



národní  
úložiště  
šedé  
literatury

## **Objektivizace charakteristik integrace infomačních systémů ve zdravotnictví**

Krsička, Daniel  
2011

Dostupný z <http://www.nusl.cz/ntk/nusl-55976>

Dílo je chráněno podle autorského zákona č. 121/2000 Sb.

Tento dokument byl stažen z Národního úložiště šedé literatury (NUŠL).

Datum stažení: 03.05.2024

Další dokumenty můžete najít prostřednictvím vyhledávacího rozhraní [nusl.cz](http://nusl.cz) .

# Objektivizace charakteristik integrace informačních systémů ve zdravotnictví

doktorand:

ING. DANIEL KRŠIČKA

Ústav informatiky AV ČR, v. v. i.  
Pod Vodárenskou věží 2  
182 07 Praha 8

dkrsicka@gmail.com

školitel:

ING. MILAN ŠÁREK, CSC.

Ústav informatiky AV ČR, v. v. i.  
Pod Vodárenskou věží 2  
182 07 Praha 8

sarek@euromise.cz

obor studia:  
Biomedicínská informatika

## Abstrakt

Cílem výzkumné práce je příprava deterministických optimalizačních metod pro řízení integrační architektury informačních systémů (IS) ve zdravotnictví. Přípravovaná metodika má poskytnout aparát ke strukturovanému vyhodnocení a porovnání dílčích návrhů integrace IS. Očekávaným přínosem je snížení celkových nákladů (*TCO - Total Cost of Ownership*) a zvýšení flexibility prostředí (*TTM - Time to Market*). Při aplikaci a kombinaci vzorů a osvědčených řešení musí architekt vždy vzít v úvahu informace o prostředí, ve kterém se integrace buduje. Právě objektivizace hodnocení prostředí by měla úkol zjednodušit a její příklad je předmětem tohoto článku.

## 1. Komunikace

Pro účely této práce definujeme 2 základní typy komunikace, které budou dále hlouběji strukturovány:

- *lokální komunikace* je definována jako řízená výměna dat mezi 2 programy spuštěnými na stejném HW.
- *síťová komunikace* je definována jako řízená výměna dat mezi 2 programy běžícími na různých strojích propojených pomocí počítačové sítě.

Motivací k prezentovanému výzkumu je i fakt, že lokální komunikace se v mnohém liší od komunikace přes síť. Toto prosté tvrzení má poměrně komplexní množinu příčin a důsledků. Jak bude ukázáno, zanedbáváním uvedeného tvrzení a jeho důsledků dochází k vytváření nevhodných integračních řešení. Některé technické standardy pro integraci přes síť byly vyvinuty evolucionálně ze standardů pro synchronní lokální komunikaci. Při extrapolaci těchto standardů do oblasti integrace přes síť

Lze téměř jistě říct, že návrh distribuovaného systému stejným způsobem jako lokálního bude mít katastrofální následky.

Přehled základních rozdílů mezi lokální a síťovou komunikací je uveden v tabulce 1. Nyní se podívejme na jednotlivé aspekty podrobněji:

### *Spolehlivost komunikační infrastruktury*

Budeme-li uvažovat o spolehlivosti, měli bychom vzít v úvahu parametry jako stav HW, množství konkurence procesů v daném prostředí, topografickou vzdálenost mezi komunikujícími procesy apod. V lokálním prostředí se nejčastěji pohybujeme na úrovni jednoho operačního systému (OS) nebo middleware, případně na úrovni společného HW, na němž jsou pomocí virtualizační technologie definovány virtuální stroje. Zásadním faktem je existence společného HW. Spolehlivost HW je zde téměř binární (stroj buď běží nebo ne, případné chyby se sdílí), na rozdíl od síťového spojení, kde za prvé procesy běží odděleně a za druhé na komunikaci se podílí řada dalších aktivních prvků. Detekce chybových stavů a alerting je řádově jednodušší lokálně než v síti. Také množství konkurenčně běžících procesů je typicky menší než při komunikaci přes obecně sdílenou síťovou infrastrukturu.

### *Rychlost komunikace*

V lokálním prostředí také prakticky zanedbáváme vzdálenost komunikujících procesů, protože sdílí stejný HW a jejich komunikace je omezena šířkou a taktem sběrnice. U dnešního hardware se pohybujeme v řádech desítek a stovek Gbps. Naopak při síťové komunikaci se v reálných hodnotách pohybujeme o řády níže, nejčastěji ve stovkách kbps, maximálně v jednotkách Mbps. Mluvíme zde o reálné komunikaci mezi procesy

na aplikační vrstvě (tzv. End-to-End), nikoli o rychlosti přenosových sítí, které mohou být mnohem vyšší.

#### *Technologická diverzita*

V návrzích integrace mezi systémy je potřeba počítat s rozdílností operačních systémů, použitých programovacích jazyků, middleware platform, kódování nebo obecně formátů dat. U síťových řešení je třeba se na všechny uvedené potenciální problémy připravit. Dnes existuje řada platformově nezávislých standardů založených na jazyce XML [10], jejich využití však má svá omezení. Jednak nejsou implementovány ve všech technologiích a dále pak nemusí být použitelné ve všech případech, především z výkonnostních důvodů.

