

APLIKAČNÍ ARCHITEKTURA INFORMAČNÍHO SYSTÉMU PODNIKU VERSUS SERVISNĚ ORIENTO VANÁ ARCHITEKTURA

M. Mišovič

Došlo 31. června 2007

Abstract

MIŠOVIČ, M.: *Application architectures of enterprise information systems versus service oriented architecture*. Acta univ. agric. et silvic. Mendel. Brun., 2007, LV, No. 6, pp. 233–242

There are two different enterprise IS architectures, older **application architecture** and younger **service oriented architecture**. The application architecture its structural element is a classical web-based application can accept a partial or complex solution of enterprise IS. The first has got problems with data-process-communication integrity disturbing among IS applications. The second is convenient for large enterprises not for small and intermediate. Classical web-based applications are too inflexible to accept necessary changes concerning a progress in the enterprise market-production environment.

The service oriented architecture of IS can be based on **enterprise web-services**. Computerization of such small and flexible units can be given by classical web-services. There is constructed a new web-based application that plays a structural unit role for service oriented architecture. This application consists of a sequence formed by enterprise web-services calling. Enterprise web-services can easily accept necessary changes concerning a progress in the enterprise market-production environment. That's why contemporary younger service oriented architecture seems to be more acceptable for any enterprise than older application architecture.

application architecture, service oriented architecture, service oriented enterprise

Hlavním cílem příspěvku je konfrontovat dvě současné architektury aplikačního software (Business software) podnikových IS (dále často PIS), stávající **aplikační architekturu** a novější **servisně orientovanou architekturu**. Konfrontace bude založena na analýze podstaty obou architektur a na porovnání, v čem

je servisně orientovaná architektura modernější a jak se vyhýbá některým negativním vlastnostem architektury aplikační. Ke splnění hlavního cíle jistě přispěje také derivovaný cíl, spočívající v dostatečném prozkoumání každé z architektur na základě sledování několika relevantních faktorů:

1. **podstata založení architektury**, výstavba aplikačního software PIS dané architektury pomocí **stavebních prvků**,
2. **respektování** datově, procesně a komunikačně orientované integrity mezi dílčími celky aplikačního software PIS,
3. **možnosti realizace změn** v aplikačním software PIS, vyvolaných změnami výrobně-obchodního prostředí podniku.

(1)

Ve svém začátku se příspěvek věnuje stručnému pojetí **globální architektury** a jednotlivých **dílčích**

architektur IS podniku a potom přechází k analýze podstaty **aplikační** a **servisně orientované** architektury

s následnou konfrontací jejich vlastností. **Architektura IS** hraje významnou roli v chápání atributů podstaty a složení informačních systémů. Není sice dodnes definována exaktně, ale její vágně podaný obsah a význam je pro pochopení její role dostatečný. Hlavní rolí architektury IS je integrovat všechny komponenty IS do jednoho konzistentního celku. Je tedy zřejmé, že základem architektury je skupina relevantních faktorů, způsob chápání jejich role a uplatnění této role ve vývoji, implementaci IS a jeho provozu v praxi. Architektura se tedy jako vícerozměrné vlákno táhne celým životním cyklem IS. Důležité je, že podstatou role architek-

tury je podpořit již známé vlastnosti IS podniku jako integrovaného celku. V odborné literatuře je nejprve zavedeno pojetí globální architektury IS podniku, která evokuje jeho přijatelné systémové složení a potom existence dílčích architektur, mezi které patří: **funkční, procesní, datové-informační, softwarové, technické-hardwarové a řízení**.

Podle (Voříšek, 2006) globální architektura IS výrobně-obchodního podniku zachycuje jeho jednotlivé komponenty a jejich vzájemné vazby ze systémového pohledu. Je založena na pěti systémových blocích:

TPS (Transaction Processing System)	podpora stěžejní činnosti podniku na operativní úrovni
MIS (Management Information System)	řízení podniku na taktické ekonomické úrovni
EIS (Executive Information System)	strategické řízení podniku
OIS (Office Information System)	kancelářská informatika
EDI (Electronic Data Interchange)	komunikace v podniku a podniku s okolím

Velmi často mají dílčí architektury následující deskripce jejich významu:

Funkční architektura je rozkladem funkcionality IS podniku do funkcionalit jednotlivých komponent globální architektury

Procesní začíná u událostí, které vyvolávají datovou komunikaci s okolím podniku, pokračuje přes procesní sety, subsety, procesy a končí na podnikových transakcích.

Datová, informační formuje představu o celkové informační základně a její struktuře. Jde o rozpoznání esenciálních skupin informace a jejich vztahů, což je nezbytné pro chod organizace.

Softwarová chápe aplikační software jako **složitý programový systém** a popisuje jeho složení ze samostatných celků, vazeb mezi nimi a způsob výstavby software z nich. Tato architektura je jednou z nejvýznamnějších dílčích architektur, protože charakterizuje stavební vlastnosti „mozku“ IS podniku. Dnes je uznáváno několik známých typů softwarové architektury: **lineární, hierarchická, vrstvená, síťová, klient-server, aplikační, servisně orientovaná a technologická**.

Technická, hardwarová se zabývá konfigurací a rozmístěním výpočetní techniky (počítače, servery, ...) podle organizační struktury nebo podle širších oblastí aktivit podniku, technickým řešením vnitřní a vnější komunikační sítě a jejich rozhraními a volbou základního programového vybavení.

Architektura řízení objasňuje, jakým způsobem se IS podniku podílí na řízení celého podniku a které služby k tomu IS dodává (plánování a distribuce řídicích dokumentů v CMS – Document management System).

Globální architektura IS podniku se projevuje ve všech jeho komponentách, tj. v aplikačním software, užitkovém software, hardware, lidských zdrojích, legislativě, fyzických opatřeních, informační infrastruktuře a dalších. Neustále je nutno respektovat to, že aplikační software IS podniku je umělý abstraktní dynamicko-statický systém zavedený pro vybraný objekt reality – podnik, který by měl nejen věrně komputerizovat stávající aktivity podniku, ale rovněž tak i budoucí změny jeho procesů a dat. Příspěvek se nezabývá všemi komponentami IS podniku, ale jen vlastnostmi jeho aplikačního software.

