

Architektury integrace pro rozsáhlé informační systémy a ekonomické aspekty systémové integrace

Michal KÖKÖRČENÝ, University of Hradec Královéⁱ

Abstract

Almost in no organization do we have information systems and applications without cooperation or communication with other systems. Usually it is necessary to integrate information systems and applications in an organization with systems of other institutions. Nowadays, integration of existing and new implemented information systems is a key subject of interest for every organization. In this paper we will introduce basic concepts of integration styles and architectures and we will point out differences, advantages and disadvantages of these approaches. Furthermore, we will present a simple economic cost model for various integration architectures and approaches.

Keywords

Costs, economic model, integration architecture, system integration.

JEL Classification: M150, A120

ⁱ Faculty of Informatics and Management, University of Hradec Králové, Rokitanského 62, 500 03 Hradec Králové 3, Czech Republic.
michal@74.cz

1. Úvod

V současné moderní době dochází k exponenciálnímu nárůstu požadavků na množství funkcí IS/ICT (Information Systems/Information and Communication Technologies) v organizaci jakožto celku – nikoliv k exponenciálnímu nárůstu požadavků na počet funkcí jednoho informačního systému či aplikace – to je důležité rozlišovat. Tím se zvyšují požadavky na provázanost a spolupráci jednotlivých informačních systémů a aplikací v organizaci. Klíčovým aspektem každého systému je tedy možnost jeho integrace do stávajícího prostředí IS/ICT v organizaci. Význam a důležitost otázky systémové integrace dále zdůrazňuje skutečnost, že výsledné integrované řešení má značný dopad na celou organizaci, svým rozsahem ovlivňuje velké množství uživatelů a současně je jeho realizace finančně značně nákladná (Sutherland a Heuvel, 2002).

Integration of software systems has during the last decades become one of the most important, as well as resource consuming, activities in enterprise software system management (Andersson a Johnson, 2001).

Většina autorů se problematikou systémové integrace zabývá spíše z pohledu technického či organizačního. V tomto příspěvku se budeme zabývat otázkou integrace informačních systémů a aplikací především z pohledu ekonomického, kdy jedním z faktorů při rozhodování o volbě architektury integrace jsou náklady na její realizaci.

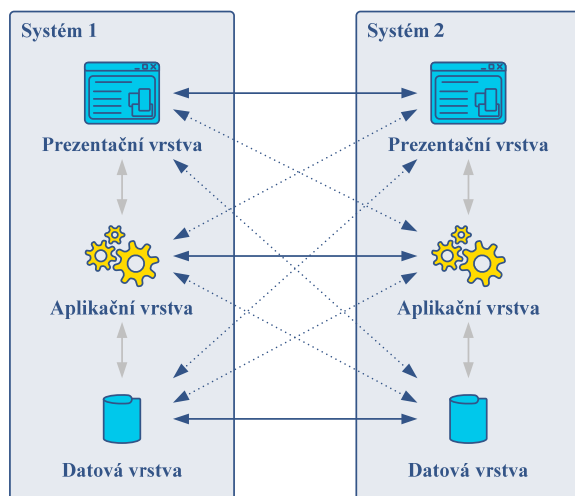
Kapitoly 2, 3 a 4 popisují možné úrovně integrace systémů, typy integračních stylů a aktuálně používané architektury integrace. V kapitole 5 je prezentován ekonomický model, který popisuje výši nákladů na integraci pro různé architektury a pro různé vstupní parametry a který umožňuje porovnat různé varianty integrace.

2. Úrovně integrace

Integraci informačních systémů a aplikací lze provádět několika způsoby. Z pohledu architektury IS ji lze realizovat na několika úrovních (Pour a kol., 2009):

- integrace na úrovni uživatelského rozhraní (prezentační vrstvy),
- integrace na úrovni aplikační vrstvy,
- integrace na úrovni datové (perzistentní) vrstvy,
- integrace mezi rozdílnými vrstvami architektury IS.

Možnosti integrace jsou znázorněny na obrázku 1.



Obrázek 1 Integrace na různých vrstvách architektury IS

2.1 Integrace na úrovni uživatelského rozhraní

V organizaci jsou obvykle využívány různé informační systémy a aplikace s odlišnými uživatelskými rozhraními. Tato rozdílnost pak z uživatelského pohledu značně omezuje intuitivnost ovládání jednotlivých systémů. Integrace na této úrovni je nejčastěji řešena prostřednictvím portálových řešení, která umožňují přístup k informacím z různých zdrojů pomocí jednotného uživatelského rozhraní. Portál obvykle představuje jednotnou bránu či vstupní bod do podnikového informačního systému a zastává funkci centrálního prvku na úrovni zobrazování informací uživatelům.

2.2 Integrace na úrovni aplikační vrstvy

Na této úrovni integrace dochází ke sjednocování obchodní logiky jednotlivých informačních systémů a aplikací. V tomto případě dochází ke vzájemné interakci systémů, čímž mezi nimi vzniká těsná vazba (tight coupling). Z pohledu komunikace mezi jednotlivými systémy je cílem především její sjednocení v rámci celé organizace (Hasselbring, 2000).

2.3 Integrace na úrovni datové (perzistentní) vrstvy

Většina informačních systémů a aplikací využívá pro trvalé uložení provozních dat svou vlastní databázi. V prostředí organizace, kde je využíváno více systémů s oddělenými datovými úložišti, pak často dochází ke vzniku redundancí a nekonzistence dat. Cílem integrace na této vrstvě je především eliminovat tyto nedostatky a poskytnout informačním systémům a aplikacím přístup k datům ostatních systémů. Rovněž se může jednat o zajištění pohledu na celopodniková data pro analytické účely.

2.4 Integrace mezi rozdílnými vrstvami architektury IS

Integrace služeb sjednocuje předchozí úrovně integrace do jedné vrstvy. Pojem služba je zde chápán jako nezávislá entita funkcionality, která zahrnuje všechny vrstvy architektury IS. Na službu lze také nahlížet jako na část business procesu dané organizace. Princip integrace na úrovni služeb je znázorněn na obrázku 2.

