

GUI pro orchestraci GeoWebových služeb

František Klímek

Institute of Geoinformatics, VSB-TU of Ostrava

frantisek.klimek.hgf@vsb.cz

Klíčová slova: GeoWeb, geoinformatika, webové služby, orchestrace, BPEL, GUI

Abstrakt

Součástí výzkumného projektu „Orchestrace služeb pro GeoWeb“ GA 205/07/0797 řešeného na Institutu geoinformatiky VŠB-TU Ostrava, zabývajícího se možnostmi orchestrace webových služeb z oblasti GIS a ověřením praktických možností dostupných jazyků pro popis a plánování obchodních procesů je i část zabývající se návrhem grafického uživatelského rozhraní, které by umožňovalo uživatelům na různých úrovních funkcionality pracovat s těmito orchestry služeb. Jaká je míra funkcionality, kterou jednotliví uživatelé požadují? Má jim být umožněno vyhledávat orchestry, spouštět je, parametrizovat, upravovat, či dokonce navrhovat? Na tyto otázky se snaží odpovědět následující řádky, ve kterých jsou shrnuty základní údaje o orchestraci v oblasti GeoWebu, analýza a popis charakteristik jednotlivých uživatelů i návrh samotného grafického rozhraní koncového uživatele a popis komponent, které by měl být v tomto rozhraní pro práci s orchestry k dispozici.

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 [PRAM].

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

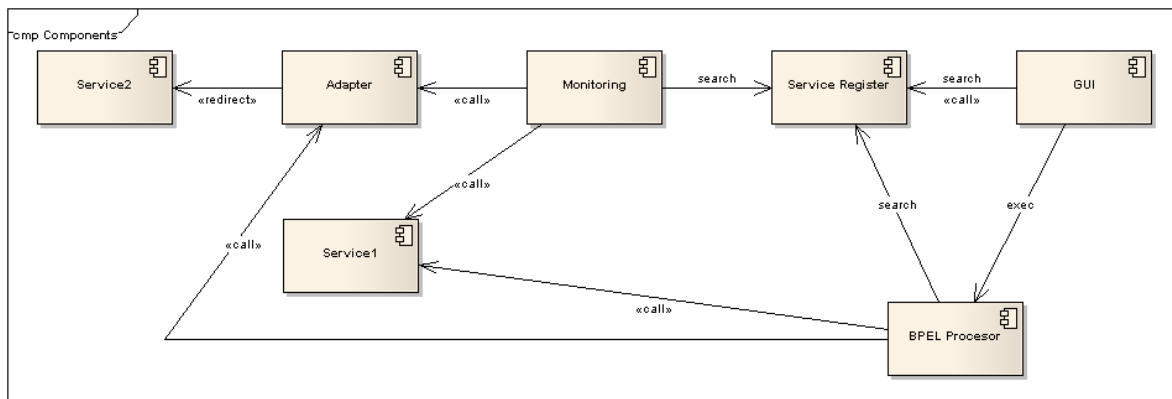
a spuštění. I když webová služba může poskytovat mnoho metod, každý WSDL soubor popisuje doslova atomické (na nízké úrovni) funkce. Co nám však tyto základní technologie neposkytují, jsou důležité detaily, které popisují chování služby jako součást větší, více komplexní spolupráce. Když se jedná o spolupráci, 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 [PRAM].

V rámci projektu proběhla analýza několika, pro orchestraci běžně používaných jazyků a po této analýze byl pro potřeby orchestrace v prostředí GeoWebu shledán jako vyhovující jazyk, jazyk BPEL. Hlavní funkcí BPEL je orchestrace webových služeb, tedy řízení souhrny funkcionality, kterou nabízí "backend" část systému, či více systémů. Tato funkcionality je dekomponována do operací, jež je možné volat přes webovou službu. Na druhé straně BPEL sám stojí za webovou službou, která definuje jeho rozhraní, tj. vstupní operace. Pro každý vstup do procesu (v BPMN objekt Start / Intermediate MessageEvent) je tedy ve webové službě, která popisuje rozhraní BPELu, jedna operace. Vstupy procesu však nemusí být výhradně na začátku, asynchronní procesy mohou mít vstupy na různých místech. Dá se tedy říci, že BPEL implementuje webovou službu. Přitom aplikace, která webovou službu používá, neví, zda se za ní skrývá proces, či zda je implementována např. EJB modulem. BPEL je rovněž nezávislý na platformě, implementace pro něj existují na platformě Java EE, .NET a jiných platformách. Proces implementovaný v jazyce BPEL pomocí jednoho nástroje by také mělo být možné přenést a spustit v nástroji jiném. Někteří výrobci byznys proces management systému (BPMS) ale používají svá vlastní rozšíření jazyka BPEL, která tuto přenositelnost znemožňují [TBPEL].

Architektura navrženého systému

Jedním z hlavních cílů grantového projektu je stanovit metodiku a popsat architekturu, jak by mohl celý komponovaný systém zahrnující služby v různorodých formách, orchestry, katalogy atd., vypadat a spolupracovat. Pro návrh grafického uživatelského rozhraní je samozřejmě nutné tuto architekturu alespoň v základní rovině znát a vědět, kde do této architektury komponenta grafického rozhraní vstupuje. V následujících několika řádcích je tedy popsána architektura systému, dle výzkumného projektu, v její aktuální podobě. Do ukončení projektu lze předpokládat ještě její další možné změny, neměly by však být nikterak dramatické. Nemělo by tedy dojít k žádné převratné změně konceptu grafického rozhraní.