#### *Rozdílný režim správy*

Se týká především organizačního zajištění integrovaných systémů a integrační technologie (včetně sítě). Je zřejmé, že 2 libovolné IS integrované přes síť mohou patřit a často patří rozdílným společnostem. Různé společnosti se vyznačují různou mírou penetrace ICT do jejich prostředí, včetně rozdílného množství organizačních procesů, které IT podporuje. Tato míra spolu s velikostí společnosti indikuje 2 druhy rozdílu v IT.

Za prvé je to využití odlišných technologií. Např. samostatně pracující praktický lékař si jistě nebude kupovat 64-jádrový server vyžadující 2kW chlazení a optický spoj s WDM. Naopak velká fakultní nemocnice si z nemůže dovolit budovat svou serverovou infrastrukturu na stolních PC, ani pro přístup do sítě používat ASDL nebo 802.11g spojení. Rozdílná úroveň používaných technologií pak implikuje i odlišné možnosti v propojení společnosti s externími subjekty. Tak vzniká riziko, že se nepodaří nalézt způsob vyhovující oběma stranám.

Za druhé existuje odlišnost v režimu správy resp. personálním zajištění. Větší společnosti mají obecně vyšší dostupnost své ICT infrastruktury a také mohou vykazovat lepší kvalitu svých ICT služeb. Naopak flexibilita při změnách bývá u větších společností výrazně menší.

Tato práce není zaměřena na organizační zajištění provozu a rozvoje integrovaných systémů. Více informací o problematice řízená architektury lze nalézt např. v [2].

## **2. Způsoby integrace**

Dnes rozlišujeme 4 základní koncepty integrace [1], které se s různou úspěšností vyrovnávají s výše uvedenými aspekty integrace:

#### *Batch File Transfer (BFT)*

Dávková výměna souborů je nejjednodušším způsobem komunikace dat. Zdrojový systém vytvoří soubor obsahující řídicí příkazy resp. data a uloží ho na persistentní úložiště (např. diskové pole). Soubor je interaktivně nebo automaticky přenesen k cílovému systému, kde ho cílový systém načte. Samozřejmě lze realizovat jedno- i obousměrnou komunikaci. Kromě obsahu dat je nutné dohodnout a respektovat řadu obslužných parametrů jako jména souborů, čas výměny, umístění, mazání souborů i mechanismy obsluhy chybových stavů.

#### *Shared Database (SDB)*

Reprezentuje archetyp dvou a více systémů sdílejících v reálném čase jedno datové úložiště. V praxi se jedná nejčastěji o společnou databázi (DB), ale stejně tak lze za příslušných podmínek využít paměť nebo diskový prostor. Nemusí se jednat o integraci s výměnou dat, protože integrované systémy sdílí fyzicky jedno úložiště.

#### *Remote Procedure Call (RPC)*

Představuje model, ve kterém jeden systém nabízí určitou svou funkci přes síťové rozhraní a jiný systém ji volá. Vznikl evolucí lokálního volání procedur / funkcí v rámci jednoho stroje resp. systému na volání v rámci počítačové sítě. Extrapoluje koncept synchronní blokující operace z prostředí jednoho počítače na infrastrukturu separátních strojů propojených v síti a přináší problémy uvedené v úvodu tohoto článku. Jedná se o synchronní blokující komunikaci v reálném čase.

#### *Messaging (MS)*

Využívá dedikovaného software k doručování zpráv. Odesílatel předává zprávu MS a sám může pokračovat ve své další činnosti. MS je zodpovědný za doručení zprávy. Tato komunikace je asynchronní a neblokující. Koncept asynchronní komunikace vznikl právě v reakci na nemožnost přistupovat k systémům propojeným sítí stejně jako k systémům lokálním. Bohužel, dodnes není na mnoha místech všeobecně přijat do praxe. Při nárůstu objemu a počtu komunikací nad udržitelnou mez není již vhodné programovat messaging ručně. Dnes se k uvedenému účelu používá téměř výhradně některý z produktů kategorie Message Oriented Middleware (MOM) [1]. MOM pak tvoří transportní základ v modelu Enterprise Service Bus (ESB) [5]. Rozbor ESB je mimo rozsah možností tohoto článku.

Uvedené 4 způsoby nelze brát jako různé evoluční úrovně. Mezi zmíněnými kategoriemi neexistuje žádné uspořádání, které by vypovídalo o kvalitě toho kterého

způsobu samo o sobě. Základní určení způsobu integrace (ještě bez vazby na jakékoli další vzory či ověřené postupy) musí být dáno vždy aktuálními podmínkami tj. jednak vyspělostí ICT infrastruktury integrovaných společností a dále především požadavky na konkrétní výměnu dat. Generalizace na jeden způsob je vhodná jen od určité úrovně komplexnosti a nikdy nemůže být úplná (tj. dogmaticky uplatňovaná).

### 2.1. Časový rozměr komunikace

Zásadní informací u výše uvedených základních způsobu integrace je určení, zda je komunikace synchronní nebo asynchronní (viz tabulku 2). Magnam partem se zajímáme o integraci systémů přes počítačovou síť. Dokonalé rozdělení není možné, protože komunikace probíhá na více vrstvách. Některé používají blokující operace a jsou tedy synchronní, některé ne. Rozdělení je tedy nutné stanovit dohodou vycházející z následujících předpokladů:

- Dělení na blokující a neblokující operace definujeme na aplikační vrstvě tj. na vrstvě volajícího a přijímajícího procesu<sup>1</sup>.
- Předpokládáme, že subsystémy realizující nižší vrstvy umožňují minimálně kvazi-paralelní zpracování více požadavků. Tzn., že žádný aplikační proces není na významně dlouhou dobu odstaven od prostředků.