3. Integrační styly

Integrační styly představují obecné přístupy k integraci informačních systémů či aplikací, resp. představují možné způsoby komunikace mezi jednotlivými systémy. Z tohoto pohledu existuje několik základních přístupů (Andersson a Johnson, 2001; Pour a kol., 2009):

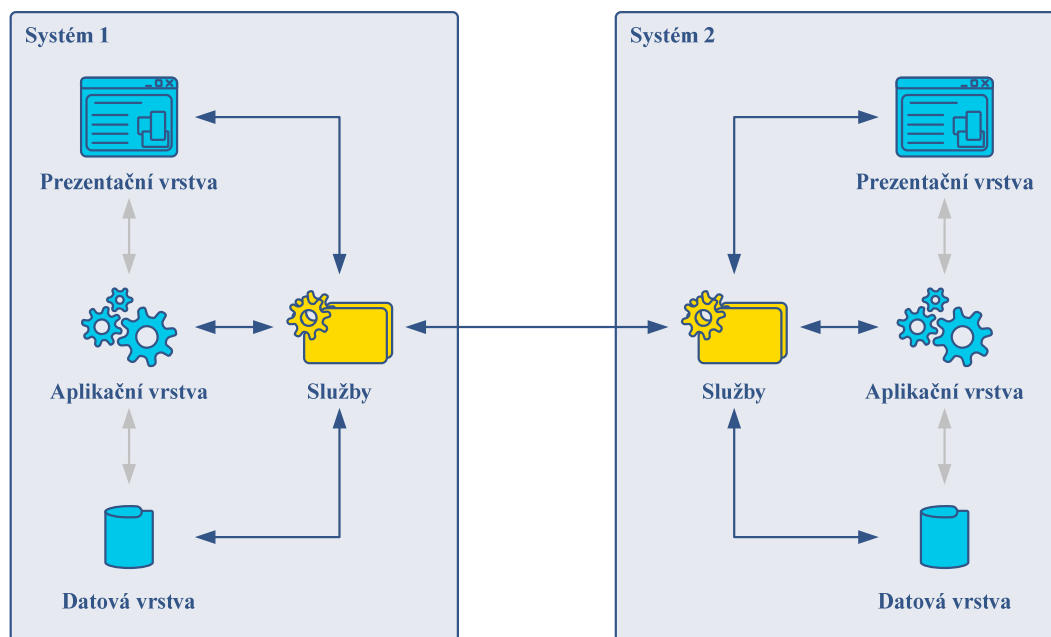
- integrace na datové (perzistentní) vrstvě: přenos souborů, sdílená databáze, sdílené soubory, replikace dat, ETL procesy,
- integrace na aplikační vrstvě: vzdálené volání procedur, messaging,
- integrace na prezentační vrstvě: portálové řešení.

Integrační styly neřeší otázku konkrétní architektury integrace, neřeší tedy způsob uspořádání vazeb mezi jednotlivými systémy či aplikacemi.

3.1 Přenos souborů

Jeden z nejjednodušších způsobů integrace na datové vrstvě je export, přenos a import souborů. Rozhraní mezi systémy je v tomto případě realizováno samotným souborem, který má určitý definovaný formát, používaný jak zdrojovou, tak i cílovou aplikací. Výhodou je zejména skutečnost, že ukládání dat do souborů podporují všechny operační systémy a pro realizaci není nutné využívat další technologie. Princip ukazuje obrázek 3.

U tohoto stylu integrace je problémem především zajištění unikátnosti názvů souborů a jejich odstranění v případě, kdy již nejsou potřebné. Již samotné určení, že soubor není dále potřebný, představuje problém,

**Obrázek 2** Integrace na vrstvě služeb**Obrázek 3** Přenos a sdílení souborů

Zdroj: Pour a kol., 2009

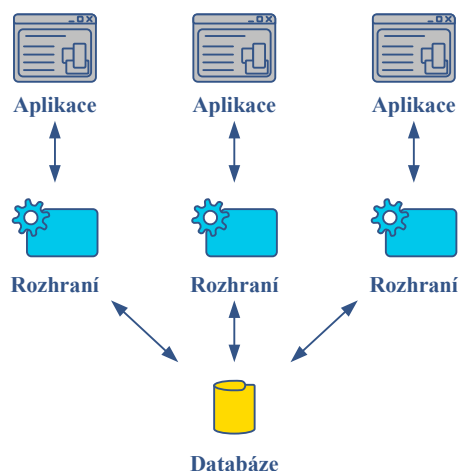
který je nutné v průběhu integrace vhodně vyřešit, včetně stanovení kompetencí k odstraňování souborů. Dalším problémem je pak souběžný přístup k exportovanému souboru. Rovněž je nutné vyřešit otázku, jak často bude export či import souborů probíhat a zda jednotlivé kroky budou prováděny automatizovaně nebo budou iniciovány ručně uživatelem. Nedostatkem tohoto stylu integrace pak může být neaktualnost dat, která jsou přenášena a následně importována s určitým časovým zpožděním (Hohpe a Woolf, 2003).

3.2 Sdílená databáze

U tohoto stylu integrace používá více informačních systémů či aplikací jednu databázi pro společné ukládání provozních dat. Jednotlivé systémy tedy využívají společnou databázi a společný datový model. Výhodou oproti předchozímu stylu integrace pomocí přenosu souborů je aktualnost dat, kdy nedochází k časovému zpoždění a veškerá data jsou vždy aktuální (Hohpe a Woolf, 2003). Způsob integrace prostřednictvím sdílené databáze znázorňuje obrázek 4.

Obecně je použití sdílené databáze slabým místem celého řešení, kdy může dojít ke zpomalení běhu jednotlivých aplikací způsobenému souběžným

přístupem k této společné databázi. Klíčovým problémem tohoto integračního stylu je zejména vhodně

**Obrázek 4** Sdílená databáze

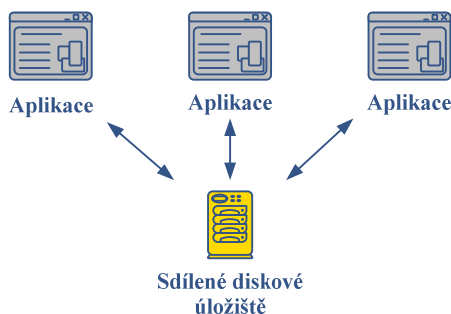
Zdroj: Pour a kol., 2009

navržený datový model a následně jeho údržba. Správně navrhnout datový model tak, aby současně vyhovoval potřebám všech integrovaných aplikací,

bývá obvykle podstatně obtížnější a časově náročnější úkol než v případě návrhu pro jeden samostatný systém. Datový model je pak obvykle poměrně rozsáhlý, což může přinášet problémy z hlediska výkonu. Problémy mohou nastat také u některých starších databázových technologií, které mnohdy nedostatečně podporují souběžné zpracování transakcí či případně vůbec nepodporují transakční zpracování dat.

3.3 Sdílené soubory

V případě integrace pomocí sdílených souborů je architektura a princip podobný předchozímu stylu, tj. integraci pomocí sdílené databáze. Zde se však jedná o využití společného diskového úložiště a společných souborů obsahujících provozní data. Jedná se o jednoduchý způsob integrace na datové vrstvě, jehož problémem je zejména zajištění a ošetření souběžného přístupu k souborům. Princip je znázorněn na obrázku 5.



