geoprvek podle hodnoty jeho identifikátoru
List<RsoPoctyOsob> osobyFindByPoint()	Vyhledá geoprvky lokalizované souřadnicí
List<RsoPoctyOsob> osobyFindByBuffer()	Vyhledá geoprvky uvnitř obálky vytvořené kolem zadané geometrie
double elevation()	Vrací nadmořskou výšku pro místo určené souřadnicí
RouteBasicInfo routeBasicInfo()	Vrací základní informace o (nejkratší nebo nejrychlejší) trase nalezené z bodu A do bodu B. Body A a B jsou zadané souřadnicemi.
String wktGeom1_within_2()	Zjistí na základě WKT geometrie zdali se jeden objekt nachází v druhém
String createWKTBuffer()	Vytvoří na základě WKT geometrie obalovou zónu kolem daného objektu
String transform2Gml()	Zajišťuje export výsledku do GML formátu

5.1 Realizace

Jedná se o klasický příklad evakuace, kdy je zapotřebí přesunout obyvatele z ohrožené oblasti do náhradních ubytovacích prostorů. Ukázkový příklad, který demonstruje kompozitní BPEL proces je situován do kraje Vysočina, kde byla uměle vytvořena ohrožená oblast. Do tohoto procesu vstupuje polygon (WKT geometrie, červený polygon, viz Obrázek 3) ohrožené oblasti (okolí obce Stoky), dále pak „minimální“ a „maximální“ (spíše doporučená) vzdálenost ve které se budou hledat náhradní prostory. Výstupem procesu je poté GML vrstva, která znázorňuje kam by se měli obyvatelé jednotlivých objektů přesunout.

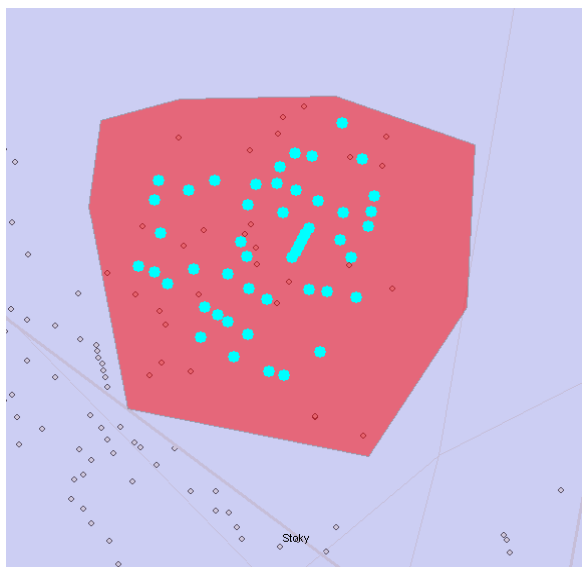


Obrázek 3 - polygon ohrožené oblasti

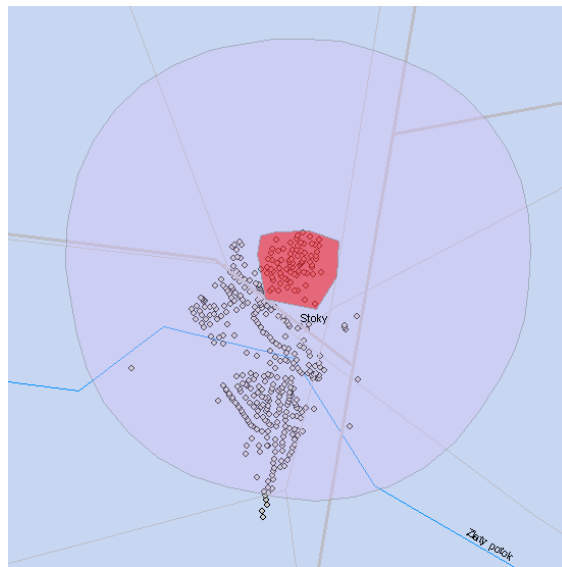
V prvním kroku procesu dochází k paralelnímu běhu dvou úloh (služeb):