Všechny uvedené způsoby lze využít k získávání informací on-demand (request/response) i k proaktivní publikaci informací (one-way). Způsob integrace tedy neimplikuje směr komunikace ani typ vyměňovaných informací.

### 3. Vzdálenost integrovaných systémů

Abychom mohli v analýze integračního scénáře zohlednit topografické rozdělení jednotlivých IS, je nezbytné zavést pojem vzdálenosti mezi integrovanými systémy. Vzdálenost musí být ordinální veličina s ostrým úplným uspořádáním, tj. na oboru hodnot musí být definována binární antireflexní, tranzitivní, antisymetrická relace uspořádání [16]. Pro objektivní hodnocení je také třeba definovat všechny hodnoty tak, aby bylo možné jednotlivé scénáře mezi sebou porovnávat. Pro účely této práce navrhuje následující kategorie  $A - F$ :

$A$  Komunikace mezi procesy jednoho OS pomocí sdílené paměti (shared memory).

$B$  Komunikace mezi procesy jednoho OS pomocí lokálního síťového rozhraní (loopback) resp. mezi 2 virtuálními stroji na stejném HW (virtual network).

$C$  Komunikace v LAN/SAN na přepínané síti včetně L3+ switchingu tj. řadíme sem komunikaci sítí, ve které jsou pouze aktivní prvky s výpočtem nad asociativní paměti (CAM).

$D$  Komunikace v LAN/SAN na směrované síti tj. přes aktivní prvky pracující s CPU výpočtem. Pro zařazení do této kategorie je významná inspekce paketů tj. řadíme sem i firewally a IPS systémy kontrolující hlavičky protokolů vyšších vrstev.

$E$  Komunikace dedikovanou (pronajatou) WAN – nad rámec uvedeného v kategorii  $D$  přibývá transportní zpoždění na síti, omezení pásma a prodleva konverze protokolů.

$F$  Komunikace přes Internet tj. WAN spojený bez záruky dostupnosti a bez možnosti aplikace QoS.

Každá z uvedených skupin je charakterizována odlišným přenosovým pásmem, dostupným pro komunikující procesy. Například komunikace 2 aplikací přes sdílenou operační paměť jistě poběží rychleji než při komunikaci v rámci počítačové sítě společnosti, nebo dokonce při volání Internetem třeba z České Republiky do Austrálie.

Výše uvedené kategorie lze hrubě charakterizovat minimálně 3 atributy: dostupným přenosovým pásmem  $f$ , transportním zpožděním na síti  $t$  a mírou vyjadřující počet paralelních/kolidujících přenosů na síti  $a$ . Na úrovni představovaného modelu neuvažujeme o kolizních přenosech ve smyslu sdílení jednoho lokálního síťového pásma, ale jako o pravděpodobnosti, že v daném časovém úseku nebude přenos dat nijak výrazně narušen využitím stejné přenosové infrastruktury jiným komunikačním procesem. Infrastrukturu je zde celá komunikační cesta ve všech svých vrstvách a v celé své délce. Ze stejného důvodu nelze definovat transportní zpoždění jako veličinu nepřímo úměrnou dostupnému pásmu, protože, především u komunikací s „delší cestou“ tj. s přenosem přes rozsáhlé sítě, je linearita relace mezi pásmem a latencí narušena využitím řady aktivních prvků.

Přenosové pásmo  $f$  lze vyjádřit v různých jednotkách, pro naše účely volíme Hz a zanedbáváme

<sup>1</sup>Je třeba dodat, že je možné a často vhodné realizovat synchronní scénář typu požadavek/odpověď pomocí MOM. Messaging implikuje asynchronní mechanismy na vrstvách nižších než je aplikační

tak rozdílná přenosová kódování. Naopak níže uvedené hodnoty frekvencí jsou vždy vztaženy k šířce dostupné sběrnice (např. násobení základní frekvence bitovou šířkou sběrnice u lokální komunikace apod.). Transportní zpoždění  $t$  je druhým parametrem, vyjádřeným v sekundách. Představuje latenci mezi odesláním zprávy odesílatelem (aplikací/systémem) a jejím přijetím na aplikační vrstvě u příjemce. Jedná se tedy o čas zpoždění mezi 2 procesy. Posledním parametrem je míra dostupnosti přenosového pásma  $a$ , který souvisí se spolehlivostí komunikační infrastruktury diskutované v úvodu tohoto článku.

### 3.1. Výpočet vzdálenosti

Shora uvedeným veličinám jsme přiřadili hodnoty. Řádové pásmo  $f$  a zpoždění  $t$  bylo stanoveno na základě parametrů dnes používaných technologií. Míru dostupnosti  $a$  stanovíme dohodou jako reálné číslo z intervalu  $\langle 0; 1 \rangle$ , kde hodnota 1 vyjadřuje 100%-ní rezervaci pásma. Vzdálenost  $d$  mezi integrovanými systémy lze tedy vypočítat jako 1:

$$d = \log \left[ \frac{\frac{1}{f} + t}{a} \right] \quad (1)$$

Pro výpočet použijeme převrácenou hodnotu  $f$  tak, abychom s narůstající šířkou přenosového pásma klesala hodnota vzdálenosti. Naopak transportní zpoždění  $t$  připočítáváme v lineárním smyslu. Mírou dostupnosti  $a$  výsledný výraz dělíme kvůli oboru hodnot, který je pro  $a$  definován  $\langle 0; 1 \rangle$ .