Jádrem orchestrace je registr služeb, který poskytuje mechanismy pro registrování, kategorizování a hlavně vyhledávání webových služeb v reálném čase. Pokud uživatel potřebuje využít nějakou specifickou službu, prohledá daný registr. Tam získá její popis a může ji začít používat. Registr je však zaměřen nejen na služby, ale i na procesy, které svým rozhraním v podstatě službám odpovídají a obsahuje i rozhraní umožňující vyhledávání služeb dle popisu, parametrů, klíčových slov, podle výkonnostních metrik, typu atd. Právě k tomuto registru, či sadě registrů spojených a potenciálně i vzájemně spolupracujících se připojuje uživatel prostřednictvím svého grafického uživatelského rozhraní (GUI) a vyhledává potřebné služby, či procesy. Hlavním požadavkem GUI aplikace je tedy možnost komunikace s registrem služeb a formulace požadavků uživatele v jemu srozumitelné podobě a následná vizualizace odpovědí registru opět v uživatelsky přívětivé formě. Celá architektura je znázorněna na obr. 1, kde jsou zobrazeny její jednotlivé komponenty.



Obr. 1: Jednotlivé komponenty navrženého systému

- Service 1..n
- Adapter
- Monitoring
- Service register
- BPEL Procesor
- GUI

GUI

Jedním z výstupů zmiňovaného grantového projektu má být i grafické uživatelské rozhraní (GUI, z Angl. Graphical User Interface). Rozhraní má umožňovat práci s orchestry. Původní plán byl, aby v ní šly orchestry i vytvářet, toto se však zdá jako nevhodné (viz. dále v textu). K tomuto úkolu je vhodnější využít externí aplikaci. GUI by tedy mělo umět "jen" vizualizovat orchestr s aktuálními instancemi služeb a dovolit uživateli zvolit jiné instance služeb (pomocí vyhledání v registru a umožnit tak uživateli optimalizovat orchestr dle jeho individuálních požadavků). Systém by mohl řešit i potřeby uživatelů, alespoň s využitím základní sady parametrů profilu uživatele. Tj. měl by být definován kontext uživatele a podle něj ve znalostech nalezen adekvátní orchestr (resp. jeho instance).

Takto navržené a popsané GUI by mělo následně být implementováno např. jako plugin do některé z desktop GIS aplikací (jako vhodná aplikace se jeví OpenJump [OJ]), nebo přístupné jako webová aplikace, což se taktéž jeví jako velmi vhodná varianta vzhledem k možnému dopadu na velké množství potenciálních uživatelů. Druhá zmiňovaná varianta by mohla být reprezentována např. implementací společně s OpenLayers [OL], což je JavaScriptová knihovna umožňující zobrazovat mapy v prohlížeči bez závislosti na serverové části.

Uživatelé

Pokud je požadavkem navržení GUI, s kvalitním, srozumitelným a intuitivním ovládáním, je třeba netradičně začít od středu – tj. od U. GUI je především navrhováno pro uživatele,

je tedy nutnost vyjít z analýzy uživatelů, kteří budou k procesu přistupovat a analyzovat taktéž jejich potřeby. Zajisté každý z nich bude mít jiné představy a požadavky jak by mělo GUI vypadat, jakou míru detailů o daném procesu má poskytovat a co vše má umožňovat. Nejdříve je tedy potřeba podívat se na role a uživatele, kteří k procesu přistupují.

Při pohledu na některé zdroje informací o tomto tématu, např. [UBPM], [RBPM], [TIL-SOA], nebo [BOSSOA] lze nalézt velké množství různorodých rolí, které jsou více, či méně nezbytné pro správné navrhování a údržbu procesů postavených na této architektuře. Pro příklad jen jmenujme některé z nich (bližší popis jednotlivých rolí a jejich kompetencí lze nalézt ve zmiňovaných zdrojích):

- Vlastník procesu
- Vrcholový (strategický, TOP) tým, nebo manažer
- Liniový manažer
- Animátor BPM
- IT specialista
- Business konzultant
- Architekt BMS
- Procesní týmy
- Agent inovace
- Centrum inovace
- Zákazník procesu

Toto dělení vychází z prostředí enterprise aplikací a firem, které obdobné technologie a procesy postavené na servisně orientované architektuře využívají. Zajisté se nejedná o kompletní a neměnný seznam, protože v každé společnosti můžou být role upravené k aktuální potřebě společnosti a podobně [RLBPM].

V námi popisovaném prostředí však omezíme množství uživatelů pouze na následující dvě skupiny, které jsou z hlediska návrhu GUI pro registr služeb a orchestraci z našeho hlediska podstatné.