- vyhledání všech obydlených objektů v ohrožené oblasti, sumarizace obyvatel určených k přesunu (viz Obrázek 5),

- vytvoření obalové zóny kolem ohrožené oblasti v minimální zadané vzdálenosti (viz Obrázek 4, v tomto případě se jednalo o 1km)



Obrázek 5 - obydlené objekty



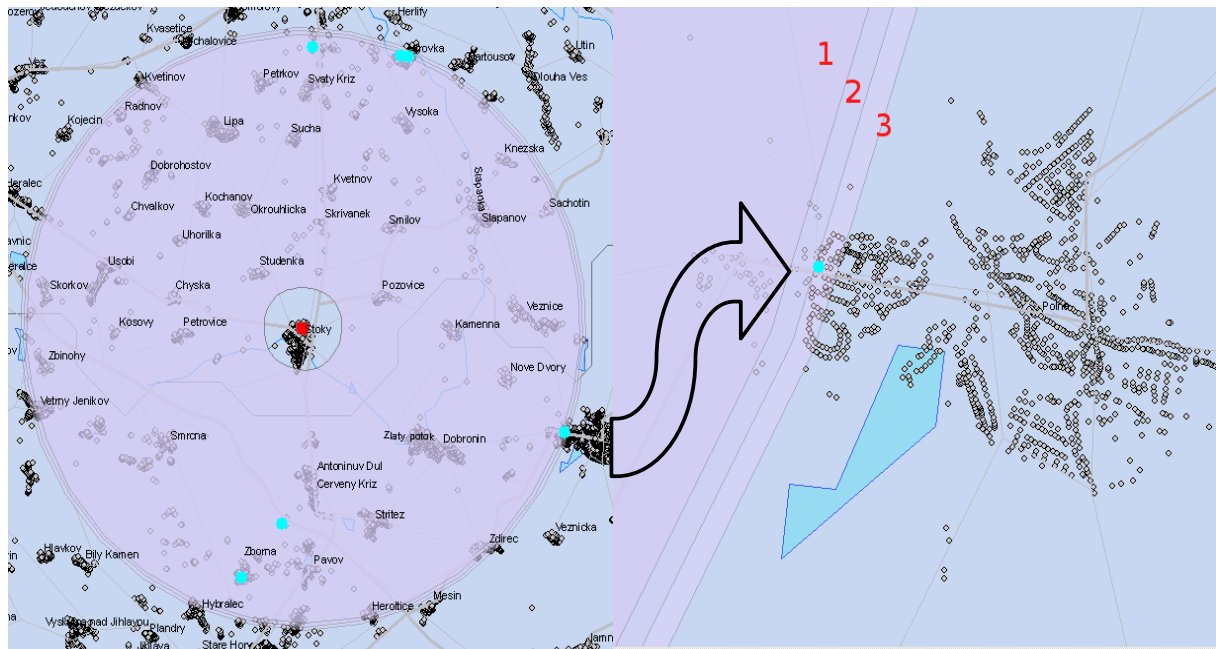
Obrázek 4 - min. obalová zóna

Druhý krok zahrnuje:

- vytvoření nové obalové zóny vymežující oblast hledání vhodných objektů pro náhradní ubytování (v tomto případě se jednalo o vzdálenost 8.4km),
- vytvoření „vyhledávacího“ polygonu rozdílem předchozích dvou obalových zón,
- vyhledání objektů s volnou ubytovací kapacitou na základě vytvořeného polygonu,
- zjištění dostupné ubytovací kapacity

V případě, že ubytovací kapacita nalezených objektů je menší nežli je zapotřebí, je tento krok opakován s tím, že v dalším cyklu je obalová zóna vždy zvětšena o určitou vzdálenost (zde 100m). Obrázek 6 vlevo znázorňuje nalezené objekty (světle modré) po třech cyklech opakování. V pravé části obrázku lze vidět detail přírůstků obalové zóny (označené červenými čísly).

V období tvorby tohoto příkladu jsme neměli k dispozici konkrétní údaje o volných kapacitách, tudíž byly tyto data náhodně vygenerována.