Po provedení výpočtů pro skutečné hodnoty frekvencí a latence výrazu  $\frac{1}{f} + t$  získáváme hodnoty, které se v krajních případech vzájemně liší o 8 řádů. Další počítání s takovými hodnotami je nepraktické, proto je vhodné provést transformaci pomocí logaritmu.<sup>2</sup>

### 3.2. Normalizace vzdálenosti

Pro praktické užití by bylo vhodné reálná čísla, která jsou výsledkem výpočtu vzdálenosti, transformovat na určitou normalizovanou stupnici. Provedeme normalizaci vzdálenosti do intervalu  $\langle 0; 1 \rangle$ . Pro provedení normalizace musíme určit mezní hodnoty vypočtené vzdálenosti odpovídající teoretické hodnotě 0 resp. 1. Hodnoty mezního výpočtu jsou uvedeny v 3.

Výsledná normalizace se pak provede transpozicí do  $R^+ + \{0\}$  a jeho projekcí na interval  $\langle 0; 1 \rangle$ . Normali-

<sup>2</sup>Byla testována i varianta s logaritmem odmocniny výrazu, nicméně výsledky se vzájemně v řádech příliš neliší a je inhibována informace o proporcí mezi jednotlivými kategoriemi. Proporcionalita hodnot může být důležitá v uplatnění veličiny vzdálenosti, a proto byla zvolena varianta výpočtu bez použití odmocniny.

zace tedy obecně vypadá následovně 2:

$$d_{norm} = \frac{d + |d_{min}|}{|d_{min}| + |d_{max}|} \quad (2)$$

Pak všechny hodnoty  $d_{norm}$  pro vstupy určené kategoriemi A - F 3 padnou do intervalu  $\langle 0; 1 \rangle$  a pro mezní hodnoty platí  $d_{norm}^{min} = 0$  a  $d_{norm}^{max} = 1$ .

### 3.3. Určení hodnot pro různá prostředí

Následuje výpočet normalizované vzdálenosti pro jednotlivé kategorie vzdáleností. Protože ICT infrastruktura se liší subjekt od subjektu v závislosti na jeho velikosti, počtu procesů podporovaných IT, počtu uživatelů, množství spravovaných dat, počtu integrovaných partnerů, míře legislativní regulace, geografické lokaci a dalších parametrech. Proto jsme příklad výpočtu rozdělili na 3 samostatné jednotky. Každá jednotka charakterizuje společnost o specifické velikosti.

#### SOHO (Small Office Home Office)

Malé subjekty, s jednotkami až desítkami uživatelů. Příkladem mohou být ambulance praktických lékařů, lékárny apod. Předpokládá se využití low-end zařízení pro síťovou komunikaci, plochá struktura místní sítě, běžné širokopásmové připojení do Internetu. Neočekáváme pronájem WAN spojů, ani dedikované serverové infrastruktury se speciálním serverovým HW. Ohodnocení vstupních veličin i vypočtené normované vzdálenosti lze nalézt v tabulce 4.

#### Mainstream

Společnosti střední velikosti s desítkami až stovkami zaměstnanců. Může se jednat o místní nemocnice, polikliniky, drobné výzkumné ústavy, menší pojišťovny, zdravotnické registry. V mainstreamu předpokládáme centralizaci dedikovaných serverů do výpočetních sálů, hierarchizaci přepínaných sítí, možnou existenci pronajatých linek s partnerskými společnostmi. Je možný výskyt systémů IPS a aplikačních firewallů. Ohodnocení vstupních veličin i vypočtené normované vzdálenosti lze nalézt v tabulce 5.

#### Enterprise

Velké společnosti. Stovky až tisíce uživatelů – fakultní nemocnice, centrální registry, velké pojišťovny, orgány státní správy etc. Očekáváme high-end výpočetní prostředky, specializované SAN sítě, optické spoje, de-

dikované linky pro spojení s detašovanými lokalitami. Ohodnocení vstupních veličin i vypočtené normované vzdálenosti lze nalézt v tabulce 6.

### 3.4. Využití normované vzdálenosti

Normovaná vzdálenost závisí na deterministických vstupech (pásmo, latence, dohodnutá míra dostupnosti). Výpočet i normalizaci zachovává proporce mezi jednotlivými kategoriemi, a proto by mělo být možné využít nejen uspořádání kategorií, ale i přímo hodnoty vzdálenosti při dalších výpočtech. Díky stanovení intervalu hodnot (oboru hodnot funkce vzdálenosti) je možné vzdálenost využít i pro případy, kdy se integrují 2 společnosti různé úrovně vyspělosti. Konečná normalizace nemá vliv na výpočty se vzdáleností, pouze usnadňuje jejich provedení a zlepšuje čitelnost.