Uživatelé vytvářející proces

Jedná se o uživatele, kteří vytvářejí určitý proces a umožňují jej využívat. Zpravidla se jedná o firmy vytvářející procesy, zahrnující např. jimi vytvářené služby. Účelem je tedy využívání jejich služeb, z čehož vyplývají např. finanční zisky, nebo reklama apod. Druhou skupinou vytvářející procesy mohou být nadšenci, které zajímají tyto technologie, nebo vytvoří proces pro vlastní potřebu a rádi se o něj podělí s jinými. Tito uživatelé zpravidla mají k dispozici lidi (nebo jsou jimi sami), kteří se vyznají v návrhu a vytváření procesů, jedná se tedy o týmy, které obsahují pracovníky, kteří nejen že mají znalosti z této problematiky, ale mají obvykle k dispozici i potřebné programové vybavení nejen pro návrh, ale i pro implementaci procesu na nějaký aplikační server. Lze je tedy označit, jako uživatele vytvářející procesy,

kteří následně proces chtějí zaregistrovat do registru služeb a mají zájem na jeho využívání. Z hlediska kontextu návrhu GUI lze konstatovat, že tito uživatelé mají již většinu potřebného – ať již ve formě komerčních řešení, nebo řešení postavených na programech s otevřeným zdrojovým kódem – k dispozici, není pro ně tedy třeba vymýšlet další nástroje, které jim umožní proces vizualizovat, upravovat, apod.

Uživatelé využívající proces

Existuje však druhá skupina uživatelů, kteří jsou konzumenty takto vytvořených procesů a chtějí je pouze spouštět, či drobně upravovat (parametrizovat) apod. Jedná se tedy o uživatele, kteří si chtějí vyhledat konkrétní proces a s tím pracovat, nejčastěji pouze získat jeho popis a spustit jej. Tato práce, která spočívá v komunikaci s registrem služeb, má být uživatelsky přívětivá a nevyžadující hlubší znalosti z oblasti SOA. Žádné takové uživatelské prostředí, zvláště pro potřeby komunikace s navrženým registrem, však v současné době není k dispozici. Jaké má být? Co má uživateli zpřístupňovat?

Požadavky uživatelů

V následujících řádcích jsou popsány možné požadavky uživatelů na toto GUI. Požadavky jsou seřazeny od těch nejjednodušších, až po pokročilejší, které sahají, až na hranici návrhu procesů – tzn. na hranici s nástroji určenými pro skupinu uživatelů vytvářející procesy.

- vyhledání potřebného procesu

Hlavním a základním požadavkem uživatelů je nalezení jimi požadovaného procesu, nebo služby. Uživatelům musí být samozřejmě nabídnuto upřesnění vyhledávání v závislostech na metrikách zjistitelných z registru služeb.

- spouštění vybraného procesu

Společně s výše jmenovaným požadavkem na nalezení procesu je spuštění procesu druhým a zároveň posledním hlavním požadavkem. Kdyby GUI odpovídalo pouze těmto dvěma požadavkům, lze předpokládat, že by bylo dostačující pro valnou většinu uživatelů využívající služeb registru.

- parametrizace procesu – úprava na základě metrik

V závislosti na míře, v jaké chce uživatel s procesem pracovat lze mluvit o jednoduché a složitější parametrizaci. Jednoduchou je myšlena pouhá úprava vstupních parametrů procesu, či výběr v závislosti na jakém kritériu má být proces upraven apod. Uživatelův požadavek může např. znít – využij pouze služby, které jsou zdarma. V případě této jednoduché parametrizace je tedy práce ponechána na straně jádra orchestrace a přebírá tedy do své režie logiku výběru. Na vstup je pouze poslána šablona, kterou jádro upraví do konkrétní podoby a výsledek opět vrátí uživateli. Naproti tomu v případě složitější parametrizace přebírá zodpovědnost a logiku již na sebe sám uživatel a vybere si např. pouze zástupnou službu za jednu konkrétní, kde vyžaduje např. vyšší přesnost.

- podpora pro workflow

Některé procesy lze definovat jako dlouho trávající procesy s lidskou interakcí (Human Task Management) [UBPM], u těchto by bylo vhodné zahrnout do tohoto jednotného GUI potřebné uživatelské rozhraní tuto interakci zprostředkující. Bude-li tedy do výsledku zahrnuta některá služba, požadující zpřesňování vstupu apod., je nežádoucí, aby uživatel nějakým způsobem hledal, kde má zpřesnění zadávat, ale je vhodné, aby uživateli byla nabídnuta, např. v rámci sledování stavu orchestru, jednoduchá možnost toto zpřesnění provést. Pokud tedy v průběhu procesu dojde např. k požadavku, aby uživatel upřesnil zda analýza má být provedena pro obec Janovice nad Úhlavou, nebo Janovice (okr. F-M), uživatel toto upřesnění provede výběrem z nabízených možností přímo v navrhovaného GUI.

- zobrazení procesu

Požadavek na zobrazení procesu se vyskytne nejen u skupiny uživatelů, kteří budou chtít složitějším způsobem parametrizovat, či upravovat nabídnutý proces, ale jistě se vyskytne i skupina uživatelů, kteří budou pouze chtít vidět, které služby jsou zapojeny apod.

