

Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE  
(magisterská)



Jakub Koňas

PODPORA SPOLUPRÁCE SE ZÁKAZNÍKY  
A SÍŤ NEZÁVISLÝCH APLIKACÍ

Katedra softwarového inženýrství

Vedoucí diplomové práce: Prof. RNDr. Jaroslav Král, DrSc.

Studijní program Informatika

Studijní obor: Softwarové systémy

Děkuji svému diplomovému vedoucímu Prof. RNDr. Jaroslavu Královi, DrSc. za vedení diplomové práce, konzultantům za poskytnutí cenných informací a svým blízkým za podporu při psaní této práce.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 20. dubna 2006

Jakub Koňas

## **Abstrakt**

*Název práce:* Podpora spolupráce se zákazníky a sítě nezávislých aplikací

*Autor:* Jakub Koňas

*Katedra (ústav):* Katedra softwarového inženýrství

*Vedoucí diplomové práce:* Prof. RNDr. Jaroslav Král, DrSc.

*E-mail vedoucího:* Jaroslav.Kral@mff.uni.cz

*Abstrakt:* Cílem práce je studium koncepce sítě nezávislých aplikací (SNA), se zaměřením na využití v oblasti podpory zákazníků (CRM, Customer Relationship Management) a podnikových informačních systémů. Práce konkretizuje pojem CRM, uvádí evoluci CRM systémů, objasňuje jejich interní strukturu a poskytované funkce s omezením ekonomických pojmů na nezbytné minimum. Vzhled do problematiky CRM uzavírá rozsáhlý přehled CRM systémů nabízených v ČR, součástí přehledu je aplikace pro nalezení CRM systému dle definovaných kritérií. V další části práce je vysvětlena koncepce SNA a provedeno zhodnocení kladů a záporů dostupných technologií pro vytváření SNA. Ze srovnání vyplývá, že SNA jsou ekvivalentem řešení na bázi SOA (Service-oriented Architecture). V závěru práce jsou analyzována konkrétní omezení, přínosy a možná další využití SNA v oblasti CRM.

*Klíčová slova:* CRM, zákazník, SNA, kompozit, služba, distribuovaný

## **Abstract**

*Title:* Customer relationship management and nets of independent applications

*Author:* Jakub Koňas

*Department:* Department of Software Engineering

*Supervisor:* Prof. RNDr. Jaroslav Král, DrSc.

*Supervisor's e-mail address:* Jaroslav.Kral@mffuni.cz

*Abstract:* The objective of this thesis is to study the concept of software confederations (networks of independent applications, NIA) orientated on its utilization in the field of customer relationship management (CRM) and the enterprise information systems. The opening chapters of the thesis specify the meaning of CRM, present the evolvement of CRM systems and clarify their internal structure and provided features. The insight into the CRM subject terminates with extensive survey of CRM systems which are being sold within the Czech Republic. The following chapters explain the conception of NIA and evaluate advantages and disadvantages of contemporary technologies for building NIA. The equivalence between NIA and SOA-based solutions is recognized. The final chapters discuss the specific limitations and possible benefits of NIA in the CRM field.

*Keywords:* CRM, customer, NIA, composite, service, distributed

# Obsah

<b>OBSAH.....</b>	<b>1</b>
<b>KAPITOLA 1 ÚVOD .....</b>	<b>3</b>
<b>KAPITOLA 2 VÝZNAM CRM.....</b>	<b>4</b>
2.1 PŘETĚŽOVÁNÍ POJMU .....	4
2.2 CO JE CRM.....	4
2.3 PROSPĚŠNOST CRM .....	6
2.4 EVOLUCE CRM .....	7
2.4.1 <i>Automatizace obchodu</i> .....	7
2.4.2 <i>Kontaktní CRM</i> .....	8
2.4.3 <i>Operativní CRM</i> .....	8
2.4.4 <i>eCRM – elektronické CRM</i> .....	9
2.4.5 <i>Analytické CRM</i> .....	9
2.4.6 <i>Komplexní CRM</i> .....	9
2.4.7 <i>CVM versus CRM</i> .....	11
<b>KAPITOLA 3 ARCHITEKTURA CRM SYSTÉMU .....</b>	<b>12</b>
3.1 APLIKAČNÍ ARCHITEKTURA.....	12
3.2 TECHNOLOGICKÁ ARCHITEKTURA .....	13
3.2.1 <i>Datový sklad</i> .....	13
3.2.2 <i>Operativní CRM</i> .....	14
3.2.3 <i>Kooperativní CRM</i> .....	14
3.2.4 <i>Analytické CRM</i> .....	15
<b>KAPITOLA 4 PŘEHLED CRM SYSTÉMŮ V ČR.....</b>	<b>16</b>
4.1 POŽADAVKY .....	16
4.2 STRUKTURA PŘEHLEDU .....	16
4.3 PROCES ZÍSKÁNÍ AKTUÁLNÍCH DAT.....	18
4.4 VÝSLEDKY.....	19
4.5 PROGRAM PRO PRÁCI S PŘEHLEDEM.....	23
4.5.1 <i>Funkce programu</i> .....	23
4.5.2 <i>Skórování</i> .....	25
4.6 SHRNUTÍ .....	25
<b>KAPITOLA 5 CRM JAKO SÍŤ NEZÁVISLÝCH APLIKACÍ.....</b>	<b>27</b>
5.1 LOGICKÁ STRUKTURA INFORMAČNÍHO SYSTÉMU .....	27
5.2 ZPŮSOBY IMPLEMENTACE IS.....	29
5.3 DEFINICE SNA .....	30
5.3.1 <i>Požadavky na NA</i> .....	30
5.3.2 <i>Komunikace v SNA</i> .....	32
5.3.3 <i>Vytvoření a správa vazeb</i> .....	35
5.4 VÝZNAM KOMPOZITŮ .....	37

---

<b>KAPITOLA 6</b>	<b>TECHNOLOGIE A SNA</b>	<b>38</b>
6.1	KOMUNIKAČNÍ PROTOKOLY	39
6.1.1	<i>Binární formát zpráv</i>	39
6.1.2	<i>Strukturovaný obsah zpráv</i>	40
6.1.3	<i>Shrnutí komunikačních protokolů</i>	43
6.2	MEDIACE	43
6.3	MIDDLEWARE	45
6.3.1	<i>Synchronní middleware</i>	45
6.3.2	<i>Asynchronní middleware</i>	46
6.4	INTEGRAČNÍ TECHNOLOGIE	48
<b>KAPITOLA 7</b>	<b>SERVISNÍ ARCHITEKTURA</b>	<b>50</b>
7.1	SOUČASNÉ IMPLEMENTACE SOA	50
7.1.1	<i>Webové služby</i>	52
7.1.2	<i>Podniková sběrnice služeb</i>	53
7.2	VZTAH SOA A SNA	55
<b>KAPITOLA 8</b>	<b>CRM DNES</b>	<b>56</b>
8.1	SOUČASNÉ ZPŮSOBY IMPLEMENTACE CRM	56
8.2	REÁLNÝ STAV CRM V ČR	58
8.2.1	<i>Příklady</i>	59
<b>KAPITOLA 9</b>	<b>MOŽNOSTI SNA V OBLASTI CRM</b>	<b>62</b>
9.1	PŘÍNOSY SNA	63
9.1.1	<i>Vnitřní</i>	63
9.1.2	<i>Vnější</i>	64
9.1.3	<i>Obecné</i>	64
9.2	PROBLEMATICKÁ MÍSTA	65
9.3	REALIZACE PŘES ESB	68
9.4	MOŽNOSTI SNA PRO ŘEŠENÍ AKTUÁLNÍCH PROBLÉMŮ CRM	68
<b>KAPITOLA 10</b>	<b>ZÁVĚR</b>	<b>69</b>
	<b>SLOVNÍK POUŽITÝCH TERMÍNŮ A ZKRATEK</b>	<b>71</b>
	<b>SEZNAM ZDROJŮ</b>	<b>75</b>
	<b>SEZNAMY OBRÁZKŮ, GRAFŮ A TABULEK</b>	<b>80</b>
	<b>PŘÍLOHA A</b>	<b>81</b>
	<b>PŘÍLOHA B</b>	<b>84</b>

## Kapitola 1 Úvod

Koncem šedesátých let prohlásil odborník na podnikání Ted Levitt že *základním principem úspěšného podnikání je schopnost získat a udržet zákazníka*. Tvrdil, že *moderní podnik by měl chápat celý obchodní proces jako pevně provázané snahy o objevování, vytvoření, povzbuzování a uspokojování potřeb zákazníka*. Levitt ak před více než třiceti lety předpověděl klíčové úlohy dnešních systémů pro podporu spolupráce se zákazníky (CRM). [12]

S rostoucím tlakem konkurenčního prostředí, vzrostl i význam CRM systémů. Implementace CRM systému byla pro podnik považována za konkurenční výhodu. Na trhu se proto objevila řada CRM produktů, které ji měly zajistit. Avšak jak uvádí studie analytické společnosti Gartner Research z roku 2003, přes 70% implementovaných projektů CRM v Evropě skončilo neúspěchem [10]. Odborníci se shodují, že jednou z příčin neúspěchu CRM byla mlhavá představa podniků o konkrétních funkcích CRM systému.

Práce si proto klade za dílčí cíl jasně definovat pojem CRM a objasnit způsob, jakým CRM systém dosahuje svého cíle – spokojeného zákazníka. Vychází přitom z tištěných i elektronických zdrojů a implementací nabízených na českém trhu. Vypracování podrobného přehledu CRM systémů v ČR, který shrnuje jejich přednosti a nedostatky, je dalším dílčím cílem práce.

Hlavním záměrem práce je studium možností použít koncepci sítě autonomních aplikací jako stavební architekturu CRM systému. Práce se rovněž zabývá využitím zmíněné koncepce při propojení s dalšími informačními systémy.

## Kapitola 2 Význam CRM

Zkratka CRM je jednou z mnoha hojně používaných třípísmenných zkratk ve světě informačních technologií (IT). Pochází z anglického originálu *Customer/Client Relationship Management* a do češtiny se nejčastěji překládá jako *Systém pro podporu zákazníků* či *Řízení vztahu se zákazníky*. Definovat, co všechno se skrývá pod označením CRM či CRM systémem, není jednoduché a zaslouží si bližší prozkoumání.

### 2.1 Přetěžování pojmu

CRM systémy doznaly velkého rozšíření v průběhu 90. let dvacátého století, kdy mnoho firem zejména v zemích Evropy a v Americe vidělo CRM systémy jako konkurenční výhodu oproti jiným firmám, čímž vytvořily obrovskou poptávku po rychle dostupných implementacích CRM. Na poptávku pochopitelně reagovali dodavatelé softwaru, kteří začali CRM produkovat. Bohužel mnoho výrobců software mělo o CRM velmi zjednodušenou představu, získanou často jen z reklamních sloganů, které sice nelhaly, neobsahovaly však důležité informace, které je potřeba mít při implementacích a nasazení na zřeteli. Výsledkem bylo, že producenti softwaru své produkty rozšířili např. o centrální adresář zákazníků, přidali strukturování prodeje podle jednotlivých zákazníků nebo doplnili jiný aspekt CRM. A pak hlavně nezapomněli označit svůj produkt přídomkem CRM. Taková situace vedla k jedinému – přetížení pojmu CRM, kdy málokdo věděl, co se pod tímto označením skrývá, jak by takový systém měl vypadat a co je vlastně cílem systémů CRM. Proto po prvotním nadšení, spojeným s nemalými investicemi ze strany odběratelů SW realizujících CRM, došlo i k patřičnému rozčarování. Když společnosti zpětně analyzovaly, co jim CRM přineslo za výhody, s překvapením došly, že návratnost vynaložených investic do CRM je v konečném součtu přinejmenším velmi diskutabilní. Tehdy došlo k procitnutí z prvního opojení a nastal čas analyzovat, o čem CRM systémy opravdu jsou, co mohou firmám přinést a naopak co od nich nelze očekávat.

### 2.2 Co je CRM

Na CRM lze nahlížet z různých pohledů, čemuž odpovídají různé definice. V této práci je kladen důraz zejména na implementační rovinu CRM systému a z principu CRM částečně i na rovinu obchodní, neboť CRM je o podnikové strategii, způsobu provádění obchodu a souvisejících činnostech, jak je uvedeno dále. Z důvodů uvedených v kapitole „Přetěžování pojmu“ je těžké najít univerzální definici, která by i pojem plně vystihovala. Podařilo se vyhledat velké množství definic, z nichž vybírám pouze podmnožinu, která se vyskytovala nejčastěji a vytváří nejucelenější dojem. Uvedené definice jsou ve většině případů doslovnou řeč citací zdrojů, spíše syntézou více myšlenek uvedených v příslušném zdroji.



Představíme-li si podnik jako množinu obchodních procesů, pak CRM je souhrn takových obchodních procesů, jejich částí a souvisejících struktur, v jejichž popisu se objevuje slovo „zákazník“. [17]

CRM zahrnuje pracovníky, podnikové procesy a technologii IS s cílem maximalizovat loajalitu zákazníků a v důsledku toho i ziskovost podniku. Je součástí podnikové strategie a jako takové se stává součástí podnikové kultury. Aplikace IS podporující CRM technologicky stále více využívají potenciálu a možností internetu. V problematice CRM se rozlišují tři základní roviny řešení:

- operační – orientované na zefektivnění klíčových procesů „kolem“ zákazníka, zejména front office úloh
- kooperační – představuje optimalizaci interakcí se zákazníkem a řešení vícekanálové komunikace,
- analytickou – zahrnující již agregace a aplikace znalostí o zákazníkovi, aplikace „business a customer intelligence“ a rovněž speciální CRM analytické aplikace na bázi datových skladů a dolování dat. [14]

Řízení vztahů se zákazníky (*Customer Relationship Management – CRM*) je podnikatelskou filosofií a strategií pro získávání a udržení nejvyšší hodnoty vztahů se zákazníky. CRM vyžaduje zákaznický orientovanou podnikatelskou kulturu pro podporu efektivních marketingových, obchodních a servisních procesů. Cílem je lepší porozumění potřebám zákazníků a kvalitnější reakce na ně, přičemž je nutné sdílet informace ze všech dostupných kanálů.

CRM je technologiemi podporované, ale není technologiemi vyvolané. Iniciované musí být podnikatelskými potřebami firmy. CRM je podnikatelská strategie vycházející z konceptu firmy zaměřené na zákazníka. Náchází a řídí hodnotové vztahy se zákazníky při zachování účelného vynakládání prostředků. Je to komplex pro efektivní marketing, prodej a péči o zákazníky složený ze čtyř složek:

- vize a firemní kultura zaměřená na zákazníka,
- vhodná podniková struktura, procesy a pravidla,
- motivovaný a kvalifikovaný personál,
- informačně-komunikační technologická podpora. [29]

CRM představuje systém obchodních, marketingových, komunikačních a servisních procesů v organizaci a příslušných technologií, který umožňuje cíleně řídit vztahy se zákazníky, uzpůsobovat nabídku podle jejich potřeb a přání a využívat všechny informace získané vzájemnými kontakty na jedné straně k výraznému zlepšení těchto vztahů a maximalizaci spokojenosti zákazníků, na druhé straně k zefektivnění a optimalizaci všech činností podniku, které se nějakým způsobem klientů dotýkají. Nejde tedy pouze o samotné technologie, neboť ty musí vždy být v souladu s celkovou strategií firmy a jejími zákaznickými procesy. [13]

Shrnuli uvedené definice, není CRM ani tak o informačním systému, jako o celkovém zaměření podniku na zákazníka, kterému jsou podřízeny veškeré činnosti podniku. Těžiště úspěšného CRM tkví hlavně v zavedení nových pracovních postupů, procesů, technologií a jejich sjednocení napříč všemi úrovněmi i sekcemi podniku, jejichž cílem je spokojený loajální zákazník podniku.

IS tedy vystupuje v roli součásti kvalitního CRM. Není pravda, že by IS sám o sobě realizoval CRM, jak je obecně vžito mezi laickou a překvapivě i částí profesní veřejnosti. IS ve spojení s dalšími technologiemi je stavební kámen vytvářející jednotnou platformu nad použitými technologiemi. Nejpokročilejší implementace CRM se posunuly prostřednictvím integrace s analytickými nástroji do fáze, kdy nejenže dokáží komplexně analyzovat historii kontaktu se zákazníkem, čímž umožňují firmě zaujmout nejvhodnější přístup k zákazníkovi, ale dokonce předpovídat i jeho budoucí chování a na základě předpovědi zákazníka cíleně oslovit speciální nabídkou.

### 2.3 Prospěšnost CRM

Asi by bylo vhodné zodpovědět otázku, v čem může být CRM firmě prospěšné nebo proč by mohla být loajalita zákazníka k podniku tak významná? Je zřejmé, že hlavním důvodem existence firmy je generovat zisk. V dnešní době existují obecně použitelné postupy, jež umožňují zvýšit zisk podniku (podpůrné prostředky pro snížení výdajů na různých úrovních, rozšíření působnosti na nové trhy...). Hlavní roli však stále hraje odběratel/zákazník, neboť on je primárním zdrojem příjmů. Není-li zákazník odebírající produkt, není příjem.

Dokud poptávka převažuje nad nabídkou, není klíčovost role zákazníka tolik viditelná jako v době, kdy dojde k nasycení trhu. Chtějí-li se podniky udržet na stávajícím trhu, musí najít způsob, jak si svou klientelu v silné konkurenci udržet, nebo ještě lépe, jak ji rozšířit.

Než osvětlím příčiny ovlivňující rozhodnutí zákazníka, zda u firmy setrvá či přejde k jiné, uvedu krátký příklad demonstrující význam loajality zákazníka v řeči peněz. Příklad je citován z materiálů společnosti AB Andersson působící jako konzultant v oblasti CRM [1]:

Vložte do kalkulačky náklady na propagaci, na provoz vašeho obchodního oddělení, služební cesty obchodníků, jejich auta, telefony, mzdy, vybavení, všechny náklady na nerealizované obchody atd., a vydělte je počtem nových zákazníků. Řekněme, že na získání jednoho zákazníka vynaložíte průměrně 600Kč (anebo také 20 000Kč, záleží na oboru, ve kterém se pohybujete). Kolik vás stojí, když zákazník přijde znovu a z jednorázového zákazníka se stane zákazník stálý? Ano, máte pravdu, nestojí vás to téměř nic, možná několik telefonátů, řekněme, že 100Kč. Opakovaný nákup má tedy pro vás ve skutečnosti o 500 Kč větší hodnotu než nákup jednorázový.

Pojďme dál. Váš průměrný prodej činí např. 10 000Kč a vaše marže je 10%. Zisk z prodeje stálému zákazníkovi činí 10% z 10 000Kč = 1 000Kč. Odečteme-li náklady 100Kč, dostáváme se na částku 900Kč čistého zisku.

Zisk z prodeje jednorázovému zákazníkovi činí také 1 000Kč, tento výnos ale musíme snížit o celých 600Kč vynaložených na jeho získání. Ve skutečnosti tedy získáte jen 400 Kč, což je v tomto ilustračním případě 2,25× méně než v případě stálého zákazníka.

Pokusme se nyní rozebrat faktory ovlivňující setrvání zákazníka u jedné firmy. Za prvé je třeba si uvědomit, že zákazník je vždy člověk či skupina lidí (tedy unikát ve smyslu vlastních emocí, zkušeností, znalostí, představ, očekávání), a proto při použití jedné obecné šablony nemůžeme nikdy dosáhnout spokojenosti všech. Jde tedy o to nalézt a aplikovat vhodnou jemnost přístupu k jedinci (= zákazníkovi), která bude stimulovat pohnutky vedoucí k uspokojení jeho potřeb, tím pádem ke spokojenosti v roli zákazníka s firmou, což se i nakonec projeví v jeho loajalitě. Uvedené pravidlo pochopitelně platí i ve smyslu, že omezíme-li stimuly vytvářející negativní vnímání podniku jedincem, posílíme tím jeho celkový pozitivní postoj. Řečeno metaforicky, jde o to zahrát zákazníkovi na tu správnou strunu.

CRM sám o sobě nezajistí nejvhodnější přístup k zákazníkovi – tíže bude ležet vždy na obchodnících podniku. Pomůže však ve shromažďování informací o zákazníkovi, ze kterých je možné při následném kontaktu se zákazníkem vycházet a zvýšit tak šanci na úspěšnost jednání. Díky zaznamenané historii kontaktu obchodník např. ví, že jeden zákazník upřednostňuje osobní jednání před telefonickým, další nerad jedná ve středu, neboť mívá schůzku vedení, jiného potěší zájem o rodinu či koníčky. V jednom z pohledů se lze na CRM dívat jako na systém pravidel určujících, jak sbírat data o zákaznících a aplikovat je.<sup>1</sup>

## 2.4 Evoluce CRM

Vývoj CRM prodělal za více než 15 let svého vývoje obrovský pokrok. Pro lepší pochopení, proč dostupné implementace CRM obsahují nabízenou funkčnost a jakou v nich hrají roli, uvádím stručný pohled do historie inspirovaný [13], [12] a [11].

### 2.4.1 Automatizace obchodu

Počátky CRM můžeme datovat do devadesátých let dvacátého století. S dostupností modernějšího HW, který byl schopen zvládnout i složitější operace v krátkých časech, byly učiněny první pokusy o automatizaci obchodních procesů (SFA, *Sales Force Automation*) prostřednictvím výpočetní techniky. Cílem bylo zvý-

---

<sup>1</sup> Uvědomíme-li si, že ke sběru a správě dat slouží majoritně informační systémy, je nasnadě, proč je v obecném povědomí CRM zakotveno jako informační systém. Je však důležité si uvědomit, že informační systém nemůže změnit chování podniku k zákazníkovi. Tato změna musí vždy vycházet ze zaměření firmy na zákazníka a teprve tehdy může být využit celý potenciál CRM.

šit efektivitu obchodního procesu, nikoliv zkvalitnit a řídit vztah se zákazníky. Systémy umožňovaly automatické generování cenových nabídek, objednávek, návrhů a smluv, dále obsahovaly podporu pro fáze obchodního procesu.

Z implementačního pohledu byl systém tvořen nezávislými systémy, každý systém si uchovával vlastní obraz dat (soubory, souborové databáze).

Uživatelé systému byli typicky obchodník/dealer, tj. pouze podmnožina zaměstnanců podniku, kteří přicházeli do styku se zákazníkem.

### 2.4.2 Kontaktní CRM

Když se podařilo zautomatizovat obchodní procesy, vynikla potřeba provázat v podniku ty činnosti, které jsou v přímém kontaktu se zákazníkem tak, aby se z pohledu zákazníka jevíly uceleně. Reklamuje-li například klient vadný výrobek v servisu, obchodník se tuto informací nedozví (systém nerealizuje provázání sekcí podniku) a v dobré víře osloví zákazníka s novým, třeba i dražším produktem, což zvýší jeho tendenci přejít ke konkurenci. Na základě situace uvedené a jí podobných došlo k centralizaci všech komunikačních kanálů zákazník – firma do jednoho prvku, tzv. kontaktního centra.

Díky sdružení komunikace s klientem do jednoho místa bylo potřeba modifikovat informační infrastrukturu. Pracovníci kontaktního centra (obchodník, podpora, servis) potřebovali aktuální ucelené informace o obsluhovaném zákazníkovi a navíc dostupné s minimální časovou prodlevou, čemuž model dosud používaných izolovaných<sup>2</sup> aplikací nevyhovoval. Problémem bylo i zaznamenání hovoru a potřebných akcí do různých systémů. Chyběla jejich provázanost.

Oddělené databáze a aplikace byly proto integrovány do jednoho centrálního systému, většinou s architekturou klient/server. Uložení dat realizovala relační databáze.

### 2.4.3 Operativní CRM

Další pokrok v oblasti IT a telekomunikací rozšířil možnosti CRM a kontaktních center. Vyšší výpočetní rychlost umožnila analyzovat data na jemnější úrovni. CRM sice na základě analýz stále neumí aktivně vybrat cílový segment (k tomu by potřebovalo mít informace BI, *business intelligence*), za to však zvládá ostatní agendu marketingových kampaní. Pomůže vybrat potenciální klienty pro marketingovou kampaň na základě demografických údajů a zakoupených produktů, v sbírá data z kampaně a v závěru dovede zhodnotit úspěšnost celé kampaně.

Kontaktní centra získala schopnost aktivního vedení hovorů s klientem (*call scripting*, instrukce pro vedení hovoru na základě výsledků analýz).

---

<sup>2</sup> Izolovanou aplikací se míní softwarový produkt, jehož vstupy a výstupy může provádět pouze člověk, tj. nelze používat funkcionalitu aplikace z jiného aplikace, např. prostřednictvím automatického sdílení dat.

Objem dat a potřeba pohotových analýz si vynutily přesun dat do datových skladů a těsnější propojení systémů (především účetního a CRM).

#### 2.4.4 eCRM – elektronické CRM

Vývojová etapa elektronického CRM reflektuje dostupnost Internetu pro širší veřejnost, což otevřelo nové komunikační kanály pro všestranné použití. Webový server společnosti má nově personalizovanou podobu podle preferencí zákazníka, funkci *Call Me Now* (pracovník kontaktního centra zavolá klientovi, přičemž na základě zaznamenané historie navštívených stránek a stránky, ze které zákazník požadavek vzešel, může pracovník oslovit klienta s předpřipraveným scénářem hovoru), on-line textovou (chat) i hlasovou (VoIP) komunikaci. Emailová komunikace dostává novou dimenzi v podobě automatických odpovědních systémů, přičemž ty nejdokonalejší dokáží automaticky detekovat a zodpovídat nejčastější dotazy.