## 4. Vazby mezi integrovanými systémy

Druhou vlastností integračních řešení analyzovanou v tomto článku je vazba mezi koncovými komunikujícími aplikacemi. Zatímco vzdálenost je příkladem charakteristiky prostředí v němž se integruje, vazbu formují samotné integrované systémy. Interaplikační vazba může být popsána sadou vlastností / atributů. Charakteristika vazby je důležitá pro objektivizaci popisu integračního scénáře, protože různé systémy mohou navazovat diametrálně odlišné vazby a to nejen na základě způsobu jejich propojení nebo vzdálenosti (viz výše). Z hlediska vazby nás zajímá míra závislosti mezi integrovanými systémy. Právě míra závislosti může být vyjádřena množstvím a vlastnostmi vazeb mezi komunikujícími IS.

V softwarovém inženýrství se nejčastěji používá klasifikace označující volnou vazbu (Loose Coupling) a těsnou vazbu (Tight Coupling). Taková klasifikace je pro formální objektivizaci nedostatečná, a proto je nutné klasifikaci dále propracovat. Základ klasifikačního modelu lze převzít z metod pro optimalizaci návrhu programového kódu [3]. Problematika architektury integračních řešení vykazuje celou řadu společných znaků, a proto můžeme základ klasifikace vazby mezi komponentami komunikujícími přes počítačovou síť postavit na modelu určeného pro programování na jednom počítači. Tento model je však nezbytně nutné dále rozpracovat, abychom zabránili automatické extrapolaci vlastností lokální komunikace na komunikaci v síti, jak bylo uvedeno dříve [1]. Vyjdeme-li z existujícího modelu, můžeme definovat následující kategorie vazeb. Ke každé kategorii uvádíme pro srovnání vždy příklad lokálního kódu i integračního řešení:

### *Content Coupling (silná těsná vazba)*

Volaná komponenta nabízí svou funkcionalitu přímo, tj. volající iniciuje přímo výkonný kód volaného. Volající musí znát přesně strukturu, ve které volaný volání přijímá. *Příkladem je volání silně typované funkce v imperativním programovacím jazyce (např. v C) nebo volání funkce přes socket, tedy situace, kdy není využit žádný vyšší protokol nad TCP a zasílaná data přijímající program přímo interpretuje a to vždy stejně (data neobsahují řídicí znaky).*

### *Common Coupling (sdílení úložiště dat)*

Představuje archetyp, v kterém 2 a více systémů sdílejí stejná data. Lze uvažovat na lokální úrovni (paměť), i na úrovni sítě (společná DB). Určujícím faktorem je i nutnost znát přesně datový model a schéma řízení přístupu. Informace drží a implementuje každý integrující systém. *Příkladem může být lokální volání funkce a předání parametru odkazem (pointer), nebo vytvoření sdíleného segmentu paměti (mezi 2 procesy OS), nebo třeba sdílení jedné databáze dvěma a více aplikacemi.*

### *External Coupling (externalizovaná společná vazba)*

Předchozí typ lze upravit exportem informací o syntaxi na společné úložiště. Export obsahuje jak datové tak řídicí schéma. *Samostatně se dnes prakticky nepoužívá, ale je nedílnou součástí masivně rozšířených případů externalizace schémat webových služeb (XSD), WS-Standardů (policies etc.), nebo třeba kaskádových stylů webových aplikací (CSS). Lze však externalizovat i jiné informace, např. o místu přístupu k datům (tsnames.ora pro Oracle DB apod.). Je tedy třeba uvažovat o jednotlivých vrstvách ISO/OSI.*

### *Control Coupling (vazba s řízením)*

Kategorie, kde volající komponenta přikládá k datovému obsahu zprávy řídicí příznak (příkaz), tj. informuje volaného, jak s daty naložit. Volající tedy nemusí znát všechny funkce volaného a s touto množinou lze dynamicky pracovat. *Příkladem je jakákoli funkce, obsahující řídicí argument. Lze vytvořit na úrovni kódu programovacího jazyka, ale stejně tak se využívá i v moderních WS-Standardech – např. hlavička `<SoapAction... />`*

### *Stamp Coupling (volnost datového schématu)*

Je analogií předchozího případu, ale v oblasti dat. Volající nemusí nutně zaslat všechny datové atributy, ale jen některé a volající data dokáže zpracovat (pokud to sémantika případu dovoluje). *Webové služby založené na standardu SOAP [14] umožňují definovat povinné a*

*nepovinné parametry stejně jako dynamicky jejich multiplicitu v dané zprávě. Využití principu Stamp Coupling predisponuje bezpečnostní slabiny systému a celkově zvyšuje nároky na robustnost výkonného kódu.*

#### Message Coupling

Komunikace probíhá přes prostředníka, který uvolňuje vzájemné závislosti mezi komunikujícími systémy. Zde se nejedná o komunikaci messagingem tj. technologií pro asynchronní persistentní komunikaci, ale o prvek typu ESB [14]. Můžeme nalézt více úrovní vyspělosti vazby Message Coupling, jejich rozbor je však předmětem výzkumné práce a je mimo rozsah tohoto článku. *Mezi lokálními programy lze využít předání přes sdílenou paměť s intervencí OS (např. pipelines na \*nixu). V síťové integraci jsou příkladem jednak komerční produkty IBM WMQ, případně produkty s vyšší logikou jako IBM WESB, SAP-PI, MS Biztalk, BEA WebLogic ..., ale i ESB řešení určená pro oblast zdravotnictví [4].*

#### 4.1. Ordinální vyjádření vazby mezi IS

Stejně jako u vzdálenosti bude vhodné pro veličiny charakterizující vazbu mezi systémy definovat relaci ostrého uspořádání. Jeho úplnost je důležitým předmětem dalšího výzkumu. Předpokládáme ovšem, že bude nutné ustoupit od úplnosti a možná i od ostrosti uspořádání množiny definující kvalitu vazby. Tj. předpokládáme, že vazba mezi systémy nebude charakterizovatelná jedinou ordinální veličinou, ale minimálně dvěma. Důvodem je vzájemná kontradikce požadavků na volnost vazby a požadavků na výkonnost celého řešení. Cílem tedy bude nalezení optimálního vyvážení těchto důležitých metrik.

