

# Sémantická integrace webových služeb

Vojtěch Svátek, Vladimír Vávra

Vysoká škola ekonomická v Praze  
katedra informačního a znalostního inženýrství  
Nám. W. Churchilla 4, 130 67 Praha 3  
e-mail: svatek@vse.cz, vava@vava.cz

## Abstrakt

*Současný stav webových služeb (WS) vyžaduje velkou míru lidských zásahů při jejich využívání. To platí i v oblasti, jejíž automatizace se zdá být poměrně schůdná: vyhledávání relevantních služeb v prostředí Internetu. Pomocí by mohly být znalostní anotace služeb, zachycující podstatu a kritéria použitelnosti služby ve formálním jazyce, vymezeném tzv. ontologií. Akademický výzkum znalostního inženýrství se ovšem převážně orientuje na ambiciózní projekty plně automatické integrace, které nejsou dobře slučitelné s procesně orientovanými standardy skládání WS orientovanými na podnikové prostředí. Článek poskytuje stručný přehled problematiky sémantického popisu (a vyhledávání) WS, jejich procesně orientovaného skládání, i kombinace obou přístupů. V závěru je zmíněn vlastní výzkumný projekt, zahrnující kombinování WS zaměřených na analýzu obsahu a struktury WWW.*

## Klíčová slova

Webová služba, integrace, UDDI, transakce, workflow, BPEL4WS, sémantický web, ontologie, OWL-S, DAML-S, OWL-QL, RDF/S, analýza WWW, skeletální plánování.

## 1 Nástup webových služeb

Internet je využíván nejen jako prostředek komunikace mezi uživatelem a aplikací, ale ve stále větší míře také jako prostředek přímé interakce mezi samotnými aplikacemi. Ačkoli na tomto poli již vzniklo několik vzájemně nekompatibilních návrhů či specifikací, žádný nezískal rozhodující majoritu na trhu, ať již z důvodu přílišné proprietárnosti konkrétního řešení či složitosti implementace lokálního řešení v prostředí otevřeného internetu. Proto byl vývoj a nástup webových služeb (WS) toužebně očekáván ze stran všech zúčastněných.

Základní architektura webových služeb předpokládá interakci tří rolí:

- Poskytovatel webové služby – poskytuje služby klientům
- Adresářové služby – umožňují registraci, vyhledání a identifikaci služby v adresáři – registru
- Klient – požaduje a využívá poskytovaných webových služeb.

V typickém scénáři poskytovatel provozuje určitou síťově dosažitelnou softwarovou komponentu (službu). Vytvoří popis této služby a publikuje jej v rámci adresářové služby, či přímo klientovi. Klient na základě požadavku vyhledá (ať lokálně, či ve veřejném adresáři webových služeb) a identifikuje vyhovující službu, se kterou naváže formu komunikace odpovídající jejímu popisu.

Součástí základního návrhu WS jsou technologie a specifikace komunikačních protokolů postavených na XML, které umožní:

- Výměnu zpráv (zpravidla protokol SOAP)
- Popis webových služeb (zpravidla jazyk WSDL)
- Publikování a vyhledávání (zpravidla jazyk UDDI).

Proces vzniku složené WS se dá rozdělit na dvě fáze: první je vyhledání dílčích WS a jejich zkombinování, druhou pak zavolání složené služby (a v jejím rámci pak i dílčích WS). Převážná část současných aktivit na poli WS je zaměřena na druhou fázi, technické realizaci volání WS, ať se jedná o oblast komunikačních standardů, zajištění bezpečnosti či podpoře transakčního zpracování. Zde se zaměříme zejména na první fázi, tj. proces publikování, vyhledání a výběru z množiny použitelných WS. Tyto aktivity jsou dnes zpravidla vykonávány člověkem, ačkoliv již existují prostředky pro jejich automatizaci, vyvinuté zejména v souvislosti s iniciativou tzv. sémantického webu.