MATERIÁL A METODIKA

Zpracovávaným materiálem jsou dvě vzájemně odlišné architektury aplikačního software IS podniku, aplikační a servisně orientovaná. První z nich je v současnosti nejužívanější architekturou podnikových IS. Je analyzována její podstata a jsou podány její pozitivní a negativní vlastnosti.

Druhá architektura, která je pro některé typy podniků perspektivní, je označovaná zkratkou SOA (Service-Oriented Architecture), viz (Woods, 2006). Je podána podstata této architektury na tzv. webových službách a je ukázáno, že negativní stránky aplikační architektury se již u SOA neobjeví. Na základě analýzy obou architektur a syntézy získaných poznatků je realizováno porovnání obou architektur ve vybraných významných faktorech (1) a učiněny z toho plynoucí závěry.

VÝSLEDKY A DISKUSE

Aplikační architektura IS podniku

Mnoho pojmů s nimiž se v problematice života podniků setkáváme, např. strategické cíle, požadované chování, funkční organizační struktura a další, můžeme specifikovat a odůvodnit na základě procesů probíhajících v podniku a existujících relevantních vztahů mezi prvky podniku a převést je do podnikové informatiky. Současný procesní přístup úspěšně roztrídil podnikové procesy do jednotlivých historicky vzniklých množin – procesních setů, jako ERP (Enterprise Resources Planning), SCM (Supply Chain Management), CRM (Customer Relationship Management), BI (Business Intelligence), B2B (Business to Business), B2C (Business to Customer) a dalších,

viz (Basl, 2002). Všechny tyto procesní sety jsou předmětem komputizace¹ aktivit podniku, jejímž výsledkem je IS podniku, který stanovuje jistý stupeň podnikové automatizace.

Vzorem pro jednu z možných komputizací libovolného z procesních setů je hierarchie, která začíná u podnikových transakcí a procesů, pokračuje na ucelené procesní řetězce (podle work-flow analýzy), prochází procesními subsety a nakonec končí procesními sety. V komputizaci potom každé z uvedených jednotek odpovídá asociovaná softwarová (SW) forma, přičemž za základní považujeme **aplikaci**. Aplikace se potom stává stavební jednotkou (cihlou) IS podniku, který se může chápat jako systém aplikací s jistými vazbami. Charakter vazeb může být velmi pestrý (datový, procesní, kauzální aj.).

Komputizační hierarchie má tvar:

AKTIVITY PODNIKU

<i>transakce</i>	<i>procedura</i> , vícenásobně využitelná programová jednotka,
<i>podnikový proces</i>	<i>procesní program</i> , který je složen z vyvolání transakčních procedur,
<i>řetězec podnikových procesů</i>	<i>aplikace</i> , která komputizuje ucelenou aktivitu podniku, tzv. řetězec procesů podle work-flow analýzy,
<i>řetězce podnikových procesů</i> v rámci procesního subsetu	<i>balíček aplikací</i> pro subset,
<i>řetězce procesů za všechny subsety</i>	<i>balík aplikací</i> pro procesní set.
téhož procesního setu	

ASOCIOVANÁ SW FORMA

Uskutečněním komputizace s uvedenou procesní hierarchií vzniká **aplikační architektura** IS podniku. Označíme-li symboly BA, BAA, A **balík aplikací**, **balíček aplikací** a **aplikaci** a použijeme-li symbol PIS, potom můžeme hierarchii skladebních rovnic pro IS podniku zapsat následovně:

$$\begin{aligned}
 \text{PIS} &= (\text{BA}^{\text{ERP}}, \text{BA}^{\text{SCM}}, \text{BA}^{\text{CRM}}, \text{BA}^{\text{BI}}, \dots) \\
 \text{BA}^{\text{ERP}} &= (\text{BAA}_1^{\text{ERP}}, \text{BAA}_2^{\text{ERP}}, \dots, \text{BAA}_k^{\text{ERP}}) \\
 &\vdots \\
 \text{BA}^{\text{BI}} &= (\text{BAA}_1^{\text{BI}}, \text{BAA}_2^{\text{BI}}, \dots, \text{BAA}_n^{\text{BI}}),
 \end{aligned} \tag{2}$$

kde $k, \dots, n$ jsou přirozená čísla označující počty procesních subsetů.

Ke každému procesnímu subsetu je po komputizaci vytvořen jistý počet asociovaných aplikací, který odpovídá počtu procesních řetězců v procesním subsetu obsažených. Tedy např. pro Majetek, subset z ERP, je

$$\text{BAA}_{\text{majetek}}^{\text{ERP}} = (\text{A}_1^{\text{majetek}}, \text{A}_2^{\text{majetek}}, \dots, \text{A}_r^{\text{majetek}}),$$

kde r je počet procesních řetězců obsažených v procesním subsetu Majetek.

Vývoj soudobých IS podniků, na platformě procesních setů, je často orientován na postupný nákup partiálních částí, tj. balíků/balíčků aplikací k jednotlivým procesním setům/subsetům, od různých malých softwarových firem. Udržování provozu a nezbytná inovace takto vzniklých podnikových IS, vedla na straně podniků k nemalým mimořádným finančním nákladům. Proč? Vysvětlení je poněkud delší.