Obrázek 5 Sdílené soubory

3.4 Vzdálené volání procedur

U tohoto stylu se jedná o integraci na aplikační vrstvě architektury IS, kdy dochází ke sdílení určitých funkcionalit jednotlivých systémů (nikoliv pouze dat jako v předchozích případech). Požadované funkcionality jsou nabízeny k využití jiným aplikacím. Jedná se o vzdálené volání procedur jedné aplikace z jiných aplikací. V tomto případě dochází ke vzájemné interakci systémů, čímž mezi nimi vzniká těsná vazba (tight coupling). V principu se jedná o synchronní komunikaci (Pour a kol., 2009), kdy volající aplikace čeká na dokončení volané procedury, teprve pak volající aplikace pokračuje v dalším zpracování. Mechanismus vzáleného volání procedur ukazuje obrázek 6.

Nedostatkem tohoto stylu integrace je především vznik těsných vazeb mezi jednotlivými aplikacemi, což způsobuje řadu problémů při narůstajícím počtu integrovaných systémů. Dalším problémem pak často bývá nižší výkon při vykonávání vzdálené procedury než v případě volání lokální procedury. Rovněž spolehlivost jednoho systému v tomto případě závisí na spolehlivosti volaných procedur a aplikací (Hohpe a Woolf, 2003). Chování a spolehlivost jednoho

díličího systému či subsystému tak může mít značný vliv na chování a spolehlivost velké části podnikového informačního systému jako celku.

3.5 Messaging

Na rozdíl od vzdáleného volání procedur představuje messaging asynchronní způsob komunikace (Pour a kol., 2009), který je do jisté míry podobný integračnímu stylu využívajícímu přenášení souborů. Oproti běžným souborům jsou zprávy obvykle menší, jejich vytvoření, odeslání a doručení pak bývá poměrně rychlé a jednoduché. V tomto případě volající aplikace nečeká na odpověď adresáta a po odeslání zprávy může pokračovat v dalším zpracování. Pokud je vyžadováno potvrzení o doručení zprávy, volající aplikace se k němu může vrátit až později, po příchodu tohoto potvrzení. Způsob zasílání zpráv mezi aplikacemi ukazuje obrázek 7.

Při tomto způsobu komunikace není nutné, aby v okamžiku odesílání zprávy byl příjemce dostupný. Zprávy jsou ukládány do odchozí a příchozí fronty, ze kterých mohou být vyzvednuty i později. Dočasné přerušení komunikace tedy nemusí zásadně ovlivnit chování podnikového informačního systému jako celku.

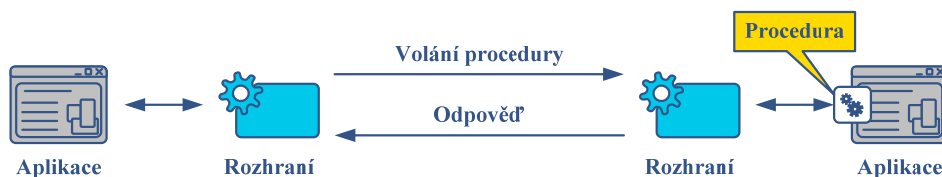
4. Architektury integrace

Architektura integrace řeší otázku, jakým konkrétním způsobem budou uspořádány vazby mezi jednotlivými systémy či aplikacemi (Pour a kol., 2009). Jedná se tedy o topologii integrovaného řešení, nikoliv o způsob komunikace mezi jednotlivými systémy.

4.1 Point to Point

Nejjednodušším způsobem jak integrovat více informačních systémů či aplikací do jednoho funkčního celku je metoda Point to Point. Princip je znázorněn na obrázku 8. Tento způsob je jednoduchý a efektivní pro malý počet integrovaných systémů, kde se nepředpokládá další rozšiřování či změny integrovaného řešení. Značnou nevýhodou tohoto přístupu je skutečnost, že každý systém musí mít vytvořené přímé rozhraní se všemi systémy, se kterými je propojen (syndrom n^2). S rostoucím počtem systémů pak dochází k velkému nárůstu počtu rozhraní mezi jednotlivými systémy a integrace se tak stává více komplexní (Andersson a Johnson, 2001).

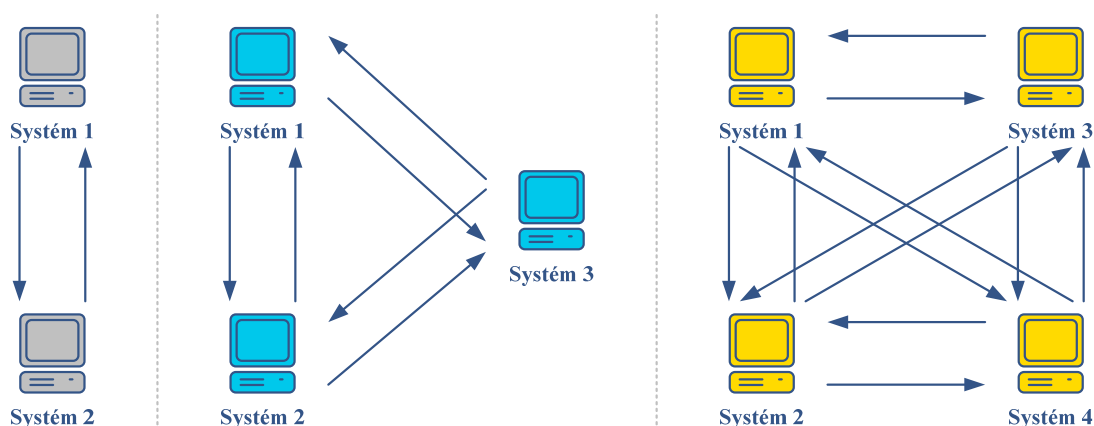
Typicky má nárůst počtu implementovaných rozhraní s rostoucím počtem systémů exponenciální charakter. Úměrně tomu pak rostou náklady jak na samotnou systémovou integraci, tak i následně na celkový provoz IS/ICT. Každé rozhraní mezi dvěma systémy představuje určité náklady na jeho implementaci a současně na jeho provoz.

**Obrázek 6** Vzdálené volání procedur

Zdroj: Pour a kol., 2009

**Obrázek 7** Messaging

Zdroj: Pour a kol., 2009

**Obrázek 8** Integrace stylem Point to Point

Integraci stylem Point to Point je možné realizovat pomocí celé řady standardních technologií. Často se však pro komunikaci mezi dvěma systémy využívají i specifické či nestandardní technologie, formáty a protokoly. Rovněž se často kombinují různé integrační styly. Tím se pak systém jako celek stává velmi heterogenním a obtížně udržitelným.

4.2 Message Broker

Message Broker, rovněž se používají termíny integrační broker, informační broker nebo integrační server, je softwarové řešení, které představuje prostředníka pro zasílání zpráv mezi jednotlivými systémy. Výhodou tohoto řešení je odstínění příjemce zprávy od jejího odesílatele a skutečnost, že Message Broker tvoří jedno komunikační rozhraní, tj. jeden komunikační bod, pro odesílání zpráv všem příjemcům (Chandra a kol., 2003). Místo odeslání zprávy přímo příjemci se zpráva odešle Message Brokeru. Ten pak směruje zprávu příslušnému příjemci, případně i více příjemcům. Současně může dojít – v případě potřeby – k validaci zprávy anebo k její transformaci do jiného formátu, čímž je možné vyřešit nekompatibilitu mezi komunikačními body.