V části 2 se nejprve pokusíme charakterizovat omezení UDDI jako prostředku pro vyhledávání WS. V části 3 uvedeme techniky rozšíření WS ve směru k transakčnímu zpracování a podpoře workflow, založené na „programování ve velkém“. V části 4 se budeme věnovat alternativnímu, převážně akademickému směru výzkumu, který vychází ze sémantického anotování WS pomocí ontologií, a na příkladu zahraničního projektu naznačíme i možnost kombinování obou přístupů. V části 5 uvedeme vlastní příspěvek k problematice, s důrazem na techniku skeletálního plánování jako možné „střední cesty“ při skládání WS. Část 6 je pak věnována shrnutí hlavních myšlenek.

## 2 Limity registrů UDDI z hlediska automatizovaného zpracování

### 2.1 Nevýhody centralizované správy

Současný stav praxe UDDI předpokládá uložení informace o libovolné WS v centrálních „autorizovaných“ registrech. V principu by ale mohla jako jednotlivý registr vystupovat libovolná webová stránka. Některé konkurenční návrhy proto předpokládají, že každý server propaguje služby přístupné přímo na něm. Tato informace by pak byla využitelná např. weeboty vyhledávačů, a to obdobným způsobem, jako při indexování HTML stránek daného sídla.

### 2.2 Nízká důvěryhodnost registrů

S předchozím bodem velice úzce souvisí otázka důvěryhodnosti údajů v konkrétním UDDI registru a odvozeně i důvěryhodnost poskytované služby. Již dnes obsahují existující UDDI registry množství neplatných záznamů, které pak „diskreditují“ funkční služby. Opět můžeme pro srovnání připomenout situaci s poli META v hlavičce stránek HTML (např. „keywords“), které měly původně sloužit ke snazší a efektivnější kategorizaci a katalogizaci; v praxi však došlo k jejich zkompromitování vinou záměrného matení indexovacích robotů (a v důsledku i uživatelů) ze strany tvůrců stránek. Další otázkou zůstává, jak bude zaručena důvěryhodnost vyhledané služby jako takové. Poznamenejme, že návrh UDDI verze 3 se snaží tyto některé z nedostatků předchozích verzí řešit<sup>1</sup> (podpora digitálního podpisu, sdílení záznamů a replikace mezi registry, podpora internacionalizace).

### 2.3 Absence sémantiky popisů WS

V současnosti je popis konkrétní služby v registru UDDI realizován pomocí několika deskripčních polí, která obsahují popis v přirozeném jazyce. Jako důsledek pak vyplývá nemožnost využití registru pro automatické vyhledávání pomocí běžných textových nástrojů, vzhledem k nejednoznačnosti přirozeného jazyka. Uváděným příkladem je např. využití termínu „quote“, který vyhledá jak služby orientované na burzovní informace, tak např. na náhodný výběr populárních citátů. Řešením problému by byl popis WS pomocí sémantické anotace odkazující na koncepty ontologie – vedle samotného vyhledávání by byl potenciálně využitelný i v procesu automatického skládání WS. Této problematice se budeme věnovat zejména v části 4.

---

<sup>1</sup> Viz <http://xml.coverpages.org/UDDIv3BetaNodes.html>

### 3 Workflow a transakce ve WS

I kdybychom pominuli problémy spojené s jejich vyhledáním, mají samotné webové služby v původní podobě návrhu jen omezenou oblast praktické využitelnosti. Vzhledem k důrazu na jednoduchost návrhu a implementace (ve srovnání se staršími standardy vzdáleného volání procedur) se v praxi hodí spíše jen pro schematické úlohy typu „aktuální meteorologické podmínky“ či „přesný čas“. Řada doplňujících návrhů se proto snaží původní model webových služeb vylepšit. Ve většině případů se jedná o výrazné rozšíření funkčnosti, či konkretizace stávajících standardů pomocí vytvoření vyšších vrstev v procesu komunikace a interakce jednotlivých WS. Jako příklady takových protokolů bychom mohli uvést např. WSIL<sup>2</sup>, WSFL<sup>3</sup>, XLANG<sup>4</sup>, ebXML BPSS<sup>5</sup>, BPML<sup>6</sup>, WSCI<sup>7</sup>, a zejména BPEL4WS<sup>8</sup> (často jen jako BPEL). Protokoly vesměs usilují o řešení následujících problémů, nezbytné pro úspěšné modelování a skládání webových služeb v běžných business aplikacích.:

- Oddělení privátní a veřejné části služby
- Podpora transakcí
- Podpora workflow
- Otázky zabezpečení
- Obsluha chyb (error handling).