- uložení procesu

Po úpravě procesu do podoby žádané uživatelem, budou někteří uživatelé chtít upravený proces uložit do registru služeb, aby si zajistili jeho znovupoužitelnost v již jednou editované podobě. Zobrazení procesu a vyhledávání v závislostech na uživateli. Tento bod naplňuje potřeby uživatelů, kteří rádi využívají práce v kontextu uživatele, kdy aplikace ví o uživateli a nabízí mu výsledky určené právě pro něj. Uživateli v jehož profilu jsou tedy informace o tom, že je „spořivý“ a využívá pouze služby zdarma, nebudou nabízeny placené služby.

- sledování stavu

Umožňuje uživateli sledovat v jakém stavu se jím spuštěný proces momentálně nachází a zobrazuje informace např. o tom, jak dlouhá doba je předpokládána do dokončení spuštěného procesu.

- monitoring

Některí uživatelé budou vyžadovat bližší informace o probíhajícím procesu a budou chtít znát informace o tom, která služba je právě zapojená, na kterou službu se čeká apod. Vhodné by bylo zobrazení procesu společně s vyznačením právě probíhajících kroků.

- debugging

V případě neúspěšného provedení orchestru budou někteří uživatelé zajisté chtít vědět, proč došlo k jeho selhání, v kterém místě apod. Debuging by jim měl umožnit provést proces krokovaně a odhalit tedy slabé místo, nalézt místo – službu, která vrací nesprávné, nebo žádné výsledky apod. Na základě toho si budou uživatelé moci vybrat zástupnou službu za slabé místo v procesu a tak provést požadovaný proces např. rychleji – po odhalení pomalé služby dojde k jejímu nahrazení za službu poskytující použitelná obdobná data rychleji.

- návrhář procesů

Pro skupinu uživatelů – konzumentů procesů se jeví jako nepotřebné – viz. výše v textu.

Prvky GUI

GUI bude složeno z jistých elementů, které by byly jednak samostatně použitelné, ale jistým způsobem i provázané. Na základě práce uživatele budou interaktivně zobrazeny aktuální prvky, které by mohly být k dané činnosti vhodné. Prvky jsou vypsány v pořadí, který se snaží korespondovat s možnými požadavky uživatelů.

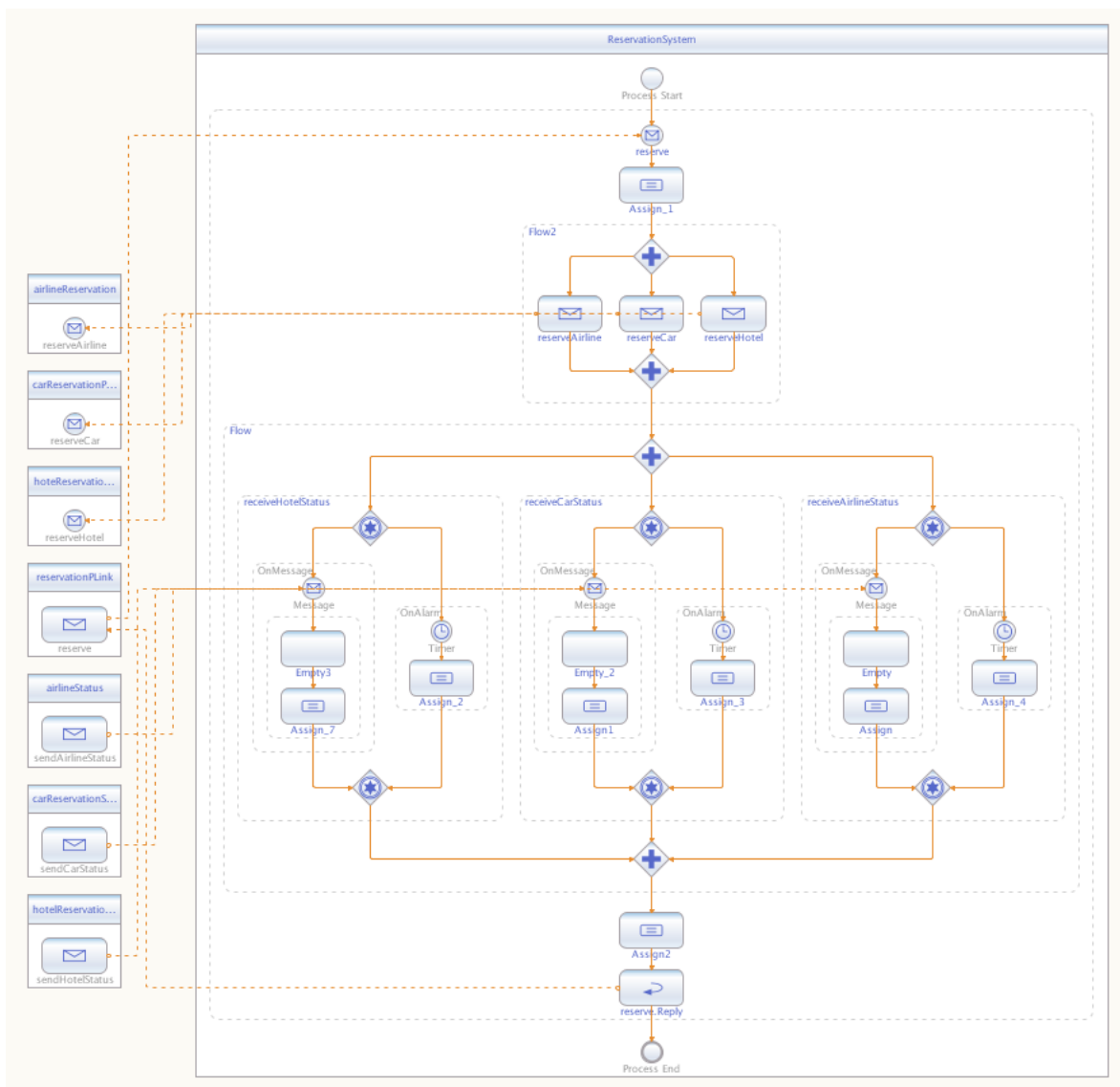
- SearchBox
- Pole s výsledky
- Dialog pro práci s procesem
- Dialog zobrazení podrobných informací o procesu
- Dialog pro jednoduchou parametrizaci
- Dialog pro vizualizaci procesu
- Mapové pole
- Tlačítko pro spuštění procesu
- Tlačítko pro uložení procesu
- Sledovač průběhu procesu
- Monitor procesu
- Debugger procesu
- Zobrazení výsledku procesu
- Přihlašovací dialog