Elektronické kanály znamenají obrovský přínos v nových datech o chování a zájmech zákazníka (navštívené webové stránky), aniž by se zvýšila náročnost zanesení dat do systému.

#### 2.4.5 Analytické CRM

Na trhu se objevují speciální nástroje sloužící k analýze klientských dat, hledání vzorů a trendů v chování klientů a jeho predikcí. Integrace s IS přináší schopnost modelovat a hlavně predikovat chování konkrétního zákazníka v různých situacích, což lze promítnout do přesného cílení marketingových kampaní včetně odhadu výsledku kampaně.

K analýze dat se používají metody ze statistiky, umělé inteligence, teorie informací, např. klasifikace, shluková analýza, asociční pravidla a logická a lineární regrese, rozhodovací stromy či neuronové sítě.

V systému dochází k implementaci analýz a optimalizacím vedoucím ke zkrácení výpočtu analýz. Výborná analýza trvající řádově desítky vteřin je například nepřijatelná pro použití v kontaktním centru, jež vyžaduje téměř okamžitou odezvu, přesto má stále své opodstatnění pro dlouhodobější plánování.

#### 2.4.6 Komplexní CRM

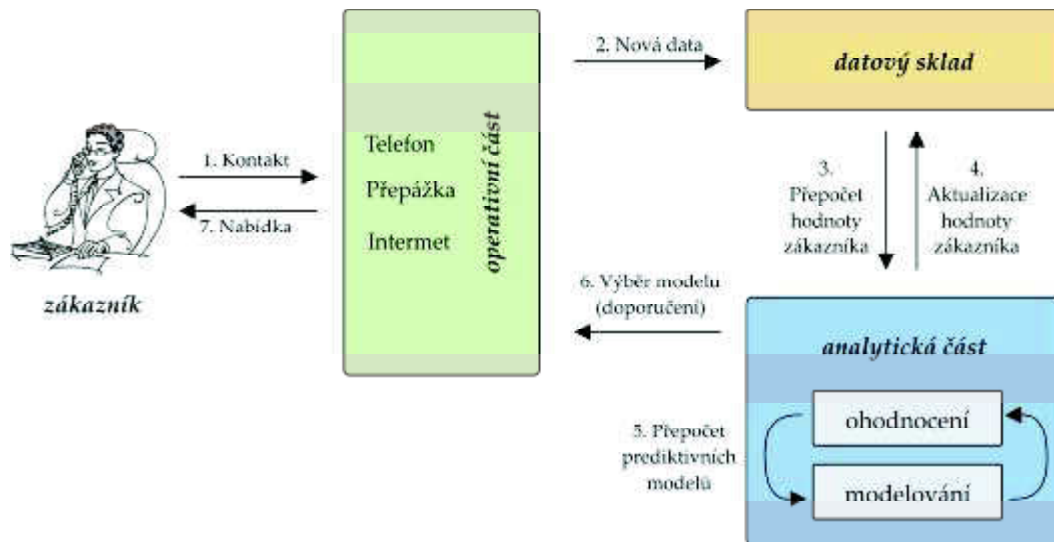
Komplexní CRM, někdy též označované jako „uzavřená smyčka“, je dosud posledním vývojovým stádiem CRM. Jeho cílem je získat údaje o aktuální ziskovosti klienta (*customer profitability*), o potenciální hodnotě klienta v průběhu jeho celého životního cyklu z hlediska organizace (*customer lifetime value*), o riziku jeho přechodu ke konkurenci (*churn propensity*) a tyto údaje využít ihned po jejich získání. Myšlenka je do praxe uvedena prováděním analýz na pozadí, přičemž výsledky nejsou pouze reportovány, nýbrž ukládány zpět do CRM k jejich využití při následných analýzách.



Aby mohlo dojít ke zpětné projekci výsledků na správná místa, obsahuje CRM systém i BI v podobě obchodních/procesních modelů. CRM systém je jednou komplexní aplikací operující nad datovým skladem.

### Magická obrazovka

Příklad fungování takto integrovaného systému pro kontaktní centrum uvádí [17].



**Obr. 1** Magická obrazovka

Obr. 1 ilustruje modelovou situaci, kdy klient volá do zákaznického centra s nějakým požadavkem (1), který operátor zaznamená v operativní části CRM. Požadavek může mít vliv na hodnotu zákazníka pro firmu, nebo může iradit k riziku ztráty zákazníka a nutnost učinit pokus klienta udržet. Proto se nově zjištěné údaje zapíší do zákaznického datového skladu (2) a spustí se analytický nástroj pro scoring klienta a aktualizaci použitého modelu (3). Výstupem tohoto procesu jsou údaje o jeho nové ziskovosti a hodnotě, resp. riziku ztráty, které se uloží zpět do datového skladu (4). Tato nová hodnota se pak stává klíčovým údajem pro řízení přístupu ke klientovi. Zároveň může být tento rozhovor příležitostí k další nabídce. Jednotlivé prediktivní modely pro daného zákazníka se proto přepočítají (5) a model, který dává nejlepší výsledek, je ve formě doporučení k dalšímu kroku předán operativní části CRM (6), která jej zobrazí operátorovi. Ten pak zákazníkovi učiní doporučenou nabídku (7). Klientova reakce se opět uloží do datového skladu a celý proces se opakuje.

Celý tento proces analýz a modelování je z pohledu operátora neviditelný. Jemu jsou prezentovány pouze konkrétní instrukce pro vedení rozhovoru se zákazníkem. To je koncept tzv. „magické obrazovky“, která zobrazuje pokyny na základě detailní znalosti zákazníka a složitého modelování jeho chování v budoucnu. Podstatné přitom je, aby tato doporučení byla zobrazována okamžitě. Zpoždění v řádu desítek vteřin je nepřijatelné, protože by se celý systém míjel účinkem.

### 2.4.7 CVM versus CRM

CVM (*Customer Value Management*, řízení hodnoty zákazníka) je vizí, kam se posune CRM. Nejpokročilejší implementace CRM se již dají označit jako CVM. CVM posunuje rovinu od řízení vztahu se zákazníkem k řízení a aktivnímu budování hodnoty vztahu se zákazníkem. Podstatou CVM je jemný rozpad parametrů zákazníka, jejich vzájemné provázání a napojení na analýzy tak, aby bylo možné rozdělit zákazníky do skupin s podobným chováním a přistupovat rozdílně k těmto skupinám, např. *one-to-one marketing*, čili oslovení zákazníka s konkrétní nabídkou podle jeho potřeb.

Modeluje se CLV (*Customer Lifetime Value*, celoživotní hodnota zákazníka), která spočívá nejen v dosavadní ziskovosti klienta, ale i v očekávané ziskovosti klienta v budoucnu, s jejíž pomocí lze ještě lépe určovat míru investic a péče k zákazníkovi.

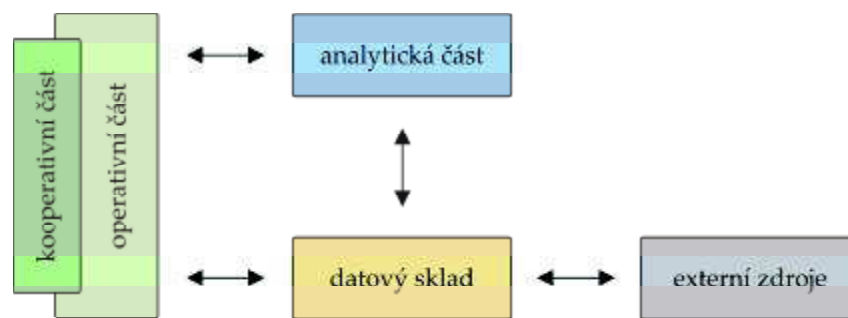
K uložení dat se používají multidimenzionální datové sklady, které umožňují kumulované pohledy na data v různých řezech podle jednotlivých dimenzí a hierarchií s variabilní detailností. Z principu CVM neexistují univerzální krabicové produkty CVM, protože řešení je vždy specifické pro konkrétní podnik a je potřeba jej navrhnout na míru. Systém je obvykle realizován kombinací produktů několika dodavatelů. [11]

## Kapitola 3 Architektura CRM systému

Na strukturu systému CRM můžeme nahlížet z různých úhlů. Jednak z pohledu funkčních celků a jejich vzájemného propojení (tzv. aplikační architektura), dále z implementačního pohledu (tzv. technologická architektura), reprezentující uspořádání informačních, komunikačních a softwarových prostředků. Rozbor aplikační architektury čerpá z [11], [16], [3].

### 3.1 Aplikační architektura

Aplikační architekturu lze rozdělit na tři až čtyři základní části – operační, analytickou (kooperativní) a dále datový sklad. Jejich vztahy ilustruje obr. 2.



Obr. 2 Aplikační architektura

*Datový sklad* integruje ostatní části systému prostřednictvím sdílení dat. Data se získávají přímým zadáním prostřednictvím modulů UI systému, syntézou již vložených dat a pomocí datových pump z externích zdrojů (ETL). Primárním zdrojem dat je operativní a kooperativní část, data jsou získávána přímým zápisem do UI, které zprostředkuje jejich uložení do datového skladu. Sekundárním zdrojem jsou výsledky analýz, jež jsou provedeny nad stávajícím daty. Výsledky analýz jsou zpětně uloženy do datového skladu. Terciární zdroj tvoří veřejně dostupné informace (např. vnější rozhraní dodavatelů pro sledování průběhu zpracování zakázek). Přenos dat z vnějšího prostředí probíhá typicky třífázově, kdy data jsou nejprve získána z vnějšího prostředí (*extract*), upravena do podoby vhodné pro systém (*transform*) a teprve pak importována do systému (*load*).

Účelem *operativní části* je zajištění co největší efektivity podnikových procesů. Je zaměřena na automatizaci a řízení podnikových procesů týkajících se marketingu, obchodu a servisních činností (automatizace tvorby nabídek a smluv, plánování a evidence aktivit, správa kontaktů, správa dokumentů...).

Některé zdroje (např. [16]) vydělují z operativní části ještě podmnožinu realizující vlastní kontakt se zákazníkem, tzv. *kooperativní část*. Kooperativní část sdružuje veškeré komunikační kanály (telefon, fax, mail, web...) pro kontakt se zákazníkem do jednoho bodu a vytváří tak komunikační rozhraní mezi podnikem a zákazní-



kem/partnerem/dodavatelem. V dnešní době je pro ni užíván název *call centrum* či *kontaktní centrum*.

Samotná data sice mohou leccos napovědět, ale jejich opravdový užitek přichází až s jejich analýzou, kterou poskytuje *analytická část*. Data v surové podobě pouze zachycují historii. Aby bylo možné vytvářet i prognózy do budoucna, je potřeba opakovaně nacházet podobnosti a souvislosti v historických datech, vytvářet abstraktní modely odrážející historický vývoj a jejich aplikací odhadovat budoucí vývoj.

## 3.2 Technologická architektura

CRM systém z pohledu implementace je souhrn použitých technologií, rozvržení do funkčních jednotek a jejich vzájemného provázání do kompaktního celku. Jinými slovy hledá odpověď na otázky co, kde, jak se má dělat a komu, jak dát vědět o výsledku.

### 3.2.1 Datový sklad

Základním kamenem je *datový sklad* jakožto centrální úložiště veškerých informací. V dnešních systémech je nejčastěji realizován formou relační či objektové databáze s možností multidimenzionálního<sup>3</sup> pohledu na data. Datový sklad je zodpovědný za uchování a zpřístupnění získaných informací. Ačkoliv systém může být výrobcem rozdělen do více (i samostatných) funkčních celků (tzv. modulů), datový sklad bývá v systémech obvykle pouze jeden. Je pak povinností modulu zajistit si přítomnost potřebných struktur ve sdíleném datovém skladu.

Existují případy, kdy CRM systém obsahuje více datových skladů. Pomineme-li případy zapříčiněné překonaným starším modelem CRM (viz kapitola „Evoluce CRM“ na straně 3), bývají častým důvodem úzce specializované moduly dodávané různými výrobci, které pro svou činnost vyžadují speciální způsoby uložení dat. Pak nastupuje na scénu synchronizace dat ve skladech, kdy data z jednoho datového skladu jsou importována do druhého a naopak. Protože jde o náročný proces, většina výrobců dává přednost vytvoření vlastního modulu s požadovanou funkcionalitou (např. zakoupením knowhow či vlastním výzkumem), jelikož vytvoření a odladění synchronizace je časově a tedy i finančně náročné (úprava dat do správného formátu, zabezpečení).

---

<sup>3</sup> Multidimenzionalitou databáze se míní její schopnost prezentovat v sobě obsažená data z různých úhlů. V klasickém případě jsou data ukládána do tabulky o dvou dimenzích, kdy jedna dimenze je typicky čas, druhá pak vhodný ukazatel (prodej, stav zaměstnanců...). Ačkoliv je možné i z klasického dvoudimenzionálního uspořádání vytvořit vícedimenzionální pohled, dostupné databáze již mají v sobě integrovánu technologii, jak přímo vytvářet vícerozměrnou prezentaci dat, což v důsledku zrychluje odezvu celého systému, neboť složité operace s daty jsou prováděny přímo databází, nikoliv až aplikací.

Příkladem využití multidimenzionální databáze může být situace, kdy nás například zajímá pohled na prodej určité komodity v daném časovém rozpětí ve zvoleném středisku/segmentu/teritoriu.

### 3.2.2 Operativní CRM

Implementaci *operativní části* lze charakterizovat jako vstupní a výstupní bod pro personál podniku. Skládá se z uživatelského rozhraní (UI), kterým uživatelé zadávají nová data do systému a pomocí kterého na dříve vložená data nahlížíjí.

Dále obsahuje pro uživatele neviditelnou složku scénářů (algoritmy), které se spouští buď automaticky na základě definovaného stavu systému (reakce) nebo manuálně podle rozhodnutí uživatele (např. informování zákazníků o nových produktech). Algoritmy jsou navrženy tak, aby automatizovaly a řídily procesy uvnitř podniku. Vedle automatizace zajišťují v součinnosti s UI i korektnost zadávaných dat podle interních pravidel podniku. Například může existovat i pravidlo, že zákazníci s nespacenými dluhy nemohou objednávat další zboží, dokud neuhradí předchozí závazky. Systém by pak před vytvořením nové zakázky kontroloval platební morálku zákazníka a zablokoval zanesení nové zakázky dlužných odběratelů do systému. Řídící role scénářů se projevuje tak, že uživateli systému jsou přístupné jen ty funkce UI, které jsou pro operaci validní (např. vydání pokynu k výrobě zakázky až po splacení zálohy).

Posledním významným prvkem operativní části jsou šablony. Šablony ulehčují zadávání nových dat. V systémech vystupují buď jako automatické dokumenty nebo průvodci. Automatický dokument obsahuje předem vyplněný text (například text smlouvy) a variabilní pole, do kterých je vložen text nový (např. údaje o smluvní protistraně). Po zadání údajů pro vyžadovaná pole je podle šablony vygenerován a uložen nový dokument, který je od toho okamžiku nezávislý na původní šabloně. Dojde-li proto následně k úpravě šablony, změny nebudou do dříve vygenerovaného dokumentu promítnuty. Vygenerovaný dokument je možné dále ručně upravit pro potřeby konkrétní situace. Průvodci jsou prvek UI používaný pro zadání údajů složitějších operací, kdy jednotlivá pole mají vzájemnou algoritmicky podchytilnou vazbu a/nebo obsahují příliš mnoho polí.

### 3.2.3 Kooperativní CRM

Jak již zmiňuje kapitola „Aplikační architektura“ na straně 3, kooperativní část je realizována formou kontaktního centra. Protože prostřednictvím kontaktního centra probíhá komunikace se zákazníkem, která je ze své podstaty real-timeovou<sup>4</sup> záležitostí, je při jeho implementaci kladen důraz na okamžitou odezvu použitých technologií. Použitými technologiemi jsou telefonie (analogová i digitální, např. IP

---

<sup>4</sup> Označení real-time, pojem v češtině používaný jako schopnost pracovat v reálném čase, charakterizuje systém/aplikaci/operaci, jejíž provedení je časově kritické. Pro real-timeové operace je typické, že přicházející data zpracovávají okamžitě bez prodlev (např. při telefonování prostřednictvím Internetu je základním požadavkem, aby hovor byl přenášen s minimálním zpožděním) resp. s jasně definovaným zpožděním.

Opakem práce v reálném čase je práce v čase virtuálním, kdy operace je sice provedena v určitý čas, který však nemusí odpovídat zažitému plynutí času (např. simulace různých procesů se často provádějí ve virtuálním čase, kdy čas je zrychlen nebo zpomalen, ovšem relativní časové souvislosti mezi operacemi simulace jsou zachovány).

telefonie), faxy, email, web i klasická korespondence, dále řízení rozložení příchozích volání na jednotlivé pracovníky. Vhodným propojením technologií lze dosáhnout automatického přesměrování příchozích hovorů na obchodníka, který je zodpovědný za obchodní případ volajícího zákazníka (detekce dle telefonního čísla), optimálního využití pracovníků kontaktního centra, automatického zpřístupnění informací o volajícím a při kompaktním propojení všech částí systému i vlastního řízení hovoru formu magické obrazovky (viz „Magická obrazovka“ na straně 3).

### 3.2.4 Analytické CRM

Analytická část doplňuje systém o podporu rozhodování, vyhodnocování a řízení s cílem dosáhnout lepšího porozumění zákazníkům. K získání výsledků se využívají metody dolování dat (*data-mining*), statistické metody, OLAP (*On-Line Analytical Processing*), skórování, modelování, umělá inteligence a další.

Dolování dat [37] slouží k vyhledávání dosud neznámých závislostí v datech. V CRM se využívá pro hledání vzorů chování.

Ze statistických metod jsou zajímavé lineární regrese (statistická technika používaná k nalezení nejlepší lineární závislosti mezi cílovou proměnnou a množinou vysvětlujících proměnných), logistická regrese (obdoba lineární regrese, ale pro kategoriální cílovou proměnnou, např. typ klienta, příznak ne/splácel u půjčky), dále rozhodovací stromy, genetické algoritmy (optimalizační techniky napodobující teorii evolučního vývoje v přírodě, základní stavební prvky těchto algoritmů jsou selekce, křížení, mutace) a neuronové sítě (nelineární prediktivní model, který vznikl jako analogie k mozkovým neuronovým sítím živých organizmů).

OLAP umožňuje pracovat s daty z různých pohledů, rozdělovat je a slučovat dle volitelných kritérií, zobrazovat do mnoha typů grafů i tabulek. Výhodou metody je její široká variabilita pohledů a okamžitá dostupnost výsledků.

Skórování průběžně ohodnocuje zákazníka pomocí sady parametrů definovaných v datovém skladu, parametry jsou vzájemně provázané. Využívá se např. pro vyhodnocování ziskovosti zákazníka, jeho CLV (*Customer Lifetime Value*) či segmentaci zákazníků podle podobných hodnot ve zvolené podmnožině sledovaných parametrů.

Analýza asociací slouží k vyhledávání souboru pravidel zachycujících vzájemný výskyt dvou či více jevů. Příkladem pravidla může být: „Jestliže si zákazník koupí chleba, koupí si v 75 procentech případů i máslo.“

Modelování doplňuje do CRM systému predikci. Modelování vychází z nalezených závislostí, pravděpodobností jejich výskytů a skórování. Nalezené modely jsou ukládány do datového skladu pro opětovné použití. Při dostupnosti nových údajů dochází k přepočtu modelů. Výstupem jsou předpokládané změny parametrů (tzv. *what-if* modelování), které lze použít například pro výběr nejvhodnější nabídky či dalšího postupu, jako podklad pro personalizaci či *one-to-one* marketing.

## Kapitola 4 Přehled CRM systémů v ČR

První část práce se věnovala obecnému popisu, co je to CRM systém, jaký je význam použití CRM v podniku a jaké funkce nabízejí nejpokročilejší implementace CRM systémů. Tato kapitola si klade za cíl zmapovat situaci s CRM systémy v České republice. Nejvhodnějším způsobem, jak dosáhnout cíle, se ukázalo vypracovat detailní přehled CRM systémů nabízených v ČR.

### 4.1 Požadavky

Základní cíl zmapovat trh CRM systémů nabízených v ČR jsem rozdělil na dva požadavky:

1. Přehled by měl být pokud možno úplný, tj. obsahovat všechny CRM systémy, které jsou v ČR nabízeny.
2. Přehled by měl obsahovat informaci o pokročilosti jednotlivých implementací ve srovnání s obecnou teorií CRM.

Vedle základního cíle jsem pro práci definoval několik dalších cílů, které mě mohou zvýšit užitnou hodnotu vypracovaného přehledu.

Přehled by měl umožnit zájemci o CRM (tj. podniku) vyhledat z nabídky CRM systémů pro něj nejvhodnější systém a doporučit i vhodného dodavatele. Na úspěšnost zavedení CRM systému do podniku má velký vliv množství zkušeností dodavatelské firmy, přehled by proto měl obsahovat informaci o zkušenostech dodavatele s procesem výběru, implementace a nasazení CRM systému. Součástí přehledu by měla být rovněž informace, pro koho je CRM systém určen.

Z pohledu další části diplomové práce jsou důležité údaje o způsobu implementace CRM systému a možnostech jeho propojení s okolním světem.

### 4.2 Struktura přehledu

Přehled jsem strukturoval tak, aby splňoval definované požadavky. Protože existuje několik veřejně dostupných přehledů CRM systémů, nechal jsem se částečně inspirovat jejich strukturou. Vedly mě k tomu dva důvody. Tím, že část přehledu bude přibližně reflektovat obecně používanou strukturu, měl by se případný zájemce o CRM systém v přehledu snáze orientovat. Druhým důvodem bylo umožnit korektní převod dat z jiných přehledů do vlastního přehledu.

Jako unikátní klíč přehledu jsem použil název CRM systému. Zde si dovoluji upozornit, že řada autorů sestavila svůj přehled dle názvu dodavatelské společnosti, což ve svém důsledku přineslo jisté problémy – v přehledu se tentýž systém objevil tolikrát, kolik bylo dodavatelských společností a naopak v případě více systémů dodávaných jedinou společností postrádal přehled informace o vlastnostech jednotlivých systémů, protože přehled obsahoval pouze jeden záznam pro jednoho dodavatele. Použití názvu CRM systému jako klíče tyto nedostatky odstraňuje.

Struktura přehledu je rozdělena do tematických skupin podle významu obsažených informací. Přesnou strukturu přehledu uvádí „Příloha A“.

### **Funkčnost systému**

Informace o schopnostech CRM systému obsahuje skupina *Funkčnost systému*. Ukazuje, které funkce CRM systém poskytuje. Oproti zbylému přehledu jsem použil ještě jednu úroveň seskupení položek podle oblastí, do kterých sledované funkce spadají (call centrum, marketing, prodej, servis, obchodní procesy).

Při zpracování informací o funkčnosti CRM systému jsem se musel vypořádat s dvěma protichůdnými problémy. Na jedné straně jsem potřeboval zjistit míru podpory funkčních oblastí CRM, tedy shromáždit velké množství dat. Na druhé straně bylo nutné zajistit aktuálnost informací, neboť data z jiných přehledů byla zastaralá a/nebo neobsahovala některé sledované údaje. Nejvhodnějším místem, kde lze získat data v požadované kvalitě, bylo buď přímo u výrobce systému nebo u jeho dodavatele. Jak se později ukázalo, po vytěžení informací z materiálů výrobce (web, články v časopisech o systému) však nebyla data úplná. Proto jsem se obrátil přímo na výrobce či dodavatele s žádostí o doplnění nových a zároveň i kontrolu již získaných dat. O vlastním procesu získání a aktualizace dat pojednává kapitola 4.3 „Proces získání aktuálních dat“. Zde pouze zmíním, že právě proces aktualizace dat byl jednou z příčin, proč jsem pro skupinu *Funkčnost systému* zvolil formu výběru konkrétních funkcí s 3-stavovou logikou určující míru podpory dané funkce (dostupná, nedostupná, připravuje se). Použitý způsob do jisté míry chrání i před subjektivností odpovědi získaných formou volného textu (např. „naše call centrum umí vše“, což může být z pohledu dotázaného pravda, ve srovnání s jiným produktem však nikoliv). Navíc jemné rozložení funkčnosti systému na dílčí funkce umožňuje zpětně syntetizovat, do jaké hloubky jsou funkční oblasti CRM daným systémem podporovány.

### **Dodavatel**

Základní informace o dodavateli (jméno, kontakt, web) obsahuje skupina *Dodavatel*. Každý CRM systém v přehledu má uveden jednoho (primárního) dodavatele, u kterého je uvedena jeho charakteristika, kontakty a dále seznam dalších dodavatelů či implementačních partnerů.

Informace o zkušenostech dodavatele je obtížné uchopit již z prosté podstaty, že zkušenost má subjektivní charakter, a tudíž ji nelze přesně měřit či kategorizovat. Proto jsem se opřel o číselné informace, které jsou objektivní a umožňují vytvořit si částečnou představu o dodavateli. Zkušenostmi dodavatele se zabývají položky „Počet konzultantů v ČR“, „Roky zkušeností s implementací CRM“, zázemí dodavatele charakterizuje „Obrat společnosti“ postihující dynamiku růstu dodavatele za posledních 5 let. Celkový obrázek o dodavateli lze doplnit prostřednictvím položky „Reference produktu v ČR“.