Kritické porovnání většiny uvedených jazyků lze nalézt např. v [Aa03]. Podrobněji se zmíníme jen o BPEL4WS (Business Process Execution Language for Web Services), který je podporován předními společnostmi z oblasti systémové integrace (BEA, IBM, Microsoft, SAP, Siebel), a vznikl sloučením předchozích iniciativ XLANG (Microsoft) a WSFL (IBM). BPEL4WS staví na existujících standardech - WSDL 1.1, dále XML Schema 1.0, XPath 1.0 a WS-Addressing. Jednotlivé interakce WS je možné skládat v rámci workflow modelu do transakcí s definovanými rolemi. Základní stavební prvky modelu jsou: *partneři*, představující jednotlivé (vyvolané či volající) WS, *proměnné*<sup>9</sup>, nesoucí hodnoty v procesu výměny zpráv, *vazby mezi partnery* (partnerLinks), definující vztahy mezi partnery, *proces* (který může využívat tradičních řídicích struktur switch, while, sequence aj.) a *obsluha chyb* (error handling) s možností *kompenzace* průběhu operací. Nezanedbatelnou výhodou BPEL4WS je fakt, že již existuje implementace nástroje BPWS4J (IBM), který pracuje s funkční podmnožinou vlastností definovaných v rámci BPEL4WS specifikace.

Z povahy jazyků použitých pro popis WS (XML, XML Schema) vyplývají možnosti a rozsah automatizace – vzhledem k absenci sémantické informace zahrnuje dynamické navázání komunikace pouze syntaktickou kontrolu. Vybraný partner v rámci procesu je předem dán (resp. vybrán v závislosti na průběhu aplikace na základě předem daných pravidel), či pochází z předem vybrané skupiny. To do jisté míry limituje „dynamičnost“ spolupráce v tak rychle se měnícím prostředí, jako

---

<sup>2</sup> <http://www-106.ibm.com/developerworks/webservices/library/ws-wsilspec.html>

<sup>3</sup> <http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>

<sup>4</sup> [http://www.gotdotnet.com/team/xml\\_wsspecs/xlang-c/default.htm](http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm)

<sup>5</sup> <http://www.ebxml.org/specs/ebBPSS.pdf>

<sup>6</sup> <http://www.bpml.org/bpml.esp>

<sup>7</sup> <http://www.sun.com/software/xml/developers/wsci/>

<sup>8</sup> <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>

<sup>9</sup> Ve specifikaci BPEL4WS 1.0 tzv. kontejnery.

je internet. Dále popis procesů není deklarativní a není zapsán způsobem umožňujícím symbolickou manipulaci a odvozování. Jelikož kontrola probíhá na syntaktické úrovni, tak např. rozdílná reprezentace dat (např. rozdíly v zápisu datumu) mohou být překážkou navázání komunikace, i když sémanticky jsou data ekvivalentní.

## 4 Sémantika ve WS

### 4.1 Sémantický web

Stávající infrastruktura WWW standardizuje pouze *syntaktické* aspekty webových dat. Podstatu obsažených sdělení musí proto analyzovat lidský uživatel – není přímo přístupná počítačovým aplikacím (když pomineme nasazení sofistikovaných nástrojů pro zpracování přirozeného jazyka, které lze použít jen v úzce vymezených oborech a se střídavým úspěchem). Cílem iniciativy *sémantického webu* je opatření každého zdroje na internetu formalizovaným popisem, jehož sémantika by byla definována pomocí *ontologií*<sup>10</sup>. Takové popisy by byly interpretovatelné softwarovými agenty s minimální mírou lidské interakce. Mezi hlavní předpokládané aplikace sémantického webu patří především inteligentní vyhledávání informací (s výrazně vyšší přesností, než fulltextové nástroje spoléhající na “hrubou sílu”), dále integrace dat, dynamická tvorba portálů apod.