Snaha managementu podniku o pokračující komputizaci podnikových procesů a zkvalitnění již komputizovaných, končila obvykle na zavádění dalších a nových modernějších balíků aplikací. To paradoxně, místo zaručení integrity PIS, konvergovalo k její stupňující se roztržitosti. Byla to roztržitost ne uvnitř každého z balíku/balíčku aplikací, ale porucha datové, procesní a komunikačně orientované integrity mezi stávajícími a nově do PIS přidávanými balíky/balíčky aplikací. Zavedení zmíněných ekonomických procesních setů a případně jejich

¹ „komputizace“ má stejný význam jako „realizace počítačem“

subsetů sice sjednotilo pohled na členění procesů podniku a vznik balíků, resp. balíčků aplikací, ale zachování procesně², datově³ a komunikačně⁴ orientované integrity obecně pro nekomplexní řešení PIS nebylo dodržováno. Pro přijatelnou, neredundantní funkcionalitu PIS je ovšem datová, procesní a komunikační konzistence mezi balíky aplikací nutnou podmínkou. Hlavní závadou vývoje PIS s aplikační architekturou je to, že se nepodařilo pro dílčí balíky aplikací zavést stejný pohled na procesy, data a komunikaci PIS s jeho klienty. Nepodařilo se tento pohled povýšit na všemi producenty software dodržovanou zásadu – normu, která by potom umožnila sdílet mezi aplikacemi data, procesy a vzory komunikace se zákazníkem. Pochopitelně, je zcela nemožné za této situace uvažovat o tom, že by jedna aplikace mohla sdílet celou funkcionalitu jiné aplikace, když to není možné ani částečně.

Nežádoucí vývoj PIS je tedy důsledkem několika následujících faktorů:

1. Komplexní řešení podnikového IS, produkované velkou softwarovou firmou (na platformě procesních setů, stylem balíčků aplikací se zabudovanou integrací mezi nimi) je pro střední a malé podniky finančně neúnosné.
2. Malá softwarová firma není schopna produkovat pro podnikový IS komplexní řešení na platformě procesních setů. Dokonce ani balíky aplikací ke stejnému procesnímu setu od různých malých softwarových firem nejsou v PIS zaměnitelné.
3. Konkurence mezi výrobci balíčků aplikací k různým a stejným procesním setům ztěžuje sdílení možných společných datových, procesních a komunikačních struktur.

Vznikly tak PIS, které byly složeny z roztříštěných balíčků aplikací s narušenou integritou orientovanou na procesy, data a komunikaci. Podle (Vokůrková, 2006) trápí samotné podniky v oblasti ICT/IS právě z narušení integrity plynoucí nepříjemné důsledky, jako:

1. Obtížné/nemožné změny v aplikacích stávajících balíčků aplikací.
2. Problematické přidávání nových aplikací do balíku aplikací a zajištění jejich propojení se stávajícími aplikacemi.
3. Nutnost transformace výstupních dat jednoho balíku na vstupní data pro jiný balík.

Tyto požadavky se velmi obtížně řeší v podnikových útvarech IT ve vazbě na požadavky zachování stability, výkonnosti, bezpečnosti a nepřetržitého provozu ICT infrastruktury a PIS. Podnikové útvary IT běžně nemají od dodavatelů balíčků žádnou dokumentaci o datových procesních a komunikačních strukturách jednotlivých aplikací balíku, proto musí rozšíření daného balíku řešit s dodavatelem (body 1, 2). Ten zabezpečí zachování integrity upravovaného balíku. Datová komunikace mezi balíky aplikací, tj. bod 3, musí být řešena buď schopným podnikovým útvarem IT nebo jako zakázka pro vybranou softwarovou firmu. V obou případech musí podnik počítat s mimořádně vzniklými náklady.

Vzhledem k uvedeným problémům s integritou balíčků aplikací v aplikační architektuře PIS je hledání koncepcí/principů pro dodatečné zabezpečení požadované integrace považováno za jeden z hlavních soudobých úkolů systémové integrace. Problematika zabezpečení integrace aplikací v PIS, tj. EAI (Enterprise Application Integration) je pro velké podniky diskutována ve studii (Sodomka, 2005), v níž je stručně popisováno sedm koncepcí/principů (point-to-point, princip middleware, transformace relačních tabulek, princip uživatelského rozhraní, princip využití aplikačního programového rozhraní-API, koncepce SOA a síťové služby, princip využití datových skladů), které umožní odstranit poruchu procesně, datově a komunikačně orientované integrity podnikového IS. Podrobnější informace o koncepcích zabezpečení EAI je v publikaci (Gála, Pour, Toman; 2006). Jisté návody na systematický vývoj PIS malých a středních podniků, které vedou na zavedení metaplatformy vývoje s dodržováním datově, procesně a komunikačně orientované integrity mezi balíky aplikací, jsou diskutovány v (Mišovič, Trenz; 2005) a (Mišovič, 2006).

Za přijatelné řešení zmíněných problémů aplikační architektury nelze považovat to, že na poli evolučního řešení PIS malých a středních podniků se domluví malé softwarové firmy na datové, procesní a komunikační integritě. Konkurenční vztahy takových softwarových firem úspěšně zmíněné řešení blokují. Na druhé straně, spíše se přikláníme k tomu, že velké softwarové firmy urychleně „rozparcelují“ svá komplexní řešení PIS pro velké podniky tak, že uspějí i v evolučním vývoji PIS malého a středního podniku. Problémy s integrací potom u malých a středních podniků – zákazníků zaniknou.

2 Duplicita ve funkcionalitě aplikací, která je dána nedisjunktními množinami procesů (redundance procesů a jejich transakcí) a nemožnost sdílení.