Zde uvádíme možné vyhodnocení míry závislosti mezi komunikujícími systémy pomocí množství vstupně/výstupních řídicích a datových parametrů. Základ číselné reprezentace míry závislosti (síly vazby) mezi systémy lze založit na [3].

$$c^{plain} = 1 - \frac{1}{D_i + 2 \cdot C_i + D_o + 2 \cdot C_o} \quad (3)$$

kde  $D_i$ ,  $C_i$  jsou počty datových resp. řídicích vstupních proměnných (parametrů volání) a  $D_o$ ,  $C_o$  jsou počty datových resp. řídicích výstupních proměnných (parametrů odpovědi).

Je zřejmé, že uvedenou číselnou reprezentaci lze velmi jednoduše ovlivnit změnou počtu datových i řídicích parametrů. Uvedený historický model určený pro vyhodnocení vazby mezi 2 lokálně běžícími programy tedy nelze použít v jeho originální podobě a musí být upraven. Shrňme nejzávažnější nedostatky:

- Pokud komunikující systémy neznají vzájemně svou vnitřní strukturu, mluvíme o volné vazbě. Teoreticky by bylo možné snížit počty datových parametrů na 1 resp. 2 (in/out) a řídicích na 0 a dosáhnout tak vazby  $c = 0,5$ . Budování takových rozhraní je však kontraproduktivní. Volná vazba znamená i možnost změn v rozhraních bez nutnosti změn mezilehlých a především protilehlých komponent. To nelze provést u nestrukturovaného rozhraní.
- V závislosti na míře vyspělosti je u menších řešení (SOHO/Mainstream) vhodné budovat integrace s přímým řízením, kde komunikující strany přímo ovládají zpracování dat a naopak u výše zmíněné kategorie Enterprise dedikovat logiku zpracování na ESB [5]. A i dále je možné řídicí atributy dělit na ty zpracované ESB a ty, které nesou informaci o sémantice dekódovatelné až konečným příjemcem.
- Řídicí informace mohou být poskytnuty v různé kvalitě v závislosti na míře jejich standardizace. U integračního řešení postaveného na obecně platných standardech [8] je zaručeně vyšší pravděpodobnost jeho opakovatelného využití a robustnosti v čase, než u těch budovaných na lokálních kódováních, číselnících, signalizaci apod. Uvedené tvrzení lze chápat na všech vrstvách od transportních protokolů, přes řídicí informace pro komunikující služby [13] až po standardizaci na aplikační úrovni [9].
- Komunikace mezi systémy může z hlediska aplikační logiky rozložena do více kroků. Nejtriviálnějším případem je komunikace požadavek/odpověď, další varianty pak znamenají již plnou statefull komunikaci s nutností udržování informací o relaci (session). Informace o počtu stavů musí být ve vyhodnocení volnosti vazby také zahrnuta.

Na základě uvedených informací upravíme algoritmus pro výpočet volnosti vazby o další aktivující i inhibující členy.

Především zavedeme *míru externalizace integračních funkcí* jako ordinální diskrétní veličinu vyjadřující mohutnost ESB. Definiční obor i navrhované hodnoty jsou uvedeny v tabulce 7. Prerekvizitou zařazení konkrétního prostředí k dané úrovni je splnění všech vlastností úrovní nižších, což někdy nemusí být automatické, především při využití orchestrace [6] procesů pomocí Business Process Engine [7].

| <i>aspekt</i>                           | <i>lokální komunikace</i> | <i>síťová komunikace</i> |
|---|---------------------------|--------------------------|
| spolehlivost komunikační infrastruktury | vysoká                    | nízká                    |
| rychlost komunikace                     | vysoká                    | nízká                    |
| technologická diverzita                 | nízká                     | vysoká                   |
| rozdílný režim správy                   | nízké riziko              | vysoké riziko            |

**Tabulka 1:** Aspekty síťové komunikace

|                             | <i>synchronní</i> | <i>asynchronní</i> |
|-----------------------------|-------------------|--------------------|
| komunikace v reálném čase   | SDB, RPC, MS      | MS                 |
| dávková komunikace off-line | -                 | BFT                |

**Tabulka 2:** Relace způsobu komunikace a jeho provádění v čase

| <i>mezí hodnota vzdálenosti</i> | <i>pásmo [Hz]</i> | <i>latence [s]</i> | <i>míra dostupnosti pásma</i> | <i>vzdálenost</i> |
|---------------------------------|-------------------|--------------------|-------------------------------|-------------------|
| $d^{min}$                       | $10^3$            | 10                 | $10^{-4}$                     | -8,995            |
| $d^{max}$                       | $10^{12}$         | $10^{-9}$          | 0,99                          | 5,000             |