Nabídka CRM systémů je dnes natolik široká, že původně univerzální řešení jsou specializována dle konkrétních požadavků zájemce. Vznikají tak CRM systémy



určené pro různá odvětví, kdy funkce systému jsou již v základní verzi uzpůsobeny pro konkrétní obor podnikání, což pro zájemce znamená zkrácení doby nasazení (není potřeba náročné customizace systému), snížení ceny (jeden specializovaný produkt prodá výrobce opakovaně, čímž mu klesnou náklady a tím většinou i koncová cena produktu) a v neposlední řadě lze očekávat i zvýšenou podporu pro dosud neimplementovaná specifika odvětví. Specializaci systému se věnuje sekce „Specializované moduly, funkce nebo nastavení pro různá odvětví“. Druhým rozměrem vedle specializace CRM systému je velikost podniku, pro kterou je systém určen (skupina „Pro jakou velikost podniku je produkt určen“, velikost podniku je měřena dle jejího ročního obrátu).

Informace o systému doplňují položky „Průměrná doba implementace u podniku střední velikosti“, „Počet instalací produktu v ČR“ a „Počet instalací produktu celosvětově“.

### Architektura a platformy

Skupina *Architektura a platformy* popisuje implementační architekturu systému (klient/server, třívrstvá, propojená aplikace spolupracujících agentů či služeb), platformu operačního systému a použitou databázi. Zamýšlená struktura dále obsahovala položky „Způsob komunikace v rámci systému / typ použité propojovací technologie v rámci produktu“ (SOAP, DCOM, CORBA...) a „Možnost propojení systému se systémy jiných výrobců“. Bohužel se mi podařilo získat informace pouze pro několik systémů, proto se ve výsledné struktuře přehledu neobjevily.<sup>5</sup>

### Různé

Strukturu přehledu uzavírá skupina *Různé*, kam jsem zařadil informace, které nebylo možné přiřadit k žádné jiné skupině. Na základě získaných dat jsem později vyčlenil položky o dvou relativně nových rysech CRM systémů – mobilním přístupu a pronájmu systému<sup>6</sup>.

## 4.3 Proces získání aktuálních dat

Naplnění přehledu aktuálními daty probíhalo ve dvou fázích, výběru produktů a aktualizace informací o produktu.

<sup>5</sup> Získané informace jsou využité v kapitole 8.1 „Současné způsoby implementace CRM“.

<sup>6</sup> V případě pronájmu se jedná o formu outsourcingu. Systém běží vzdáleně u výrobce či dodavatele. Jeho povinností je spravovat systém, provádět jeho aktualizace, přizpůsobení, zálohy apod. Nájemce systém pouze používá, nevlastní. Na straně nájemce tedy odpadají náklady na osobu/y odpovědnou za korektní chod systému. Ty jsou nahrazeny náklady na pronájem systému (většinou paušál), jež však jsou nižší. Navíc nájemce má takto garantovanou kvalitu služeb, čehož lze při správě (cizího) systému vlastními lidmi těžko dosáhnout.

Hrubý výběr produktů se opíral o existující seznamy CRM systémů. Bohužel žádný z autorů seznamů nereferoval na nabídku poskytnout offline databázi dat výměnou za jejich aktualizaci a doplnění. Proto jsem byl nucen vytvořit program, který mi potřebné informace z online databáze zjistil a připravil k dalšímu ručnímu zpracování. Nejprínosnější byly seznamy:

- (1) přehledu CRM České společnosti pro systémovou integraci (ČSSI)<sup>7</sup>,
- (2) přehledu CRM časopisu System On Line (SOL)<sup>8</sup>,
- (3) profilu firem poskytovaných služeb v oblasti CRM<sup>9</sup>, Jan Dohnal
- (4) přehledu CRM časopisu Business World 7/2005.

Hrubý výběr jsem rozšířil o produkty odkazované z knih, časopisů, článků na Internetu a přímým vyhledáním na Internetu.

*Aktualizace a doplnění dat* čerpaly z informací na Internetu, následně byly doplněny o data získaná prostřednictvím dotazníku. Těsně před uzavřením sběru dat byla provedena aktualizace daty z nového přehledu uveřejněného v časopise Business World 7/2005.

Abych zvýšil šanci na získání vyplněného dotazníku, kladl jsem důraz na jeho jednoduchost a rychlost vyplnění, což se zpětně promítlo i do struktury přehledu, jak zmiňuje kapitola 4.2 „Struktura přehledu“. Jak se později ukázalo, prozíravým krokem bylo předvyplnění dotazníků veřejně dostupnými údaji. Údaje sice byly často zastaralé, přesto jejich použití zvýšilo počet doručených odpovědí.

Protože v oblasti obchodu se často používají produkty MS Office, byl dat azní k formát Excelovské tabulky, kde první sloupec obsahoval popis požadované informace, druhý byl určen pro odpověď. Použití formátu Excelu mělo tu výhodu, že respondent obdržel vše potřebné do emailové schránky, mohl úkol delegovat na zodpovědného pracovníka prostým přeposláním emailu a navíc nebylo potřeba dlouze vysvětlovat, jak tabulku správně vyplnit, neboť práce s Excelem patří v oblasti obchodu k základním dovednostem. Další výhodou byla snadná manipulace s daty při jejich slučování do výsledného přehledu.

## 4.4 Výsledky

Přehled obsahuje informace o 96 CRM systémech nabízených v ČR. Výsledný přehled není kvůli velkému rozsahu uveden v tištěné podobě, pouze na přiloženém elektronickém nosiči.

<sup>7</sup> [http://www.cssi.cz/trh\\_produkty.asp?hodnot a=030E801&typakce=typproduktu](http://www.cssi.cz/trh_produkty.asp?hodnot a=030E801&typakce=typproduktu)

<sup>8</sup> <http://prehledy.systemonline.cz/index.php?skip=1&kat=2>

<sup>9</sup> <http://rb.vse.cz/~dohnal/texty/firele%202004.doc>

Z informací jsem získal následující údaje:

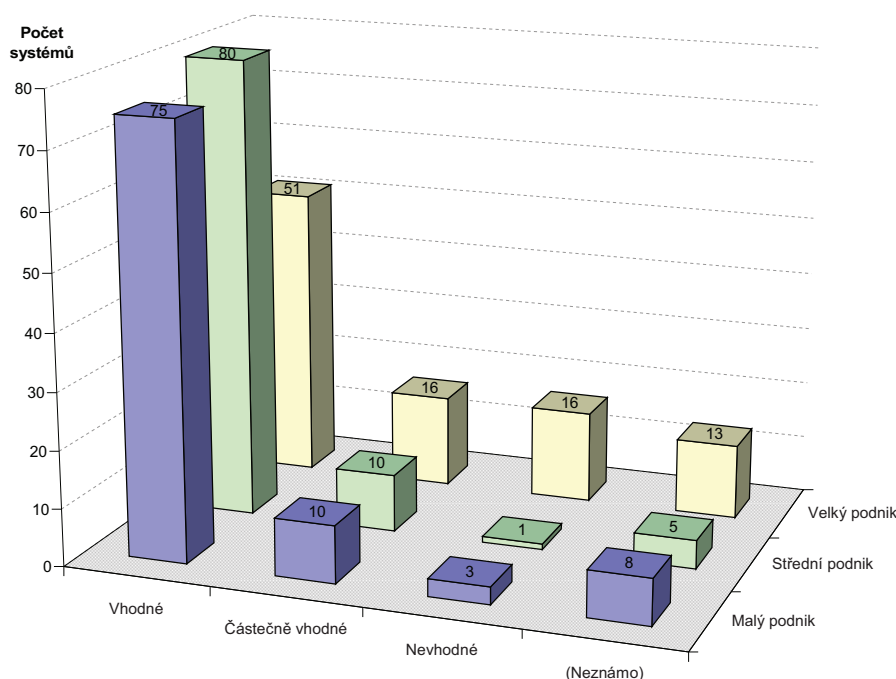
- srovnání CRM systémů podle velikosti podniku, pro který je systém určen
- srovnání očekávané rychlosti nasazení CRM systému do středně velkého podniku
- informaci o podpoře datových skladů různých výrobců
- rozdělení CRM systémů dle architektury systému

### Vhodnost CRM systému podle velikosti podniku

Tab. 1 ukazuje rozložení CRM systémů, pokud sledovaným kritériem je velikost podniku, do kterého má být systém nasazen. Protože každý systém může být použit pro jednu až více velikostí podniku, srovnání je provedeno pro každou kategorii kritéria zvlášť. Řádky tabulky určují míru podpory pro danou velikost podniku, sloupce sledují kategorii kritéria, hodnoty uvádějí počet systémů vyhovujících kritériím řádku i sloupce.

	Malý podnik	Střední podnik	Velký podnik
Vhodné	75	80	51
Částečně vhodné	10	10	16
Nevhodné	3	1	16
(Neznámo)	8	5	13

**Tab. 1** Rozdělení systémů podle velikosti cílového podniku



**Graf 1** Rozvrstvení systémů podle velikosti cílového podniku



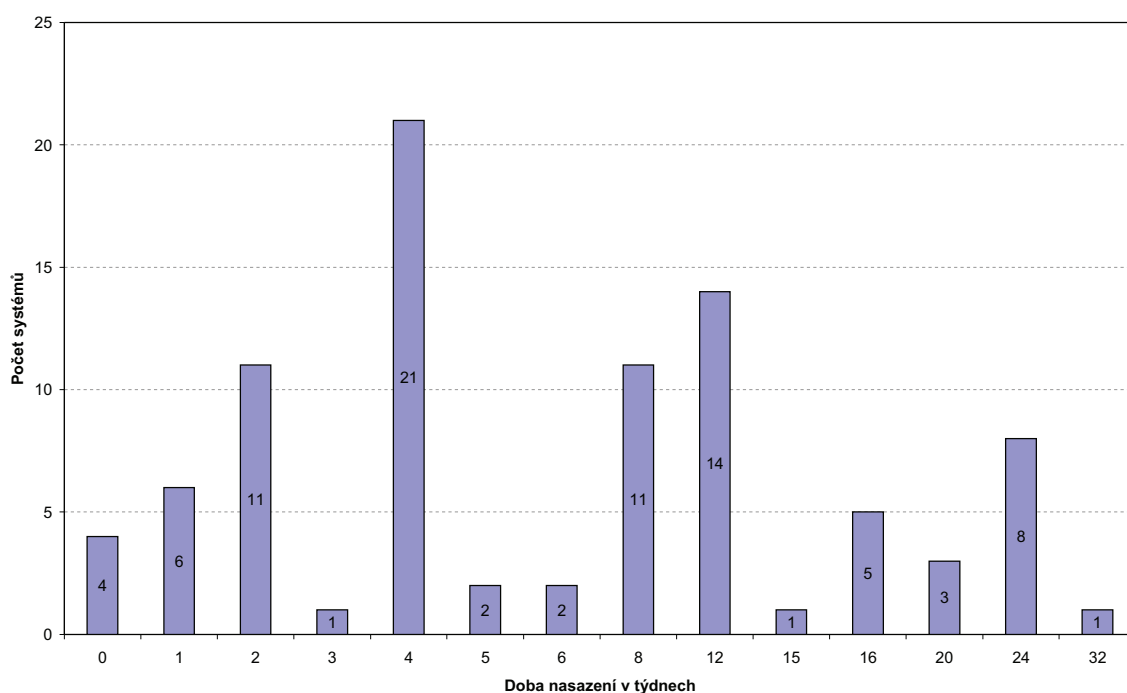
Ze srovnání vyplývá, že největší zájem mají výrobci CRM systémů o středně velké podniky (90 systémů), následují malé podniky (85 systémů), uzavírají podniky velké (67 systémů).

### Doba nasazení CRM systému

Graf 2 znázorňuje, jak rychle může být kolik CRM systémů nasazeno v podniku. Pokud položka „Průměrná doba implementace u podniku střední velikosti“ obsahuje interval namísto jednoho čísla, pak byla z intervalu použita nižší hodnota. Graf proto ukazuje rozdělení podle nejkratší doby implementace.

Vodorovná osa v grafu zobrazuje počet týdnů potřebných k nasazení. Pro její hodnoty (uvedeny jako horní řádek tabulky pod grafem) platí, že pokud jsou hodnoty větší než 6 týdnů, pak každé čtyři týdny odpovídají jednomu měsíci. Pokud nasazení systému může proběhnout do 3 dnů včetně, byl systém přiřazen do kategorie „nula týdnů“.

Svislý rozměr grafu (a dolní řádek tabulky pod grafem) uvádí celkový počet systémů, které lze v nejkratším možném čase nasadit právě za X týdnů.



**Graf 2** Rozdělení CRM systémů podle nejkratšího času potřebného k nasazení systému

Z údajů lze vyčíst, že nejběžnější doba nasazení současného CRM odpovídá jednomu měsíci a dále, že většinu systémů lze nasadit do jednoho roku (70 systémů).

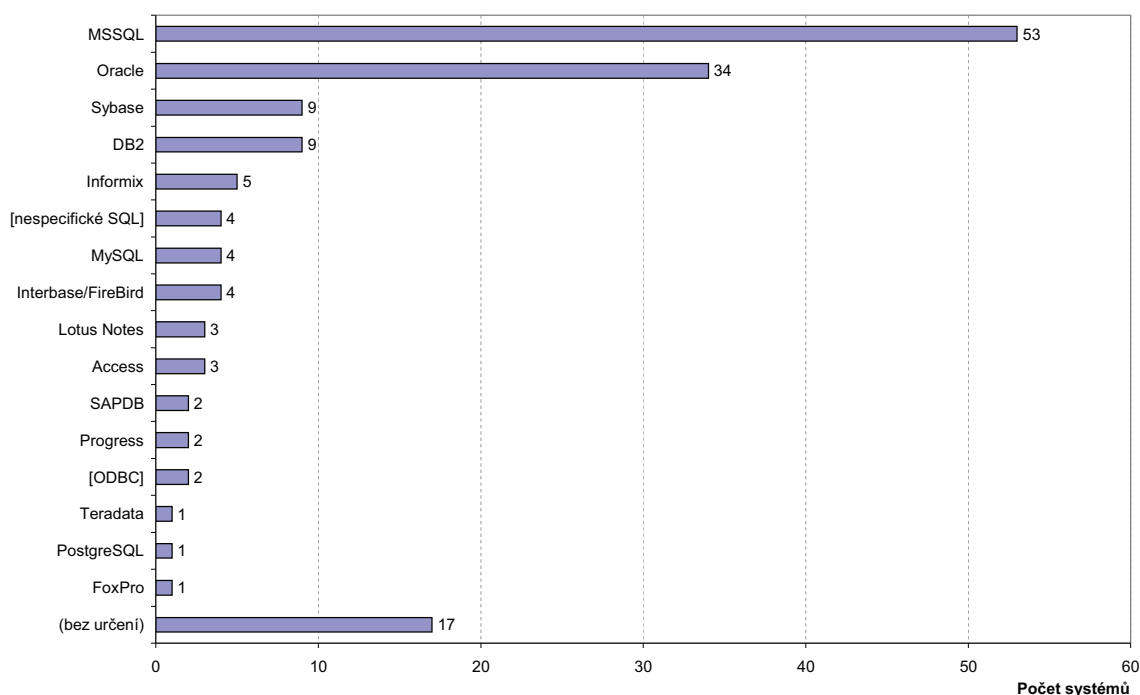
Zajímavý je fakt, že existují 4 systémy, které lze používat prakticky bez customizace (kategorie „nula týdnů“).<sup>10</sup>

Poznámka: Šest zkoumaných systémů nemá v přehledu uvedenou odhadovanou dobu nasazení, proto v grafu figurují pouze 9 systémů.

### Podporované datové sklady

Tato statistika ukazuje, které datové sklady jsou používány CRM systémy pro uložení dat včetně četnosti jejich použití.

Na svislé ose jsou vyneseny názvy databázových úložišť, vodorovná osa udává počet CRM systémů, které mohou dané úložiště používat (graf 3).



**Graf 3** Podporované datové sklady

Z grafu 3 je patrná jasná převaha MicrosoftSQL serveru (podporuje jej 53 systémů), následuje Oracle (34 systémů), s handicapem 20 systémů uzavírají rovnocenně Sybase a DB2 (9 systémů).

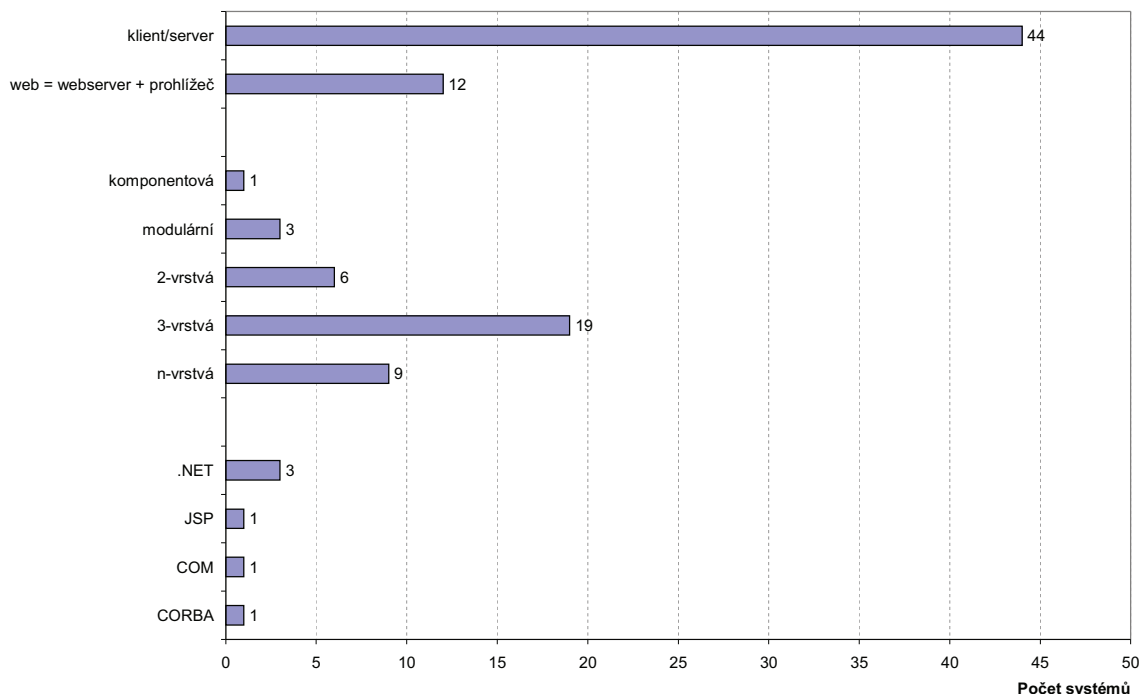
### Architektura systému, komunikační infrastruktura

Se zjištěním architektury systému se pojí dva problémy. Jednak chyběla potřebná data pro zhodnocení, protože respondenti odmítli zodpovědět příložené dotazy, a za druhé, způsob vyplnění položky „Architektura systému“ brání vynesení patřičného závěru. U většiny systémů totiž položka obsahuje hned několik údajů

<sup>10</sup> Produktům, které lze nainstalovat a používat bez další customizace se říká *krabicové produkty*.

ohledně architektury, přičemž některé údaje se vzájemně překrývají (častou odpovědí je „2-vrstvá/3-vrstvá, n-vrstvá“, zatímco jinde je uvedena pouze „n-vrstvá“.

I přesto výsledky uvádím (graf 4), protože alespoň rámcově ukazují, jaké architektury a technologie se používají v současných CRM systémech.



**Graf 4** CRM systémy podle architektury

## 4.5 Program pro práci s přehledem

Program *CRM Finder* umožňuje jednoduché vyhledání CRM systému zájemcem dle jeho konkrétních potřeb.

Otázka vyhledání vhodného CRM systému není jednoduchá. Pomineme-li, že zájemce by měl mít alespoň základní znalost problematiky CRM, ani toto nemusí být zárukou úspěšného výběru CRM systému. Už pouhý fakt, že existují společnosti, které se specializují na konzultační a poradenskou činnost v oblasti CRM systémů, poukazuje na složitost otázky výběru CRM systému. Proto si program neklade za cíl najít jeden nejvhodnější systém (zájemce častokrát ani nemá konkrétní představu, co požaduje – ta se formuje až při konzultacích s dodavatelem či poradcem), nýbrž zúžit širokou nabídku na systémy, které nejvíce odpovídají potřebám zájemce. Pro tento účel jsem navrhl a implementoval systém výběru se skórováním.

### 4.5.1 Funkce programu

Nezbytným krokem pro smysluplné vyhledání je dostatečné množství vyhledávacích informací. Velká část uživatelského rozhraní proto slouží k vytěžení informací od zájemce.

### Informace o zájemci

Na záložce *Kdo jsem* popisuje zájemce svoje vlastnosti. Vytváří tím základní filtr, kterým budou vyloučeny systémy určené pro podniky jiného zaměření či velikosti. Navíc může omezit i čas, do kdy by chtěl mít systém nasazen.

### Požadované funkce

Záložka *Požadované funkce* slouží k určení požadovaných funkcí. Zájemce je veden přes řadu položek funkcí, u kterých může zvolit důležitost dané funkce. Má na výběr z hodnot 0 až 10, kde nula označuje nepotřebnou funkci, zatímco desítka důležitou funkci. Použitím škály hodnot namísto prostého ano/ne tedy definoval e prioritou potřebnosti dané funkce. Seznam funkcí obsahuje pouze ty položky z přehledu CRM systémů, u kterých lze počítat skóre (viz dále). Vybrané funkce jsou seskupeny obdobně jako v přehledu CRM systémů.

Program umožňuje pojmenovat a uložit více sad priorit. Obsahuje předdefinované sady *Základní funkčnost* (nejdůležitější funkce CRM systému), *Dostupnost funkcí* (zobrazení funkcí systému bez prioritizace).

### Interpretace neznámých dat

Protože se nepodařilo získat všechna data do přehledu (někteří dodavatelé odmítli vyplnit kolonky v dotazníku či neodpověděli vůbec, případně dostupné informace si protirečily), existují v přehledu i místa bez hodnoty. Tento stav vadí zejména při vytváření skóre pro funkci potažmo celý systém, protože nelze říci, zda funkce v systému dostupná je či není. Obdobný problém se týká i funkcí, o kterých výrobce prohlásil, že se připravují do další verze. Namísto finální odpovědi „ano“ či „ne“ s kterým stavem vypořádat, obsahuje program záložku *Interpretace*, kde uživatel nastavuje, jak se má který stav interpretovat při vytváření skóre. Ve výchozím nastavení jsou neznámé hodnoty považovány za „ne“ (funkce/vlastnost není dostupná), hodnoty „připravuje se“ odpovídají částečné podpoře funkce/vlastnosti.

### Výsledky

Záložka *Výsledky* zobrazuje nalezené CRM systémy. Tabulka výsledků je aktualizována buď automaticky po změně kritérií nebo je možné ji aktualizovat manuálně.

Struktura tabulky odpovídá struktuře přehledu (včetně seskupení položek), skupiny je možné individuálně zobrazit či skrýt. Řádky odpovídají položkám přehledu, sloupce nalezeným CRM systémům. Systémy jsou řazeny sestupně dle dosaženého skóre, přičemž nejvhodnější systém (má nejvyšší skóre) je zobrazen vlevo. U položek, které má význam skórovat, je zobrazeno jejich skóre dle nastavení priority funkce. Skórované jsou všechny položky ze skupiny *Funkčnost systému a Další vlastnosti systému*. Pro snazší čitelnost výsledků jsou skórované položky podbarveny dle dosaženého skóre (barevná škála od bílé po sytě zelenou, zelená odpovídá nejvyššímu skóre v řádku).

## 4.5.2 Skórování

Pro každý systém je vytvářeno celkové skóre, jehož hodnota přímo úměrně odpovídá vhodnosti systému dle zadaných kritérií (záložka *Požadované funkce*). Celkové skóre je získáno jako součet hodnot dosažených skóre u skórováných položek.

Skórování probíhá u položek, na které bylo možné v dotazníku odpovědět ano/ne, resp. ano/ne/nevím/připravuje se, tedy položky ze skupiny *Funkčnost systému a Další vlastnosti*.

Mějme skórovanou položku s číslem 12. Její hodnotu pro pevně daný systém označme jako  $F_{12}$ . Hodnota  $F_{12}$  nabývá jedné ze čtyř možných hodnot (ano, ne, nevím, připravuje se) resp. dvou hodnot (ano, ne). Dále nechť priorita položky 12 má hodnotu  $P_{12}$ , která je z intervalu  $\langle 0, 10 \rangle$ . Pak funkční předpis pro získání skóre položky 12 ( $score_{12}$ ) lze vyjádřit vztahem

$$score_{12} := \text{Normalize}(F_{12}) * P_{12},$$

kde funkce  $\text{Normalize}()$  zajišťuje správnou interpretaci informace dle nastavení ze záložky *Interpretace*. Funkce  $\text{Normalize}()$  provádí transformaci vstupní 4-stavové informace takto:

ano	→	10
ne	→	0
nevím	→	UNKNOWN()
připravuje se	→	FUTURE()

přičemž funkce  $\text{UNKNOWN}()$  a  $\text{FUTURE}()$  vracejí hodnotu dle nastavení na záložce interpretace (ano → 10, částečně → 5, ne → 0).