3 Významově a obsahově nekonzistentní množiny datových entit se širokou redundancí, bez možnosti sdílení.

4 Nejednotnost GUI jednotlivých aplikací, bez závislostí a sdílení.

Dílčí závěr

1. Vývoj PIS s aplikační architekturou je velmi transparentní a silně svázaný s procesy podniku a jejich seskupováním do známých procesních setů. Podstatě aplikační architektury rozumí nejen informatici, tedy např. i management podniku různých úrovní.
2. Komputerizace podnikových procesů je prováděna na základě tvorby klasických webových aplikací, které jsou určujícími stavebními prvky. V balících aplikací může být využito odlišných dílčích bází dat a balíky jsou typu Data driven.
3. PIS může být realizován jako komplexní řešení a potom nejsou problémy s procesně, datově a komunikačně orientovanou integrací mezi balíky aplikací (výrobce to zabezpečí). Systém má jednotný pohled na data, na procesy a komunikaci.
4. Jestliže PIS není komplexním řešením, tj. vzniká evolučně, evokuje to možnost poruchy procesně, datově a komunikačně orientované integrity mezi balíky aplikací od různých softwarových výrobců. Na zabezpečení zmíněné integrity se používají již známé koncepce z EAI, to ale vyvolá mimořádné výdaje podniku.
5. Koncepce z EAI neřeší podstatu obecného požadavku na aplikační architekturu: mít velmi flexibilní systém vývoje aplikací s předem stanovenými principy vzájemných vazeb a technologií jejich zabezpečení. EIA pouze odstraňuje závady stávajícího vývoje podnikového IS.
6. Aplikační architektura podnikového IS (bez ohledu je-li to komplexní nebo parciální řešení) neumožňuje pružné změny v aplikacích. Jsou to příliš velké celky a změna v jedné aplikaci vyvolá nutnost změny v mnoha dalších aplikacích. Není tedy možné v aplikacích snadno zavádět pružné reakce na změny v podnikových procesech, které jsou důsledkem změn ve výrobně-obchodním prostředí podniku.

Podnikové služby

Mnozí informatici a IT manažeři podniků chápou vznik servisně orientované architektury PIS jako odpověď na řešení nezabezpečené flexibility stavebních prvků aplikační architektury. V mnoha člancích, jejichž autory jsou zejména informatici, je servisní orientace funkcionality IS podniku přisuzována až aplikačnímu software, které aktivity podniku komputerizuje. Soudí se tedy, že je to apriori až vlastností software a ne podniku samotného. Podnik se potom této architektuře musí přizpůsobit.

Problém je ovšem poněkud složitější. Zapomíná se totiž na již dlouhodobě fungující podniky/systémy založené na konzumaci služeb jiných podniků/sys-

témů. A právě na tuto skutečnost chceme dále upozornit. Následující text je názorem informatika a může vyvolat námitky zejména ze strany expertů na management.

V současné době existují *poradenské systémy*, které mají funkcionalitu vůči svým zákazníkům postavenou na tzv. *poradenských službách*. Jedna poradenská služba může být sestavena z jednoho nebo více poradenských procesů. Sekvence poradenských služeb poskytovaných stejnému zákazníkovi je považována za *poradenský případ*. Podobně fungují i orgány veřejné správy ČR. Poradenské systémy ovšem nejen poskytují, ale rovněž konzumují služby poskytované jinými podniky, případně veřejnou správou.

Je teď otázka, jestli by se nemohla postavit funkcionalita podniku na poskytování vzájemných *podnikových služeb* mezi jeho odděleními (případně divizemi/úseky/sekcemi), včetně chápání toho, že oddělení může poskytovat jisté služby samo sobě. Půjde tedy o jakýsi způsob (se stanovenými pravidly) poskytování podnikových služeb na teritoriu podniku. Je možné, i když vágně, chápat jednu *podnikovou servisní službu jako jednotku aktivity, která může být složena z diskrétní funkcionality podniku (a discrete business functionality) nebo z jistého počtu vzájemně propojených podnikových procesů*.

V souladu s vyslovenou myšlenkou je v podnicích již chápána funkcionalita mnoha oddělení. Patří mezi ně např. *Personální oddělení, Oddělení údržby, Poradenské centrum* (zdravotní, pracovní, psychologické a finanční poradenství), *Podniková hypoteční a úvěrová banka* a další, které mohou být o speciální služby požádány z jiných oddělení. Je zřejmé, že jednotlivá oddělení mají své interní služby a četné, jakoby externí služby, které poskytují jiným oddělením.

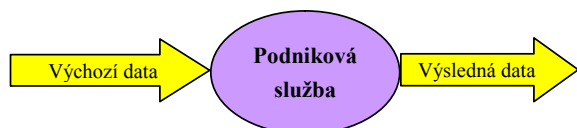
Ne vždy vystačí podnik jen se svými interními podnikovými službami. Je často nucen použít rovněž externí služby jiného podniku, např. *nájemného dopravce* v logistickém řetězci, *marketingových služeb* jiné firmy pro propagaci svých výrobků nebo prodávaného zboží, *finančních služeb, uklízací služby* a pod. Ukazuje se, že servisní kooperace podniků navzájem a oddělení v podniku na základě podnikových služeb má reálné opodstatnění. *Podniková služba* by se tak mohla stát jednotkou pro vyjádření funkcionality podniku jako systému, který konzumuje poskytované služby. Dokonce by se pro stejné obově orientované podniky mohly tyto služby považovat za typové. Služby by tak mohly vytvořit:

- buď jinou podnikovou rovinu než jeho oddělení/divize/úseky/sekce a zaměstnanci
- nebo rovinu zcela vzdálenou a s podnikem vůbec neasociovanou.

Obecně se dá mluvit o zavedení speciálního abstraktního systému, systému podnikových služeb –

SPS (ESS – Enterprise Service System), jehož základními prvky jsou podnikové služby a vazby mezi nimi. Aktivita všech organizačních celků podniku, patřících do podnikových procesních setů, se pak dají vyjádřit pomocí poskytnutí služeb ze systému SPS. Podnik potom buď systém SPS vlastní, anebo systém může vlastnit organizace mimo podnik samotný. Tato organizace se pak stává **poskytovatelem podnikových služeb** a podnik jejich **konzumentem**. Je mnoho podniků, jejichž reálný život je postaven jen na konzumaci služeb, tj. jejich aktivita je značně suplována poskytovanými podnikovými službami. Poskytování služeb, jinými slovy **správa podnikových služeb**, je jistě spojeno s platebním systémem, na kterém je založen profit poskytovatele.