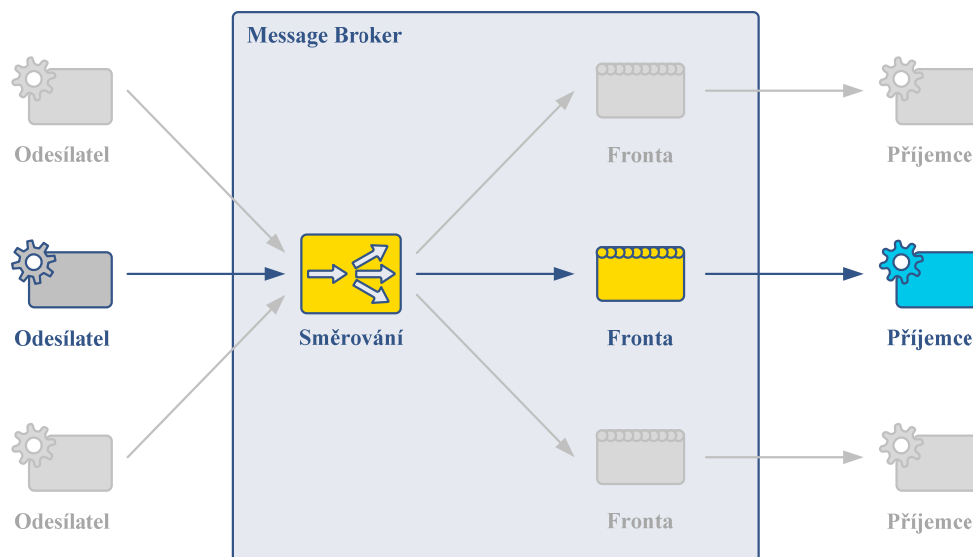
Způsob zasílání zpráv u Message Brokeru ukazuje obrázek 9.

Základem komunikace prostřednictvím Message Brokeru je zpráva, která je zasílána příjemcům na základě jejich logického jména registrovaného na serveru. Message Broker zasílá zprávu pouze příslušným příjemcům a nezatěžuje tak komunikační kanály ostatních systémů. Message Broker je typicky centralizované řešení a tím představuje potenciální riziko a slabé místo této architektury. Problémem může být výkon, dostupnost a spolehlivost při vysokém vytížení tohoto centrálního prvku. Případný (dlouhodobější) výpadek může mít značný dopad na funkčnost velké části podnikového informačního systému jako celku.

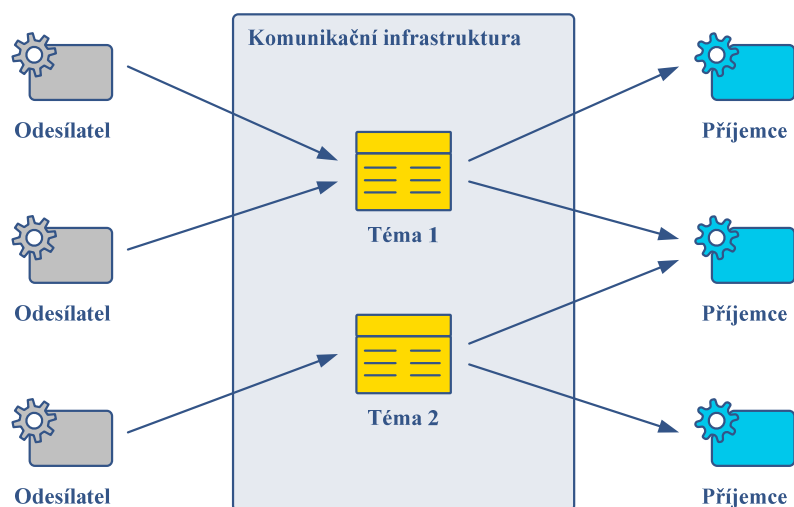
4.3 Publish/Subscribe

Metoda Publish/Subscribe představuje zcela odlišný přístup k zasílání zpráv mezi systémy. Namísto klasického zasílání na základě adresy či jména příjemce probíhá doručování zpráv podle jejich obsahu (Trowbridge a kol., 2004). Princip je znázorněn na obrázku 10. V tomto případě hovoříme o tzv. tématech (topics). Jednotliví příjemci se na

serveru přihlašují k odběru konkrétních témat, která jim jsou následně



Obrázek 9 Message Broker



Obrázek 10 Publish/Subscribe

Zdroj: Trowbridge a kol., 2004

automaticky doručována. Odesílatel tak může zaslat jednu zprávu současně více příjemcům, resp. odesílatel zprávy nemá přímou kontrolu nad tím, jakým příjemcům je zpráva doručena.

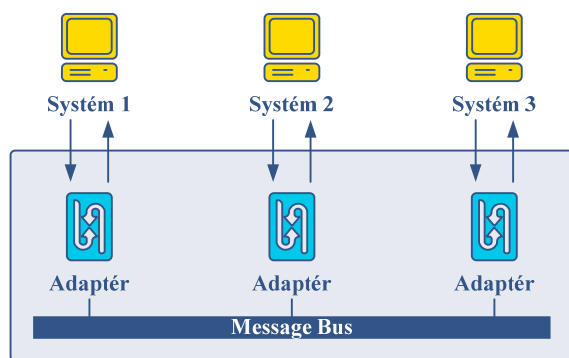
Výhoda tohoto přístupu spočívá v tom, že je možné integrovat nový systém, aniž bychom museli modifikovat stávající systémy. Přidání nového systému nemusí nutně ovlivnit stávající systémy, protože zprávy se nezasílají konkrétním příjemcům – nový systém se pouze zaregistruje k odběru příslušných témat. Při tomto způsobu integrace je možné, oproti architektuře Point to Point nebo Message Broker, dosáhnout volnější vazby (loose coupling) mezi jednotlivými systémy.

4.4 Message Bus

Princip tohoto přístupu je podobný jako u Message Brokeru, tzn. je zde použit jeden centrální bod komunikace, který představuje společnou infrastrukturu zprostředkávající přenos dat mezi jednotlivými systémy či subsystemy. I u této technologie je klíčovou vlastností vzájemné odstínění jednotlivých komunikačních bodů a určitá platformní nezávislost. Oproti Message Brokeru však Message Bus poskytuje větší funkcionalitu. Na rozdíl od Message Brokeru představuje Message Bus decentralizované řešení. Většina logiky je implementována v tzv. adaptérech, které představují rozhraní mezi jednotlivými systémy, samotná sběrnice

(bus) pak zajišťuje pouze přenos zasílaných zpráv. Princip technologie Message Bus ukazuje obrázek 11.