### Poznámka

Původně byl plánován složitější systém skórování, který by dokázal navíc penalizovat systém, pokud by požadovanou funkci (tj. položka zkoumané funkce má hodnotu větší než 0) neměl. Protože by však bylo uživatelské rozhraní programu složitější (u položek by se musely zadávat dva údaje – jeden jako priorita, druhý jako preference) a přínos z řešení by nebyl tak velký, rozhodl jsem se pro jednodušší, výše zmíněnou variantu skórování.

## 4.6 Shrnutí

Jako součást diplomové práce jsem vytvořil přehled CRM systémů nabízených v ČR, i když se nepodařilo shromáždit všechny zamýšlené informace. Podařilo se shromáždit detailní informace o 96 CRM systémech. Přehled tak rozsahem i hloubkou překonal ostatní veřejně dostupné přehledy. V rámci průzkumu bylo dotazníkem obesláno 80 firem. Přestože firma byla obzvláště se skutečností, že údaje poskytují za účelem dosažení cílů diplomové práce a nejde tedy o možnost levně propagovat svůj produkt, poskytla více než ¼ dotázaných aktuální informace (26 kladných odpovědí + 1 záporná). Někteří respondenti uvedli, že odpověděli jen díky předvyplnění dotazníku relativně aktuálními údaji, které jim ušetřilo

cenný čas. K další spolupráci bylo ochotných 8 firem. Firma Net Genium dokonce Vě nabídnula licenci ke svému produktu pro studijní účely zdarma.

Při aktualizaci přehledu údaji z (4) byly zjištěny výrazné rozdíly oproti informacím získaným z emailové korespondence, byť časový rozdíl pořízení údajů byl menší než 1 měsíc. V případě produktu K2amitec bylo provedeno srovnání s informacemi na webu dodavatele, které potvrdilo platnost údajů z emailové korespondence<sup>11</sup>. Na základě tohoto zjištění byly při aktualizaci přehledu upřednostněny údaje z emailové korespondence.

Pro využití dat přehledu jsem sestavil program, kterým může zákazník vyhledat vhodný CRM systém podle svých potřeb. Program včetně přehledu v Excelu byl zaslán několika konzultačním společnostem specializujícím se na oblast CRM. Jedna společnost kladně ohodnotila vypracovaný přehled.

---

<sup>11</sup> Jednalo se o položku „Počet instalací v ČR“. Respondent v dotazníku uváděl více než 300, kdežto (4) pouze 11 – 25.

## Kapitola 5 CRM jako síť nezávislých aplikací

V předchozích kapitolách jsem vysvětlil, co je to CRM a jak vypadá nabídka CRM systémů v ČR. Zbývající část práce je věnována analýze možností použití sítě nezávislých aplikací v oblasti realizace CRM systémů.

Nejprve bude vysvětleno, co se pod pojmem *síť nezávislých aplikací* (SNA) skrývá a jak zapadá do kontextu implementace informačního systému.

Další část uvede do souvislostí myšlenky SNA s pokrokem v oblasti informačních technologií umožňujících realizovat informační systém či aplikaci formou SNA, shrne hlavní přednosti a nedostatky dostupných technik z pohledu SNA. Bude ukázáno, že architektura orientovaná na služby (SOA, *Service-oriented architecture*) je zatím posledním vývojovým stupněm na cestě k SNA. Samostatná část bude věnována dvěma implementacím SOA, tzv. podnikové sběrnici služeb (ESB, *Enterprise Service Bus*) a webovým službám (WS, *Web Services*).

Aby bylo možné analyzovat možné přínosy SNA při implementaci CRM, uvádím pohled do implementace CRM systémů různých dodavatelů, jakožto obraz stávajícího stavu. Prostřednictvím návrhu komplexního CRM diskutuji možné přínosy použití koncepce SNA pro stavbu CRM.

Pojem *síť nezávislých aplikací* (SNA) slouží k označení architektury informačního systému. Než vysvětlím, co pojem SNA přesně znamená, uvedu jeden z pohledů na informační systém, o který se bude popis opírat. Jedná se o strukturální pohled na IS.

### 5.1 Logická struktura informačního systému

Představme si obecný IS. Pro záměry práce přidejme požadavek, aby byl schopen komunikovat s okolním světem i bez přímé interakce s člověkem. Jak takový IS bude vypadat?

Rozhodně bude obsahovat výkonnou část, tj. aplikační logiku, která poskytuje funkce pro řešení problematiky doménové oblasti<sup>12</sup>, pro níž je určena. Aby bylo možné se systémem pracovat, musí obsahovat rozhraní pro vyvolání poskytovaných funkcí. Již první IS obsahovaly rozhraní pro obsluhu člověkem, tzv. UI (*User Interface*, uživatelské rozhraní). UI dnešních systémů je tvořeno vizuálními prvky,

---

<sup>12</sup> Doménovou oblastí míním určitou množinu činností se společným rysem. Hranici mezi doménovými oblastmi pak tvoří vzdálenost mezi rysy. O doménových oblastech se hovoří zejména ve spojení s lidským konáním, kde jedné doméně odpovídá obor specializace. Není přitom vyloučeno, že se doménové oblasti překrývají či dokonce vnořují – záleží jen míře specializace.

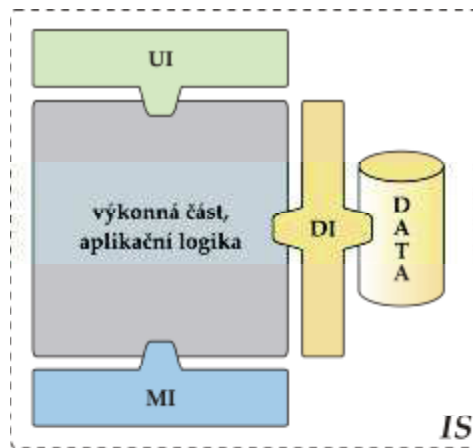
Příkladem doménových oblastí může být účetnictví, zpracování textů, ale stejně tak i kompletní ekonomická agenda z pohledu manažera.



kteřé lze ovládat pomocí klávesnice a myši (resp. jiného typu ukazovacího zařízení).

Mnohem později se začala objevovat rozhraní pro strojové použití systému. V této práci bude strojové rozhraní označováno jako MI (*Machine Interface*). Rozdíl mezi UI a MI spočívá především v jeho podobě. Zatímco UI je typicky grafické, s variabilním rozmístěním ovládacích prvků, MI má zcela odlišný charakter. Rozhraní je typizované, má pevně danou strukturu a přesná pravidla způsobu použití, což plně odpovídá potřebám strojového zpracování.

Poslední částí systému jsou data, nad kterými výkonná část operuje. Jsou-li data systému přístupná i mimo jejich mateřský systém a lze-li chování výkonné části ovlivnit skrz data nebo vyvolat funkci přímo změnou dat, lze datovou složku považovat za další druh rozhraní, tzv. datové rozhraní (DI, *Data Interface*). Smysluplný systém v minimální konfiguraci by měl obsahovat výkonnou část (provádí logiku systému) a alespoň jeden druh rozhraní (musí se určit, co, případně s čím, by měl operaci provádět).



**Obr. 3** Obecná struktura informačního systému

Pro označení softwarového díla se v praxi používá vedle pojmu *informační systém* pojem *aplikace*, resp. *softwarová aplikace*. Přesnou definici ani jednoho z pojmů se mi nepodařilo najít, proto se musím opřít o intuitivní výklad. Oba pojmy slouží k označení softwarového díla o němž lze prohlásit, že má strukturu uvedenou na obr. 3. Rozdílné je použití pojmů podle velikosti softwarového díla IS typicky označuje větší celek, který bývá složen z menších částí. Oproti tomu pojem aplikace se používá pro menší díla, která jsou dále nedělitelná. Vztah mezi pojmy IS a aplikace lze charakterizovat tak, že IS je složen z aplikací či jiných informačních systémů. Protože z pohledu této práce velikost díla nehraje významnou roli a z pohledu struktury označují oba pojmy totéž, jsou ve zbytku práce pojmy *informační systém* a *aplikace* považovány za ekvivalentní.



## 5.2 Způsoby implementace IS

Nyní se podívejme na IS z pohledu rozložení IS do funkčních celků. Rozdělení IS na celky je určeno architektem systému. Hovoříme o tzv. *architektuře IS*.

Dobře známá je architektura *monolitické aplikace*. Monolit je tvořen jedním spustitelným celkem, který obsahuje všechny potřebné funkce, tj. veškerou logiku, správu dat i rozhraní v sobě, nepotřebuje nic dalšího pro svou činnost. Pracuje typicky na jednom stroji, propojení s vnějším světem bývá malé až žádné.

Na potřebu sdílet informace, vysokou cenu výkonného HW a nové možnosti komunikace reagoval svět IT vytvořením *2-vrstvé* architektury. V době vzniku 2-vrstvé architektury se pro její označení používal i termín *klient/server*. Teprve s postupem času byl pojem klient/server specializován pro popis tzv. rozmístění výkonných celků („deployment“), kdežto „vrstevnatost aplikace“ se ujala pro popis logického členění systému. IS implementovaný 2-vrstevně sestává z vrstvy výkonné a prezentační. Výkonnou vrstvu realizuje typicky jeden aplikační server, ke kterému se připojuje řada klientů sloužících pro přístup k funkcím serveru. Podle množství funkcí implementovaných v klientovi se hovoří o *tenkém klientovi* (obsahuje pouze funkce pro vyvolání akce na serveru a funkce pro prezentaci výsledků obdržených od serveru) a *tlustém klientovi* (část funkcí logiky systému provádí sám klient s minimální interakcí se serverem). 2-vrstevná architektura tedy oddělila logiku systému a její prezentaci. Výhodou této architektury bylo vedle snížení nákladů na HW i zjednodušení správy systému prostřednictvím jednoho centrálního bodu – serveru.

V *3-vrstvé* architektuře dochází k oddělení logiky systému od datového úložiště. Použití odděleného úložiště přineslo několik výhod. Z pohledu implementace odpadla řada problémů spojených s daty (zajištění bezpečnosti dat, odolnost proti výpadkům, paralelní přístup k datům...), protože je na svá bedra převzalo samostatné datové úložiště realizované např. SQL databází. Databáze již měla dostatečně obecné a použitelné rozhraní, takže nebyl vážný problém ji pouze zapojit do systému. Vyčlenění datového úložiště zjednodušilo i činnosti jako zálohování a obnovu dat, které sice nejsou pro systém na první pohled zásadní, jak ale ukazuje praxe, v době potíží se systémem jejich význam exponenciálně roste. Dvousečně se jeví přínos v podobě datového rozhraní (viz kapitola 5.1 „Logická struktura informačního systému“), díky kterému lze sice propojit různé systémy, na druhé straně lze takto jednoduše narušit integritu dat a narušit tak celý systém.

Při dalším rozložení funkcí dle podobných rysů do vrstev hovoříme o *n-vrstvé* architektuře. Rozdělení bývá logické, tj. jde o vnitřní členění implementace, zvenku většinou skryté. Příkladem budiž členění na vrstvy perzistentní (stará se o načítání/ukládání dat), komunikační (propojení s ostatními částmi systému), obchodní/logickou (výkonné funkce). Konkrétní členění je odvislé od doménové oblasti IS. Použití vícevrstvé architektury zjednodušuje implementaci rozsáhlých systémů, na kterých se podílí řada vývojových týmů či dokonce externích dodavatelů.

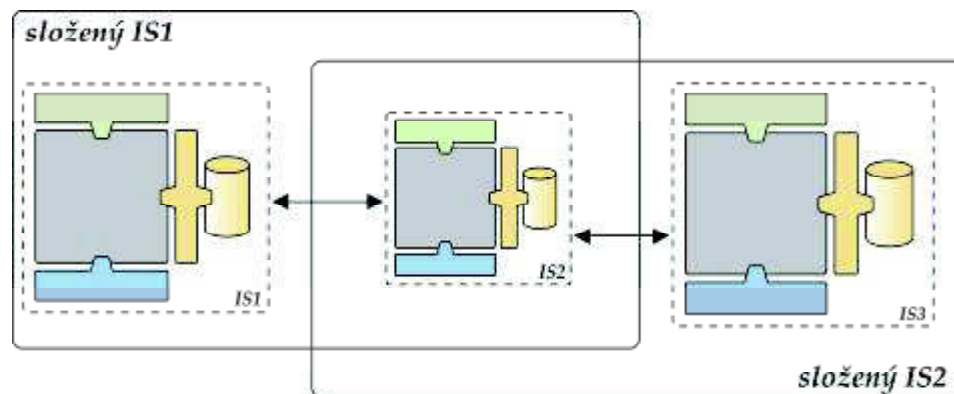
Takto lze rozpadnout velký, těžko uchopitelný problém na menší části, které lze realizovat snadněji a rychleji v menším vývojovém týmu.

Mladou architekturou je tzv. *kompozitní systém*. Jedná se o systém, který vznikl složením jiných informačních systémů do jednoho celku. Důvodem pro vznik kompozitních systémů je přirozený tlak ze strany uživatelů informačních systémů po integraci a propojení stávajících systémů. Existuje několik způsobů, jak integrovat systémy do větších celků, každý se svými pro a proti. V této práci bude diskutován přístup označovaný jako *síť nezávislých aplikací*.

### 5.3 Definice SNA

Stavebním prvkem *sítě nezávislých aplikací (SNA)* je nezávislá aplikace (NA). Důležitou charakteristikou NA je její autonomnost, tj. schopnost samostatně vykonávat funkci (na základě vstupů poskytnout adekvátní výstup), pro kterou byla navržena bez potřeby těsných vazeb na své okolí. Vezmeme-li např. monolitickou aplikaci či server z klient/server architektury, lze je označit jako nezávislé aplikace. Oproti tomu samotný klient nikdy nebude nezávislou aplikací, protože nedovede bez serveru vytvořit ze vstupů výstup. Propojením nezávislých aplikací vznikne síť nezávislých aplikací.

Vzhledem ke skutečnostem uvedeným v kapitole 5.1 „Logická struktura informačního systému“ se může stejně tak jednat o síť nezávislých IS. Strukturu systému sestaveného jako SNA ukazuje obr. 4.



Obr. 4 Kompozice informačních systémů

Jak obrázek naznačuje, je v obecném případě možné, aby jedna NA byla součástí více sítí nezávislých aplikací.

#### 5.3.1 Požadavky na NA

Interní struktura nezávislé aplikace nehraje při jejím použití v SNA významnou roli. Co je však důležité, je způsob, jakým je možné aplikace mezi sebou propojit, aby:

- vzájemně dokázaly vyvolat implementovanou funkčnost,

- dokázaly získat případné výsledky,
- uměly korektně interpretovat získané výsledky.

## Rozhraní

Vyvolání funkcí probíhá prostřednictvím rozhraní. Z kapitoly 5.1 „Logická struktura informačního systému“ vyplývá, že aplikace má tři typy rozhraní – uživatelské (UI), datové (DI) a strojové (MI). Podívejme se na vhodnost jednotlivých rozhraní pro vytváření vazeb v SNA.

*Uživatelské rozhraní* je určeno primárně pro uživatele, čemuž odpovídá jeho tvář. Je uzpůsobeno pro ovládání systému prostřednictvím klávesnice, případně ukazovacího zařízení. Nemalý důraz při definování UI je kladen na jeho líbivost a efektivitu použití člověkem. Byť se do návrhu UI promítají zjištěné návyky uživatelů, přesto nelze říci, že všechny systémy mají jednotný způsob ovládání. A to dokonce ani při srovnání dvou systémů určených pro stejnou doménovou oblast. Z tohoto pohledu se UI jeví jako velmi náhodně sestavené rozhraní. Navíc vizuální způsob reprezentace výsledků je vhodný pouze pro člověka, nikoliv stroj. Proto není UI pro propojení s jiným systémem příliš vhodné.

Pro *datové rozhraní* vyznívá bilance lépe<sup>13</sup>. Nezbytnou podmínkou pro použití DI je znalost struktury a významu dat poskytovaných přes DI. V dobách úsporného binárního uložení dat šlo DI používat jen s velkými komplikacemi. Dnes, kdy se data ukládají do databází s obecným standardizovaným rozhraním (SQL, ODBC...), je situace mnohem lepší. Díky obecnému rozhraní databáze lze informace z databáze číst a dokonce i zapisovat a upravovat. Ze struktury databáze lze často zjistit význam jednotlivých položek. Sice ani tak nelze v obecném případě vyvolat konkrétní funkci aplikace, přesto modifikace dat v databázi může vést k vyvolání funkce aplikace, je-li výkonná část funkce obsažena v databázi v podobě obslužné rutiny (tzv. *triggeru*).

Přesto však DI není vhodné pro propojení systémů v obecné rovině. Způsob uložení dat systému je závislý na konkrétním systému a dokonce v případě jednoho systému se většinou liší mezi svými verzemi. Navíc i zdánlivě nepatrná nekorektní změna dat může narušit činnost celého systému. Riziko narušení integrity dat systému je natolik závažné, že DI nebývá aplikacemi příliš vystavováno a v případě, že je, tak s patřičnou ochranou (uživatelská práva pro přístup k datům zvenčí) a dokumentací struktury dat včetně jejich sémantiky.

Existují však speciální situace, kdy je použití DI vhodné. Typickým případem je potřeba přenést větší objem informací z jednoho systému do jiného. Pro řešení se vžilo označení datová pumpa (*data pump*). Vlastní přenos dat probíhá mecha-

<sup>13</sup> Datovým rozhraním se míní možnost přistupovat k datům aplikace díky přístupnému rozhraní např. datového úložiště (SQL databáze, ODBC...), viz kapitola 5.1 „Logická struktura informačního systému“.

nizmem ETL (*Extract, Transform, Load*), tj. extrahují se data ze zdrojového systému, získaná data se transformují do formátu vhodného pro cílový systém a nakonec se nahrají do cílového systému.<sup>14</sup> Nutno upozornit, že častým důvodem použití datových pump je absence sofistikovanějšího řešení.

*Strojové rozhraní* je relativně mladý způsob, jak zpřístupnit funkčnost IS. Již při návrhu MI se počítá s tím, že MI bude sloužit především k použití jiným IS (přesněji strojem, na kterém jiný IS běží) a je k tomuto účelu proto patřičně uzpůsoben.

Pro MI je charakteristická absence grafických prvků známých z UI, nepracuje přímo s daty jako DI. Starší verze MI byla popsána ve vývojové dokumentaci systému (tj. pro programátora), což pro účely SNA není postačující, neboť textový popis není interpretovatelný strojem. Proto bylo potřeba najít vhodný způsob popisu MI, který je strojově zpracovatelný. Jazyk popisu musí být dostatečně obecný, aby dovedl popsat funkce nabízené systémem včetně popisu struktury výsledku<sup>15</sup>. Bylo by vhodné, kdyby popis definoval i způsob komunikace a sémantiku dostupných funkcí. Vezmeme-li v úvahu pokrok v oblasti umělé inteligence, bylo by s její pomocí možné, aby se na základě popisu systém automaticky propojil s jiným systémem jen na základě požadované sémantiky.<sup>16</sup>

Použití funkce prostřednictvím MI probíhá ve stylu „požadavek – odpověď“ nebo „příkaz“, není-li k požadavku generována odpověď.

### 5.3.2 Komunikace v SNA

Rozhraní nezávislých aplikací vytváří body, skrz které lze vytvořit propojení. Použijeme-li popis v terminologii komponentových systémů, rozhraní tvoří konektory. Konektor sám o sobě k vytvoření vazby nestačí, k tomu je zapotřebí vytvořit spoj mezi konektory, jehož prostřednictvím dochází k výměně informací, tj. vyvolání požadavku a doručení případné odpovědi.

Vedle klasických problémů, jako a/synchronnost komunikace, způsob vyvolání akce (posílání zpráv/vzdálená volání procedur), objevuje se u SNA nový fenomén, který je zapříčiněn různorodostí komunikace.

Mějme IS postavený jako síť  $N$  nezávislých aplikací. V obecném případě může komunikovat každá NA se všemi ostatními, tj. jedna NA může vytvářet až  $(N-1)$  různých způsobů komunikace. Můžeme předpokládat, že formát komunikace mezi každými dvěma částmi je shodný pro oba směry komunikace. Celkem tedy v rámci systému může probíhat až  $(N*(N-1))/2$  druhů komunikace, s rostoucím

<sup>14</sup> Jednu aplikaci uvádí kapitola 3.1 „Aplikační architektura.“

<sup>15</sup> Pro MI se též používá označení *deklarativní rozhraní*, neboť jeho popis jasně deklaruje dostupné funkce.

<sup>16</sup> O takovou funkčnost se pokouší například výzkumná skupina soustředěná kolem WSMO (*Web Service Modeling Ontology*) [56].

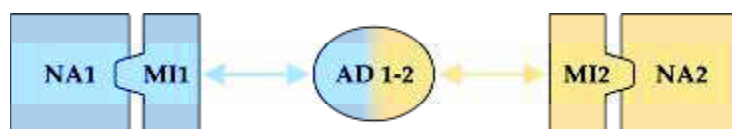
počtem komunikujících stran roste potenciální počet druhů komunikace exponenciálně. Polynomiální závislost složitosti komunikace na velikosti IS může být i při dnešních parametrech výpočetní techniky kritickou pro přenosovou infrastrukturu, a proto je potřeba ji řešit.

V principu se nabízejí tři přístupy, jak se s různorodostí komunikace vypořádat. Je důležité si uvědomit, že různorodost komunikace má několik příčin. Vedle známých komplikací způsobených *komunikačním modelem* (protokol, a/synchronnost volání, obsluha volání...) je potřeba řešit i otázku správné *interpretace předávaných informací*.

První přístup je triviální. Spočívá v prostém přidání požadavku na NA, aby všechny NA v SNA komunikovaly stejně, tj. měly shodný komunikační model i strukturu předávaných dat. Omezení je obtížně překonatelné u existujících aplikací, do jejichž MI nelze zasáhnout (např. z důvodu, kdy aplikaci poskytl externí dodavatel), u nově vznikajících systémů však lze s tímto omezením počítat již při návrhu MI a vyhnout se tak diskutovanému problému.

Druhý přístup by se dal nazvat přístupem mrtvého brouka, tj. ponechme nezávislé aplikace tak, jak jsou a propojme jen ty komunikačně kompatibilní. Byť se tento přístup může zdát na první pohled nesprávný, má své opodstatnění. Dnes, v době vzniku a implementace prvních servisních architektur (viz kapitola 7 „Servisní architektura“), se zvyšuje šance, že bude existovat NA, která řeší potřebnou doménovou oblast a přitom má vhodné komunikační rozhraní. Nelze na to sice spoléhat, pro menší IS však může tento přístup zjednodušit, tudíž zkrátit a zlevnit vývoj IS, což v oblasti obchodu stále platí za bernou minci.

Třetí přístup nabízí nejosifikovanější řešení. Ve z měme dvě NA s různým druhem MI, do jejichž struktury nemůžeme zasáhnout, a přesto je potřeba je vzájemně propojit. Řešením je vytvořit mezičlánek, který rozdíl v komunikaci překlene. Na výběr jsou dvě možnosti. V prvním případě se vytvoří jeden adaptér, který bude transformovat komunikaci jedné NA přímo na komunikaci druhé NA (obr. 5; „AD X-Y“ označuje adaptér provádějící transformaci mezi komunikací typu X a Y).



**Obr. 5** Přímá adaptace

Druhá možnost využívá místo jednoho adaptéru dva, u každé NA po jednom. Jeden pár adaptéru pak transformuje komunikaci z formátu jedné NA do formátu druhé NA prostřednictvím transformace na univerzální formát komunikace (obr. 6).

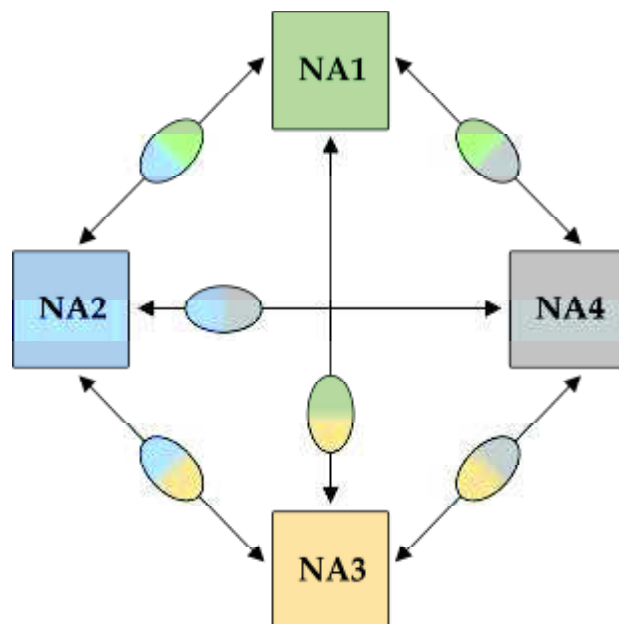




**Obr. 6** *Adaptace s využitím univerzálního meziformátu*

Výhodou řešení s jedním adaptérem na jednu komunikační osu je vyšší propustnost komunikace než v případě použití dvou adaptérů (transformace probíhá pouze jednou). Řešení s párem adaptérů začne nabývat na významu v okamžiku, kdy se SNA bude skládat z většího počtu NA. Představme si situaci, že máme síť  $N$  aplikací, přičemž každá komunikuje s každou a žádné dvě aplikace nemají kompatibilní komunikaci. Po  $N$  aplikací tedy existuje  $C(N,2)$  komunikačních spojů, které je nutné adaptovat, tedy SNA bude obsahovat při použití scénáře „jeden adaptér na jeden spoj“  $N!/(2*(N-2)!) = N*(N-1)/2$  adaptérů. Oproti tomu ve scénáři „jeden adaptér k jedné aplikaci“<sup>17</sup> stačí pro  $N$  komunikujících stran pouze  $N$  adaptérů. Rovnost v počtu potřebných adaptérů nastává již pro  $N = 3$ . Po  $N > 3$  je počet adaptérů ve druhém scénáři nižší, byť se na první pohled jevil lepší první scénář. Právě nižší počet potřebných adaptací ve druhém scénáři vedl ke vzniku architektury, které lze nazvat *sběrnicevé*.