Podnikové služby nejsou rozsáhlými jednotkami aktivity a tedy je snadné je modifikovat a skládat, na čemž je postavena flexibilita systému SPS. Každá služba žádá jistá **výchozí data** (dodává je management konzumenta), která zpracovává a zobrazuje je do jistých **výsledných dat**, která vydává. To je v podniku velmi závažné pro jeho management, který skládání řídí a rovněž výsledná data jedné služby převádí na výchozí data jiné služby, která je pro předchozí následníkem. Pochopitelně, **management správy podnikových služeb** (integrátor podnikových služeb) musí mít dokonalou představu o funkcionalitě poskytovaných služeb a využívat ji pro zabezpečení datové, procesní a komunikační integrity ve funkcionalitě postavené na podnikových službách. To může vést i na požadavek, aby poskytovatel danou službu adaptoval na specifické prostředí konzumenta (např. výrobně – obchodní).



1: Struktura podnikové služby

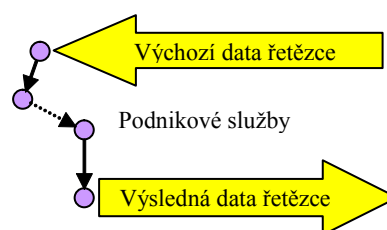
Z již uvedeného plyne, že podnikové služby by měly vyhovovat následujícím požadavkům:

1. Celá funkcionalita podniku (business logic) je rozdělena na podnikové služby, které jsou potenciálně znovupoužitelné a představují nedělitelnou část funkcionality podniku, tj. např. transakci/proces, resp. jinou diskrétní neohraničenou aktivitu.
2. Podnikové služby jsou navzájem nezávislé, lze je sdružovat do celků (např. řetězců), které mohou být považovány za nové podnikové služby.
3. Podnikové služby tvoří systém podnikových služeb se stanoveným rámcem pro vzájemnou spolupráci.
4. Popisy služeb, tj. výchozí a výsledná data, funkcionalita a rámec pro vzájemnou spolupráci jsou

volně přístupné nejen managementu správy podnikových služeb.

Přesto, že podnik značně konzumuje podnikové služby, nemusí mít rozsáhlý management jejich správy. To vše je dáno stupněm komputizace podniku. Pro takový podnik je užitečné z trvalých sekvencí podnikových služeb vytvořit větší jednotky aktivity (business elements), např. **řetězec podnikových služeb**, které jsou opět považovány za služby. S takto vytvořenými řetězci se pracuje jednodušeji, než se samotnými elementárními podnikovými službami.

Řetězec podnikových služeb – nová služba



2: Struktura řetězce podnikových služeb – nová služba

Na základě rychle se měnícího výrobně-obchodního prostředí podniku je nutno adekvátně měnit rovněž podnikové služby, a to na základě změny do služby včleněných transakcí/procesů, popřípadě včleněné diskrétní aktivity. Nejsou pochybnosti o pružnosti, s kterou by to měl příslušný management podniku provádět pro své interní služby a management poskytovatele pro služby externí.

Na základě uvedených myšlenek je zřejmé, že orientaci na správu interních a poskytovaných externích podnikových služeb a postavením veškeré funkcionality podniku na správě podnikových služeb přechází podnik na tzv. servisně orientovaný podnik (SOE – Service-Oriented Enterprise). Tato orientace, která je postavena na podnikových službách, má relevantní dopad na jeho organizační strukturu. Podnik již nemá ty své subjekty a asociovaný management, které by jinak zabezpečovaly konzumované podnikové služby. Vedle toho tato orientace vede na specifický způsob podnikového řízení a ustavení nového typu managementu – managementu správy podnikových služeb, pečujícího o zdárnou manipulaci se všemi podnikovými službami v souladu s **rámcem jejich vzájemné spolupráce** (podstatou tohoto rámce je zachování datové, procesní a komunikační integrity ve funkcionalitě podniku).

Dílčí závěr

1. Bylo ukázáno, že pojetí podnikové služby je reálné, ne zavádějící. Je reálné ovšem i to, že více podniků žije z poskytování podnikových služeb,

- tj. z jejich plného funkcionálního zabezpečení a prodeje a mnoho podniků služby zase konzumuje. Aktivitu podniku se dají reprezentovat podnikovými službami, i když některé z nich realizuje podnik sám.
2. Podniková služba se stává malou, pružnou a znovupoužitelnou jednotkou aktivity, dobrým stavebním kamenem pro širší aktivitu podniku, snadno uplatnitelnou pro více podniků a snadno modifikovatelnou a udržitelnou v souladu s rychle se měnícím výrobně-obchodním prostředím podniků.
 3. Z podnikových služeb lze vytvořit systém, který může zcela suplovat aktivity mnoha subjektů podniku – konzumenta služeb. Za manipulaci s výsledky poskytovaných podnikových služeb je odpovědný specifický management konzumenta. Za uskutečnění poskytované služby je odpovědný poskytovatel.

Podnikový IS se servisně orientovanou архитектурou

Pochopitelně, obecným základem komputizace servisně orientovaného podniku je naprogramování především jeho transakcí, procesů a diskrétních aktivit, které jsou použity v podnikových službách. V úvahu nemusí přicházet jen podnikové procesy, transakce a diskrétní aktivity. Např. od nich odlišné softwarové procesy/transakce mohou vzniknout v době návrhu aplikačního software podniku (Registrace, Přihlášení se a další). Uvažujeme, že pro další implementační úvahy dojde k uplatnění paradigmatu objektového modelování a programování. Při jeho použití jsou transakce a procesy obecně považovány za metody různých, objektovým návrhem vzniklých tříd. Servisně orientovaný podnik ovšem pracuje na podnikových službách a tedy implementace objektového paradigmatu se bude vztahovat na ně.