Podoba zobrazeného procesu

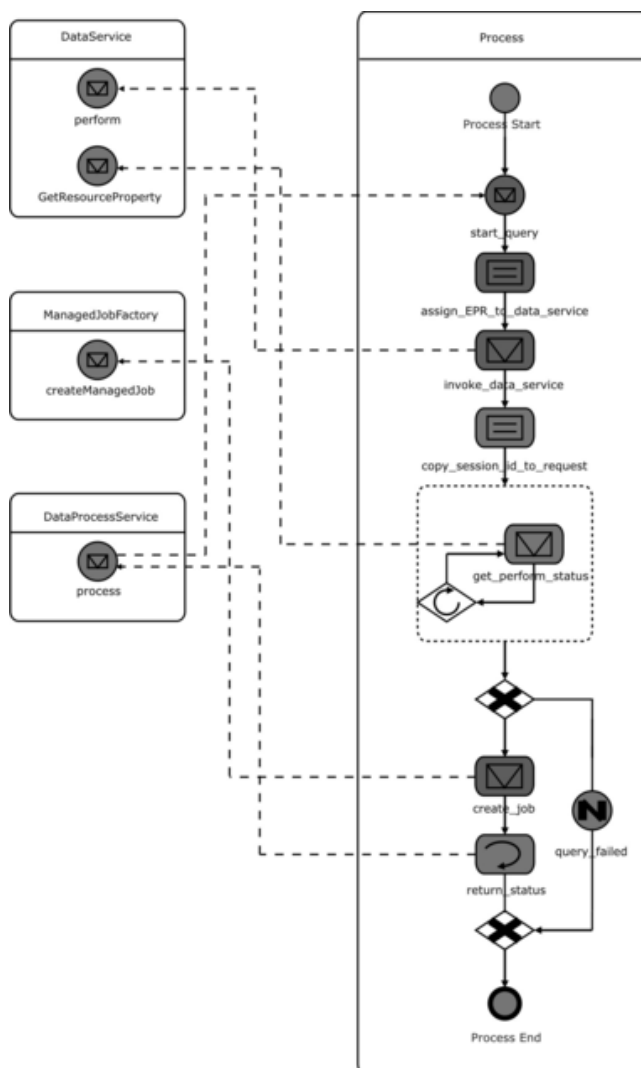
Při návrhu nového procesu se obvykle používá BPMN. Primárním cílem BPMN je však poskytnout notaci, která je snadno srozumitelná všem business uživatelům: business analytikům, kteří navrhují procesy, technickým vývojářům, kteří implementují technologie pro vykonávání procesů a managerům, kteří tyto procesy monitorují a řídí. BPMN vytváří standardizovaný most mezi návrhem business procesů a jejich implementací. Dalším cílem BPMN je umožnit vizualizaci XML jazyků určených pro návrh a vykonávání procesů (jako např. BPEL4WS) prostřednictvím business-orientované notace [REEN].

Až potom je obvykle tento zápis navrhovaného procesu, převeden do jeho implementace v BPEL, BPML, či jiném jazyce pro spouštění procesů. BPMN tedy definuje, jak převádět jednotlivé elementy a sekvence těchto elementů do jazyka BPEL. Je tedy možné model procesu do jeho spustitelné podoby převést. Díky poměrné volnosti modelování v BPMN však nebývá obvykle možné vygenerovat BPEL automaticky, některé BPMS nástroje však tuto funkci nabízejí, a to za cenu určitých omezení při samotném modelování procesu [UBPM3]. Možnost Automatické generování lze zajistit i striktním dodržením pravidel definovaných v BPMN.