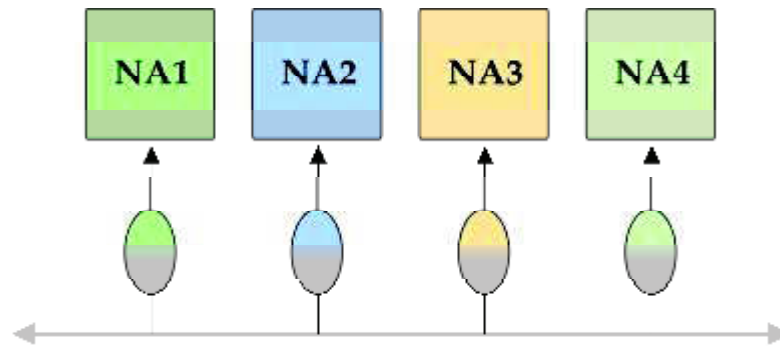
Situaci pro čtyři nezávislé aplikace propojené přímou adaptací ilustruje obr. 7, nepřímou adaptaci stejného počtu aplikací znázorňuje obr. 8<sup>18</sup>.



**Obr. 7** *Úplné propojení nezávislých aplikací přímou adaptací.*

<sup>17</sup> Jde pouze o jinou formulaci druhého případu „pár adaptérů pro jeden komunikační přes univerzální meziformát“.

<sup>18</sup> Z obr. 8 je patrné, proč je nepřímá adaptace spojována se sběrnicevou architekturou.



Obr. 8 Úplně propojení nezávislých aplikací nepřímou adaptací.

### 5.3.3 Vytváření a správa vazeb

Dosavadní pohled na SNA přes vlastnosti NA i komunikaci byl do jisté míry statický, tj. popisoval stav, jak SNA vypadá, když je sestavená. Obrázek by však nebyl úplný, kdyby postrádal popis, jak SNA vzniká.

Jak uvádí [28], existují obecně dva způsoby vytvoření SNA. Jedná se o tzv. *softwarové konfederace* a *softwarové aliance*. Rozdíl mezi konfederací a aliancí určuje míra předchozí dohody o vazbách mezi nezávislými aplikacemi před spuštěním SNA.

#### Softwarová aliance

Pro *softwarové aliance* je charakterické ad-hoc propojení, které se vytváří až za běhu systému. Před jeho spuštěním si partnerské komponenty nejsou známy.

Aby aliance mohla vzniknout, musí v systému existovat možnost vyhledat partnera s potřebnou funkcionalitou. Dále veškerá komunikace mezi partnery (včetně volacího modelu a formátu zpráv) musí být založená na shodném komunikačním formátu. Jakákoliv forma proprietární komunikace je nepřijatelná, protože partneři by se nedomluvili. Převáděno do praxe, komunikace by měla probíhat prostřednictvím obecně akceptovaného a podporovaného standardu.

Komponenty by měly pracovat ryze autonomně. Je žádoucí, aby byly nestavové<sup>19</sup>. Důvodem pro tento požadavek je ad-hoc propojení mezi komponentami. V alianci totiž nejenže není dopředu známo schéma propojení komponent, dokonce ani po vytvoření vazeb nemusí být použité komponenty *fixní*.

<sup>19</sup> Mějme komponentu  $K$ , která se stala součástí jedné SNA vzniklou ad-hoc propojením, označme ji  $SNA1$ . Protože  $K$  se stala součástí  $SNA1$ , není vyloučeno, že bude potřeba ji zapojit i do nějaké jiné SNA ( $SNA2$ ). Kdyby však  $K$  používala při generování výsledku stavové informace získané z předchozích výpočtů (řekněme z výpočtů v  $SNA1$ ), náhodně vložený výpočet pocházející z  $SNA2$  by byl ovlivněn výpočtem z jiné SNA (zpětně by dokonce ovlivnil výpočet v  $SNA1$ ), což je typicky nežádoucí.

Ze stejných důvodů usnadňuje bezstavovost komponenty její použití i v *softwarových konfederacích*.

Standardizovaná forma komunikace, autonomní charakter komponent a jejich bezstavovost předurčuje aliance ke strojovému zpracování. Tento názor podporuje i fakt, že vytvořit uživatelsky přívětivé rozhraní pro alianci je obtížné.

### **Softwarová konfederace**

*Softwarová konfederace* je pravým opakem softwarové aliance. Partneři tvořící konfederace se znají předem.

Dopředná znalost komunikujících stran přináší některé možnosti, které jsou u aliancí těžko realizovatelné. Standardní formát komunikace, který může být kvůli své univerzálnosti neefektivní, může být na potřebných místech nahrazen jinou formou komunikace. Lze tak optimalizovat komunikaci pro lepší přenosovou efektivitu, náročnost zpracování či vyladit spolupráci partnerů pro rychlejší odezvu. Dokonce je možné využít služeb jiné komunikační infrastruktury (např. jiného middleware), vhodnějšího pro danou oblast použití (např. reálného času či soft-reálného času zpracování). Jelikož je dopředu známa konfigurace systému, snáze se vytváří její UI.

Konfederace díky své povaze mohou vhodně řešit problematiku vytváření IS, jehož složky mají vynucená omezení použití (různý vlastník, citlivá data...). Vezmeme-li například IS přesahující hranici jednoho podniku či organizace (IS mezi odběratelem a dodavatelem, IS propojující zdravotnická střediska), můžeme úspěšně použít schéma softwarové konfederace k vytvoření nového systému. Poskytovatel (= vlastník) komponenty sám určí množinu služeb, kterou zpřístupní vnějšímu světu, případně může omezit, kdo smí kterou službu používat. Poskytovatel má tak plně ve své režii způsob použití svého systému. Lze tak zachovat vynucená omezení použití systému třetí stranou a přitom systém nezavřít před světem úplně. Jinými slovy, granularita přístupu k poskytovaným funkcím je říditelná.

Příkladem systému přesahujícím hranice více podniků jsou dodavatelské řetězce (*Supply Chain Management*, řízení dodavatelských řetězců) či vazby s partnerskými systémy.

### **Orchestrace**

S vytvářením vazeb v SNA souvisí i správa a řízení komunikačních kanálů mezi uzly sítě.

Vezměme si libovolnou SNA. Ta je určitě vytvořena pro vyřešení určitého problému, tj. pro daný vstup má vygenerovat patřičný výstup. Schématu, který určuje postup, jak výsledku dosáhnout, se říká *proces*. Proces v SNA odpovídá proceduře při klasickém procedurálním programování.

Procedura dostane na vstupu hodnoty, které během výpočtu transformuje podle různých okolností, a to jak vnitřních (daných hodnotami vstupu, interními proměnnými), tak vnějších (zjištěných z vnějšího prostředí, např. globálních proměnných). Řízení výpočtu sestává z klasických konstrukcí, jako větvení, skoky apod. V případě SNA je průběh vytváření výsledku analogický, jen okolnosti se liší od



těch u procesu – vnitřní okolností ovlivňující průběh výpočtu je stav NA, vnější okolností stav jiné NA<sup>20</sup>.

Proces, kterým SNA transformuje vstup(y) na výstup, musí být někde definován a následně někým řízen. Definií procesu jakožto volání rezaví slých aplikací, se říká *orchestrace*. Termín se někdy používá i v souvislosti s řízením procesu (vytvoření vazeb, obsluha volání), přesnější je označení vytvoření *instance orchestrace* či *orchestrování*.

## 5.4 Význam kompozitů

Kompozity nabízejí vlastnosti, kterých lze při použití jiné architektury dosáhnout jen obtížně či vůbec.

Aliance umožňují díky ad-hoc propojení *vysokou odolnost proti narušení*. V případě, že selže některá NA, může aliance dohledat jinou funkční službu s podobnou funkcionalitou (stejně jako při svém vytvoření) a transparentně ji použít. Samozřejmě pro zajištění transparentní záměny je potřeba další podpora ze strany komunikační infrastruktury, na které je SNA postaveno.

Konfederace bohužel nenabízejí tak vysokou odolnost proti výpadku, protože jsou těsněji provázány. Přípustná optimalizace komunikace odolnosti systému nepřispívá, neboť nastolený způsob komunikace snižuje šance nahradit nefunkční složku jinou. Na druhé straně díky optimalizaci lze *vylepšit kritické parametry komunikace*, jako rychlost odezvy, propustnost apod.

V obou modelech skládání systémů do větších celků umožňuje *získat novou hodnotu, která je v dílčích systémech nedostupná*. Příkladem budiž systém mateřské společnosti složený ze systémů dceřiných společností.

Proti použití kompozitních aplikací se staví nedostupnost opravdu funkčních vývojových nástrojů, ověřené technologie, chybí tzv. „best practices“ (vyzkoušená modelová řešení z praxe). Svou roli hraje i lidský faktor v podobě strachu o vlastní data, který vede k neochotě sdílet potenciál vlastního systému.

---

<sup>20</sup> Nezávislou aplikací může být i platforma, na které SNA pracuje, tj. operační systém, middleware apod., neboť splňují definií *NA*. p

## Kapitola 6 Technologie a SNA

Myšlenka vytvořit systém z menších funkčních celků není nová. Vezmeme-li např. princip objektově orientovaného programování (OOP), již u něj lze hovořit o jisté autonomnosti funkčních celků (objektů), které jsou spojovány dohromady, čímž vytváří systém analogickým způsobem jako SNA. Na rozdíl od SNA je propojení v rámci systému *fixní* . g

Přidání možností komunikace mezi implementačními entitami technologie (objekty, moduly, služby) a zajištění nových charakteristik komunikace, vedlo ke vzniku nových technologií pro vytváření distribuovaných systémů, čímž se dále přibližují k principům SNA.

Protože svět IT nabízí široké spektrum technologií, je zajímavé položit si otázku, do jaké míry která technologie usnadňuje tvorbu SNA?

Na základě popisu SNA uvedeného v kapitole 5.3 „Definice SNA“ jsem vybral několik vlastností, které tvorbu SNA podporují<sup>21</sup>. Seznam vlastností obsahuje jen ty nejdůležitější vlastnosti, zajímavé schopnosti technologií pro SNA jsou zmíněny později přímo u zkoumaných technologií.

Sledované vlastnosti:

- komunikace
  - zajistit přenos informace (bez ní by SNA nebyla možná)
  - formát přenášené informace – binární/univerzální (ovlivňuje snadnost tvorby adaptérů, viz 5.3.2 „Komunikace v SNA“)
- uzly = NA
  - řešení výpadku uzlu (jak se technologie vypořádá se situací, kdy cílová strana komunikačního spoje nežije)
  - vyhledání uzlu/registr uzlů (důležité pro aliance, použití i pro konfederace)
  - adresace uzlů (jsou komunikační spoje vytvářeny přímo, např. IP:port adresou či je možná virtualizace umístění uzlu)
  - podpora životního cyklu uzlu (aktivace/deaktivace, perzistence stavu)
- procesy
  - orchestrace (definice schématu procesu)
  - instance orchestrace (provedení schématu procesu)

---

<sup>21</sup> Seznam by mohl obsahovat řadu dalších sledovaných charakteristik, resp. detailnější rozpad. Cílem této kapitoly však není detailní rozbor nabízených technologií, nýbrž zmapování podpory SNA v existujících IT technologiích a směrech.

- správa a monitoring (možnost řídit SNA)

Průřez technologiemi postupuje odspoda vzhůru, tj. od přenosové vrstvy (komunikační protokoly) s postupným přidáváním dalších funkcí nad komunikační protokol tak, jak je se historicky objevovaly v nových technologiích (middleware), až po nový směr integračních technologií, které doplňují mozaiku middlewaru o další schopnosti.

## 6.1 Komunikační protokoly

Pokud se podíváme na potřeby SNA a pokusíme se je rozložit do vrstev, kdy horní vrstva potřebuje pro dosažení své funkčnosti spodní vrstvu, pak nejspodnější vrstvu reprezentují komunikační protokoly. Komunikační protokol zajišťuje základní přenos informace z jednoho místa do druhého. Pro něj nehraje roli, zdali jde o přenos souboru (např. FTP, NFS) či vzdálené volání (RPC, ORPC, GIOP/IIOP...). Stará se pouze o to, aby předanou informací korektně dopravil adresátovi.

### 6.1.1 Binární formát zpráv

Komunikační protokoly lze rozdělit na dva typy podle toho, jakým způsobem přenáší informaci. V dobách pomalejších síťových spojů převládaly protokoly přenášející data v úsporné *binární* formě. Díky tomu dosahovaly výborného poměru „objem přenášené informace“ ku „obsažená informace“.

Binární formát přenášených zpráv s sebou nese jeden problém, který se jeví pro budování SNA zásadním – binárnímu formátu není rozumět. Závažnost tohoto problému vyplývá z kapitoly 5.3.2 „Komunikace v SNA“. Aby si dvě NA porozuměly, tj. aby výstup funkce jedné NA mohl být vstupem druhé NA, musí být zajištěn nejen fyzický přenos informace, ale rovněž musí příjemce rozumět obdržené informaci po sémantické stránce.<sup>22</sup> Překlad informace má v SNA na starosti adaptér. Právě tvorba adaptéru je tím problémovým místem, které prakticky vylučuje použít komunikační protokol, který přenáší informaci v binárním formátu, tj. formátu zvenku nečitelném. Pro každý komunikační spoj by se totiž musel vytvářet nový adaptér ušitý na míru každé přenášené informaci, protože do obsahu přenášené informace není zvenku vidět. Navíc adaptér by musel být umístěn dovnitř aplikace (musí vědět, co se přenáší, aby mohl vhodně transformovat obsah

<sup>22</sup> Příklad pro ilustraci významu sémantik.

Aplikace A odesílá vizitku zákazníka. Pro jednoduchost nechť vizitka obsahuje jen 3 položky – křestní jméno, příjmení a email. Všechny 3 údaje jsou text. Příjemce rovněž očekává vizitku. Pomineme-li, že vizitka může obsahovat různé údaje (tj. odesílaná vizitka i vizitka očekávaná na straně příjemce obsahují sémanticky tutéž informaci), přesto není zárukou, že si komunikující aplikace porozumí. Stačí, aby vizitka odesílatele byla ve formátu „jméno;příjmení;email“ a očekávaná vizitka „příjmení;jméno;email“. Sémanticky jde o tutéž informaci, syntaxe se však liší.

Další rovina nesrozumitelností zpráv může pocházet z různého kódování obsahu zprávy.

zprávy – ten při napojení na komunikační protokol nezjistí), což taktéž není žádoucí<sup>23</sup>.

Komunikační protokoly s binárním formátem přenášených zpráv nejsou obecně vhodné pro tvorbu SNA. Jak z důvodu špatných možností adaptovat zprávy, tak kvůli své nízkoúrovňovosti postrádající dodatečné služby. Výjimkou jsou speciální situace, kdy potřeba rychlosti komunikace hraje dominantní roli a okolnosti umožňují binární komunikaci použít.

### 6.1.2 Strukturovaný obsah zpráv

Protokoly se strukturovaným obsahem zpráv lze rozdělit na dvě podskupiny – strukturované binární a strukturované textové. U těchto protokolů standard poskytuje prostředky pro definici formátu zprávy, které je možné získat informace potřebné pro sémanticky správnou interpretaci zprávy.

Příkladem protokolu se *strukturovaným binárním formátem zprávy* je protokol RPC (*Remote Procedure Call*, vzdálené volání procedur) [51], [53]. Pro něj existuje standardizovaný popis struktury označený jako XDR (*eXternal Data Representation*) [52]. Definiuje jednotný způsob reprezentace přenášených dat, rezavíslý architektuře odesílatele i příjemce. Součástí XDR je jazyk pro nezávislý popis přenášených dat. Zajímavostí u RPC v souvislosti s SNA jsou tzv. *XDR filter*<sup>24</sup>, které lze považovat za velmi primitivní formu adaptéru zabudovanou přímo v protokolu. Bohužel práce s formátem zprávy je těžkopádná (definice zprávy se de facto kompiluje do XDR filtru, který je pak potřeba programově zapojit do aplikace).

Nejslibnější z rodiny komunikačních protokolů pro oblast SNA se jeví protokoly, které přenášejí *zprávy ve strukturovaném textovém formátu*. Textová reprezentace informace přináší dobrou čitelnost pro člověka, strukturovanost informace výborné možnosti pro interpretaci obsahu zprávy. Zmíněné vlastnosti společně vytvářejí vhodné prostředí pro vytváření adaptérů převádějících obsah zprávy mezi odesílatelem a příjemcem při zachování stejného významu zprávy. Tento fakt umocňuje skutečnost, že byly definovány standardy a vyvíjeny nástroje speciálně pro práci se strukturovaným textem (viz 6.2 „Mediace“).

Snad všechny komunikační protokoly pracující se strukturovaným textem se opírají o XML. XML (*eXtended Markup Language*, rozšířený značkovací jazyk)<sup>25</sup> patří do rodiny SGML (*Standard Generalized Markup Language*, standardizovaný obecný

<sup>23</sup> Propojovanou aplikací totiž může být aplikace či systém, do jehož vnitřností není možné zasáhnout. Důvodů může být mnoho – nedostupnost zdrojového kódu, odmítavý postoj dodavatele aplikace, nemožnost pozastavit běh aplikace...

<sup>24</sup> *XDR filter* je rutina provádějící konverzi odchozích dat na straně odesílatele do binárního formátu a analogicky na straně příjemce zpět do formátu srozumitelného příjemci. Jde tedy o adaptaci z/do binárního formátu, nikoliv o sémantický převod.

<sup>25</sup> <http://www.w3.org/XML/>

značkovací jazyk). XML i SGML jsou tzv. „meta“ jazyky, protože slouží k definici značkovacích jazyků. Značkovací jazyk definovaný pomocí XML či SGML má specifický slovník (názyv elementů a atributů) a deklarovanou syntaxi (gramatiku definující hierarchii elementů a další vlastnosti) a [9]. Vhodným použitím XML lze dosáhnout stavu, kdy zprávy přenášené prostřednictvím definovaného značkovacího jazyka nesou vedle hodnotové informace (např. „12“, „Mirek“) i označení významu hodnoty (teplota, křestní jméno), tj. sémantiku.

Ačkoliv XML umožňuje vnést do zprávy sémantiku, neznamená to, že toho protokoly skutečně využívají. Příkladem protokolu s omezeným využitím XML je XML-RPC, oproti tomu SOAP těží maximálně snad ze všech schopností XML a souvisejících standardů.

### XML-RPC

XML-RPC [61] je specifikace a sada implementací, které umožňují software u běžícím na různých systémech a v různých prostředích volat procedury vzdáleně přes Internet. Jedná se o RPC mechanismus využívající protokol HTTP pro transport požadavků a odpovědí, pro kódování využívá XML. Jak uvádí jeden z autorů protokolu Dave Winner [60], XML-RPC byl uvolněn jako zjednodušení pracovní specifikace SOAPu. a

XML-RPC bylo navrženo jako nejjednodušší možné řešení umožňující přenos a zpracování komplexních datových struktur. Cíle návrhu XML-RPC lze shrnout do tří bodů [43]:

- umožnit vzdálená volání přes rozhraní počítačových sítí, tj. propustnost komunikace skrz ochranné síťové prvky bez dodatečné konfigurace (firewally),
- jednoduchost použití při zachování rozšiřitelnosti (celá specifikace včetně doplňujících informací se vejde na sedm stránek textu, rozšiřitelnost zaručuje XML),
- jednoduchost implementace do různých prostředí a operačních systémů (implementace HTTP i RPC jsou běžnou součástí většiny systémů, další potřebné nástroje jsou volně k dispozici).

XML-RPC definuje skalární datové typy (int, boolean, float, datetime, base64) a i jeden složený typ (struct).

Kladem XML-RPC je jeho rychlost (v porovnání s dále zmíněným SOAPem), nenáročnost na systémové prostředky, jak v průběhu de/kódování zprávy, tak i malé nároky na přenosovou kapacitu síťové infrastruktury. Nedosahuje sice přenosové efektivity protokolů s binárním formátem zpráv, vezmeme-li však v úvahu čitelnost přenášených dat strojem i člověkem, který je pro SNA důležitý, nejde o zásadní nedostatek. Dalším kladem je využití rozšířeného protokolu HTTP pro přenos zpráv. Jednak je tak zajištěna průchodnost komunikace síťovými firewally dále pro obsluhu požadavků lze použít většinu webových serverů s pomocí rozši-



řujících modulů, takže aplikace propojené do SNA pomocí XML-RPC mohou přesáhnout hranice chráněných sítí.

Záporem může XML-RPC (podobně jako u RPC) je jeho nízkoúrovňovost. Protokol nedefinuje žádné doplňující služby (např. transakční zpracování), což při orchestraci více aplikací v SNA, která se má provést buď celá nebo vůbec, bude problémem, který by se musel řešit externím mechanismem.

Jednoduchost zpráv XML-RPC má i své zápory. Ačkoliv má XML-RPC obsah zprávy uložen v XML a může jednoduše definovat složitější datové struktury, nelze pojmenovávat přenášené položky zprávy, čímž dochází z pohledu SNA k jeho degradaci téměř na úroveň protokolů se strukturovaným binárním uložením, protože zprávy neobsahují přímo ve zprávě sémantický rozměr. Ten sice lze určit z pořadí a typu hodnot ve zprávě v kombinaci se znalostí významu složek u odesílatele, pro tvorbu adaptérů do SNA však oproti přenosově efektivnější strukturovaným binárním zprávám nic nového nepřináší.

## SOAP

Dle specifikace [48] je SOAP „lehký protokol zamýšlený pro výměnu informací v necentralizovaném distribuovaném prostředí“. Nakolik lze s tímto tvrzením souhlasit je věcí názoru. Každopádně specifikace SOAPu se reje dě na sedm stránek textu jako v případě protokolu XML-RPC.

Protokol SOAP pokračuje tam, kde XML-RPC končí. Pokračuje ve využití XML technologií, s jejichž pomocí definuje rozšiřitelný framework zasílání zpráv nad rozmanitými nižšími transportními protokoly. Cílem návrhu protokolu byla jeho jednoduchost a rozšiřitelnost, dále nezávislost na programovacím modelu či prostředí. V rámci jednoduchosti protokol v základní podobě nezahrnuje vlastnosti jako spolehlivost, bezpečnost, souvztažnost/transakce či směrování komunikace známé např. z middleware.

K definici a přenosu zpráv používá ve všech možnostech jazyka XML také schémata nad XML, jmenné prostory a atributy. Obdobně jako XML-RPC definuje SOAP základní datové typy, navíc však poskytuje možnost prakticky neomezené definice vlastních datových typů. Každou formu dat zpráv, umožňující SOAP hodnotu pojmenovat prostřednictvím standardního XML atributu ID. Právě pomocí atributu ID doplňuje významnou informaci o sémantice předané hodnoty, kterou XML-RPC postrádá. Možnosti datových typů uzavírá otevřenost pro definici vlastních jednoduchých i komplexních datových typů. [36]

Užitek ze silných možností protokolu SOAP však není zadarmo. Za prvé, textová reprezentace zprávy je velmi neefektivní, což do poměru objemu přenesených dat ku velikosti informace. Např. pro přenos jednoho celého čísla (integer) potřebuje SOAP alespoň 266 znaků, zatímco XML-RPC pouze 176 znaků<sup>26</sup>. Za druhé, zpra-

<sup>26</sup> Výpočet zjištěn sestavením a srovnáním dvou zpráv. Do počtu znaků není započítáno kódování vlastního údaje (čísla), které je ve zmíněných případech totožné, velikost je počítána pouze jako



cování strukturované textové informace zabírá pochopitelně více procesorového času než v případě binárního formátu. Tento problém není tak podstatný na straně klienta/volajícího, jako spíše na straně serveru/volaného. Typicky totiž volající generuje jeden požadavek za jednotku času, zatímco volanému se za stejnou jednotku může sejít požadavků řádově více, čímž se vytváří potencionální úzké místo systému.

Užitečným rysem pro SNA je rozdělení zprávy na více oblastí, zde zmíním pouze obálku/hlavičku a tělo zprávy. Již na úrovni protokolu je možné do obálky zprávy začlenit prostřednictvím hlavičky doplňující údaje pro autorizace či transakční zpracování (jméno a heslo volajícího, identifikátor transakce, do které volání spadá) a mnoho dalších údajů, čímž SOAP předbíhá ostatní dosud zmíněné protokoly.

Byť defní SOAP u reur čuj e transpor tří protokol, jehož prostřednictvím se má jí zprávy přenášet, v praxi se nejvíce ujal protokol HTTP. Pro jeho použití hovoří snadná průchodnost síťovými firewally a jeho kryptovaná verze HTTPS transparentně doplňuje ochranu přenášené informace před vnějším znehodnocením a odposlechem.