Naprogramování externí služby provede poskytovatel a podniku dodá způsob využití její funkcionality (např. ve formě instance pro klienta). Kód podnikové služby je „zpřístupněn“ prostřednictvím jeho volání a současného dodání výchozích dat. Zde do popředí vystupuje problém distribuce dat (*výchozích* od konzumenta a *výsledných* od poskytovatele) nebo distribuce pro klienta vytvořené instance. Přirozeně se na splnění takového požadavku hodí právě internet a jeho nativní služba WWW, která požadovanou distribuci zaručuje prostřednictvím svého protokolu HTTP. Ten poskytuje např. přenos mezi uživatelskou aplikací, která je konzumentem podnikové služby (aplikace vyvolává službu) a samotnou službou uloženou na internetovém serveru poskytovatele. Z těchto pochopitelných důvodů jsou poskytované externí podnikové služby nejčastěji naprogramovány jako tzv. *podnikové*

w-slouby (enterprise web-based services), přičemž realizačním prostředkem jsou obecné internetové w-slouby (web-based services). Pokud se programují interní služby podniku, potom vzniklý kód má charakter lokálního použití. Přesto je žádoucí, pro jednotnost použití kódu interních a externích webových podnikových služeb, zřídit pro interní podnikové w-slouby prostor pro jejich uskladnění (repository) na lokálním internetovém serveru podniku a z tohoto prostoru je volat k činnosti různými podnikovými subjekty stejným způsobem. Tím dojde nejen k jednotnému pohledu na oba diametrálně odlišné typy podnikových w-slouby, ale umožní to konzumentovi stát se rovněž poskytovatelem svých, původně interních podnikových služeb. Na základě uvedených myšlenek lze konstatovat, že se podnikové w-slouby stávají základními stavebními kameny aplikačního software podniku.

Podívejme se teď velmi stručně na obecné vlastnosti realizačních internetových w-slouby (Erl, 2004), protože jak později uvidíme, bude jimi evokováno několik velmi prospěšných vlastností na nich založených aplikací a rovněž interpretací pro podnik a jeho PIS.

Konzultační společnost Stencil Group, zdroj (Gála, Pour, Toman; 2006), chápe internetovou w-sloubu jako „*volně spojené, znovupoužitelné softwarové komponenty, které sémanticky zapouzdřují oddělenou funkcionalitu a které jsou distribuovány a jsou programově přístupné přes standardní internetové protokoly*“.

W-slouba jako internetová objektová programová jednotka je plně nezávislá na jiných w-sloubách. Je hostována na zvoleném webovém serveru podobně jako HTML stránky. Ve své specifikaci obsahuje w-slouba následující velmi důležité popisy (v souborech typů WSDL a WSML), z nichž 1. až 3. představují specifická metadata:

1. Popis požadavku a odpovědi:

- a. Obsahuje v XML popis svého rozhraní, popis formátu požadavků a odpovědí, tj. zpráv, včetně formátu výchozích a výsledných dat a jejich omezení, dává k dispozici jednoduchý mechanismus pro realizaci změn svého rozhraní, vše v souladu s XML specifikací WSDL (Web Service Definition Language). To vše tvoří tzv. WSDL dokument w-slouby.
- b. Na základě popisu rozhraní w-slouby je konzument schopen sestavit *požadavek na službu* (včetně dodání výchozích dat) a umí zpracovat odpověď s příslušnými výslednými daty.
- c. Požadavek na službu odpovídá přenosovému protokolu (HTTP, SMTP, FTP) a používá jen data popsaná ve WSDL dokumentu.
- d. Vedle toho musí požadavek dodržet formát daný protokolem SOAP (Simple Object Access Protocol).

2. **Popis komunikace:** obsahuje popis mechanismus komunikace s w-sluzbou. Uvádí se typ komunikace (jednosměrná – není odpověď, požadavek – odpověď, žádost o odpověď, oznámení – zpráva zasílaná více příjemcům a očekává se potvrzení přijetí zprávy).
3. **Popis odkazu:** Je uvedena konkrétní typická síťová URL adresa na fyzickou reprezentaci w-sluzby.
4. **Vlastní kód:** Je velmi často součástí specifikace w-sluzby a pomůže k pochopení její funkcionality. Nativní kód w-sluzby je v péči jejího poskytovatele.

Všechny uvedené popisy jsou veřejné a konzument jich může plně využít. V každém případě jsou postaveny na standardech a lze na nich provádět požadované změny bez nutnosti změn v aplikaci, která je v roli konzumenta. Tato flexibilita je cestou k modifikaci podnikových w-sluzeb takovým způsobem, aby odpovídaly progresivním změnám ve stávajícím prostředí podniku. Z dosud uvedených poznatků plyne, že aplikační software PIS postavené na podnikových w-sluzbách dostává charakter nové architektury ozna-

čované SOA (Service Oriented Architecture), což je pochopitelně něco jiného než dříve objasňovaný SOE. Jestliže SOE je postaven na podnikových sluzbách (interních, externích), je SOA postavena na podnikových w-sluzbách, tj. způsobu komputerizace běžných podnikových sluzeb.

Jestliže v PIS s aplikační architekturou bylo použito klasické programování a aplikace tvořila těžko dělitelný celek, tak aplikace v PIS se servisně orientovanou architekturou je postavena na volání podnikových w-sluzeb se široce přístupnou jejich separovanou modifikací. Nová objektová aplikace, volající podnikové w-sluzby, se stává základní stavební jednotkou aplikačního software PIS se servisně orientovanou architekturou, zcela nezávislou od změn kódu podnikových w-sluzeb při zachování jejich interface. Odlišnosti mezi aplikacemi v předmětných architekturách jsou značné, ovšem strukturalizace (2) může být respektována. Je rovněž zřejmé, že komputerizační hierarchie podniku pracujícího na bázi SOE bude jistě technologicky odlišná od obdobné hierarchie vedoucí na aplikační architekturu, viz následující seznamy aktivit podniku a s nimi asociovaných softwarových forem.

AKTIVITY PODNIKU

<i>transakce</i>	<i>procedura</i> , vícenásobně využitelná programová jednotka pro metodu jisté třídy objektů,
<i>podnikový proces</i>	<i>procedura/-y</i> , vícenásobně využitelné programové jednotky pro metodu jisté třídy objektů,
<i>podniková služba</i>	spolupráce objektů různých tříd na základě zpráv, převedená na podnikovou w-sluzbu,
<i>řetězec podnikových sluzeb</i>	<i>objektová aplikace</i> , jejíž kód obsahuje volání interních a externích podnikových w-sluzeb. Tato aplikace má architekturu SOA.