**Tabulka 3:** Mezní hodnoty vzdálenosti

| <i>případ</i> | <i>pásmo [Hz]</i> | <i>latence [s]</i> | <i>míra dostupnosti pásma</i> | <i>vzdálenost</i> | <i>normovaná vzdálenost</i> |
|---------------|-------------------|--------------------|-------------------------------|-------------------|-----------------------------|
| A             | $10^{10}$         | $10^{-7}$          | 0,9                           | -6,954            | 0,146                       |
| B             | $10^9$            | $10^{-5}$          | 0,7                           | -4,845            | 0,297                       |
| C             | $10^8$            | $10^{-4}$          | 0,1                           | -3,000            | 0,428                       |
| D             | $10^7$            | $10^{-2}$          | 0,1                           | -1,000            | 0,571                       |
| E             | n/a               | n/a                | n/a                           | n/a               | n/a                         |
| F             | $10^6$            | 1                  | 0,0001                        | 4,000             | 0,929                       |

**Tabulka 4:** Vypočtené hodnoty normované vzdálenosti pro kategorii SOHO

| <i>případ</i> | <i>pásmo [Hz]</i> | <i>latence [s]</i> | <i>míra dostupnosti pásma</i> | <i>vzdálenost</i> | <i>normovaná vzdálenost</i> |
|---------------|-------------------|--------------------|-------------------------------|-------------------|-----------------------------|
| A             | $10^{10}$         | $10^{-7}$          | 0,9                           | -6,954            | 0,146                       |
| B             | $10^9$            | $10^{-5}$          | 0,7                           | -4,845            | 0,297                       |
| C             | $10^9$            | $10^{-4}$          | 0,1                           | -3,000            | 0,428                       |
| D             | $10^8$            | $10^{-2}$          | 0,01                          | 0,000             | 0,643                       |
| E             | $10^6$            | $10^{-1}$          | 0,01                          | 1,000             | 0,714                       |
| F             | $10^6$            | 1                  | 0,0001                        | 4,000             | 0,929                       |

**Tabulka 5:** Vypočtené hodnoty normované vzdálenosti pro kategorii Mainstream

| <i>případ</i> | <i>pásmo [Hz]</i> | <i>latence [s]</i> | <i>míra dostupnosti pásma</i> | <i>vzdálenost</i> | <i>normovaná vzdálenost</i> |
|---------------|-------------------|--------------------|-------------------------------|-------------------|-----------------------------|
| A             | $10^{11}$         | $10^{-8}$          | 0,9                           | -7,954            | 0,074                       |
| B             | $10^{10}$         | $10^{-6}$          | 0,7                           | -5,845            | 0,225                       |
| C             | $10^9$            | $10^{-4}$          | 0,1                           | -3,000            | 0,428                       |
| D             | $10^8$            | $10^{-2}$          | 0,01                          | 0,000             | 0,643                       |
| E             | $10^7$            | $10^{-1}$          | 0,001                         | 2,000             | 0,789                       |
| F             | $10^6$            | 1                  | 0,0001                        | 4,000             | 0,929                       |

**Tabulka 6:** Vypočtené hodnoty normované vzdálenosti pro kategorii Enterprise



| úroveň externalizace  | popis   | navrhovaná hodnota metriky |
|-----------------------|---|----------------------------|
| P2P                   | ad-hoc integrace. Pro $n$ systémů max. $\frac{n(n-1)}{2}$ integrací | 1                          |
| adaptéry              | standardizace L4 protokolů  | 2                          |
| ad-hoc messaging      | MOM pro P2P spojení   | 3                          |
| content-based routing | agregace, publish/subscribe a směrování dle řídicích dat            | 4                          |
| dynamic routing       | externalizace řídicích pravidel (konfigurovatelnost)                | 5                          |
| BPM                   | řízení org. procesů dedikovaným SW nas ESB                          | 6                          |

**Tabulka 7:** Kvantifikace míry externalizace integračních funkcí mimo integrované aplikace

| úroveň DTD jazyka | popis                                  | navrhovaná hodnota metriky |
|-------------------|--|----------------------------|
| žádný DTD         | syntaxe není externalizována           | 1                          |
| vlastní DTD       | syntaxe určena vlastní definicí ad-hoc | 2                          |
| syntaktické XSD   | využití CSA standardizace [2], [11]    | 3                          |
| sémantické XSD    | využití IS standardizace [2], [9]      | 4                          |

**Tabulka 8:** Kvantifikace míry standardizace dat

Dalším důležitým parametrem je *míra standardizace jazyka pro definici dat s*. Opět se jedná o ordinální diskrétní veličinu. Definice je dána tabulkou 8.

Posledním parametrem musí být počet komunikací v rámci relace mezi integrovanými systémy  $n$ . Pro zjednodušení zanedbáváme případ bezstavové komunikace typu požadavek/odpověď a hodnotu parametru definujeme jako celkový počet přenosů dat nezávisle na směru. Při *one-way komunikaci* bude tedy  $n=1$ , při *request/response* bude  $n = 2$ , při 2 komunikacích tam a zpět bude  $n = 4$  etc.