Dalším rozdílem mezi oběma přístupy je to, že Message Broker umožňuje, mimo jiné, směrování zasílaných zpráv příslušným příjemcům – Message Bus naopak umožňuje vytvoření definovaných komunikačních kanálů, nejedná se tedy o směrování zpráv. Message Bus obvykle poskytuje prostředky pro připojení a odpojení systému bez vlivu na ostatní systémy. Přidání nebo odebrání systému tak nemusí nutně ovlivnit stávající systémy, protože zprávy se nezasílají konkrétním příjemcům – nový systém se pouze zaregistruje k odběru příslušných témat. Zasílání zpráv tedy probíhá podobně jako u metody Publish/Subscribe (Trowbridge a kol., 2004). Při tomto způsobu integrace je možné dosáhnout volnější vazby (loose coupling) mezi jednotlivými systémy.



Obrázek 11 Message Bus

Zdroj: Pour a kol., 2009

V případě technologie Message Bus se zprávy jednotlivým příjemcům nezasílají přímo, ale pomocí fronty pro příchozí zprávy a fronty pro odchozí zprávy. Odesílaná zpráva je předána do odchozí fronty, Message Bus pak zprávu vyzvedne a předá do příslušné příchozí fronty adresáta či adresátů.

5. Ekonomické aspekty integrace

V této kapitole se budeme zabývat ekonomickými aspekty systémové integrace, tj. jakým způsobem ovlivňuje použitá architektura a počet integrovaných systémů náklady na realizaci. Většina organizací obvykle stojí v určité době před zásadním rozhodnutím, jakou architekturu integrace, technologii, dodavatele apod., zvolit, přičemž toto rozhodnutí je pro budoucí vývoj velmi významné.

5.1 Volba architektury integrace

Obvykle se nesetkáme s případy, kde by informační systémy a aplikace byly provozovány zcela samostatně, bez jakékoli integrace s jinými systémy v rámci organizace nebo i mimo organizaci. Z historických důvodů, kdy byly v organizaci postupně

zaváděny nové a nové systémy, jsou jednotlivé aplikace typicky integrovány stylem Point to Point. Tento způsob je ze své podstaty nejjednodušší, pro malý počet integrovaných systémů efektivní, méně nákladný a z technického hlediska přímočarý.

S postupným rozvojem IS/ICT v organizaci se však integrace každého nového systému či aplikace, ale také údržba podnikového informačního systému jakožto celku stávala nákladnější a z technického hlediska komplikovanější. Z těchto důvodů je mnohdy nutné zvážit změnu architektury integrace informačních systémů a aplikací v rámci organizace tak, abychom zajistili dostatečnou efektivitu a pružnost pro budoucí změny a zavádění nových systémů. Přestože existuje mnoho různých integračních architektur, nejčastěji existují v principu dvě možná řešení, resp. zvažují se v principu dvě odlišné varianty: zachování současného stavu, tj. pokračovat v integraci stylem Point to Point, nebo přechod na jinou architekturu integrace (například ESB, příp. obdobnou, viz Schmid a kol., 2005), která umožňuje vytvářet volnější vazby mezi jednotlivými systémy a snižovat počet rozhraní nutných pro funkci podnikového informačního systému jakožto celku.

The Enterprise Service Bus (ESB) is the infrastructure which underpins a fully integrated and flexible end-to-end service-oriented architecture (SOA). (Schmid a kol., 2005)

Obě dvě uvedené alternativy mají své opodstatnění a jsou více či méně vhodné pro různé situace. Mezi faktory ovlivňující rozhodnutí o volbě architektury integrace a současně finanční náklady na realizaci patří zejména:

- počet integrovaných systémů a aplikací,
- míra integrace (provázanosti) jednotlivých systémů a aplikací, tj. počet rozhraní mezi jednotlivými systémy,
- předpokládaný počet nově zaváděných (integrovaných) systémů v budoucnu,
- náklady na realizaci a údržbu jednoho rozhraní,
- náklady na zavedení nové architektury integrace.

5.2 Počet rozhraní mezi systémy

Celkový počet rozhraní mezi systémy při integraci stylem Point to Point (Ondruška, 2009) je v extrémním případě:

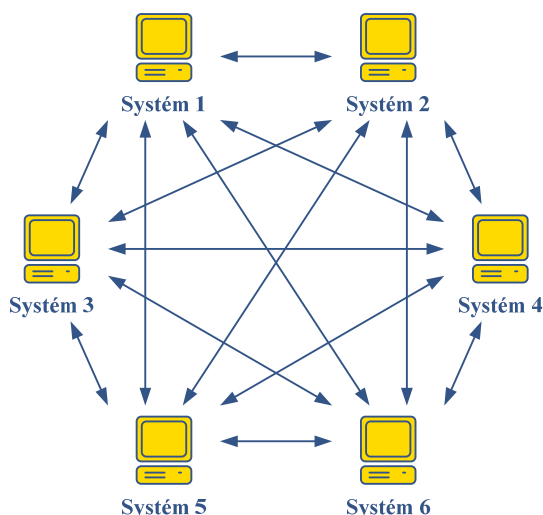
$$I_{PiPEx} = \frac{n(n-1)}{2}, \text{ kde } n \geq 1. \quad (1)$$

Proměnná n představuje počet integrovaných systémů, I_{PiPEx} má exponenciální charakter. Uvedený vztah představuje situaci, kdy je každý informační systém v rámci podnikového informačního systému provázán se všemi ostatními systémy či aplikacemi

(Ondruška, 2009). Provázanost systémů v případě integrace stylem Point to Point ilustruje obrázek 12.

Reálně se ale jedná o ojedinělé případy, obvykle je míra integrace (provázanosti) nižší, typicky není potřeba integrovat všechny systémy navzájem. Reálný počet rozhraní mezi systémy při integraci stylem Point to Point je pak:

$$I_{PIP} = k \cdot I_{PiPEx} = k \cdot \frac{n(n-1)}{2}, \text{ kde } k \in \langle 0; 1 \rangle. \quad (2)$$



Obrázek 12 Integrace stylem Point to Point – extrémní případ

Parametr k představuje míru integrace jednotlivých systémů. Hodnota 1 znamená úplnou provázanost všech systémů (extrémní případ), hodnota 0 pak znamená, že všechny systémy jsou provozovány zcela samostatně (žádná integrace). Konkrétní hodnotu parametru k lze zjistit při analýze stávajícího stavu integrace, kdy je potřeba zmapovat veškerá rozhraní a výměnu dat mezi jednotlivými informačními systémy a aplikacemi v rámci organizace.

$$k = \frac{2 \cdot I_{PIP}}{n(n-1)}. \quad (3)$$

Proměnná I_{PIP} zde představuje skutečný počet identifikovaných rozhraní mezi n systémy. Empiricky zjištěné hodnoty parametru k se zhruba pohybují v intervalu:

