Kolokvium KSI MFF UK

5. dubna 2011

Architektura v SOA

Michal Žemlička

SOSG – http://www.ksi.mff.cuni.cz/sosg/

SOA – co je to?

- servisně orientovaná architektura
- v IT velmi frekventované slovo
- ”stříbrné kladívko”?
- architektura pro rozlehlé a komplexní systémy
- cesta k integraci stávajících aplikací

Různé chápání SOA

- Každý autor má vlastní představu o tom, co SOA je.

- Shoda – systém složený z relativně samostatných spolupracujících částí

Rozdíly v chápání SOA

- granularita a autonomie služeb
- způsob komunikace
- organizace systému jako celku (architektura)

SOA v odborné literatuře

- V první velké vlně byla SOA spojována s webovými službami – a tedy s XML, SOAP, WSDL a spoustou dalších WS-standardů.

- Druhá vlna začala připouštět, že SOA nemusí být složena jen z WS, ale může obsahovat i zapouzdřené stávající (legacy) systémy

Problém

V literatuře je spousta pozitiv servisní orientace, přesto je příliš mnoho neúspěšných projektů (nepovede se je dokončit, neposkytují to, co se od nich čekalo, nebo nejsou včas nebo v rozpočtu).

Jak je to možné?

Příčiny

- Cokoliv se pozitivního napsalo o SOA, chápe se jako vlastnost všech SOA systémů (je to jako kdybychom z razance bouracího kladiva a přesnosti hodinářského nástroje odvodili, že každé kladivo má raznaci toho bouracího a přesnost hodinářského)

- Vezme se první SOA v nabídce a nerozlišuje se, odkud přišla a co vlastně umí.

Standardizace a SOA

- Standardizace v SOA běží velmi rychle – podle všeho dokonce příliš rychle: mnohé standardy je třeba předělat už po velmi krátké době.

- Mnoho standardů (desítky) se váže k WS; prakticky se rozšířily jen základní 4

- Objevují se standardy i pro architekturu (referenční modely; aneb jak mít uspořádané služby v systému)

Odkud SOA přišla?

Zdrojů SOA je více:

- objektové technologie
- databáze
- dávkové systémy
- řídící systémy

Každý z nich vede k jinému pojetí SOA!

SOA odvozená z objektových technologií

Složitost a velikost OO systémů rostla, až začalo být neúnosné mít diagramy tříd tak velké, že se nevejdou ani na zeď, natož na obyčený papír $\Rightarrow$ třídy se začaly zapouzdřovat do větších celků a vznikly komponenty, případně služby.

U těchto autorů bývají rozhraní mezi službami blízká komunikaci mezi objekty a zachovává se použití stávajících vývojových nástrojů.

SOA odvozená z objektových technologií (2)

V mnoha případech je systém dost těsně provázaný – to má důsledky na konečné omezení velikosti i na obtížnou údržbu.

Tento způsob chápání SOA je v odborné litaratuře i v nabídkách firem dominantní

Struktura systému vychází z programátorského světa a struktura organizace se tomu musí přizpůsobit.

SOA odvozená z objektových technologií (3)

Pro uspořádání služeb v systému existují různá doporučení:

- Mnohavrstevnatý model (OASIS, Open Group)
- Síť služeb postavená kolem Enterprise Service Bus

Obojí vychází z toho, že tomu velí silná centrální autorita (s odpovídajícími prostředky), která může vnutit přesně definovanou komunikaci i strukturu systému.
Taková autorita je ve velkých firmách; chybí v malých firmách i ve státní správě

SOA nad centrální databází

Je možné (a v praxi používané) postavit systém nad centrální databází s vloženými procedurami zajišťující základní funkce systému a nad tím udržovat skupinu specializovaných aplikací (služeb) poskytujících koncové služby

Tento způsob se hodí pro využití v rámci jedné organizace, může mít velkou propustnost, dá se vytvořit a udržovat i s nepříliš početným týmem.

Dávkové systémy

- Systémy hromadného zpracování dat (psané typicky v COBOLu) vykazují mnohé znaky servisní orientace: je to síť autonomních spolupracujících aplikací, aplikace mohou být tvořeny nezávisle ...