Jazykem navrženým pro zápis sémantických popisů (metadat) v prostředí WWW je *RDF*<sup>11</sup>. *Tvrzení* RDF jsou uspořádané trojice “subjekt, predikát, objekt”, které společně tvoří orientovaný graf. Zjednodušeným příkladem tvrzení může být např.: „zaměstnanec351“ (subjekt) „je-absolventem“ (predikát) „VŠE Praha“ (objekt). *RDFS* (RDF Schema) je pak rozšířením RDF o možnost definovat konstrukce známé z rámcových (či objektových) systémů, tj. *třídy* a binární *relace* (vlastnosti) včetně definičního oboru a oboru hodnot; nad třídami i relacemi může být definována hierarchie. RDF i RDFS využívá syntaxi XML, čímž je zajištěna přenositelnost mezi různými systémy a platformami.

V roce 2000 vznikl projekt DAML (DARPA Agent Mark-up Language), s cílem vytvořit sémantickou nadstavbu nad RDF s větší vyjadřovací silou než má RDFS. Jazyk označený jako DAML-ONT již vycházel z předchozího výzkumu v oblasti *ontologických jazyků*, a zahrnoval celou řadu konstruktů pro vymezení vztahů tříd a hodnot slotů. Částečným sloučením s paralelním projektem OIL (Ontology Inference Layer) vznikl jazyk s nepříliš elegantním názvem DAML+OIL. V roce 2002 byl vývoj zastřešen konsorciem W3C, v jehož rámci byla ustanovena Ontology (WebOnt) Working Group<sup>12</sup>; výsledkem její práce je standardizace jazyka (formou doporučení W3C) pod názvem *OWL* (Ontology Web Language). K dotazování nad ontologiemi DAML+OIL byl mezitím vypracován návrh dotazovacího jazyka DQL (DAML Query Language), jenž byl následně přejmenován na OWL-QL [Fi03]. OWL-QL umožňuje specifikovat sémantické relace mezi dotazem, odpovědí a bází znalostí (primárně definované pomocí OWL).

### 4.2 OWL-S – ontologie pro popis WS

Vzhledem k náročnosti sémantické analýzy existujících zdrojů (zejména statických stránek) internetu je velmi obtížné zajistit jejich zpětný popis<sup>13</sup>. Pravděpodobnějším scénářem prosazení vize sémantického webu do praxe je využití sémantického popisu při vytváření nových síťových zdrojů, nejlépe založených na nové technologii, která by byla od počátku sémanticky orientována. Mezi

---

<sup>10</sup> O sémantickém webu a ontologiích podrobně včetně odkazů ve [Sv02].

<sup>11</sup> Resource Description Framework, viz <http://www.w3.org/RDF>.

<sup>12</sup> <http://www.w3.org/2001/sw/WebOnt>

<sup>13</sup> Existují projekty, které se o to přesto snaží – jedním z nich je projekt *Rainbow* popsáný v kap. 5.

nej důležitější zdroje na internetu patří právě zdroje poskytující služby. Výzkum v oblasti sémantického webu si proto mj. klade za cíl vyvinutí prostředků a nástrojů umožňujících automatické vyhledání, výběr, navázání spojení a interakci webových služeb<sup>14</sup>.

Pokud bychom 3. kapitolu chápali jako vývoj řešení integrace webových služeb „zdola nahoru“ (směrem od jednoduchých ale rozšířených standardů), pak návrhy z oblasti sémantického webu usilují spíše o přístup „shora dolů“, tj. o využití předem navržené doménové ontologie. V případě webových služeb se jedná o *OWL-S*<sup>15</sup> (Ontology Web Language for Services).

OWL-S (návrh ve verzi 1.0) vychází ze svého předchůdce DAML-S (0.9). Struktura ontologie se skládá ze tří základních stavebních kamenů:

- *Service profile* - Vyjadřuje věcnou podstatu služby, tj. poskytuje vyhledávacímu agentovi informaci, na základě které se může rozhodnout, zda daná služba splňuje dané požadavky. Kromě toho může obsahovat i definici potřeb agenta (vstupní podmínky), který by mohl danou službu využívat, kategorizaci typu služby, či hodnocení kvality služby podle některého standardního bodovacího systému.
- *Service model* - Popisuje chování služby při jejím vyvolání. Službu si můžeme zobrazit jako proces, v případě služby skládající se z několika kroků může vyhledávací agent využít tuto informaci např. k hlubší analýze, ke skládání popisů více služeb či k monitorování běhu služby.
- *Service grounding* - Obsahuje konkrétní popis rozhraní služby (zatímco Service profile a Service model tvořily abstraktní reprezentaci). Typicky jde o komunikační protokol, formát zpráv, použité porty apod.