### 6.1.3 Shrnutí komunikačních protokolů

Pro potřeby SNA jsou nejvhodnější protokoly přenášející informaci ve formě strukturovaného textu a obsahující sémantickou složku, což dnes prakticky umožňují protokoly budované s využitím XML. Jak bude ukázáno v kapitole „Mediace“, existuje řada nástrojů pro manipulaci s obsaženou informací, což zjednodušuje tvorbu adaptérů potřebných pro transformaci zpráv. Navíc použití čitelného textového formátu umožňuje začlenit adaptér na komunikační spoj vně aplikace, tedy do aplikace není potřeba zasahovat.

## 6.2 Mediace

V kontextu SNA míním mediaci<sup>27</sup> přizpůsobení NA tak, aby mohla tvořit společně s dalšími aplikací sítí NA.

V předchozí kapitole byla zmíněna jedna rovina mediace, a to tzv. *informační mediace*, kdy cílem mediace bylo zachovat sémantickou hodnotu informace při přenosu mezi dvěma nekompatibilními aplikacemi. Informační mediace se v současnosti provádí většinou prostřednictvím XML, pro které existují dostupné nástroje. Nástroje se opírají o rodinu specifikací XSL (*eXtended Stylesheet Language*), která obsahuje specifikace XSLT (*XSL Transformations*, jazyk pro transformaci XML), XPath (*XML Path Language*, jazyk pro tvorbu výrazů použitý v XSLT pro přístup či odkazování na části XML dokumentu) a XSL-FO (*XSL Formatting Objects*, XML slovník

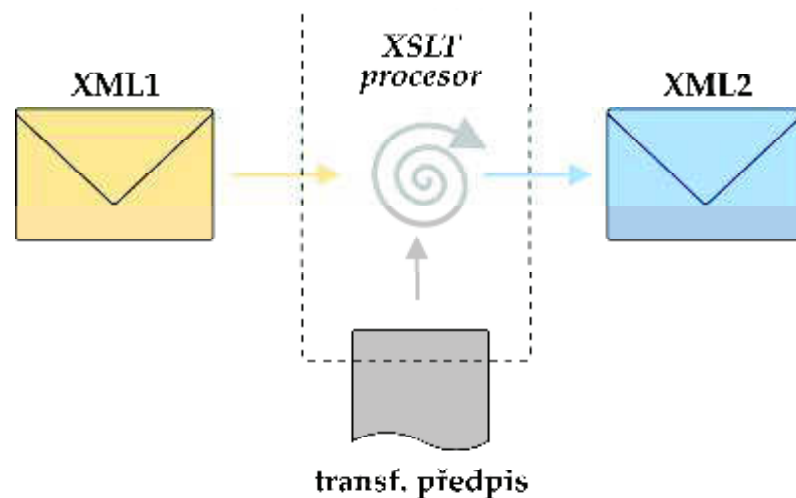
---

overhead protokolu. Při srovnání s binárním přenosem, kdy integer zabírá velikost 4 byty, a overhead binárních protokolů se vejde do desítek bytů, je navýšení rezie markantní.

<sup>27</sup> z anglického *mediate* = zprostředkovat, sjednat, smířit

pro specifikaci séna ntiky formá tu XSL-FO nemá pro informační mediaci význam). Jako náhrada za XPath se může použít XQuery (dotazovací jazyk pro získání informací z XML dokumentu). Všechny uvedené specifikace jsou dílem t frakce *XML Activity* spadající pod W3C. Až na mladší XQuery jsou uvedené specifikace ve verzi 2.0 a v přípravě pro tzv. „doporučení“ („Recommendation“, jedna z posledních fází schvalovacího procesu, kdy specifikace je již schválena), jedná se proto o reálně použitelné vyspělé specifikace. [35]

Dostupnost ověřeného a v praxi používaného standardu má velký význam, protože umožnila vytvořit dostatek spolehlivých nástrojů provádějících transformaci XML dokumentů, tj. informační mediaci. Nástroje se navíc vyskytují i ve formě implementačních balíčků, takže tvůrce adaptéru se musí postarat jen o vytvoření správného předpisu pro transformaci zprávy a napojení adaptéru na komunikační linku. Co víc, pomocí uvedených standardů a dostupných nástrojů lze pro XML komunikaci vytvořit univerzální adaptér, jehož strukturu ukazuje obr. 9 Daní za univerzálnost řešení zůstává procesorová náročnost provedení transformace, pramenící z použití textové reprezentace (její použití bylo zdůvodněno v kapitole 5.3.2 „Komunikace v SNA“).



**Obr. 9** Univerzální adaptér pro informační mediaci

Vedle mediace informační má význam hovořit i o *mediaci komunikační*. Ta má za úkol překlenout rozdíl mezi synchronním (blokujícím) a asynchronním (neblokujícím) voláním<sup>28</sup>. Například počítá-li volající s asynchronní obsluhou a volaný pracuje synchronně, dojde buď k posunu asynchronního modelu na synchronní nebo dokonce k selhání volání – záleží na způsobu implementace patřičných rozhraní<sup>29</sup>.

<sup>28</sup>Blokující/neblokující volání je zde chápáno z pohledu aplikace generující požadavek.

<sup>29</sup> Obdobná situace nastane při přístupu synchronní služby k asynchronní. Volající očekává, že bude pokračovat v činnosti až po vyřízení svého požadavku. Bohužel volaná asynchronní služba vrátí řízení volajícímu ihned, což může vést k závažným chybám systému jako celku.

V rámci mediace lze zabudovat mechanismus, který sníží počet případů, kdy rozdílná a/synchronnost rozhraní vede k chybám<sup>30</sup>. Aby však naznačený mechanismus plně ošetřil rozdílný způsob volání, musel by být adaptér provádějící komunikační mediaci zabudován přímo do aplikace. Pokud by totiž byl zavěšen až za generování volání (tj. mimo aplikaci) a aplikace počítala se synchronním voláním, stále by mohlo dojít k navrácení řízení aplikaci před tím, než bude požadavek vnější (asynchronní) aplikací skutečně vyřízen.

Z uvedeného je patrné, že ani komunikační mediace nemůže vyřešit problém s rozdílným způsobem volání, pokud není podpora pro mediaci zabudována přímo do aplikace.

## 6.3 Middleware

Middleware lze charakterizovat jako přenosovou infrastrukturu (analogická ke komunikačním protokolům), jež poskytuje další služby nad komunikací.

Po uvedení architektury klient/server do praxe byly rozpoznány problémy, které omezovaly možnosti komplexnosti, heterogenity, transparentnosti, rychlosti, kapacity, správy, vývoje, údržby a způsobovaly tak nežádoucí neefektivitu. Objevila se proto myšlenka řešit uvedené problémy vytvořením mezivrstvy mezi klientem a serverem. Mezivrstva dostala pojmenování middleware, k naznačeným problémům přistoupila zapouzdřením nižších infrastruktur (OS, síť, transportní protokoly) a poskytnutím služeb pro potřeby distribuovaných prostředí.

### 6.3.1 Synchronní middleware

Největšího rozšíření se dočkaly technologie CORBA (*Common Object Request Broker Architecture*, autorem OMG) a COM/DCOM (*Distributed Component Object Model*, Microsoft). Pro obě je charakteristická synchronní služba volání<sup>31</sup>, vlastní binární komunikační protokol (GIOP/IOP, ORPC) a řada speciálních služeb, z nichž důležitá je podpora transakcí, dále možnost generovat požadavky i na funkční entity, jejichž volací rozhraní nebylo v době kompilace volajícího definováno (*rDynamic Invocation Interface* u Corby, *QueryInterface* u DCOMu), registr instancí s možností vyhledávání („Zlaté stránky“, CORBA dokonce prostřednictvím *TradingService* dokáže vyhledávat instance v sémantických kategoriích). V čem CORBA předstihla DCOM, je virtualizace lokace instance (*IOR, Interoperable Object Reference*) a správa životního cyklu instance (aktivace, deaktivace, podpora pro perzistenci stavu instance<sup>32</sup>).

<sup>30</sup> Princip je založen na zpoždění (resp. včasné) odpovědi v případě asynchronní (resp. synchronní) služby, čímž umožňuje komunikace mít synchronní (resp. asynchronní) charakter.

<sup>31</sup> Později bylo doplněno i asynchronní volání.

<sup>32</sup> Díky zabudování podpory pro perzistenci stavu přímo do infrastruktury je možné vytvářet instance, jež se navenek tváří jako neustále dostupné, byť ve skutečnosti jsou aktivní třeba jen po dobu vyřízení požadavku. Při správné implementaci je možné požádat aktivní instanci o uložení

Při pohledu na sledovaná kritéria pro SNA (viz úvod kapitoly „Technologie a SNA“) splňuje synchronní middleware všechna kritéria s výjimkou podpory pro mediaci (kvůli binárnímu formátu komunikace) a orchestrace (žádná podpora). Z dosud jmenovaných se tedy jedná o technologii, která nejvíce odpovídá potřebám SNA.

Co však není vidět na první pohled, je praktická rovina. Praxe potvrdila, že CORBA i DCOM jsou skutečně mocné a komplexní nástroje podporující tvorbu distribuovaných systémů. Bohužel trpí i nedostatky, které jim brání v běžném užití. CORBA je velmi složitý systém (specifikace má stovky stran), existuje jen několik produkčně použitelných implementací (VisiBroker, Orbix...), přičemž cena kvalitních implementací odráží jejich výjimečnost. Vedle toho DCOM funguje jako černá krabice, která v ukázkových příkladech pracuje dle popisu, jakmile však dojde na situace, kdy je potřeba použít speciálních (zdokumentovaných) technik DCOMu, přestávají výsledky odpovídat korektním výstupům. Pak je obtížné rozpoznat, je-li chyba v aplikaci, systému či infrastruktuře DCOMu. Na samostatnou kapitolu by bylo pojednání o kompatibilitě jednotlivých verzí DCOMu, kdy novější verze není kompatibilní se starší. Co tento jev přináší do správy distribuovaného prostředí, jistě netřeba popisovat. Dvouznačně se u DCOMu jeví úzká provázanost s produkty autorské dílny Microsoftu.

Stručně řečeno, synchronní middleware má svými cíli velmi blízko ke splnění požadavků pro tvorbu SNA, reálné technologie však nikoliv.

### 6.3.2 Asynchronní middleware

Do kategorie asynchronního middleware spadají ty technologie, kde vyřízení požadavku (tj. generování požadavku volajícím, předání požadavku volanému, generování případné odpovědi volaným a její předání zpět volajícímu) probíhá asynchronně. Na rozdíl od synchronního middleware volající vygeneruje požadavek, předá jej infrastruktuře k vyřízení a pokračuje dále ve své činnosti, aniž by čekal na doručení odpovědi od volaného<sup>33</sup>. Protože obsluha požadavku odpovídá zaslání zpráv (požadavek, odpověď), ujal se pro asynchronní middleware ozna-

---

svého stavu (= perzistence) a následně ji „vypnout“. Po příchodu požadavku přes infrastrukturu je požadavek pozdržen, instance spuštěna, obnoven její stav a teprve pak je jí předán požadavek k vyřízení. Z pohledu volajícího i volaného jde tedy o plně transparentní záležitost, která umožňuje šetřit systémové zdroje. Za jistých okolností by se jí dalo použít i k migraci instance na jiné místo v distribuovaném prostředí (musela by být vhodně vyřešena adresace a směrování požadavků, což technologie nativně nepodporují).

<sup>33</sup> Je důležité si uvědomit rozdíl mezi operacemi „předat infrastruktuře požadavek k doručení“ a „čekat na obsluhu požadavku“. První operace je pochopitelně synchronní (volající musí vědět, zda se podařilo požadavek předat infrastruktuře, výsledek operace je dostupný okamžitě), kdežto teprve druhá operace je důvodem, proč je celá operace obsluhy požadavku označena za asynchronní (volající není pozastaven do doby doručení odpovědi k požadavku).

čení MOM (*Message-Oriented Middleware*, middleware pracující na principu zasílání zpráv) [32].

Asynchronnost přináší oproti synchronnímu middleware nové schopnosti. Již z definy asynchronnosti obchází vyk odbourání blokování klienta po dobu vyřizování jeho požadavku, klient se tedy může věnovat další činnosti, což může vést ke zvýšení jeho výkonu, protože může mezitím zpracovat jiné požadavky. Díky principu předávání požadavků a odpovědí formou zpráv umožňuje MOM tzv. *perzistence komunikace*<sup>34</sup> (perzistence komunikace vznikla jako vedlejší vlastnost zapříčiněná způsobem implementace asynchronní služby<sup>35</sup>). Částečnou perzistenci komunikace poskytovala i CORBA, jak uvádí poznámka 32.

Zprávy v MOM nemají z pohledu middleware definovanou svoj i strukturu. MOM chápe zprávu jako nedělitelný balíček informací, který má doručit. Struktura zprávy je proto plně v režii komunikujících stran, a lze proto využít poznatků uvedených v kapitole 6.2 „Mediace“ k vytvoření prostředí vhodného pro SNA.

MOM přináší novou schopnost i v oblasti adresace. Poskytuje základní způsob *přímé adresace*, kdy volající při odesílání zprávy určí adresáta. Princip interních front však dokáže zajistit, že adresát nemusí být pouze jeden. Použitím terminologie počítačových sítí, MOM dokáže vytvářet *broadcastové* spojení, přičemž lze dokonce vyčlenit pouze určitou skupinu adresátů, kterým je zpráva doručena<sup>36</sup>. V praxi se používají přístupy *publish-subscribe* a *topic-based*. Přístup *publish-subscribe* (vydavatel-odběratel) funguje stejně jako princip doručování časopisů do poštovních schránek, kdy si zájemce předplatí titul od vydavatele a ten mu jej bude na základě toho doručovat. Existuje tedy jeden vydavatel a více příjemců (vztah 1:N). V *topic-based* přístupu je odstraněno omezení na jednoho vydavatele. Místo vydavatele se zavádí tzv. téma (*topic*), ke kterému se zájemci registrují. Zprávy, které jsou označeny registrovaným tématem, jsou pak automaticky rozesílány všem odběratelům tématu, pod které zpráva spadá. Jedná se o vztah N:M<sup>37</sup>.

V tradičním MOM jsou zprávy adresovány svým příjemcům. Může být však výhodné (a SOA z toho i těží) rozvolnit vztah odesílatel – příjemce ještě více, dokonce do té míry, že jsou si komunikující strany vzájemně anonymní. Typickým

<sup>34</sup> Je-li komunikace perzistentní, je vhodným způsobem zajištěno uložení zprávy v případě nemožnosti doručit zprávu okamžitě. Doručení bude automaticky pokračovat po nastolení vhodných podmínek.

<sup>35</sup> Princip implementace asynchronního volání formou zpráv se opírá o interní fronty technologie, kdy odesílaná zpráva je po doplnění adresy cíle zařazena synchronně do fronty a teprve pak asynchronně technologií vyjmuta, doručena adresátovi a zařazena do jeho příchozí fronty. Je-li mechanismus ukládání zpráv do front zajištěn perzistentně (např. prostřednictvím databáze), pak komunikace má perzistentní charakter.

<sup>36</sup> Implementačně se nejedná o nic jiného, než že interní fronty jsou částečně vystaveny programátorovi, samozřejmě s potřebnou ochranou před nekorektním použitím.

<sup>37</sup> V případě *topic-based* přístupu musí být z těla zprávy oddělena položka označující téma, aby MOM dokázal se zprávou správně naložit.



způsobem, jak anonymity dosáhnout, je využít mechanismus vydavatel – odběratel (*publish – subscribe*), kdy vydavatel publikuje zprávy do sítě a MOM je doručuje jen těm odběratelům, kteří se u vydavatele zaregistrovali jako zájemci. V modelu vydavatel – odběratel je potřeba, aby transportní služba rozuměla obsahu zpráv. Jinou realizací jsou tzv. témata (*topic-based*), kdy transportní systém potřebuje přečíst pouze předmět zprávy a zbytek zprávy může ignorovat.

Příkladem MOM jsou MQSeries od IBM, MicrosoftMQ (dodáván s OS Windows), Sonic MQ, Fiorano MQ, JMS a další. Produkty bývají často součástí větších celků, kde plní roli chytrého komunikačního rozhraní, nad kterým jsou poskytovány další služby. Je možné, aby MQ pracovala samostatně bez přidáných služeb.

## 6.4 Integrační technologie

Integrační technologie jsou novým fenoménem reagujícím na potřebu propojit roztráštěné IS do většího celku. Jedná se o balík služeb, obsahující:

- přenosový protokol (i více), často zapouzdřený do middlewaru (novější obsahuje výhradně MOM),
- podporu pro začlenění zděděných i nových systémů prostřednictvím speciálních konektorů,
- jednotné rozhraní pro předávání informací (vychází z přenosového prostředí),
- podporu pro správu a monitoring připojených systémů,
- mechanismus pro řízení výměny informací mezi systémy (orchestrace).

Ne každý balík obsahuje všechnu zmíněnou funkčnost, míra podpory pro zmíněné funkce je též velmi odlišná.

Integrační technologie byly zpočátku určeny pro integraci existujících systémů. Cílem integračních technologií bylo propojit několik informačních systémů podniku do jednoho celku. Byla vyzkoušena řada postupů, počínaje propojením přes speciální konektory, adaptací stávajících systémů (dokódování synchronizačního rozhraní do systému) nebo datovou synchronizací. Každý přístup měl svá úskalí (absence konektoru, nemožnost zasáhnout do systému, žádná podpora od tvůrce systému, absence API či jiného vhodného rozhraní, nepřístupná či neexistující centrální datová úložiště), slíbené plné integrace dosáhl málokdy. Co se při integracích ukázalo jako zásadní problém, byla malá podpora integrace ze strany systémů. Starší systémy byly totiž navrženy jako uzavřené celky s žádnou nebo velmi omezenou možností propojení s jiným systémem. Na základě (nejen) tohoto zjištění vznikl nový způsob tvorby IS opírající se o tzv. servisní architekturu (SOA, *Service Oriented Architecture*), jejímu popisu je věnována samostatná kapitola 7 „Servisní architektura“. Protože novější integrační technologie podporují SOA (což do značné míry určuje jejich vlastnosti), srovnání integračních technologií a SNA je uvedeno v rámci odkázané kapitoly.



Z komerční produkce se ujaly integrační technologie například MS BizTalk, IBM WebSphere, Sonic ESB, Fiorano SOA Platform, vospělými zástupci z OpenSource iniciativy jsou Mule a OpenESB.

## Kapitola 7 Servisní architektura

*Servisní orientace* je přístup, kdy prostředky jsou jasně rozděleny a konzistentně reprezentovány. Aplikováno na IT architekturu, servisní orientace vytváří univerzální model, ve kterém se automatizační a dokonce i obchodní logika přizpůsobuje této vizi. Pojem „servisní orientace“ byl používán již nějaký čas v různých kontextech a odlišných významech. Společným rysem významů bylo jasné rozdělení zájmů či zodpovědností. Jinými slovy, logika potřebná k vyřešení velkého problému může být snáze sestavena, provedena a ovládána, pokud je rozdělena na soubor menších souvisejících částí. Každá z těchto částí řeší konkrétní zájem nebo specifikou část problému. [19]

*Architektura orientovaná na služby* (či *servisní architektura*, SOA, *Service Oriented Architecture*) je realizací servisní orientace v prostředí IT. Vlastní provedení se může lišit, konstantním však zůstává základní stavební prvek architektury – služba. Služba je uzavřená logická jednotka splňující následující charakteristiky [30], [19]:

- je *uzavřenou entitou* – zvenku se jeví jako černá skříňka provádějící definovanou činnost, její vnitřní mechanismus je skrytý před okolím (vlastnost bývá v označována i jako *abstrakce*)
- je *autonomní* – služba má kontrolu nad zapouzdřenou logikou,
- má jasně deklarované rozhraní svého použití,
- je *objevitelná* – služba je navržena tak, aby měla zvnějšku popisná rozhraní tak, aby ji bylo možné nalézt prostřednictvím dostupného vyhledávacího mechanismu,
- *nemá pevné vazby* – služba má jen tolik vztahů, kolik je nezbytně nutné pro udržení vzájemného povědomí o službách navzájem,
- je *znovupoužitelná* – logika je rozdělena do služeb se záměrem podporovat znovu použití,
- služby lze *skládat* – kolekce služeb může být sestavena do nové složené služby.

Jak uvádí [19], definice servisní architektury není jiným režimem návrhových principů, ze kterých lze vycházet při návrhu konkrétní implementační technologie, která dovolí skládat služby dohromady za účelem vytvoření automatizovaných servisně-orientovaných řešení.

### 7.1 Současné implementace SOA

První implementace SOA se datují do poloviny 90. let, kdy došlo k rozmachu integračních řešení určených pro podnik, tzv. EAI (*Enterprise Application Integration*, integrační podniková platforma). Nejednalo se o nativně servisní architekturu, při zpětném pohledu však bylo možné nalézt v EAI technologiích řadu rysů typických pro SOA, proto i tato řešení lze považovat přinejmenším za předchůdce SOA. Známa řešení jsou *IBM WebSphere* a *TIBCO Rendezvous*.

Na svou dobu se jednalo o velmi vyspělé systémy s vysokými integračními schopnostmi, v řadě podniků se používají dodnes. Problémem EAI technologií byla jejich proprietárnost, projevující se ve vzájemné nekompatibilitě produktů různých dodavatelů. Příčina překvapivě nespočívala na bedrech autorů produktů, nýbrž v nedostupnosti potřebných specifikací a standardů, které by definovaly i formát kontaktního rozhraní mezi technologiemi různých výrobců. Použitím EAI řešení se proto méně či více úspěšně vytvářely větší ostrovy softwarových produktů z menších. Až postupem času se objevily specifikace, které definovaly řadu atributů servisní architektury. [21]

Seznam charakteristik současné SOA uvádí [19]. Současná SOA:

- zvyšuje *kvalitu služeb* – Je způsobeno potřebou posunout SOA do stavu, kdy bude moci implementovat celopodnikovou funkčnost tak bezpečně a spolehlivě, jako zavedené distribuované architektury (CORBA, DCOM...).
- je zásadně *autonomní* – Princip servisní orientace vyžaduje, aby individuální služba byla nezávislá a uzavřená ve významu správy implementované logiky. Toho se dosahuje pomocí autonomie na úrovni zpráv, pomocí kterých služby komunikují. Předávané zprávy jsou dostatečně inteligentní, aby mohly ovlivnit způsob, jakým je zpráva zpracována na straně příjemce.
- je založena na *otevřených standardech* – Podpora otevřených standardů je nejdůležitějším rysem současné SOA. Výměna zpráv probíhá standardizovaným způsobem, nezávislým na prodejci. Vlastní obsah zprávy je též standardizován (SOAP, WSDL, XML, XML Schema), standardy dovolují vytvářet „inteligentní“ zprávy, čímž odpadají starosti platformě závislé starosti (jako typový systém) a podporují volné vazby mezi službami.
- podporuje *rozmanitost prodejců* – Otevřený komunikační framework (díky standardům) dokáže překlenout heterogenní prostředí uvnitř i mezi podniky. Organizace si tak může zvolit vhodné prostředí pro specifické aplikace.
- propaguje *objevitelnost*
- umožňuje *tvorbu federací* – Použití SOA v podniku nemusí nutně znamenat nahrazení stávajících řešení. Prostřednictvím široké palety adaptérů lze abstrahovat logiku existujících aplikací a systémů do služeb, kterých lze dále využít přes standardizovaný komunikační framework. Nabízí se tak možnost integrovat dříve izolované systémy do nových hybridních celků.
- podporuje *znovupoužitelnost* – Separace logiky do samostatné služby umožňuje její znovupoužití, aniž by v době implementace služby byla její znovupoužitelnost požadována.
- podporuje servisně-orientované paradigma *modelování obchodních procesů* – Služby lze vzájemně propojit do procesů realizujících obchodní logiku.
- stále *dospívá* – Standardy pro SOA nejsou dosud ustálené, vyvíjejí se, což se promítá i do nástrojů a technologií, které se o ně opírají.

Zmíním dvě implementace vycházející z principů SOA.

### 7.1.1 Webové služby

Historicky nejstarší úspěšnou implementací principů SOA jsou webové služby (*Web Services*, WS). WS jsou určeny k propojení obecných služeb hostovaných v prostředí Internetových technologií, zejména webu.

Standardy definující WS pocházejí z dílny organizace W3C<sup>38</sup>. První generace architektury WS vznikla na základě tří standardů opírajících se o XML. Pro popis struktury služby se používá jazyk WSDL (*WS Description Language*, obdoba IDL známého z DOT, kódovaný v XML), komunikace probíhá prostřednictvím protokolu SOAP (*Simple Object Access Protocol*, viz „SOAP“ na straně 42, typicky provozovaný na přenosovém protokolem HTTP, respektive HTTPS při potřebě zabezpečení informace při průchodu přenosovým kanálem). Informace o službách poskytuje služba prohlédávatelného registru, využívající mechanismus UDDI (*Universal Description, Discovery and Integration*), kterým lze registrovat i vyhledat konkrétní služby.<sup>39</sup>

Protože první verze byla příliš obecná, byla základní specifikace WS doplněna o WS-I BP (*WSI Basic Profile*) obsahující doporučení, jak a které konstrukty tří standardů používat, přičemž výběr podmnožiny funkcí původních specifikací byl prováděn s ohledem na nejpoužívanější postupy v praxi [4].