Oproti BPMN nemá BPEL, žádnou implicitní grafickou reprezentaci a slouží k popisu procesu už na vykonatelné úrovni, v podstatě jde o programový kód. Právě BPEL však bude pro potřeby vizualizace procesu v GUI přístupný z registru. Některé z programových nástrojů sloužících pro potřeby tvorby aplikací založených na SOA, jako jsou např. NetBeans [NB], nám usnadňují přechod z BPMN na BPEL tím, že se snaží používat stejné grafické prvky, to ale rozhodně nebývá pravidlem [TBPEL]. Tato cesta se vzhledem k tomu, že v registru budou služby uloženy ve formě jazyka BPEL, jeví jako vhodná. Proces je vizualizován v podobě, který je při troše snahy pochopitelný i pro mírně pokročilé uživatele a lze předpokládat, že právě pokročilejší uživatelé budou vyžadovat pokročilejší funkcionalitu práce s orchestry. Na následujícím – obr. 2 – je zobrazen ukázkový proces vytvořený a vizualizovaný právě v programovém produktu NetBeans a na obr. 3 je proces vizualizovaný pomocí WEEP Engine [WEEP], který umožňuje konverzi souboru BPEL do podoby SVG, nebo PNG. Tento engine by mohl být dobře využitelný pro potřeby funkční implementace popisovaného GUI.



Obr. 2: BPEL proces vizualizován pomocí NetBeans [NB]

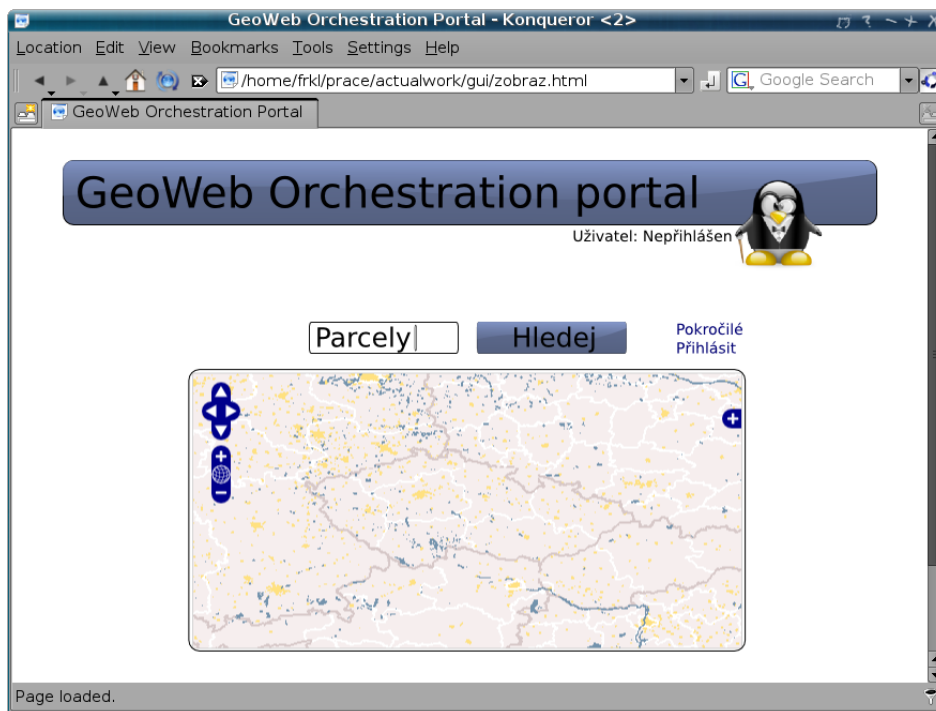


Obr. 3: BPEL proces vizualizován pomocí WEEP Engine [WEEP]

Scénář práce

V následujících řádcích je popsán možný scénář práce s grafickým rozhraním pro orchestraci GeoWebových služeb. V závislosti na formě – desktop aplikace, či webovém rozhraní, uživatel zahájí práci vyvoláním nabídky menu v aplikaci, pro niž bude např. vytvořen plugin, nebo spustí internetový prohlížeč a zadá webovou adresu, kde bude klientská aplikace ve formě webové aplikace. Následně bude uživateli zobrazeno následující výchozí dialogové okno, které bude obsahovat textové pole a mapové pole, oboje určeno k vyhledávání služeb. Bude zde i volba pokročilá, které umožní zpřesnit požadovaný vyhledávaný výraz, nebo již v tuto chvíli určit, aby výsledné orchestry byly vráceny parametrizované, např. dle ceny. Měla by zde být i možnost přihlášení uživatele, kterou by následně byly ovlivněny vyhledávané služby a procesy.

Uživateli budou následně zobrazeny vyhledané služby a orchestry s možností zobrazení si více podrobnosti. Pro zobrazení podrobností geografických bude využito opět komponenty zprostředkovávající mapové výstupy.