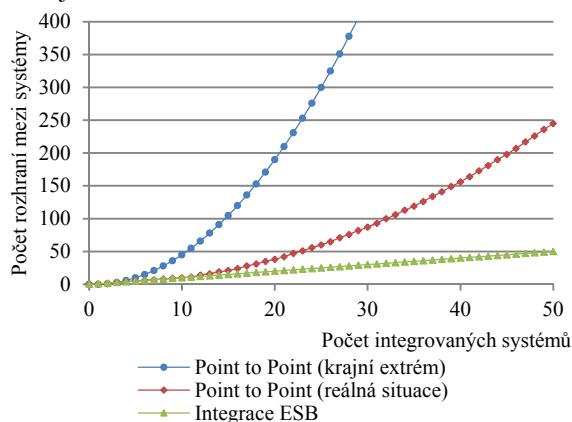
$$k \in \langle 0,1; 0,2 \rangle.$$

Tyto hodnoty byly zjištěné na základě několika realizovaných projektů systémové integrace v rámci analýzy současného stavu IS/ICT u několika větších společností v ČR. Konkrétní hodnota tohoto parametru závisí na konkrétní organizaci a může se značně lišit.

Celkový počet rozhraní při integraci prostřednictvím ESB je:

$$I_{ESB} = n, \text{ kde } n \geq 1. \quad (4)$$

Hodnota I_{ESB} má lineární charakter a je přímo úměrná počtu integrovaných systémů n . Každý systém má v tomto případě vytvořené pouze jedno rozhraní s integrační platformou namísto několika přímých rozhraní s ostatními systémy či aplikacemi. Srovnání počtu rozhraní při různých způsobech integrace ukazuje obrázek 13.



Obrázek 13 Počet rozhraní mezi systémy při různých způsobech integrace

5.3 Předpoklady pro realizaci pomocí ESB

Základním předpokladem integrace prostřednictvím ESB je, aby počet rozhraní mezi integrační platformou a jednotlivými systémy nebyl vyšší, než když bychom realizovali integraci stylem Point to Point:

$$I_{PIP} \geq I_{ESB}. \quad (5)$$

V opačném případě by se jednalo o natolik malou míru integrace jednotlivých systémů, kdy nemá smysl zvažovat náhradu přímých rozhraní (kterých je velmi málo) řešením, kdy každý systém bude mít vytvořené jedno rozhraní s integrační platformou. Aby tedy integrace prostřednictvím ESB měla opodstatnění (neuvažujeme-li jiné technologické aspekty), musí platit vztah:

$$k \frac{n(n-1)}{2} > n, \text{ kde } n \geq 1. \quad (6)$$

Po jednoduché úpravě:

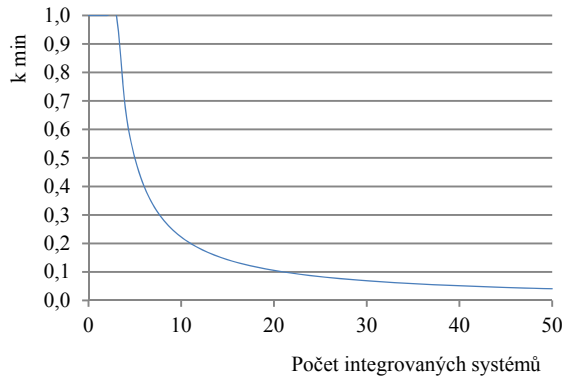
$$k > \frac{2}{(n-1)}. \quad (7)$$

Minimální hodnota parametru k , pro který má smysl zvažovat zavedení integrační platformy a integraci prostřednictvím ESB, je tedy:

$$k_{\min} = \frac{2}{(n-1)}. \quad (8)$$

Graficky ukazuje závislost hodnoty $k_{\min}$ na počtu integrovaných systémů n obrázek 14.

Z průběhu uvedeného obrázku 14 je patrné, že integrace prostřednictvím ESB nebo jiných obdobných technologií je vhodná spíše při větším počtu integrovaných systémů, resp. není přínosná – pokud neuvažujeme i jiné aspekty – pro velmi malý počet aplikací.



Obrázek 14 Minimální hodnota parametru k pro integraci prostřednictvím ESB

5.4 Náklady na systémovou integraci

Náklady na systémovou integraci jsou ovlivněny několika parametry, zásadní vliv však má především počet integrovaných systémů, který spolu s mírou provázanosti determinuje počet implementovaných rozhraní a tím i celkové náklady na realizaci.

Celkové náklady

Celkové náklady integrace stylem Point to Point, přičemž uvažujeme reálnou situaci, nikoliv extrémní případ:

$$TC_{PIP} = k \frac{n(n-1)}{2} C_I = \frac{k \cdot C_I}{2} n^2 - \frac{k \cdot C_I}{2} n. \quad (9)$$

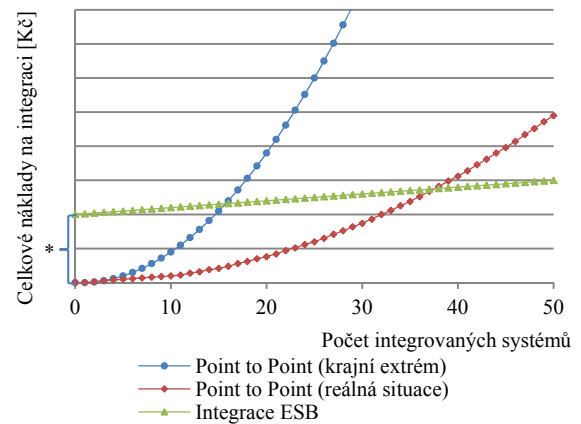
Celkové náklady integrace prostřednictvím ESB:

$$TC_{ESB} = FC_{IP} + VC_{IP} = FC_{IP} + n \cdot C_I. \quad (10)$$

V obou dvou případech představuje parametr C_I náklady na realizaci jednoho rozhraní. V tomto případě se jedná o teoretický model, konkrétní (reálné) náklady se mohou u různých rozhraní lišit, tento parametr je nutné chápat spíše jako průměrnou hodnotu. Z pohledu modelu se zabýváme spíše charakterem a závislostí jednotlivých proměnných, nikoliv jejich konkrétními hodnotami.

Parametr FC_{IP} představuje náklady na pořízení a zavedení integrační platformy do struktury podnikového informačního systému (tzn. zakoupení a implementaci potřebných technologií apod.). Obvykle se jedná o jednorázovou počáteční investici, která má fixní charakter, tj. není závislá na počtu následně

integrovaných systémů. Naopak variabilní náklady VC_{IP} jsou přímo úměrné počtu integrovaných systémů. Výši celkových nákladů při různých způsobech integrace graficky znázorňuje obrázek 15.



Obrázek 15 Celkové náklady realizace při různých způsobech integrace

*Náklady na zavedení IP (ESB)

Mezní náklady

Mezní náklady udávají, jak se změní celkové náklady, pokud se zvýší počet integrovaných systémů o jeden.