WS byla doplněna o řadu rozšíření přidávajících nové vlastnosti do schématu WS. Prostřednictvím rozšíření je možné zajistit v prostředí WS virtuální adresaci (*WS-Addressing* – definuje koncové body komunikace služeb pro použití SOAPem a virtualizuje konkrétní umístění), omezení komunikace (*WS-Policy*), spolehlivé doručování zpráv (*WS-Reliable Messaging*), orchestraci (*WS-BPEL*, *WS-Choreography*), souvztažnost mezi zprávami/transakční zpracování a řadu dalších<sup>40</sup>. Většina rozšíření je určena pro zajištění QoS v oblasti svého určení, čímž se WS otevřely i světu podnikových aplikací. [19]

Rozšíření WS jsou navržena modulárně. Máme-li dvě různé služby používající odlišné podmnožiny dostupných rozšíření, k tomu, aby se vzájemně domluvily, postačuje, podporují-li obě ta rozšíření, která vyžaduje partnerská služba.

Díky použití široce rozšířených internetových technologií, podpoře ze strany výrobců operačních systémů, webových serverů i programátorských nástrojů a jednoduchosti použití, se WS staly běžnou součástí nových softwarových produktů

<sup>38</sup> <http://www.w3.org/2002/ws/>

<sup>39</sup> Obdobnou koncepcí jako WS má REST (*Representational State Transfer*). REST vznikl jako reakce na „zbytečné“ funkce SOAP protokolu, které lze realizovat již na úrovni HTTP. Konkrétně jde o operátory PUT, GET, POST a DELETE, které sémanticky odpovídají databázovým operacím SELECT, INSERT, UPDATE, DELETE a lze je tak přímo použít například pro přístup k souborům dat (databáze, soubor...). [49] Vzhledem k nedostupnosti materiálů o REST (v porovnání s WS) se zdá, že REST byl slepou větví vývoje.

<sup>40</sup> Více lze najít na webu WS Interoperability Organization, <http://ws-ei.org>.

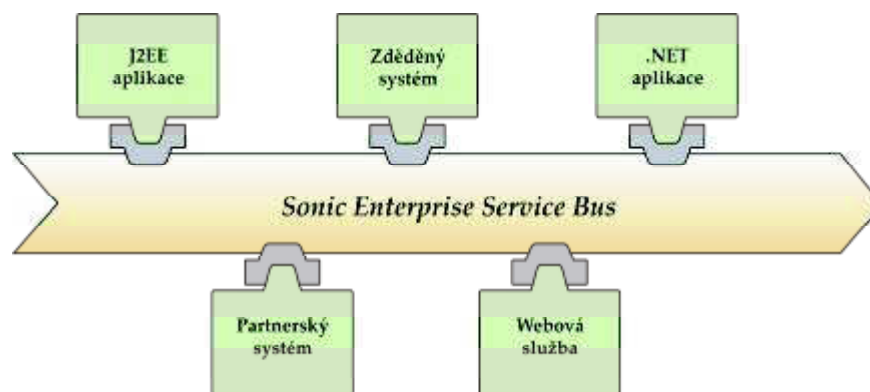
### 7.1.2 Podniková sběrnice služeb

*Enterprise Service Bus* (ESB, podniková sběrnice služeb) je koncept, který publikovala společnost *Sonic Software*<sup>41</sup> v roce 2001, implementace je dostupná od března 2002 [22]. Koncept ESB není novinkou, historie zná více obdobných řešení. V čem se však ESB liší od jiných řešení, je založení na otevřených standardech a obecně uznávaných technologiích.

ESB implementace používá [21]:

- standardizovanou komunikační infrastrukturu (JMS),
- standardizovaný způsob propojení (Web services, J2EE, .NET<sup>42</sup>),
- standardizované transformační jádro (XSLT, XQuery),
- standardizovaný způsob zabezpečení (SSL),
- princip SOA pro rozmístění a běh aplikací.

Popis vlastností *Sonic ESB* včetně obr. 10 vycházejí z materiálů firmy *Sonic Software* [50], [45].



**Obr. 10** Schéma *Sonic ESB*

ESB je softwarová infrastruktura. Jejím základem je sběrnice (odtud pochází část názvu), která zprostředkovává komunikaci hostovaných služeb. Umožňuje bezpečně propojit hostované služby<sup>43</sup>, překlenout komunikační rozdíly mezi službami a ovládat je.

<sup>41</sup> <http://www.sonicsoftware.com>

<sup>42</sup> J2EE (Java2 Enterprise Edition, Sun) a .NET (Microsoft) jsou dva dominantní frameworky pro distribuované výpočty. J2EE poskytuje přenositelnost aplikací díky jazyku Java napříč HW architekturami a operačními systémy. Naopak .NET podporuje širokou škálu jazyků, je však primárně svázan s operačním systémem Microsoft Windows a architekturu rodiny procesorů Intel.

<sup>43</sup> Dle Thomase a Daviese [54] se sběrnice chová jako asociovaná paměť sdílená neomezeným počtem procesů.

Vlastnosti komunikace vycházejí z osvědčené komunikační infrastruktury JMS (*Java Messaging System*), která je obohacena o další schopnosti. Vedle standardních vlastností jako spolehlivost, zabezpečení (SSL), podpora transakcí a perzistence, může poskytnout i garanci přenosových parametrů, případně zajistit nepřetržitý provoz (tzv. *high-availability*). JMS patří do MOM, podporuje klasický unicast, multicast i broadcast (*topic-based* a *publish-subscribe* schéma), dostupná jsou i real-time rozšíření.

Velkým krokem vpřed je zabudovaný framework pro správu sběrnice i hostovaných služeb. Framework usnadňuje aktualizace služeb (i po skupinách), sbírá a zprostředkovává výkonnostní statistiky infrastruktury, může automaticky provádět diagnostiku hostovaných služeb (logování, audit, selhání, aktuální stav). Nabízí tak kvalitní nástroj, který významně usnadňuje správu distribuovaného řešení<sup>44</sup>.

Pomocí ESB lze propojit širokou škálu systémů, služeb a aplikací (J2EE, .NET, webové služby, zděděné a partnerské systémy). Pro všechny hostované prvky platí jediný požadavek – poskytnout rozhraní pro zpřístupnění své funkcionality (tj. MI ve významu, jak pojem zavádí kapitola 5.1 „Logická struktura informačního systému“). Optimální je, používá-li komunikace XML formát dat (SOAP, XML-RPC), množství přizpůsobení před hostováním je pak minimální. Rozdíly ve způsobu komunikace jednotlivých služeb překlenuje vrstva mediátoru umístěná mezi hostovanou službu a služební sběrnici. Veškerá komunikace mezi službou a ostatními částmi systému je mediátorem oboustranně automaticky konvertována. Mediací tak stírá rozdíly nekompatibilních protokolů, datových formátů (transformace přes XSLT či vlastní řešení závislé na službě) a komunikačních modelů (včetně rozdílu a/synchronnosti) připojených služeb. Skutečnost, že mediací je definována v konfiguraci ESB (níkolis v kódu) a je vázána ke konkrétnímu hostu, dává možnost vytvářet univerzální migrovatelné<sup>45</sup> stavební bloky. Jde o jistou obdobu servantů (serverových implementací distribuovaných objektů) zmíněných v DOT, tentokrát ovšem na vyšší úrovni abstrakce, na úrovni služeb poskytovaných aplikacemi a IS.

Bohužel, stejně jako při boomu CRM, existuje řada firem, které svůj starý produkt označují jako ESB. Poznat kvalitní implementaci není snadné. Naštěstí existuje způsob, jak alespoň částečně odlišit marketingovou kampaň od reality. Na strán-

---

<sup>44</sup> Pro označení správy interakcí hostovaných služeb se vžilo označení *orchestrace* (z anglického *orchestrate*). I pro oblast orchestrace byly v nedávné době vyvinuty standardy, které dále posouvají možnosti ESB směrem k větší univerzálnosti (např. BPEL4WS).

Produktu *Sonic ESB* bývá vyčítáno, že místo podpory standardu používá proprietární řešení orchestrace. Překvapivě se kritiky opominají zmínit, že v době uvedení proprietárního řešení nebyl standard k dispozici.

<sup>45</sup> Schopnost migrace musí být podporována i hostovanou službou.



kách společnosti *Cape Clear* je dostupný *ESB Truth Test*<sup>46</sup>. Test obsahuje 10 otázek na klíčové vlastnosti ESB produktu i dodavatele, podle nichž provede vyhodnocení produktu. Formě dotazníku lze vytknout řadu věcí (např. množství reklamy dotazníku snižuje objektivitu výsledku), přesto dotazy s asociovanými odpověďmi cíleně vystihují stěžejní vlastnosti ESB a jsou vhodným místem, odkud zahájit zhodnocení konkrétního produktu.

## 7.2 Vztah SOA a SNA

Porovnání vlastností služby a nezávislé aplikace (jak ji popisuje kapitola 5.3 „Definice SNA“), ukazuje, že se jedná prakticky o ekvivalentní pojmy. Pojmy od sebe odlišuje pouze směr, odkud se k definici šlo SNA a definována jako logická jednotka, která řeší podproblém většího problému (dekompozice). Oproti tomu definice SNA postupoval a obráceně, tj. od nezávislé aplikace k SNA, které tak dovede řešit i problémy, na které uzly SNA nemohou dosáhnout (skládání). V pojmech servisní architektury odpovídá síti nezávislých aplikací služba vytvořená z jiných služeb.

---

<sup>46</sup> <http://www.esbtruthtest.com>

## Kapitola 8 CRM dnes

### 8.1 Současné způsoby implementace CRM

Abych mohl analyzovat možné přínosy koncepce SNA pro oblast CRM systémů, potřeboval jsem zjistit, jak se CRM systémy implementují dnes. Proto jsem při získávání dat do přehledu CRM systémů nabízených v ČR (viz kapitola 4 „Přehled CRM systémů“) přiložil čtyři doplňující otázky, které měly zjistit:

- jaká je implementační architektura systému (monolitická; modulární; soubor spolupracujících agentů či služeb; jiná)
- jakým způsobem probíhá výměna informací v rámci systému, o jakou technologii se opírá (proprietární; konkrétní middleware – DCOM, CORBA, MQ; služby – WS, ESB; jiná)
- schopnosti systému propojit se s jinými systémy
  - zda jsou jasně definována jejich rozhraní
  - použité komunikační technologie/protokoly
  - rozsah vystavených funkcionalit
- komplikace či zajímavá řešení, která se projevila při návrhu, implementaci nebo nasazení systému související s tématem práce<sup>47</sup>

Neočekával jsem, že by bylo více firem ochotných podělit se o informace, jež lze považovat za interní, vzhledem k obecnému charakteru dotazů mě však počet došlých reakcí nemile překvapil. Z celkového počtu 80 respondentů pouze 5 zodpovědělo přiložené dotazy, z nichž pouze jeden respondent reagoval na všechny 4 otázky<sup>48</sup>.

Díky statisticky neprůkaznému vzorku<sup>49</sup> nelze z došlých odpovědí vyvodit průkazné tvrzení o způsobu implementace CRM systému, přesto lze z došlých odpovědí (a ve spolupráci s daty přehledu) usoudit na jistý trend.

#### Implementační architektura

CRM systémy se staví typicky modulárně v třívrstevném modelu, přičemž CRM systém jako takový bývá sám modulem většího ERP (*Enterprise Resource Planning*) systému, který může být v případě požadavku zájemce od většího systému odseparován a dodán samostatně (obraz situace u větších systémů jako SAP, LCS Norris). Menší či „krabicové“ produkty (např. K2) bývají samostatným systémem.

<sup>47</sup> Stručný popis cílů práce byl součástí dotazů.

<sup>48</sup> Počet došlých odpovědí na dotazy (5) v porovnání s počtem vyplněných dotazníků pro přehled (26) mimo jiné potvrzuje významnost kroku *předvyplnění dotazníku*, který zkrátil čas zaměstnance firmy k jeho vyplnění.

<sup>49</sup> Dorazilo několik málo odpovědí.

Jako deploy architektura byla nejčastěji uváděna klient/server (viz kapitola 4.4 „Výsledky“).

### Vnitřní rozhraní

Ve způsobu výměny informací v rámci systému se mi nepodařilo najít převažující trend. Pro výměnu informací se používají snad všechny existující způsoby, počínaje přímou komunikací přes TCP-IP, RPC, přes middleware (zejména DCOM a CORBA), po řešení opírající se o .NET a EJB. Přístup formou služeb ani message-oriented middleware neuvedl ani jeden výrobce<sup>50</sup>.

Charakteristickým rysem vnitřní výměny informací je její proprietárnost, která brání přístupu k jednotlivým funkcím systému zvenčí.

Srovnáním data uvolnění první verze CRM systému a času dostupnosti technologií pro komunikaci lze usuzovat, že návrh vnitřní komunikace v systému vycházel z nejpokročilejší ověřené technologie své doby, přičemž na ní setrvává doposud.

Spekulovat lze o příčinách setrvačnosti v zažitých kolejkách, za které lze považovat vložené investice, odladěnost komunikační infrastruktury, ověřenou spolehlivost a potřebu podporovat dříve nasazené systémy.

### Vnější rozhraní

V případě vnějšího rozhraní se projevuje zájem výrobců o podporu servisně-orientovaného přístupu. Z pěti došlých odpovědí čtyři uvádějí, že pro vnější komunikaci poskytují definovaná rozhraní s výměnou informací v XML formátu (tři z nich prostřednictvím WebServices), pátý pro výměnu informací používá vlastní protokol. Tolik k možnostem propojení směrem ze systému ven.

Naopak při potřebě napojit vnější systém k sobě, bylo ve dvou případech zmíněno datové rozhraní, tedy přístup k datům vnějšího systému, nikoliv abstraktněji k informacím. Tento stav odpovídá současné situaci v podnikovém SW, kdy vnější systémy (často staršího data) neposkytovaly jinou možnost napojení, než přímo přes datové úložiště.<sup>51</sup>

Shrnuli získané informace, SOA je podporováno od úrovně podnikových systémů výš (podpora SOA je zpětně doplňována i do starších systémů), podsystémy jsou postaveny proprietárně.

<sup>50</sup> Zde je důležité poznamenat, že minimálně komunikace v rámci CRM a za MOM považují veřejně dostupný (koupitelný) SW, nikoliv proprietární řešení výrobce. Např. produkty SAP staví na vlastní architektuře *SAP NetWeaver*, která má jisté charakteristiky jak MOM, tak SOA. Zásadní však je, že *NetWeaver* je proprietární systém určený zejména pro propojení modulů ERP systému, nikoliv pro propojení částí uvnitř modulu. Komprimovaný výťah o technologii *SAP NetWeaver* uvádí „Příloha B“.

<sup>51</sup> Tento stav zjištěn při konzultacích s odborníky.

## 8.2 Reálný stav CRM v ČR

Ačkoliv při čtení kapitoly „Přehled CRM systémů“ může čtenář nabýt dojmu, že oblast CRM v ČR prosperuje a drží krok s vývojem CRM v západních zemích, ve skutečnosti tomu tak není. Přesněji řečeno, nabídka CRM systémů v ČR bohatá je a se zahraničím se může směle rovnat, jak dokazuje vypracovaný přehled CRM systémů. Co však přehled odráží jen částečně, je míra nasazení CRM v českých podnicích (položka „Počet instalací produktu v ČR“), o konkrétních problémech CRM v ČR mlčí úplně<sup>52</sup>.

Podařilo se mi získat zdroj, který mi poskytl dostatek informací z oblasti nasazení, vytváření a použití CRM systémů v praxi, čímž doplnil mozaiku o CRM z pohledu praxe. Byl jsem požádán o utajení identity zdroje. Z důvodu ochrany soukromí subjektů jsem záměrně zobecnil získané informace s cílem zachovat informační hodnotu důležitou pro tuto práci a zároveň ji odstínit od konkrétních subjektů.

Jsem si vědom skutečnosti, že informace pocházejí z jednoho zdroje, což obnáší subjektivnost plynoucí z jednostrannosti pohledu. Na druhé straně, zdrojem je několik pracovníků nadnárodní společnosti, zabývající se nasazením CRM řešení partnerů, jeho customizací, vývojem vlastních řešení i výzkumnou činností na poli IT. Díky bohatým praktickým zkušenostem lze tedy informace považovat za dostatečně kvalitní<sup>53</sup>, alespoň pro potřeby této práce.

Z diskuzí se zdrojem vyplynulo, že CRM systémy, jakožto systémy po zajištění spokojenosti zákazníka v takové podobě, jak je popisují v úvodní části této práce, v ČR prakticky neexistují. Tvrzení neznamena, že v ČR není podnik používající CRM systém (těch je celá řada, jak dokazují reference výrobců obsažené v samotném přehledu CRM). Tvrzením bylo míněno, že jen několik málo podniků skutečně má a používá CRM systém ke sledování a řízení spokojenosti zákazníka. Je prý patrná snaha podniků tento stav zlepšit.

Z předložených konkrétních příkladů jsem sestavil seznam příčin, které patrně vedly k současnému stavu s CRM systémy v ČR (dokládající příklady následují):

- vysoká cena CRM systému pro větší podnik
- špatná dostupnost hostingu (outsourcing CRM)
- licenční politika velkých výrobců
- český naturel
- vysoká roztržitost informací v podniku

---

<sup>52</sup> Což není chybou přehledu, neboť byl navržen s jiným záměrem.

<sup>53</sup> Pro použití informací hovoří i fakt, že o konkrétních problémech CRM systémů dodavatelé systému mlčí (byť se jedná o výzkumnou činnost, přesto by označení problematických partií jejich systémů pro ně bylo „negativní reklamou“, čemuž se samozřejmě každý výrobce snaží vyhnout).

## 8.2.1 Příklady

### Český naturel

Zákazník si chce pořídit vůz na leasing. Zajde do pobočky společnosti, kde sestaví smlouvu (dealer zkontroluje, zdali má zákazník v pořádku závazky vůči společnosti) a odjede v novém voze. Smlouva pak putuje do klientského centra, jehož prostřednictvím je sledována platební morálka zákazníka.

Kontrolní mechanismus však neřeší situaci, kdy vychytralý (a znalý) zákazník obejde v rychlém sledu více poboček, protože smlouvy jsou odesílány do centra až na konci pracovního dne. Proto se z pohledu společnosti jedná o bezproblémového zákazníka a bez obtíží jsou mu vystavovány další smlouvy.

Vzhledem k četnosti výskytu tohoto a podobných případů pak CRM systém, (jakožto rozhraní mezi zákazníkem a podnikem) slouží místo prostředku pro řízení spokojenosti zákazníka jako *ochranný prvek* před nekalým jednáním. Dochází tedy ke značnému posunu role CRM systému a je otázkou, zdali lze v uvedeném případě hovořit o CRM systému jako systému pro „podporu“ vztahu se zákazníkem.

Druhý příklad se týká CRM systému společnosti Siebel. Siebel se věnuje vývoji CRM systému již řadu let, čemuž odpovídá i velmi propracovaný produkt, poskytující podporu pro řadu odvětví. Ve specializovaných produktech je předdefinována celá škála obchodních procesů pro zajištění konzistentního přístupu k zákazníkovi (procesy při customizaci stačí za běžných okolností parametrizovat).

Při nasazení CRM systému do podniku v ČR bývá nutné předdefinované procesy místo pouhé parametrizace dokonce předdefinovat, aby nedocházelo k tak častému zneužití ochoty podniku zákazníkem.

### Cena rozsáhlejšího CRM

Tento příklad odráží reálné ceny produktu *mySAP* společnosti SAP. Jedná se o modul většího ERP systému, který může být použit i samostatně.

Cena jedné uživatelské licence vychází na cca 105 tis. Kč. Potřebuje-li podnik CRM pro deset uživatelů, budou ceny přibližně následující:

▪ 10 uživatelských licencí	1 mil. Kč
▪ produkční a testovací hardware	0,5 mil. Kč
▪ customizace/implementace <sup>54</sup>	1 mil. Kč
⇒ CELKEM	2,5 mil. Kč

<sup>54</sup> Cena customizace je určena koeficientem vzhledem k ceně uživatelské licence. SAP tvrdí, že koeficient odpovídá 0,7-násobku ceny uživatelské licence. Praxe prý ukazuje na koeficient mezi 1,1 až 1,4.

Vzhledem k obecnému povědomí o předraženosti produktů společnosti SAP je otázkou, zdali její CRM systém dostatečně reprezentuje cenovou politiku CRM systémů. Pokud by tomu tak nebylo a cena produktu skutečně přesahovala poskytovanou funkcionalitu v porovnání s konkurencí, nabízí se otázka, proč je o produkty SAPu stále zájem.

### **Cena za správu, outsourcing CRM a licenční politika**

CRM systém je potřeba po nasazení do podniku nadále přizpůsobovat (nové potřeby podniku) a spravovat. Používají se tři modely řešení podle toho, kdo je za operace zodpovědný (možné jsou i kombinace):

- a) vlastními silami = pracovník podniku
- b) dodavatelem systému u klienta
- c) formou outsourcingu, tj. externím specialistou, který se specializuje na úpravu<sup>55</sup> více systémů

Z pohledu nejnižších finančních nákladů i díky odezvy od požadavku na přizpůsobení do doby realizace je outsourcing vnímán jako zlatá střední cesta. Bohužel v ČR je outsourcovaných CRM systémů poskrovnu, úspěch slaví pouze společnost K2<sup>56</sup>.

Jak jsem vyrozuměl z konzultací s odborníky, za špatnou nabídkou outsourcingu je nezájem výrobců CRM systémů o outsourcing vlastních produktů. Bohužel outsourcing CRM systému třetí stranou (redistributorem) naráží na komplikace z oblasti licenční, finanční a administrativní.

Společnost, ve které jeden z konzultantů působí, uvažovala o hostingu CRM systému. Představa byla taková, že pro každého klienta by společnost poskytla customizovanou verzi CRM systému včetně HW, tedy přesně podle modelu uvedeného v poznámce 55. Vzhledem k faktu, že společnost nebyla výrobcem hostovaného systému, bylo nutné vyřešit pře prodej licencí. Protože cena za licenci pro jednoho uživatele se u větších systémů pohybuje řádově v desítkách tisíc korun (viz „Cena rozsáhlejšího CRM“, strana 59), varianta zakoupit  $x$  koncových licencí a přeprodávat je klientům nepřipadala v úvahu (multilicence či obdobný model výrobce nenabízel). Pak by bylo nutné licence obstarávat přímo u výrobce pro každého klienta, což by obnášelo značnou administrativní zátěž. Vhodný kompromis mezi dopředným nákupem licencí a nákupem až na základě zájmu klienta se nepodařilo nalézt.

<sup>55</sup> V praxi se v zahraničí osvědčil takový outsourcingový model CRM, kdy součástí dodávky CRM systému je i HW, který je fyzicky umístěn u hostitele, jež outsourcing poskytuje. Systém pak fakticky běží vně podniku, s ostatními podnikovými systémy je propojen přes počítačovou síť.

<sup>56</sup> [http://www.k2atmitec.cz/technologie/out\\_sour\\_cing\\_sht\\_m\\_D](http://www.k2atmitec.cz/technologie/out_sour_cing_sht_m_D)  
[http://www.k2atmitec.cz/technologie/technologie\\_outsourcing.shtml](http://www.k2atmitec.cz/technologie/technologie_outsourcing.shtml)



### **Datová roztržitost**

Pro optimální funkci CRM je potřeba, aby data o zákaznících (seznamy, historie kontaktu...) byla logicky centralizovaná, tj. tvořila jasně čitelnou a ucelenou informační základnu. Data o zákaznících v podniku bývají však rozdělena do řady menších podsystémů, což komplikuje nasazení CRM systému.

Roztržitost dat do menších systémů je odrazem nedostupnosti celopodnikového IS v době, kdy byl potřeba, rozdělení dat podpořil i rok 1990. Po revoluci se na trhu objevila řada nových produktů, které poskytovaly dosud nedostupné lákavé služby. V rámci zápalu polistopadového boje o silnější pozici na trhu a zvýšené důvěře ve splnění slibů (míněno slibů dodavatelů SW) se řada podniků rozhodla pro přechod na nový sofistikovanější systém. K jejich vlastní škodě často bez potřebného zvážení všech okolností. Výsledkem bylo, že nový systém sice přinesl řadu nových funkcí, leč často nedokázal plně nahradit původní, optimalizovaný a léty prověřený systém. V podniku se proto objevil nový systém. Namísto původního záměru sjednotit podnik prostřednictvím jednoho systému tak došlo k rozmělnění dat do dalšího systému.

Měl jsem možnost prostudovat dvě případové studie zabývající se vytvořením call-center pro bankovní sektor. V obou studiích bylo zapotřebí získat data o zákaznících z několika zděděných systémů, v jednom případě se jednalo dokonce o sladění 20 informačních subsystémů. Velká část studie se proto zabývala synchronizací dat mezi systémy.

## Kapitola 9 Možnosti SNA v oblasti CRM

Představme si CRM, které by pracovalo jako síť nezávislých aplikací, tj. takových aplikací, jak je definuje kapitola 5.3 „Definice SNA“. Vzhledem k ekvivalenci pojmů „nezávislá aplikace“ a „služba“ zdůvodněné v kapitole 7.2 „Vztah SOA a SNA“, budu pro označení nezávislé aplikace používat kratší označení služba.

Z kapitol „Evoluce CRM“ a „Architektura CRM systému“ vyplývá, které části musí CRM obsahovat. Protože pro zhodnocení přínosů SNA v oblasti CRM nebude potřeba rozebírat jednotlivé funkce CRM, bude skladba CRM systému definována na „meta“ úrovni funkčních celků, realizaci funkčních celků budu označovat jako (softwarové) moduly.