Výše uvedené veličiny zavedeme do výpočtu volnosti vazby  $c$  následující úpravou vzorce 3:

$$c^{plain} = \frac{n}{e \cdot s} \left[ 1 - \frac{1}{D_i + 2C_i + D_o + 2C_o} \right] \quad (4)$$

Je možné, že pro praktické využití bude nutné upravit hodnoty  $e$  a  $s$  tak, aby lépe vyjadřovaly proporcionalitu mezi definovanými kategoriemi. K úpravě je možné přistoupit až po provedení testovacích výpočtů na reálných scénářích, což zatím nebylo provedeno. Hodnoty těchto veličin mají přímý vliv na obor hodnot volnosti vazby. Z toho důvodu zatím není vhodné navrhnout *normalizovanou míru vazby* tak, jak jsme to provedli pro vzdálenost mezi integrovanými 2.

#### 4.2. Využití normované míry vazby

Na rozdíl od normované vzdálenosti vyjadřuje míra vazby nejen vlastnosti existujícího prostředí, ale dotýká se i samotného návrhu konkrétní integrace (vstupy  $C_i$ ,  $D_i$ ,  $C_o$ ,  $D_o$  a  $n$ ). Normalizovanou míru vazby by tedy

mělo být možné využít přímo ve vyhodnocování kombinací integračních vzorů.

#### 4.3. Degradace výkonu

Nezanedbatelným markerem při objektivizaci integračních řešení je míra degradace výkonu, která úměrně souvisí s volností vazby mezi komunikujícími IS. Uvolnění vazby mezi systémy vynucuje jednak strukturalizaci rozhraní včetně datových formátů a dále pak použití dalších mezilehlých komunikačních prvků pracujících na vyšších vrstvách modelu ISO/OSI [15]. Degradace výkonu je pak dána především:

- Prolongací vytvoření resp. parsování zprávy ve všech bodech scénáře (volající, ESB intermediaries, volaný). V analýze je nutné zvážit jednotlivé vrstvy ISO/OSI, nicméně lze očekávat, že řádově nejvýznamnější bude práce s dokumenty ve formátu XML způsobená DOM transformací [12] a dále pak konverze datových typů do/z řetězového zápisu (viz dále).
- Zvýšení doby potřebné na transport informace sítí kvůli existenci mezilehlých prvků pracujících přímo s aplikačními daty.

Vyhodnocení degradace výkonu již indikuje nutnost rozdělit výpočet po jednotlivých vrstvách modelu ISO/OSI, což překračuje možnosti tohoto článku. Do výpočtu degradace výkonu bude zahrnuta i normovaná vzdálenost integrovaných systémů 2.

## 5. Závěr

Uvedli jsme přehled a rozbor některých vlastností prostředí, v němž jsou budovány komunikace mezi informačními systémy. Ukázali jsme možnosti jejich strukturalizace a ordinálního ohodnocení. Ukázali jsme

možnosti transformace původně kategoriálních dat na normované číselné hodnoty. Připravili jsme tak část podkladů pro vytvoření metodiky objektivního deterministického hodnocení integračních scénářů založeného na integračních vzorech. Rozbor samotných integračních vzorů nebyl obsahem tohoto článku.

## Literatura

- [1] G. Hohpe and B. Woolf, “*Enterprise Integration Patterns*”, Courier Westford, Westford, Massachusetts, 2009.
- [2] The Open Group, “*TOGAF Version 9*”, Van Haren Publishing, Zaltbommel, 2009.
- [3] W. Stevens, G. Myers, and L. Constantine, “*Structured Design*”, IBM Systems Journal, 13 (2), 115-139, 1974.
- [4] Integrační middleware pro zdravotnictví MIRTH, <http://www.mirthcorp.com>.
- [5] Enterprise Service Bus na Wikipedia, [http://en.wikipedia.org/wiki/Enterprise\\_service\\_bus](http://en.wikipedia.org/wiki/Enterprise_service_bus).
- [6] Orchestrace procesů na Wikipedia, [http://en.wikipedia.org/wiki/Orchestration\\_%28computers%29](http://en.wikipedia.org/wiki/Orchestration_%28computers%29).
- [7] Business Process Engine na Wikipedia, [http://de.wikipedia.org/wiki/Business\\_Process\\_Engine](http://de.wikipedia.org/wiki/Business_Process_Engine).
- [8] Object Management Group, <http://www.omg.org>.
- [9] Web konsorcia HL7, <http://www.hl7.org/>.
- [10] R.H. Elliot, “*XML Bible*”, Wiley, Hoboken, New York, 2001.
- [11] Standard XSD na W3C, <http://www.w3.org/TR/xmlschema-0/>, <http://www.w3.org/TR/xmlschema-1/>, <http://www.w3.org/TR/xmlschema-2/>.
- [12] Standard XSLT na W3C, <http://www.w3.org/TR/xslt>.
- [13] Web Service standardy na W3C, <http://www.w3.org/2002/ws/>.
- [14] Protokol SOAP na W3C, <http://www.w3.org/TR/soap12-part1>.
- [15] R. Pužmanová, “*TCP/IP v kostce*”, Kopp, České Budějovice, 2004.
- [16] Teorie uspořádání množin na Wikipedia, [http://cs.wikipedia.org/wiki/Kategorie:Teorie\\_uspo%C5%99%C3%A1d%C3%A1n%C3%AD](http://cs.wikipedia.org/wiki/Kategorie:Teorie_uspo%C5%99%C3%A1d%C3%A1n%C3%AD).