Vidíme, že specifikace OWL-S se částečně překrývá s již dříve zmíněným BPEL4WS, i když se ke styčným bodům dospělo z jiného východiska.

### 4.3 Kombinace sémantického vyhledání a procesního skládání

Myšlenka sémantického webu je zatím stále spíše myšlenkou, proto je větší pozornost věnována zajištění syntaktické a procesní interoperability, která může přinést praktické výsledky dříve. Jako zajímavý postup v současném stavu, kdy není na obzoru definitivní řešení, se jeví kombinace obou, do jisté míry komplementárních vývojových větví, formou relativně oddělených, kooperujících vrstev. Příkladem je aplikace vyvíjená na Stanfordské univerzitě pod vedením Dr. Sheily McIlraith [MM03]. Vychází z modelu BPEL4WS, ovšem rozšířeného o sémantickou nadstavbu zajišťující vyhledání předem neznámého partnera a sémantický překlad potřebných informací – ve fázi návrhu složené služby je proto definována pouze role, a konkrétní partner se do ní dosazuje až za běhu. Pro formální popis charakteristiky a funkce konkrétní webové služby je využito DAML-S, a dotazy nad katalogy DAML-S profilů služeb jsou formulovány v DQL. Z implementačního hlediska se jedná o doplnění stroje BPWS4J o speciální vrstvu SDS (Semantic Discovery Service), fungující jako agregátor webových služeb. SDS vystupuje v roli transparentního proxy objektu, skrze který jsou zasílány jednotlivé zprávy vstupující nebo vystupující z/do BPWS4J. Pokud SDS zachytí žádost o navázání spojení s partnerem, zabalí DAML-S charakteristiky žádosti do DQL dotazu a zašle jej DQL serveru, na základě kterých DQL server vrátí možné vyhovující partnery. SDS vybere konkrétního partnera (v případě potřeby přitom využije sémantickou překladovou službu), a naváže s ním spojení pomocí parametrů spojení poskytnutých od BPWS4J. Po proběhnutí operace partner vrací výsledek, opět

---

<sup>14</sup> Propagátorem webových služeb jako “kick-off” aplikace pro sémantické anotování je mj. Tim Berners-Lee, tvůrce koncepce jak “obyčejného”, tak i sémantického webu. Viz jeho přednáška na světové konferenci WWW2003 v Budapešti, online na <http://www.w3.org/2003/Talks/0521-www-keynote-tbl>.

<sup>15</sup> <http://www.daml.org/services/owl-s/1.0>

prostřednictvím SDS, stroji BPWS4J. Ten obnoví řídicí tok dané aplikace a pokračuje ve vykonávání procesů. SDS je pak opakovaně volán v případě, že je požadováno spojení s určitou WS.

Příkladem je automatické zprostředkování žádosti o půjčku. Předpokládejme, že se žadatel nedávno přestěhoval z Velké Británie do USA, a poskytovatelem kreditu má tedy být (z daňových důvodů) firma z Kalifornie, ovšem informace potřebné pro zhodnocení schopnosti splácet je nutné získat z Británie. Vzhledem k sémantické nekompatibilitě amerických a britských ‘credit reports’ by pouze pomocí BPWS4J nebylo možné službu realizovat. SDS v takovém případě nejprve sémanticky (tj. dotazováním do profilů DAML-S) vyhledá příslušnou překladovou službu, zapojí ji do řetězce zpracování, a na základě výsledku překladu (opět sémanticky) vyhledá poskytovatele půjčky.