Mezní náklady integrace stylem Point to Point:

$$MC_{PIP} = TC'_{PIP} = k \cdot C_I \cdot n - \frac{k \cdot C_I}{2} = k \cdot C_I \left(n - \frac{1}{2} \right). \quad (11)$$

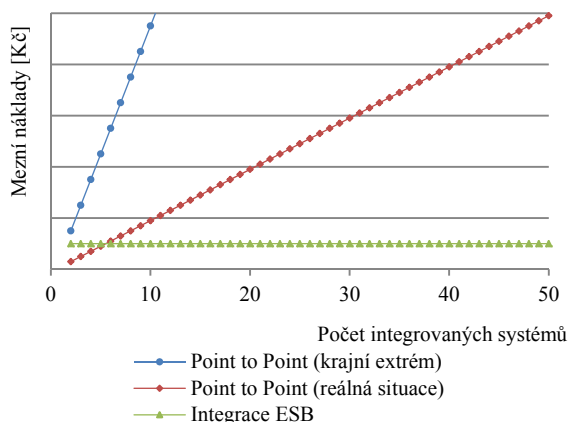
Mezní náklady integrace prostřednictvím ESB:

$$MC_{ESB} = TC'_{ESB} = C_I. \quad (12)$$

Celkové náklady při implementaci ESB

Při rozhodování, zda zachovat současný stav a pokračovat v integraci stylem Point to Point nebo zda přejít na jinou architekturu integrace, nejčastěji ESB nebo obdobnou, která umožňuje snižovat počet rozhraní mezi jednotlivými systémy, je nutno zvažovat několik faktorů. S přibývajícím počtem nově integrovaných systémů mohou náklady na integraci stylem Point to Point značně narůstat. Naopak integrace prostřednictvím ESB vyžaduje určité fixní náklady na zavedení integrační platformy a současně také určité variabilní náklady na modifikaci všech nebo části stávajících rozhraní. Prostřednictvím ESB je však možné mnohem snadněji a efektivněji integrovat nové informační systémy a aplikace při výrazně nižších mezních nákladech oproti stylu Point to Point. Závislost mezních nákladů na počtu integrovaných systémů ukazuje obrázek 16.

V tomto ohledu je klíčová analýza nákladů obou alternativ. Při zavedení integrační platformy a ESB je nutné zvažovat celkové náklady na integraci nově



Obrázek 16 Mezní náklady při různých způsobech integrace

plánovaných systémů, ale také na úpravu stávajících rozhraní pro potřeby ESB. Současně je ale nutné zvažovat úspory, které vzniknou tím, že integrace nových systémů prostřednictvím ESB je méně nákladná než pomocí stylu Point to Point. Tyto úspory nám tak snižují celkové náklady na integraci. Celkové náklady při implementaci ESB při započtení úspor vzniklých se změnou architektury integrace lze obecně vyjádřit vztahem:

$$TC_{IP} = FC_{IP} + \int_1^{n+n_{New}} MC_{ESB} dn - \int_n^{n+n_{New}} MC_{PiP} dn, \quad (13)$$

$$TC_{IP} = TC_{ESB}(n+n_{New}) - [TC_{PiP}(n+n_{New}) - TC_{PiP}(n)] \quad (14)$$

Po úpravě:

$$TC_{IP} = FC_{IP} + (n+n_{New})C_I - \frac{k \cdot C_I}{2} (n_{New}^2 + 2 \cdot n \cdot n_{New} - n_{New}), \quad (15)$$

Celkové náklady TC_{IP} zde zahrnují i úspory vzniklé změnou architektury integrace. Jedná se o funkci dvou proměnných a několika dalších parametrů, kde n představuje počet stávajících, již integrovaných systémů (prozatím stylem Point to Point), a n_{New} počet plánovaných nových systémů, které bude nutné rovněž integrovat (nově již prostřednictvím integrační platformy a ESB). Celkové náklady na zavedení integrační platformy se skládají:

- z fixních nákladů na pořízení integrační platformy (FC_{IP}),
- variabilních nákladů na integraci jak starých, tak i nově plánovaných systémů prostřednictvím ESB ($n+n_{New}$ systémů),
- úspor (se záporným znaménkem) vzniklých při nižších mezních nákladech u integrace nových systémů prostřednictvím ESB (n_{New} systémů).

Při rozhodování o způsobu a architektuře integrace systémů, resp. při rozhodování, zda přejít ze stylu Point to Point k zavedení jednotné integrační

platformy a integraci prostřednictvím ESB, hledáme bod (resp. body), ve kterém jsou náklady na změnu stejné jako úspory vzniklé přechodem na architekturu ESB. Kritériem pro rozhodování je pak porovnání výše nákladů a úspor. Možné situace popisují následující vztahy (16), (17) a (18).

Vyrovnané náklady a úspory:

$$TC_{IP}(n, n_{New}, k, FC_{IP}, C_I) = 0. \quad (16)$$

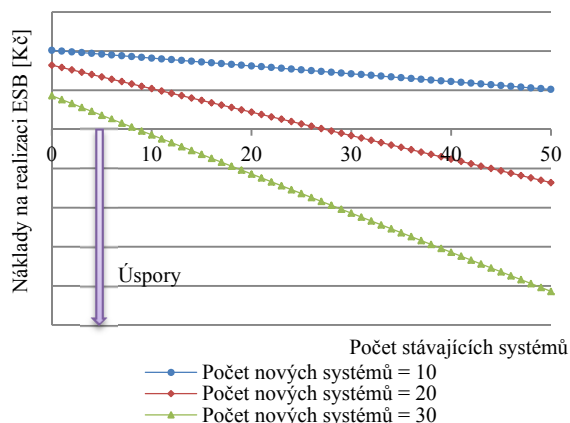
Náklady na zavedení IP jsou vyšší než úspory:

$$TC_{IP}(n, n_{New}, k, FC_{IP}, C_I) > 0. \quad (17)$$

Náklady na zavedení IP jsou nižší než úspory – požadovaný stav:

$$TC_{IP}(n, n_{New}, k, FC_{IP}, C_I) < 0. \quad (18)$$

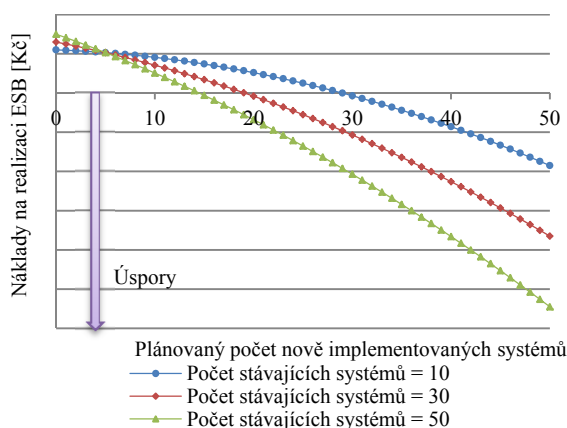
Obrázek 17 znázorňuje náklady na realizaci ESB při přechodu z architektury Point to Point v závislosti na počtu stávajících systémů při konstantním plánovaném počtu nově integrovaných systémů (v tomto případě 10, 20 a 30). Hodnoty nižší než 0 představují požadovaný stav, kdy úspory realizované změnou architektury převyšují náklady na realizaci a zavedení integrační platformy a ESB. Z uvedeného obrázku 17 je patrné, že vyšších úspor dosáhneme při větším počtu nových systémů (posun křivky směrem dolů) a současně při větším počtu stávajících systémů (posun po křivce doprava dolů).