Na základě dosud uvedených poznatků o SOE a SOA je možno soudit, že jádrem výstavby PIS se servisně orientovanou architekturou je tvorba nově pojatých objektových aplikací – konzumentů podnikových w-sluzeb. SOE a SOA tak vlastně koncipují nové paradigma pro tvorbu takových aplikací. Roli nově pojatých aplikací v PIS se servisně orientovanou architekturou ilustruje obrázek 3.

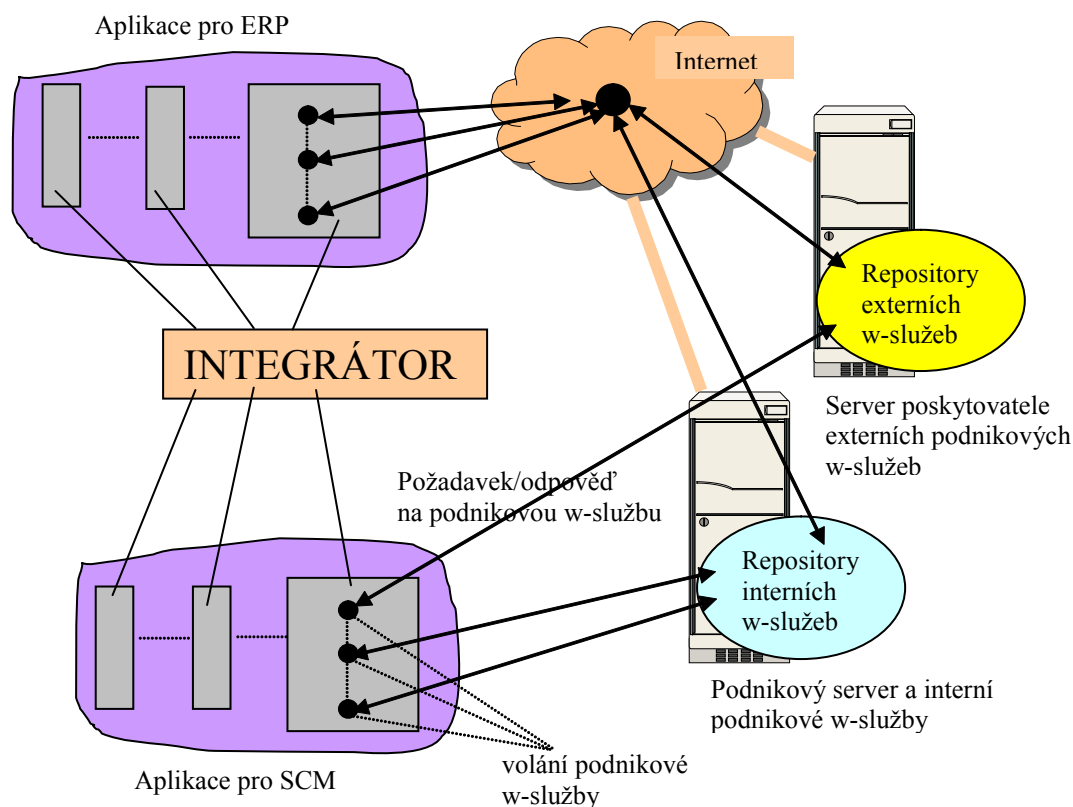
Jestliže tvoříme PIS z aplikací, které mají servisně orientovanou architekturu, nesmíme zapomenout na vzájemnou komunikaci mezi aplikacemi. Datový, procesní a komunikační charakter je potom řešen společným *integrátorem*, pro který může být vymezen všem aplikacím dostupný pracovní rámec (framework). Za podstatu servisně orientovaných aplikací nelze považovat jen volání podnikových w-sluzeb. Aplikace musí respektovat komputerizovaný rámec

ASOCIOVANÁ SW FORMA

vzájemné spolupráce podnikových sluzeb, na kterém je postavena datově, procesně a komunikačně orientovaná integrita PIS. Přesná formulace tohoto rámce není předmětem příspěvku.

Souběžně s vývojem nových aplikací pro SOA je zájem softwarových firem orientován také na vývoj metodiky převodu aplikací z aplikační architektury na podnikové w-sluzby (např. metodika zapouzdření aplikací). Tím sice vznikají poněkud robustnější podnikové w-sluzby, ale další operace s nimi již lze provádět podle SOA.

Vedle toho, mnohé velké softwarové firmy pokusně připravují podnikové w-sluzby různých kategorií a dávají je k veřejné dispozici. Není rovněž u nich opomíjen ani seriózní výzkum SOE – SOA paradigmatu a tvorby nových objektových aplikací se servisně orientovanou architekturou.



3: Volání podnikových w-sluzeb z aplikací PIS pro dva procesní sady ERP a SCM

Jsou uznávané prognózy, že výrobci aplikačního software postupně opustí aplikační architekturu a přejdou na modernější SOA. Ukazuje se, že tento přerod bude pružnější u větších softwarových firem, které dokážou rychle organizovat přípravu svých výkonných programátorů. Zdá se, že tyto firmy mohou „oplatit“ malým softwarovým firmám to, že dílčím řešením balíků aplikací k vybraným procesním setům podniku jim „vypálili rybník“ a ony se svým komplexním řešením IS podniku se nedokázaly uplatnit u malých a středních podniků.

Tyto prognózy je ovšem nutné chápat velmi citlivě, protože ne všem organizacím se SOA hodí. Organizace se zažitým a spolehlivým PIS nebudou chtít vynaložit náklady na jeho převod na SOA.