## 5 Projekt Rainbow – skládání webových služeb pro analýzu WWW

### 5.1 Projekt Rainbow

Přestože jsou hlavní očekávání spojená s webovými službami orientována na integraci aplikací napříč organizacemi (zejména B2B), tato technologie se dobře uplatní i v rámci distribuovaných aplikací vytvářených v rámci jediného projektu. To platí zejména pokud se jedná o projekt s omezenou mírou centrálního řízení a komponentami vyvíjenými různými osobami na různých platformách. Příkladem je *Rainbow* (Reusable Architecture for INtelligent Brokering Of Web information access): akademický projekt<sup>16</sup> sdružující větší počet dílčích studentských a doktorandských projektů zaměřených na analýzu obsahu a struktury *webových prezentací* menších firem [Sv03]. Komponenty vzniklé v rámci dílčích projektů se od sebe liší *typem zpracovávaných dat* (volný text, značky HTML, topologie odkazů, bitové mapy obrázků atd.), *druhem analyzovaného objektu* (věta textu, jednotlivá stránka, jednotlivý odkaz, jednotlivý obrázek, množina obrázků, množina stránek atd.), *typem řešené úlohy* (klasifikace objektu, vyhledání relevantních objektů, extrakce cílového textu), a v některých případech i *věcnou oblastí*. Tou je u většiny komponent obecně problematika nabídky zboží a služeb, některé jsou však orientovány na specifické typy nabídek: momentálně se jedná zejména o cyklistické produkty a pornografii (ta ovšem není zbožím v běžném smyslu).

Komponenty průběžně vznikají v různých programovacích jazycích (Java, Python, PHP, ASP) a na různých platformách (Unix, Windows). Jejich propojení formou *webových služeb* bylo proto prakticky jediným schůdným řešením. V současnosti zahrnuje architektura *Rainbow* sedm komponent již „zabalných“ ve webové službě, a několik dalších, pro které se toto řešení připravuje.

### 5.2 Role ontologií v Rainbow

Vzhledem k prozatím poměrně malému počtu služeb v *Rainbow* zahrnutých není formální modelování jejich sémantiky (na rozdíl od příkladu zmíněného v kap. 4.3) příliš potřebné pro primární *vyhledání* vhodné služby. Jednotlivé služby jsou navíc natolik elementární, že by externímu uživateli v případě jejich samostatného volání mohly přinést jen nepatrný užitek. O to významnější se však jeví role formální (nebo alespoň semi-formální) ontologie při *skládání* služeb do složitější aplikace. V rámci projektu *Rainbow* vznikla soustava několika propojených ontologií lišících se předmětem modelování. *Základní ontologie struktury WWW* zachycuje vzájemné vztahy různých druhů objektů tvořících webové prezentace. Na ni navazují *ontologie typů dat* (tzv. „pohledů“ na WWW, např. kódu HTML nebo topologie odkazů), *ontologie úloh* (např. klasifikace nebo extrakce), a *doménové ontologie* (např. cyklistických produktů). Jednotlivé služby jsou *anotovány* z hlediska všech relevantních ontologií. Při skládání služeb do funkční aplikace pak lze zřetěžit jen takové dvojice služeb, které si odpovídají z hlediska druhu objektů na vstupu/výstupu „svých“ úloh, a navíc množiny

---

<sup>16</sup> Projekt je řešen ve spolupráci katedry informačního a znalostního inženýrství a Laboratoře inteligentních systémů, na fakultě informatiky a statistiky Vysoké školy ekonomické.

ontologických tříd na vstupu/výstupu mají neprázdný průnik. Pokud např. komponenta pro extrakci cenových informací z kódu HTML firemních katalogů navíc potřebuje ukládat obrázky odpovídajících produktů, musí se obrátit na službu rozpoznávání obrázků produktů, a nikoliv na službu rozpoznávání reklamních obrázků či pornografie.

V současnosti probíhá skládání služeb ručně (programováním „natvrdo“), a ontologie a anotace jsou tudíž využívány jen nepřímo. V budoucnu se však počítá se vznikem *uživatelského prostředí pro skládání služeb*, které by mohlo formální modely využívat automatickým způsobem. Ontologie jsou napsány v jazyce OWL, pro který je v rámci iniciativy sémantického webu vyvíjena řada odvozovacích nástrojů. Díky tomu nebude zřejmě nutné *odvozovací složku* zmíněného prostředí (pro doporučení služeb k zařazení na určité místo aplikace, resp. kontrolu konzistence aplikace jako celku) vyvíjet od základu.