- Komunikace probíhá výměnou souborů

- Takové systémy vynikají stabilitou a obrovskou propustností; bohužel nejsou interaktivní

Řídící systémy

V řídících systémech bývá základním úkolem propojit nějaká zařízení do jednoho funkčního celku – ta zařízení typicky musíme brát, jaká jsou; tedy včetně jejich rozhraní

Musíme hlídat nejen správnou funkci (na daný podnět to nějak zareaguje), ale i správné časování (nemůžeme připustit, aby vlak začal brzdit až půl hodiny po stisku rychlobrzdy).

Řídící systémy (2)

Mnohdy nemůžeme testovat, jak se nám zachce – mnohá zařízení nebo alespoň vybrané subsystémy prostě musí fungovat napoprvé (jaderná elektrárna spadlá na GPF?) $\Rightarrow$ jiné způsoby ladění aplikací a vyšší nároky na vývojáře

Světy informačních systémů a řídících systémů bývají mnohdy odděleny – zkušenosti se mezi nimi moc nepřenáší; ti vývojáři prostě myslí jinak)

Softwarové konfederace

- kombinace různých přístupů
- primárně – integrace větších celků (stávající aplikace, aplikace třetích stran)
- lidé jako součást systému
- struktura systému odpovídá struktuře služeb reálného světa – rozhraní mezi službami vychází z existujících rozhraní mezi lidmi nebo organizacemi
- nový prvek: architekturní služby

Služby v softwarových konfederacích

- aplikační služby
- předřazené brány
- portály
- datová úložiště
- hlavy kompozitních služeb
- správci procesů

Aplikační služby

- realizují vlastní funkcionalitu systému
- mohou to být zapouzdřené stávající aplikace i aplikace nové vytvořené
- mívají procedurální rozhraní poskytující přístup ke všemu, co ta aplikace dovede
- bývá rozumné zachovat i stávající rozhraní

Předřazené brány

- aplikační služby mají procedurální rozhraní, služby reálného světa spíše deklarativní $\Rightarrow$ potřebujeme převodník
- aplikační službu můžeme vybavit více takovými převodníky (například pro každou specifickou skupinu uživatelů jedním)
- rozšířená aplikační služba tak může poskytovat rozhraní na míru – každý má k dispozici právě takové rozhraní, jaké potřebuje (a nic navíc)

Portál

- Rozhraní systému pro koncové úživatele

- Systém může mít (a typicky i mít bude) více portálů (intranet, extranet, technická rozhraní)

Hlava kompozitní služby

- Abychom se v systému vyznali, má smysl uzavřené skupiny služeb organizovat do kompozitních služeb, poskytujících navenek iluzi, že se jedná o službu jedinou
- Rozhraní celé skupiny je tak reprezentováno touto unikátní službou, která pak rozděluje požadavky mezi konkrétní služby ve skupině
- kompozitní služba může být opět vybavena více předřazenými branami

Datové úložiště

- V dávkových systémech jednotlivé aplikace spolupracují přes úložiště (soubory, databáze); chceme-li je integrovat, potřebujeme je také
- Umožníme-li přístup k datům v úložišti nejen hromadný, ale i po záznamech, můžeme pak propojovat služby zpracovávající požadavky jednotlivě a v dávkách.
- úložiště se vyplatí i pro organizaci požadavků

Správci procesů

- za byznys procesy by měl být někdo právně odpovědný $\rightarrow$ měl by mít možnost BP sledovat a ovládat
- je-li BP zaznamenán tak, aby tomu jeho vlastník rozuměl, může jej i pozměnit – v případě nouze se rychlost změny z týdnů může zkrátit na minuty až hodiny.
- BP v organizaci mohou být různé povahy a složitosti to je rozumné podporovat různé notace BP

Vlastnosti konfederací

- stabilní rozhraní (vychází z vyzkoušených a zavedených rozhraní reálných služeb) $\Rightarrow$ lokalita změn, tj. snadnější vývoj i údržba
- použitelné v e-governmentu i SMB
- podpora agilních byznys procesů
- hladší přechod na SO systém

Závěr

- vhodným zmapováním typů SOA a jejich použitelnosti můžeme zkvalitnit výběr prostředků pro různé typy úloh (zvýšíme šanci na úspěch projektu)
- softwarové konfederace rozšiřují použitelnost SOA do dalších prostředí a pro další typy úloh
- použitím architekturních služeb můžeme dosáhnout i zlepšení vlastností dalších typů SOA