Obrázek 17 Náklady na realizaci ESB v závislosti na počtu stávajících systémů

Obrázek 18 znázorňuje náklady na realizaci v závislosti na plánovaném počtu nově integrovaných systémů při konstantním počtu stávajících systémů (v tomto případě 10, 30 a 50) – tento model či pohled se nejvíce přibližuje reálným situacím při rozhodování o změně architektury integrace. Z uvedeného obrázku 18 je patrné, že vyšších úspor dosáhneme při větším počtu stávajících systémů (posun křivky směrem dolů) a současně při větším počtu nových systémů (posun po křivce doprava dolů). V tomto případě však má křivka exponenciální charakter, tj. při rostoucím počtu

nově plánovaných systémů jsou realizované úspory výraznější.



Obrázek 18 Náklady na realizaci ESB v závislosti na počtu nových systémů

6. Závěr

Při výběru a implementaci konkrétního řešení integrace je nutné zvažovat mnoho aspektů (Pour a kol., 2009), nicméně jedním z klíčových faktorů bude téměř vždy minimalizace nákladů jak na integraci nových systémů, tak i na provoz stávajících a nových rozhraní mezi systémy. Obecně by mělo být vždy cílem směřovat spíše k vytváření volných vazeb mezi jednotlivými systémy namísto mnohdy existujících těsných vazeb, které obvykle přinášejí řadu komplikací. Mezi další faktory při výběru konkrétního řešení pak bude mimo jiné patřit i otázka vlivu na chování a spolehlivost podnikového informačního systému jako celku při výpadku či nedostupnosti části integrovaného řešení, kterým může být dílčí informační systém, aplikace nebo prvek integrační infrastruktury. Celkově má systémová integrace a výsledné řešení značný dopad na celou organizaci a svým rozsahem ovlivňuje velké množství uživatelů.

Prezentovaný ekonomický model představuje jednoduchý popis problematiky systémové integrace, kdy zvažujeme dvě alternativy – integraci stylem Point to Point a zavedení integrační platformy a přechod k architektuře ESB. Celkové náklady na integraci oběma styly definují rovnice (9) a (10). Při zavedení integrační platformy a ESB je nutné zvažovat celkové náklady na integraci nově plánovaných systémů, náklady na úpravu stávajících rozhraní a úspory, které vzniknou v důsledku nižších nákladů na integraci nových systémů. Celkové náklady na implementaci ESB při započtení úspor vzniklých se změnou architektury integrace vyjadřují rovnice (13), (14) a (15). Pokud jsou tyto náklady nižší než 0, pak nově zavedená architektura integrace představuje dosažení úspor oproti zachování

stávajícího způsobu integrace stylem Point to Point. Naopak pokud jsou tyto náklady vyšší než 0, pak zavedení integrační platformy a přechod k architektuře ESB nepředstavuje pro organizaci přínos v podobě požadovaných úspor. Celkové náklady (úspory) na zavedení integrační platformy představují funkci několika proměnných. Zejména se jedná o počet stávajících, již integrovaných systémů a plánovaný počet nově integrovaných systémů. Výši nákladů (úspor) v závislosti na těchto dvou proměnných ukazuje obrázek 17 a obrázek 18. Vyšších úspor lze dosáhnout především při vyšším počtu nově integrovaných systémů. Naopak zavedení ESB či jiné obdobné architektury není přínosem při menším počtu systémů, jak stávajících, tak i nově integrovaných.

Uvedený model nezahrnuje a neřeší některé další aspekty. Největším nedostatkem je skutečnost, že model řeší pouze nákladovou stránku dané problematiky, nikoliv příjmovou stránku. Změna či zavedení integrační architektury jakou je například ESB, může kromě úspor při integraci nově dodávaných systémů mít rovněž pozitivní dopady na fungování IS/ICT a tím i na organizaci jako celku v podobě větší efektivity, lepší dostupnosti poskytovaných služeb apod. Tyto skutečnosti pak mohou mít dopad na celkově nižší náklady anebo vyšší příjmy podniku. Dalším nedostatkem je zjednodušený pohled na náklady spojené s realizací jednoho rozhraní – tento parametr se může značně lišit dle použitých technologií. Rovněž implementace nových rozhraní při integraci nových systémů může představovat jiné náklady než úprava stávajících rozhraní. Nižší počet rozhraní v architektuře ESB současně představuje nižší náklady na údržbu podnikového informačního systému jako celku, které budou mít v budoucnu dopad v podobě nižších nákladů na provoz IS/ICT. Tyto úspory však v popisovaném modelu nejsou zahrnuty.

Literatura

- ANDERSSON, J., JOHNSON, P. (2001). Architectural integration styles for large-scale enterprise software systems. *Enterprise Distributed Object Computing Conference, EDOC '01*. Seattle.
- HASSELBRING, W. (2000). Information system integration. *Communications of the ACM archive* 6 (43): 32–38.
- HOHPE, G., WOOLF, B. (2003). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Boston: Addison-Wesley.

ONDRUŠKA, M. (2009). Architektura informačního systému z pohledu integrace v kontextu podnikové integrace. In *Systémová integrace*. Praha: Česká společnost pro systémovou integraci, 3 (16): 143–158.

POUR, J., GÁLA, L., ŠEDIVÁ, Z. (2009). *Podniková informatika*. Praha: Grada.

SCHMIDT, M. –T., HUTCHINSON, B., LAMBROS, P., PHIPPEN, R. (2005). The Enterprise Service Bus: Making service-oriented architecture real. *IBM Systems Journal* 4 (44): 781–797.

<http://dx.doi.org/10.1147/sj.444.0781>

SUTHERLAND, J., HEUVEL, W.–J. (2002). Enterprise application integration and complex adaptive systems. In *Communications of the ACM* 10 (45): 59–64.

Další zdroje

CHANDRA, D., LIU, A., ROXBURGH, U., MASON, A., NADHAN, E.G., SLATER, P. (2003). *Guidelines for Application Integration*. Microsoft Patterns & Practices, URL: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag/html/eappint.asp>

TROWBRIDGE, D., ROXBURGH, U., HOHPE, G., MANOLESCU, D., NADHAN, E.G. (2004). *Integration Patterns*. Microsoft Patterns & Practices, URL: <http://msdn.microsoft.com/en-us/library/ms978729.aspx>