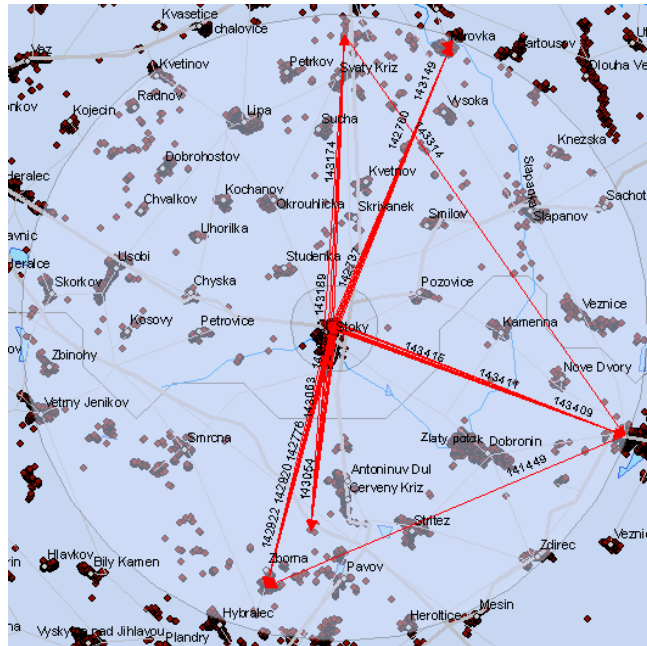
Obrázek 6 - hledání ubytovacích kapacit

Krok tři:

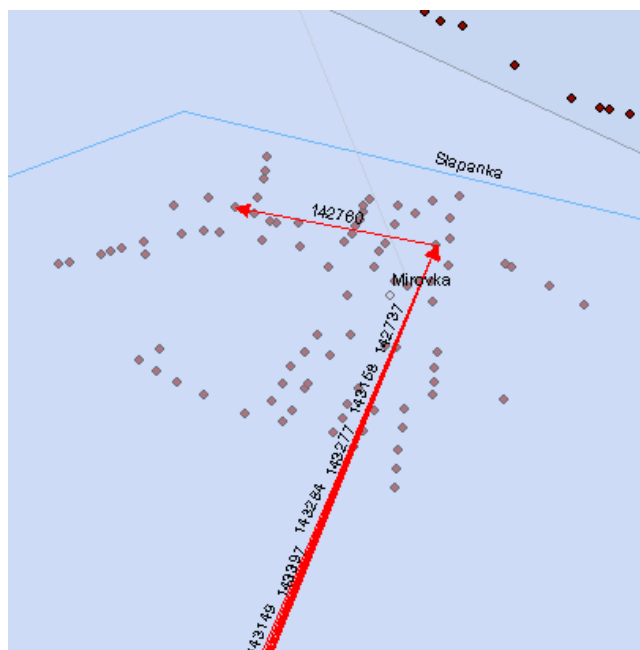
- postupně je pro každý ohrožený objekt vytvořen seznam spočítaných vzdáleností (jedná se o silniční vzdálenost) ke všem dostupným objektům poskytujícím ubytovací kapacity,
- poté je tento seznam vzestupně seřazen a může začít „virtuální“ evakuace,
- obyvatelé z ohrožených objektů jsou přesouváni do náhradních ubytovacích prostorů (začíná se přesouvat do nejbližšího objektu),
- upravují se hodnoty dostupných kapacit jednotlivých objektů (zmenšování o již ubytované osoby),
- v případě, že již v daném objektu není místo je zbytek obyvatel přesunut do následujícího objektu v seznamu s volnou kapacitou (viz Obrázek 8)

Krok čtyři:

- vytvořený seznam s přehledem kam přesunout obyvatelé z jednotlivých objektů je v tomto případě převeden na GML vrstvu (viz Obrázek 7)



Obrázek 7 - výslední GML vrstva procesu



Obrázek 8 - přesun do 2. nejbližšího objektu

Konkrétní využití jednotlivých služeb pro jednotlivé operace spolu se zmíněnými kroky této úlohy lze vyčíst z příloženého BPEL diagramu z prostředí *Oracle jDeveloper*, nebo přímo ze zdrojových kódů.