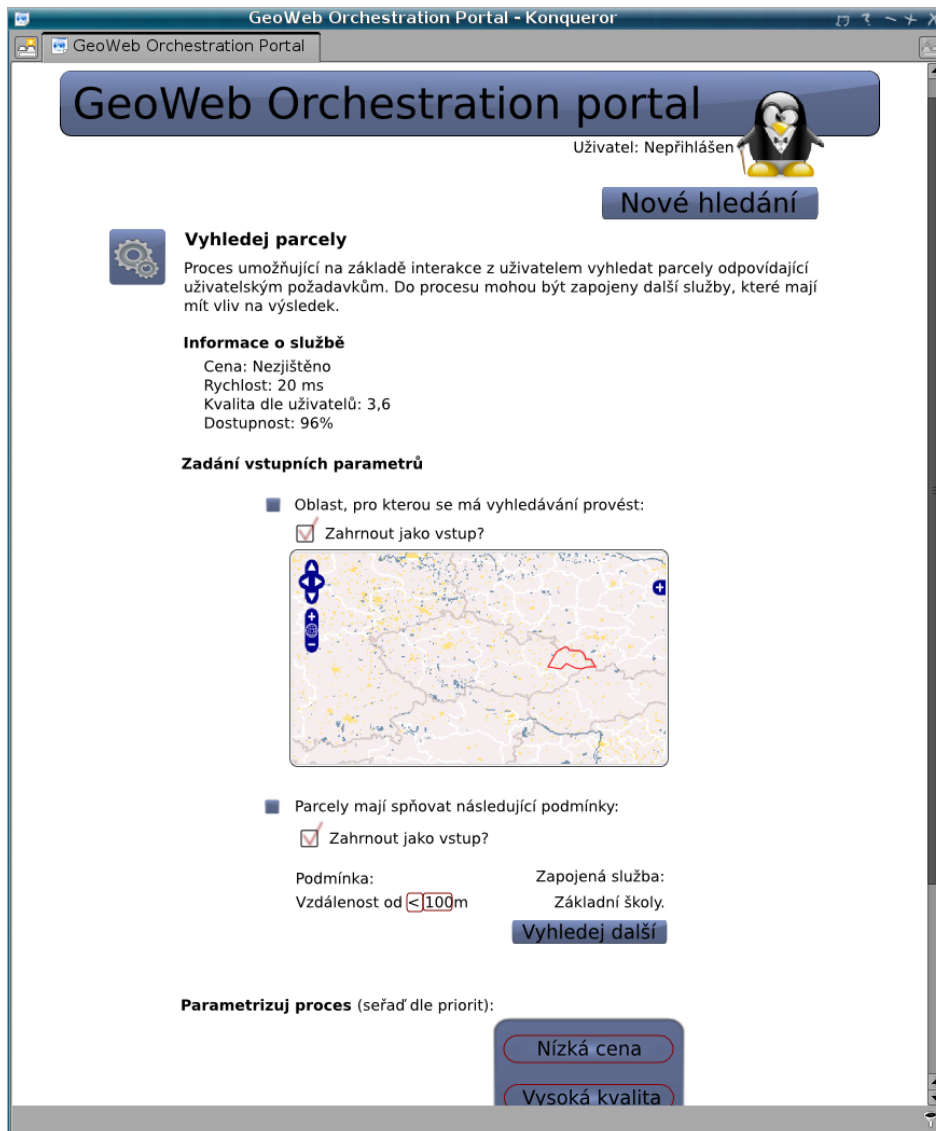


Obr. 4: Návrh úvodní stránky portálu sloužícího běžným uživatelům



Obr. 5: Zobrazení vyhledaných služeb

Po vybrání daného orchestru bude uživateli přímo umožněna jeho jednoduchá parametrizace, nebo spuštění vybraného orchestru. V případě požadované úpravy procesu bude proces registrem upraven a opět vrácen v obdobném dialogovém okně (webové stránce) a parametrizace se může stále opakovat, dokud nebude uživatel spokojen.



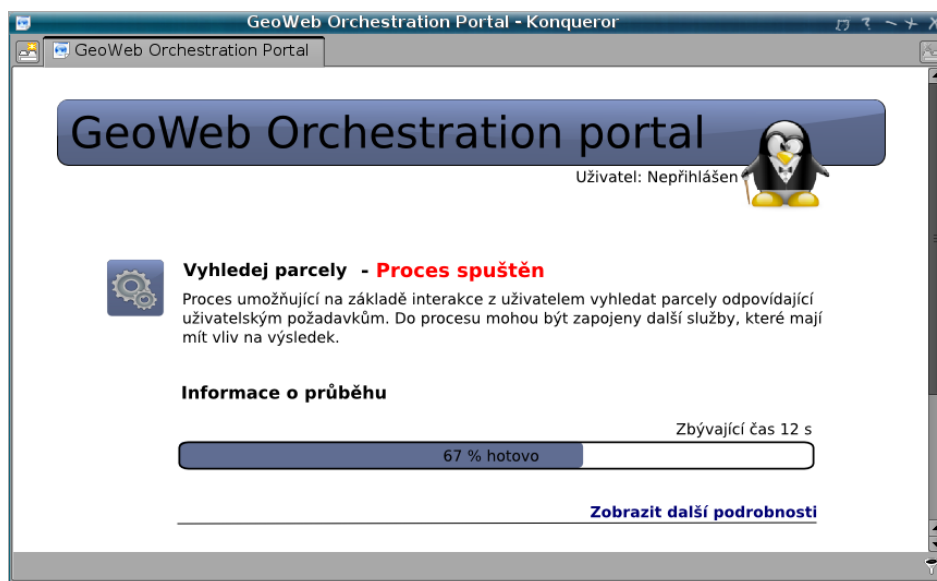
Obr. 6: Zobrazení všech podrobností o procesu, včetně možnosti parametrizace a spuštění

Při volbě složitější parametrizace bude uživateli zobrazen proces v jeho grafické podobě – viz. obr. 2, nebo obr. 3. Při požadavku záměny služby za jinou bude opět využíván dialog pro vyhledávání služeb a jejich volba. Po spuštění procesu bude uživateli zobrazen dialog o průběhu a následně zobrazen výsledek.