### 5.3 Skeletální plánování – řešení pro analýzu v reálném čase?

Ačkoliv již v současnosti existuje uživatelské prostředí pro aplikaci jednotlivých nástrojů *Rainbow* na stránky *aktuálně otvírané* v běžném webovém prohlížeči, hlavní úsilí je orientováno na vývoj demonstrační aplikace pro sémantické *vyhledávání v databázi faktů* (o nabídkách cyklistických produktů) *předem* získaných z WWW. Jak proces získání primárních dat, tak jejich znalostní analýza, zde probíhají mimo časový rámec seance s koncovým uživatelem. Rychlost odezvy proto víceméně závisí jen na složitosti vyhledávání relevantních faktů v databázi; v současnosti je jako „faktuální databáze“ používán systém *Sesame*<sup>17</sup>, pracující s fakty v jazyce RDF a dotazovacím jazykem SeRQL (jako sémantickou nadstavbou nad relačně-databázovou vrstvou s jazykem SQL).

Složení analytických služeb lze v takovém modelu realizovat přímočaře, formou procedurální programové aplikace. Ta postupně zavolá participující služby, jako je analýza URL (pro rychlé vytipování „slibných“ stránek v rámci celé prezentace), analýza kódu HTML statistickým modelem natrénovaným na danou věcnou oblast (pro extrakci cenových informací z katalogu), nebo povrchově-lingvistická analýza volného textu (pro extrakci slovně formulovaného profilu firmy). Některé služby potom dále zavolají služby jiné – analýza HTML typicky analýzu bitových map vybraných obrázků a analýzu URL pro vnořené odkazy. Výsledky budou sloučeny a uloženy do výstupního souboru, ze kterého ho systém *Sesame* načte.

Situace se ovšem radikálně změní v okamžiku, kdy uživatel začne od (pokročilejší) aplikace požadovat zpřístupnění informací z WWW v *online* režimu. Nyní již bude nutné vyvažovat na jedné straně kvalitu výsledků získatelných různými způsoby, na straně druhé různou časovou náročnost těchto způsobů (uvažme např. na jedné straně analýzu URL nebo topologie odkazů, na straně druhé komplexní lingvistický rozbor textu) a omezenou trpělivost uživatele na tyto výsledky čekajícího. Předem popsat všechny možné situace v rámci procedurální aplikace zjevně nebude možné, tj. bude nutné skládat služby do jisté míry *dynamicky*. Na druhé straně však zůstanou i zde platné určité stereotypy: některé posloupnosti volání služeb mají smysl jen jako celek, a tvoří ucelený *plán*. Jednou zvolený plán ovšem může být nutné pozastavit a nahradit jiným, pokud se jeho realizace ukáže neúnosná nebo průběžně dojde ke zpřísnění časového limitu. Nabízí se zde analogie s lékařským prostředím, kde jsou rovněž určité postupy doporučeny, ale lékař (popřípadě softwarový systém pro podporu rozhodování, který mu asistuje) je často musí upravit podle změn aktuálního stavu pacienta i dostupných zdrojů. Právě v lékařském prostředí se také v posledních letech rozvíjí koncepce *skeletálního plánování* [Mi99]. Na rozdíl od úlohy plánování v tzv. umělé inteligenci je zde kladen důraz na předchozí existenci skeletů plánů, ty tedy nejsou vytvářeny od základu z atomických komponent, nýbrž jen spouštěny, pozastavovány, synchronizovány, a podle potřeby navzájem nahrazovány. Automatizace tohoto procesu je zajištěna pomocí filtrovacích, umožňovacích („set-up“),