- BPEL diagram:
http://gisak.vsb.cz/~pra089/GIS_Ostrava_2008/BPEL_diagram.png
- Zdrojové kódy procesu:
http://gisak.vsb.cz/~pra089/GIS_Ostrava_2008/evakuace.zip

6 Závěr

Zvolený algoritmus uvedeného praktického příkladu není zcela jistě nejlepší a nejefektivnější, ale to ani nebylo jeho cílem. Tento příklad poukazuje na využití SOA v ISKŘ, konkrétně na orchestraci webových služeb pomocí jazyka BPEL. Veškerá procesní logika je zde vykonávána právě na jeho straně. Což je samozřejmě pro demonstraci jazyka plusem, ale na druhé straně ne vždy to nejlepší řešení. To, že když budeme chtít propojit dostupně služby „nebudeme“ muset nic dalšího programovat (zařazovat do procesu naše vlastní služby např. pro transformaci dat, řazení, filtrování a jiné zpracování) a vše si „naklikáme“ prostřednictvím nějakého grafického nástroje je pěkná představa. Bohužel, jak se v praxi ukázalo zmíněné procedury jsou časově mnohonásobně náročnější v případě nativního zpracování v BPEL nežli s využitím externího zpracování. „Jednoduché naklikání“ procesu v grafickém nástroji se také neobejde bez velmi dobré znalosti XML a jiných dovedností.

Podpora BPEL je v současné době již velmi široká a tento jazyk se jeví jako použitelný v praxi. Jak bylo uvedeno výše, skoro každá větší společnost poskytuje SOA řešení v podobě vlastního editoru a spouštěcího stroje. Při výběru konkrétního řešení, které nám bude vyhovovat je nutno myslet na zmíněnou nekompatibilitu jednotlivých řešení. Dále pokud chceme úspěšně implementovat tento jazyk měli bychom volit kompromis mezi nativním a externím zpracováním. Poté lze docílit dobrých výsledků při vytváření dynamických procesů a aktivně se podílet na rozvoji servisně orientované architektury.

7 Reference

- [DAML] DAML. [online]. 2007. Dostupný z WWW: <http://www.daml.org/>
- [ebWS] GERSTBACH, P. ebXML vs. Web Services. [online]. 2006. Dostupný z WWW: <http://www.gerstbach.at/2006/ebxml-ws/ebxml-ws.pdf>
- [OASIS_BPEL] OASIS. [online]. 2007. Dostupný z WWW: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
- [OASIS_ebXML] OASIS. [online]. 2007. Dostupný z WWW: <http://ebxml.xml.org/>
- [PRADP] PRAGER, M. Řetězení webových služeb v prostředí open source GIS. *Diplomová práce*. 2007. Ostrava. Dostupný z WWW: http://gisak.vsb.cz/~pra089/texty/DP_pra089_v1_0.pdf
- [WSCDLFG] RUŽIČKA, J; MARŠÍK, V; ŠELIGA, M; PRAGER, M; ORLÍK, A; HORÁKOVÁ, B. WS-CDL for Geoinformatics. [online]. 2006. Dostupný z WWW: <http://gisak.vsb.cz/wscs/publikace/RuzickaPaper.pdf>
- [THANV] THAN, V. Web Service Orchestration - An open and standardised approach for creating advanced services. [online]. 2003. Dostupný z WWW: http://www.eurescom.de/message/messageJun2003/Web_Service_Orchestration.asp
- [W3C_2006] W3C. [online]. 2006. Dostupný z WWW: <http://www.w3.org/>
- [W3C_WSCI] W3C. Web Service Choreography Interface (WSCI) 1.0. [online]. 2002. Dostupný z WWW: <http://www.w3.org/TR/wsci/#TOC1>
- [WfMC] WfMC. [online]. 2007. Dostupný z WWW: <http://www.wfmc.org/>
- [WSeb] YAN, Y; KLEIN, M. Web Services vs. ebXML. [online]. 2006. Dostupný z WWW: http://www.flydragontech.com/publications/2006/ws_ebxml_esc.pdf