Navržené rozhraní

Z výše zmíněných řádků je patrné, že GUI bude přistupovat k Service Registru a BPEL procesoru. V následujících řádcích je popsáno základní rozhraní vůči těmto dvěma zmiňovaným komponentám.

GUI – Registr Služeb



Obr. 7: Zobrazení informací o průběhu spuštěného procesu

- `getServices()` – vrací seznam procesů/služeb upravený v závislostech na metrikách, šablonách, či uživateli apod. Součástí vráceného seznamu jsou i základní metriky a informace o procesech a službách.
- `getDetail()` – Vrací všechny dostupné informace o procesu, či službě. Umožňuje vrátit proces ve formě BPEL souboru, který je možno následně vizualizovat.
- `save()` – slouží k uložení upraveného procesu do registru služeb, k pozdějšímu znovupoužití.

GUI – BPEL Procesor

- `execute()` – umožňuje zavolat BPEL procesor, aby spustil konkrétní službu uloženou v registru služeb, nebo službu, která je upravena uživatelem a není žádané její uložení v registru služeb.

Závěr

V současnosti je grafické uživatelské rozhraní navrženo v teoretické rovině a byly modelově vytvořeny dialogy a komponenty, které by mohly být při práci s orchestry využitelné. Pro potvrzení použitelnosti a uživatelské přívětivosti však bude nejdůležitější interakce tohoto návrhu přímo s uživateli. Až po této interakci s vybranou různorodou skupinou uživatelů – v první fázi realizované taktéž v rovině teoretické je vhodné přistoupit k realizaci GUI, její praktickou implementací. Následně je vhodné provést druhé kolo interakce z uživateli a zanést jejich připomínky vzniknuvší při reálné práci s navrženým GUI. Současný návrh vychází ze současně navržené architektury, která se ještě může drobně upravit, což se může projevit i v navrženém grafickém rozhraní.

Reference

- [BOSSOA] Bose S., Bieberstein N., Fiammante M., Jones K., Shah R., SOA Project Planning Aspects, [online](#)¹.
- [NB] Domovská stránka produktu NetBeans, [online](#)².
- [OJ] Domovská stránka projektu OpenJump, [online](#)³.
- [OL] Domovská stránka projektu OpenLayers, [online](#)⁴.
- [PRADP] Pager, M., Řetězení webových služeb v prostředí open source GIS. Diplomová práce. 2007. Ostrava. [online](#)⁵.
- [PRAM] Prager M., Maršík V., Využití orchestrace služeb pro řešení úloh v rámci ISKŘ, [online](#)⁶.
- [RBPM] Role BPM, BPM Portál, [online](#)⁷.
- [REEN] BPMN & BPEL for business analysts, Úvod do kurzu, [online](#)⁸.
- [RLBPM] Organizační struktury v procesním řízení, BPM slovníček, [online](#)⁹.
- [TBPEL] Vašíček P., Seriál BPM prakticky, 5. část: Tvorba BPEL modulu, [online](#)¹⁰.
- [TILSOA] Tilkov S., Roles in SOA Governance, [online](#)¹¹.
- [UBPM] Vašíček P., Seriál BPM prakticky, 1. část: Proč BPM s open source nástroji, [online](#)¹².
- [UBPM3] Vašíček P., Seriál BPM prakticky, 3. část: Úvod do BPMN, [online](#)¹³.
- [WEEP] Domovská stránka projektu WEEP, [online](#)¹⁴.

¹<http://www.informit.com/articles/article.aspx?p=422305&seqNum=5>

²<http://www.netbeans.org/>

³<http://openjump.org/wiki/show/HomePage>

⁴<http://www.openlayers.org/>

⁵http://gisak.vsb.cz/pra089/texty/DP_pra089_v1_0.pdf

⁶http://gis.vsb.cz/GIS_Ostrava/GIS_Ova_2008/sbornik/Lists/Papers/093.pdf

⁷<http://www.procesy.cz/Methodiky/Role-BPM.htm>

⁸<http://www.reengine.cz/index/bpmn-and-bpel-for-business-analysts.do>

⁹<http://bpm-slovník.blogspot.com/2007/09/organizace.html#Role>

¹⁰<http://bpm-sme.blogspot.com/2008/04/5-tvorba-bpel-modulu.html>

¹¹<http://www.infoq.com/articles/tilkov-soa-roles>

¹²<http://bpm-sme.blogspot.com/2008/02/1-uvod-do-bpm-pro-sme.html>

¹³<http://bpm-sme.blogspot.com/2008/03/3-uvod-do-bpmn.html>

¹⁴<http://weep.gridminer.org/index.php/About.WEEP>