---

<sup>17</sup> <http://sesame.aidadministrator.nl>

aktivačních, pozastavovacích („suspend“), reaktivačních, uspokojovacích („complete“) a zrušovacích („abort“) podmínek, přiřazených k jednotlivým plánům. V prostředí *Rainbow* si můžeme např. představit, že objevení značek <table> na stránce s nabídkou produktů způsobí *aktivaci* plánu orientovaného na analýzu tabulkově koncipovaného katalogu. V okamžiku, kdy se ukáže, že tabulka neobsahuje ceny produktů, bude plán *pozastaven*, a místo něj *aktivován* jiný plán, zaměřený na vyhledávání slovních indikátorů (např. „model“, „typ“) a extrakce cenových údajů z jejich okolí. Mezitím ovšem může uživatel upřednostnit požadavek získání informace o co největším počtu firem nabízejících určitý okruh produktů, před požadavkem na konkrétní cenové údaje. V takovém případě bude naopak *pozastaven* (případně zrušen) nový plán, a *reaktivován* plán původní.

Myšlenka využít skeletální plánování pro skládání webových služeb představuje zajímavý kompromis mezi zcela volným skládáním služeb v otevřeném prostředí (vesměs předpokládaném v akademických projektech typu OWL-S), a programováním „ve velkém“ a „natvrdo“, typu BPML4WS (rozšířeným ve firemním prostředí). Přestože se „laboratorní“ prostředí webových služeb *Rainbow* (i jejich cílová úloha – analýza webových prezentací) do značné míry liší od běžně zdůrazňovaných aplikací webových služeb pro transakce B2B, předpokládáme, že výsledky chystaných experimentů se skeletálním plánováním mohou být přínosné i pro širší komunitu tvůrců složených služeb.

## 6 Závěr

Problematika skládání webových služeb pro integraci aplikací je v současnosti velmi frekventovaným tématem. V tomto článku jsme se pokusili naznačit její souvislost s problematikou formálně popsané sémantiky služeb. Navzdory velkému počtu akademických projektů v této oblasti předpokládáme i v budoucnu vůdčí roli (jakkoli nedokonalých) standardů navrhovaných velkými softwarovými firmami. Některé výsledky akademického výzkumu by se však mohly dílčím způsobem uplatnit, při respektování omezení daných realitou komerčního prostředí. V této souvislosti upozorňujeme na variantu kombinace procesního skládání se sémantickým vyhledáváním, a na technologii skeletálního plánování, se kterou v našem vlastním výzkumném projektu začínáme experimentovat.

**Poděkování:** Autoři by na tomto místě rádi poděkovali všem kolegům, kteří se podíleli na vzniku webových služeb a ontologií v rámci projektu *Rainbow*, zejména pak Jirkovi Koskovi a Martinu Labskému. Práce na článku byla částečně podporována grantem GAČR č.201/03/1318.

## 7 Literatura

[Aa03] van der Aalst, W.M.P.: Don't go with the flow: Web services composition standards exposed. *IEEE Intelligent Systems*, Jan/Feb 2003, <http://www.tm.tue.nl/it/research/patterns/ieeewebflow.pdf>.

[Fi03] Fikes R., Hayes P., Horrocks I.: OWL-QL – A Language for Deductive Query Answering on the Semantic Web, online na [http://ksl-web.stanford.edu/KSL\\_Abtracts/KSL-03-14.html](http://ksl-web.stanford.edu/KSL_Abtracts/KSL-03-14.html)

[MM03] Mandell D. J., McIlraith S. A.: Adapting BPEL4WS for the Semantic Web: The Bottom-Up Approach to Web Service Interoperation. In: ISWC2003, Sanibel Island, Florida, 2003.

[Mi99] Miksch, S.: Plan Management in the Medical Domain, *AI Communications*, 12(4), pp.209-235, 1999.

[Sv02] Svátek V.: Ontologie a WWW. In: Sborník konference Datakon 2002, s.27-55. Rozšířená verze k dispozici online na <http://nb.vse.cz/~svatek/onto-www.pdf>.

[Sv03] Svátek V., Kosek J., Labský M., Bráza J., Kavalec M., Vacura M., Vávra V., Snášel V.: *Rainbow – Multiway Semantic Analysis of Websites*. In: *2nd DEXA International Workshop on Web Semantics*, Praha 2003. Los Alamitos : IEEE, 2003, s. 635–639.