Dílčí závěr

1. Podnikové služby obou typů (interní a externí) se dají komputertizovat na stejné internetové platformě do podnikových w-sluzeb, což vede na jednotný způsob jejich použití. Aplikační software pro PIS tak získává novou architekturu SOA. Tato architektura také umožní, aby se konzument stal rovněž poskytovatelem podnikových w-sluzeb. Produkce komputertizovaných interních podnikových služeb může být svěřena

skupině IT odborníků podniku. To jaksí evokuje názor, že komplexní, krabicová řešení PIS nebudou již v budoucnu tak implementačně úspěšná. Uznává se, že SOA umožní lépe asociovat IT na výrobně-obchodní potřeby podniku.

2. Podnikové služby komputertizované na podnikové w-sluzby přebírají všechny vlastnosti obecných internetových w-sluzeb, kterými jsou realizovány.
3. Základem PIS se servisně orientovanou architekturou jsou podnikové aplikace, tj. komputertizace řetězců podnikových služeb k různým procesním setům. Každá taková aplikace může být vlastně konzumentem obou typů podnikových w-sluzeb (interních, externích). SOE a SOA jsou základem nového architektonického paradigmatu tvorby PIS.
4. Velmi přílehlavý PIS malého podniku orientovaný jen na vybraný procesní subset může být sestaven jako spolupracující aplikace pomocí podnikových externích a interních w-sluzeb malou skupinou podnikových IT odborníků. Integrace aplikací je zabezpečena specifickým integrátorem, který je součástí PIS.
5. SOE umožňuje silnou fragmentaci aktivit podniku a SOA je zabezpečí IT technologiemi. Tato fragmentace je úzce spojena s vytvářením flexi-

bility podnikových služeb a jejich komputerizace – podnikových w-sluzeb. Pružná modifikace podnikových služeb, přenesená rovněž na podnikové w-sluzby, umožní podniku přizpůsobení PIS na dynamické progresivní změny ve výrobně-obchodním prostředí podniku.

6. Na základě současného vývoje zkušebních verzí podnikových IS se dá předpokládat, že architektura SOA rychle nahradí aplikační architekturu a změní životní cyklus vývoje IS podniku.

SOUHRN

Příspěvek ukázal problémy s datovou, procesní a komunikační integrací u PIS s aplikační архитектурou pro malé a střední podniky, u nichž PIS vznikl evoluční cestou. Příspěvek vysvětlil podstatu podnikové služby a na ní založeného funkcionálního SOE (Service Oriented Enterprise), která poskytuje podniku jiné možnosti ve vývoji jeho informačního systému než dřívější aplikační architektura. Komputerizace podnikových služeb pomocí obecných internetových w-sluzeb otevřela cestu k servisně orientované architektuře aplikačního software PIS, se změnou filosofické koncepce jeho tvorby. Mimo jiné, tato architektura netrpí problémem poruch datové, procesní a komunikační integrity svých částí.

Bylo také ukázáno, že nová orientace na SOA je spojena se třemi odlišnými projevy. Prvním je problém technologický, který je silně provázán s technologií pro tvorbu webových služeb. Druhým je SOE, tj. nový pohled na podnikové procesy, které se začínou chápat jako služby, pro jejichž komputerizaci se použije právě platformy SOA, čímž IS podniku získá nový typ architektury. Třetí projev je čistě obchodní, tj. jaký je, resp. bude odbyt IS podniků založených na SOA a jak rychle tento odbyt potlačí prodej IS dřívější aplikační architektury.

aplikační architektura, servisně orientovaná architektura, servisně orientovaný podnik

LITERATURA

- BASL, J.: *Podnikové informační systémy. Podnik v informační společnosti*. Grada Publishing, Praha, 2002. ISBN 80-247-0214-2.
- WOODS, D., MATTERN, T.: *Enterprise SOA. Designing IT for Business Innovation*. O'REILLY, 2006. Tokyo. ISBN 0-596-10238-0.
- ERL, T.: *Service-oriented Architecture. A Field Guide to Interacting XML and Web Services*. Prentice Hall, USA, 2006. ISBN 0-13-142898-5.
- FIALA, F., MINISTR, J.: *Průvodce analýzou a modelováním procesů*. Ostrava: VŠB-TU, 2003. 110 s. ISBN 20-248-0500-6.
- SODOMKA, P., HABÁŇ, J.: *Integrace podnikových aplikací. Analytická studie CVIS*, 2005. Uvedeno na adrese http://www.cvis.cz/index_cz.htm
- MIŠOVIČ, M., TRENC, O.: *Malé SW firmy a komplexní řešení podnikového IS*. Konference Informatika XVII, Karlov pod Pradědem, 2005. Sborník s. 126–133. ISBN 80-7302-110-2.

- VOKŮRKOVÁ, L.: *Co trápí firmy v oblasti IT?* COMPUTERWORLD č. 5, roč. XVII, IDG Czech, Praha, 2006.
- GÁLA, L., POUR, J. – TOMAN, P.: *Podniková informatika*. Grada Publishing, Praha, 2006. ISBN 80-247-1278-4.
- VOŘÍŠEK, J.: *Strategické řízení informačního systému a systémová integrace*. Management Press, Praha, 2006. ISBN 80-85943-40-9.
- ŘEPA, V.: *Podnikové procesy. Procesní řízení a modelování*. Grada Publishing, Praha, 2006. ISBN 80-247-1281-4.
- MIŠOVIČ, M.: *Produkce integrovaných balíků aplikací pro podnikový IS*. In: Sborník z konference Svět informačních systémů 2006. Zlín, Univerzita Tomáše Bati ve Zlíně, 2006. ISBN 80-7318-400-1.
- KRÁL, J., ŽEMLIČKA, M.: *Architektury orientované na služby jako nové paradigma: mýty, problémy, výzvy, přínosy*. In: Sborník z konference Systems Integration 2004. Praha, 2004. Vysoká škola ekonomická.

Adresa

Prof. RNDr. Milan Mišovič, CSc., Ústav informatiky, Mendelova zemědělská a lesnická univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika, e-mail: misovic@pef.mendelu.cz