Skladba komplexního CRM (2.4.6 „Komplexní CRM“) by měla obsahovat následující prvky:

- Modul *kontaktního centra* (či *call-centra*, jak se kontaktnímu centru stále říká), které bude tvořit kontaktní rozhraní mezi podnikem a zákazníkem. Kontaktní centrum bude sdružovat všechny komunikační kanály, tj. telefon, fax, email, web, papírovou korespondenci. (3.2.3 „Kooperativní CRM“)
- *Datový sklad* pro uložení všech informací souvisejících se zákazníkem, jako jejich seznamy, historii kontaktu (nejen prostřednictvím komunikačních kanálů kontaktního centra, ale též prostřednictvím obchodníků<sup>57</sup>). Jak bude ukázáno dále, datový sklad nemusí být při realizaci CRM formou SNA fyzicky přítomen, respektive může mít jinou tvář, než jak je zažito pro pojem datový sklad.
- *Analytický modul* provádějící vyhodnocení dostupných dat o zákaznících a zpětnou projekci výsledků do systému. (viz 2.4.5 a 3.2.4)<sup>58</sup>
- *Operativní část* formou vhodného uživatelského rozhraní umožňující zaměstnancům podniku pracovat s CRM systémem. (viz 2.4.2, 3.2.2)
- *Definice pod množiny obchodních procesů (BP)*, které určují závislosti a posloupnosti operací se zákazníkem<sup>59</sup>.

---

<sup>57</sup> Na obchodníky lze nahlížet jako na speciální typ komunikačního kanálu, který se však principem svého fungování natolik odlišuje od komunikačních kanálů sdružených do kontaktního centra (vlastní přístup do systému → SFA, odlišné možnosti poskytovaných operací pro zákazníka, lidé nelze automatizovat a softwarově řídit), že není zvykem zahrnovat je do schématu kontaktního centra.

<sup>58</sup> Nepovinnou součástí analytického modulu může být funkčnost pro hodnocení přínosů CRM systému.

<sup>59</sup> Zákaznické BP jsou součástí operativního CRM. Pro potřeby zhodnocení přínosů SNA jsou záměrně z operativního CRM vyčleněny.

- *Propojení* na další vnitropodnikové a externí či partnerské informační systémy.
- *Administrativní modul* pro správu CRM systému (definice uživatelů systému, jejich rolí, zodpovědnosti, propojení modulů...).

Protože každý z uvedených prvků poskytuje/zajišťuje množinu funkcí, které mají volnou vazbu k ostatním prvkům, lze je realizovat formou služeb (resp. nezávislých aplikací, viz definice pojmu a v úvodu této kapitoly). Dnes již splnitelným předpokladem je standardizované rozhraní služeb (viz 7.1 „Současné implementace SOA“).

Protože CRM obsahuje citlivá data o zákaznících, obchodní procesy je potřeba navázat na konkrétní služby poskytované moduly, mělo by mít CRM konfederativní charakter, kde lze dopředu zajistit potřebné hodnoty. V případě softwarové distribuce by se například zbytečně zvyšovalo riziko úniku citlivých informací (služba by mohla pocházet z neověřeného zdroje a provádět nekalé praktiky).

Komunikace by měla z důvodů diskutovaných v kapitolách 6.1.3 a 6.3.2 probíhat formou výměny zpráv s textovým strukturovaným obsahem. Z dostupných technologií je nejvhodnějším kandidátem protokol SOAP pracující nad libovolným MOM, který podporuje transakční zpracování zpráv včetně perzistence komunikace.

Uvedený návrh CRM bude v dalším textu vystupovat jako „distribuované CRM“, zkráceně *dCRM*.

## 9.1 Přínosy SNA

Možné přínosy plynoucí z použití koncepce SNA rozdělují na *přínosy vnitřní* a *vnější* podle toho, týká-li se přínos vnitřních částí *dCRM* (moduly, mezimodulární komunikace) nebo vnějších vazeb *dCRM* k ostatním systémům.

U každého přínosu uvádím krátké zdůvodnění či odkaz, kde lze potřebné zdůvodnění přínosu koncepce SNA nalézt, není-li důvod triviálně zřejmý.

### 9.1.1 Vnitřní

Díky standardizovanému rozhraní je možné, aby tentýž modul poskytovalo více dodavatelů. Tím se otvírá prostor pro konkurenční boj mezi dodavateli, což by se mělo projevit *kvalitnějšími moduly* (špatné moduly konkurenční boj vytlačí z trhu) a *nižší cenou modulu* než v případě, že by modul poskytoval jediný dodavatel. Zároveň by se měla *snížit závislost na jednom dodavateli*.

Z pohledu implementátora (výrobce) modulu by měla uzavřenost modulu *zjednodušit* a *zrychlit vývoj* modulu. Jak známo z programátorské praxe, v menším programovém celku se snadněji hledají a odstraňují chyby, než ve velkém či těsně provázaném.

Řada výhod by měla plynout z možností orchestrace služeb, tj. způsobu vytváření procesů. Použití autonomních strojově ovladatelných prvků podporuje tvorbu *plně automatizovaných činností*. Při využití nástrojů pro vizuální návrh orchestračního předpisu<sup>60</sup> může uživatel systému *modifikovat a vytvářet procesy bez zásahu programátora*. Tím zároveň dochází k *přesunu zodpovědnosti za korektní funkci od programátora k uživateli systému*.

Volné vazby mezi moduly, jejich bezstavovost (viz poznámka 19 na straně 34) a perzistentní komunikace *zjednodušují aktualizaci modulů*. Pro nahrazení modulu (např. novou verzí) stačí pozdržet zprávy putující do modulu (zajistí perzistence komunikace v MOM) a po skončení všech aktivních operací v modulu modul nahradit. Je-li rozhraní nového modulu zpětně kompatibilní, pak po spuštění nového modulu MOM doručí pozdrženou zprávu, čímž se naváže na pozdrženou (nikoliv přerušenu) činnost. Pro nahrazení modulu tedy *není nutné pozastavit celý systém*.

### 9.1.2 Vnější

CRM vytvořené jako SNA obsahuje implicitně rozhraní pro strojové použití (MI, viz 5.1 „Logická struktura informačního systému“), které vznikne sloučením MI jednotlivých modulů, v případě potřeby je možné rozšířit či omezit složené rozhraní nadřazeným rozhraním<sup>61</sup>. Díky dostupnosti MI je *snazší integrovat dCRM do ostatních systémů podniku*.

Ze stejného důvodu by mělo dojít i ke *zlepšení B2B<sup>62</sup> a B2C<sup>63</sup> komunikace*, zejména k její automatizaci.

Dostupnost veřejného MI *odbourává potřebu přístupu k systému přes DI*. Odpadají tak problémy způsobené nekorektní změnou interních dat externím systémem. (problémy viz kapitola 5.3.1, odstavec „Rozhraní“).

### 9.1.3 Obecné

Koncepce SNA přináší *neměnnost komunikačních rozhraní*, jakožto důsledek informační mediace. Při potřebě modifikovat předávané informace není nutné měnit signaturu volání (jako např. v RPC, CORBA a dalších), nýbrž plně postačuje pozměnit obsah předávané informace. Proto není nutné upravovat<sup>64</sup> služby, jež modifikované rozhraní používá, stačí aktualizovat jejich modální předpisy.

<sup>60</sup> Nástroje jsou obsaženy v novějších verzích SOA implementací a vývojových nástrojích (např. MicrosoftB zřetel).e

<sup>61</sup> V případě potřeby omezit přístup k interním funkcím dCRM je nutné zajistit, aby vnitřní rozhraní nebylo vně dCRM přístupné.

<sup>62</sup> Business To Business – komunikace mezi podniky

<sup>63</sup> Business To Customer – komunikace mezi zákazníkem a podnikem

<sup>64</sup> V současném způsobu vývoje recompileovat.

Zajímavým jevem je posun *od dat směrem k informacím*. Zatímco většina dosud používaných systémů musí komunikovat prostřednictvím DI<sup>65</sup>, tj. musí se vyznat ve struktuře cizích dat (jejich syntaxi), při použití MI s textovým strukturovaným protokolem se sémantickou informací probíhá komunikace na úrovni jasně interpretovatelných informací, tj. sémanticky. Díky uzavřenosti zdroje informace se příjemce navíc nemusí zajímat, jakým způsobem zdroj k informaci došel. Představíme-li si nyní rozptýlenost dat o zákaznících v různých systémech (CRM, účetnictví, fakturace...), je možné vytvořit službu, která při požadavku na informaci o zákazníkovi zjistí dílčí informace z různých systémů, složí je do jedné komplexní informace a tu předá tazateli. Tímto způsobem je možné vytvořit *virtuální informační vrstvu*, která dovede v mnoha ohledech zastoupit centrální datový sklad<sup>66</sup>.

## 9.2 Problematická místa

Rozdělení systému na nezávislé aplikace s sebou přináší několik komplikací.

### Datová základna

Charakteristickým rysem komplexního CRM je orientace na data. Data musí být ve vhodné míře sdílená a pro některé operace i velmi rychle dostupná (např. viz „Magická obrazovka“ na straně 10). Distribuce však přináší přesně opačné vlastnosti.

Ukládání dat v dCRM lze realizovat dvěma hraničními způsoby, případně jejich kombinací – ukládat data v rámci modulu nebo využít službu centrálního datového úložiště<sup>67</sup>. Obě možnosti umožňují definovat vlastní strukturu dat modulu. Spornou je rychlost přístupu k datům v případě, že jeden fyzický stroj hostí více modulů, které si ukládají data ve vlastní režii.

Scénář *data u modulů* zvyšuje robustnost celého systému (vyplývá z autonomie služby, která není vázána na externí službu datového skladu), modul má plnou kontrolu nad daty, data jsou rychle dostupná pro lokální funkce, komplikuje se však vytváření analýz. Potřebuje-li služba data kontrolovaná jiným modulem, získává je použitím k tomu určené služby cizího modulu.

Scénář *data v centrálním datovém skladu* se na první pohled jeví velmi nebezpečným – při havárii služby realizující datové úložiště selžou všechny služby, které sklad používají. Vývoj v oblasti databázových systémů však pokročil do té míry, že totální výpadek služby nehrozí. Navíc ESB obsahuje mechanismus vysoké dostupnosti (*high-availability*). Nevýhodou centrálního skladu je náročnost na přenosové

<sup>65</sup> V době vývoje starších systémů neexistovala univerzální sofistikovanější řešení.

<sup>66</sup> Existují produkty, které virtualizují obdobným způsobem data z různých zdrojů. Příkladem buď *Lotus Integrator* či *IBM db2 Information Integrator*. Druhý jmenovaný dokáže extrahovat informace nejen z DB, ale též z XML, emailu a WS.

<sup>67</sup> Centrální datové úložiště může být standardní SQL server rozšířený o potřebné komunikační rozhraní.



cesty, naopak velkou výhodou je použití pokročilých databázových technik (*data mining*) a rychlé vytváření analýz.

V praxi budou používány pravděpodobně obě varianty. Data budou uložena v rámci modulu v případech, kdy je modul realizován zděděnou aplikací či informačním systémem. Naopak služba datového skladu bude upřednostňována pro uložení dat v rámci modulů tvořících společně jeden systém.

### Kapacita přenosové infrastruktury

S místem uložení dat se pojí otázka, zdali bude přenosová kapacita infrastruktury dostatečná. Význam otázky se dostává do popředí při využití funkcí náročných na množství zpracovávaných dat, např. při činnosti modulu analytického CRM. Zdali bude kapacita infrastruktury pro přenos dostatečná, lze zodpovědět rozbořem existujících softwarových architektur a srovnáním s novým modelem.

Nejpokročilejší implementace větších podnikových systémů obsahují speciální cluster realizující pouze služby datového skladu. Data takového systému jsou tedy umístěna v oddělených datových skladech a zpřístupněna aplikaci vhodným rozhraním. Funkčnost tohoto schématu byla již prověřena v praxi<sup>68</sup> a není proto důvodu se obávat, že by scénář *data v centrálním datovém skladu* selhal z důvodu nedostatečné přenosové kapacity.

V případě scénáře *data u modulů* není odpověď tak snadná, protože koncepce distribuovaných dat se dosud běžně nepoužívá. V situaci, kdy jsou data požadována jejich rodičovským modulem, jsou nároky na přenosovou infrastrukturu mírné (většinou aktualizace, vložení menšího objemu dat). Má-li však dojít k paralelnímu přenosu většího množství dat mezi moduly (např. analytický modul si pro potřeby vyhodnocení vyžádá data od účetního a obchodního modulu), je situace komplikovanější. Na rozdíl od datového skladu, kde lze mnoho operací s daty realizovat uvnitř služby a přenést pouze výsledek operace, zde musí minimálně jedenkrát dojít k přenosu dat mezi každým párem zdrojový – cílový modul. Počet opakování přenosu stejných dat závisí na inteligenci cílového modulu (stupeň cachování, chytrá replikace dat, předvypočítané parametrické modely) a částečně i na schopnostech zdrojových modulů zodpovědět dotaz adekvátním množstvím dat.

Jestli by použití scénáře „data u modulů“ znamenalo zvýšení zátěže přenosové infrastruktury, záleží proto na konkrétní skladbě modulů a jejich inteligenci.

S přenosovou kapacitou se pojí další efekt. V obou scénářích dochází k přenosům dat mezi moduly, rozdílné je časové rozložení přenosů. Zatímco pro scénář *data v centrálním datovém skladu* je charakteristické rovnoměrné rozložení datového přenosu (správa vlastních dat rodičovským modulem, analýzy prováděné v rámci

---

<sup>68</sup> Příkladem je jakýkoliv rozsáhlejší IS či aplikace zpracovávající data, která jsou uložena v datovém skladu umístěném na jiném stroji než logika aplikace (např. řešení dodávaná firmou SAP).<sup>4</sup>



datového skladu), použití scénáře *data u modulů* generuje datové přenosy nárazově (klid při práci modulů s vlastními daty, extrémní přenos při potřebě zpracování dat z více modulů). Amplituda extrémů může být zmenšena zabudovanou inteligencí modulů.

### Náročnost informační mediace

XML transformace jsou časově náročné operace, které probíhají při zpracování každé zprávy. Náročnost padá buď na bedra komunikační infrastruktury (tj. transformaci do cílového tvaru provede KIS automaticky po cestě) nebo se o ni stará až adaptér, kterým je připojena volaná aplikace. V obou případech však mediace spotřebuje stejné množství výkonu. Případné přetížení HW způsobené náročností informační mediace je možné řešit přesunem provádění mediace do KIS, které pak může zajistit její provedení na jiném fyzickém HW.

### Zabezpečení přístupu

Protože rozhraní modulu je zvenku viditelné a jeho struktura je založená na otevřeném standardu, existuje riziko, že se na rozhraní napojí nežádoucí element. CRM systém operuje s velmi citlivými daty, a protože potřeba zabezpečení informací roste přímo úměrně s jejich citlivostí, je nutnost ochrany přístupu k informacím přes rozhraní zásadní.

Proti odposlechům se lze bránit prostým zašifrováním komunikačních kanálů tak, jak se dnes běžně používá. O šifrování se může transparentně postarat KIS.

Závažnější je poznat, zdali volající strana je oprávněna vyvolat funkci rozhraní. Vzhledem k nespojovanému charakteru komunikace bude nutné ověřovat každou zprávu zvlášť, což s sebou nutně přináší další režii.

Ověřovací informace může být přenášena na dvou úrovních, ty lze vzájemně kombinovat:

- na úrovni KIS, kdy každá zpráva bude obsahovat označení, zda pochází z věrohodného/oprávněného zdroje; označení bude vně aplikační zprávy → validaci zprávy provádí KIS,
- na úrovni logiky procesu (např. v bankovníctví je potřeba ověřit, že bankovní operaci provádí oprávněný klient), kdy ověřovací informace je součástí aplikační zprávy → validaci informací provádí příjemce zprávy.

Samotný proces ověření může probíhat *centralizovaně* (jedna autorita, obdoba dnešních autorizačních serverů v bankovníctví) či *decentralizovaně* (každá entita musí obsahovat vlastní mechanismus pro zjištění, zdali volající splňuje kritéria pro přístup k funkci). Protože proces ověřování může být náročný výpočetně (kryptování) i z pohledu přenosových cest (ověření každé zprávy u centrální autority), je vhodné proces ověření zprávy optimalizovat.

Nejschůdnější variantou se jeví delegovat problém ověřování na KIS, kdy k ověření věrohodnosti zprávy a oprávnění odesílatele zprávy by docházelo pouze při vstupu primární zprávy do KIS a KIS by garantovala, že zpráva bude beze

změn doručena volajícím<sup>69</sup>. Pokud by navíc bylo zajištěno, že orchestrační schémata jsou korektní a nemohou vést k úniku informací, pak by stačilo, aby po dobu zpracování orchestrace byly všechny zprávy v rámci instance orchestrace validovány oproti unikátní identifikaci orchestrace, což lze realizovat příznakem příslušnosti zprávy k instanci orchestrace<sup>70</sup>. Zkontrolování příznaku při zajištění uvedených podmínek je pak jednoduchou časově nenáročnou operací.

### 9.3 Realizace přes ESB

Pokud by jako KIS byla pro dCRM použita podniková sběrnice služeb (viz 7.1.2), dCRM by transparentně získalo nové vlastnosti plynoucí z vlastností ESB:

- nativní podporu pro redundanci, high-availability
- garantovaný čas doručení zprávy, realtime komunikaci
- podporu pro správu a monitoring služeb
- vizuální orchestrační nástroje pro vytváření business procesů umožňující vytvářet kompozitní služby

### 9.4 Možnosti SNA pro řešení aktuálních problémů CRM

SNA bohužel nemůže plně vyřešit všechny současné problémy uvedené v kapitole 8.2, může však napomoci s jejich řešením.

Snížování ceny modulů CRM (viz 9.1.1) by se mělo projevit snížením ceny celého CRM. Zde však hodně záleží na ceně KIS<sup>71</sup>.

Proti českému naturelu lze bojovat vybudováním „kontrolního rozhraní“ nad rozhraním realizujícím přímou funkci, které před vyvolání operace nejprve zkontroluje korektnost požadavku. Ochrana proti případu z leasingové společnosti (strana 59) by měl napomáhat asynchronní on-line charakter vazeb mezi subjekty, tedy včasná dostupnost kontrolních informací.

Datovou roztržitost může SNA řešit vytvořením virtuální informační vrstvy (VIV, viz 9.1.3), která bude vytvářet sloučenou informaci kompozicí dat z jednotlivých subsystémů. Pro vybudování VIV je však nezbytné, aby do subsystémů bylo doplněno komunikační rozhraní dodávající potřebné informace. Z konzultace s odborníkem v oblasti integrace systémů bohužel vyplynulo, že šance doplnit do starších systémů jakékoliv rozhraní je mizivá. Jediným dostatečně pružným způsobem je použít DI, tj. přímý přístup do databáze systému.

---

<sup>69</sup> V tomto případě by KIS musela obsahovat podporu pro validaci zpráv.

<sup>70</sup> Zde je nutný požadavek, aby komunikace uvnitř KIS byla odolná proti vnějšímu napadení.

<sup>71</sup> Při pokusu zjistit cenu technologie *Sonic ESB* jsem se pouze dozvěděl, že „cena ESB odpovídá 40% ceně konkurenčního produktu, přičemž poskytuje 80% funkcí konkurenčního produktu“. U konkurence bylo určení ceny obdobné.

## Kapitola 10 Závěr

V úvodu práce je konkretizován pojem „CRM“ jakožto podniková filozofie a strategie zaměřená na získání a udržení nejhodnotnějších vztahů se zákazníky. Oproti tomu pojem „CRM systém“ je vymezen jako technologický prostředek napomáhající dosažení cílů firmí strategie. Syntézou zdrojů a studiem CRM systémů jsem i vytvořil pohled do nitra problematiky CRM systémů, který definoval CRM dále prohlubuje. (kapitoly 2 a 3)

Informačně obsáhlou částí práce je detailní přehled CRM systémů nabízených v ČR, který rozsahem (96 systémů) a granularitou převyšuje obdobné veřejně dostupné studie. Přehled obsahuje podrobné informace o poskytovaných funkcích, uskutečněných nasazeních, zázemí společnosti i přibližný odhad doby potřebné k implementaci CRM systému. Pro práci s daty přehledu jsem vytvořil aplikaci, jež umožňuje zájemci o koupi CRM systému zvolit na základě definovaných kritérií vhodný CRM systém. (kapitola 4)

Přes malou ochotu výrobců podělit se o implementační detaily se podařilo s pomocí informací z přehledu CRM systémů a na základě konzultací s odborníky částečně vysledovat používané způsoby implementace CRM systému. Konzultace zároveň umožnily shrnout současný stav CRM systémů v ČR a označit nejzávažnější problémy při nasazení CRM do českých podniků. (kapitola 8)

Srovnáním různých technologií se ukázalo, že koncepce SNA je nejen teoreticky použitelnou architekturou pro tvorbu IS, ale že dokonce existují standardy a technologie, na kterých lze SNA skutečně stavět. Provedené srovnání mezi koncepcí SNA a servisně-orientovaným přístupem ukázalo, že SNA je ekvivalentem řešení založených na SOA. (kapitola 6)

Diskuze možností použití koncepce SNA v oblasti CRM ukázala, že použití SNA může pomoci s problémy současného českého CRM jen částečně (finanční stránka). Pro řešení největšího problému (datová roztržitost) nabízí SNA nové prostředky, jejich aplikace je však závislá na možnosti doplnit do ostatních systémů potřebná rozhraní.

Použití SNA pro stavbu CRM by mohlo přinést zvýšení modularity systému, snížení závislosti na jednom dodavateli, zjednodušení aktualizace systémových komponent, nativní podporu automatizace a možnost definovat aplikační logiku a (business procesy) uživatelem. Vně CRM by díky SNA mohlo dojít ke zlepšení automatizovaného propojení jak na úrovni B2B (partnerské systémy), tak B2C (zákazník).

S využitím SNA obecně se pojí několik problémů – jedná se o ochranu přístupu k funkcím prostřednictvím veřejných rozhraní, kapacitní limity přenosové infrastruktury a zátěž způsobenou informační mediací. Otevřenou otázkou je lokalita

uložení dat, která poukazuje na obecnější problém – najít hranici, kdy je vhodné operovat centrálně a kdy je naopak přínosnější decentralizované řešení. (kapitola 9)

Pro plné využití potenciálu SNA, jmenovitě typu softwarové aliance, dosud neexistuje ověřený způsob, jak sémanticky popsat funkce SNA tak, aby bylo možné automatizovat vytváření softwarových konfederací<sup>72</sup>. Lze totiž očekávat, že softwarové konfederace budou mít úzkou souvislost s tzv. *PSA (Problem Solving Architecture, architektura zaměřená na řešení problémů)*, o které se hovoří jako o dalším evolučním stupni softwarové. [5]

Cenným zdrojem informací o problematice se ukázal Internet, bílá literatura<sup>73</sup>, a články v odborných časopisech (elektronických i tištěných). Praktické informace pocházely hlavně z konzultací s odborníky. Knih o CRM publikovaných v českém jazyce je poskrovnu, navíc u čtenáře předpokládají hlubší znalosti z oblasti ekonomie. V reakci na toto zjištění byla úvodní část koncipována jako vstupní brána do problematiky CRM pro čtenáře bez ekonomického vzdělání.

---

<sup>72</sup> Problematikou sémantického popisu funkce se zabývají skupiny *Web Service Modeling Ontology* [56] a *Semantic Web* (<http://www.w3.org/2001/sw/>).

<sup>73</sup> Oficiální dokumentace k produktům od autora produktu.

## Slovník použitých termínů a zkratek

### analýza asociací

Analýza hledající soubor pravidel, které zachycují vzájemný výskyt (vztah) dvou nebo více jevů, například: „Jestliže si zákazníci koupí chléb, potom si v 75% případů koupí i máslo.“

### BI (*Business Intelligence*)

soubor nástrojů a postupů umožňující efektivní analýzu obchodních dat

### BP (*Business Process*, obchodní proces)

sled činností v podniku, který má zajistit dosažení požadovaných výsledků (např. zajistit efektivní provoz podniku, zaručit kvalitu poskytovaných služeb a produktů, určit příčinu nedosažení cíle)

### *call scripting*

operátorovi se zobrazují instrukce pro vedení hovoru s klientem

### CI (*Customer Intelligence*)

vhled do struktury a chování zákazníka

### CLV (*Customer Lifetime Value*)

souhrn současné ziskovosti klienta a jeho potenciální ziskovosti v budoucnu po dobu celého jeho životního cyklu s organizací

### CRM (*Customer Relationship Management*, řízení vztahů se zákazníky)

viz kapitola 2.2

### CTI (*Computer Telephony Integration*)

propojení hlasových služeb s informační technologií

### CVM (*Customer Value Management*)

ohodnocení zákazníka z hlediska jeho přínosu dodavatelské firmě

### *data mining* (dolování/vytěžování dat)

Spočívá v pokročilé analýze velkého množství dat, jejímž cílem je zjistit jinak nepostřehnutelné vnitřní souvislosti a trendy. Příklady častých aplikací jsou segmentace zákazníků, analýza a predikce odchodu zákazníků ke konkurenci, analýza chování návštěvníků webových stránek, detekce podvodného chování a analýza rizik. Data-mining se využívá také k přesnému zacílení marketingových kampaní, podpoře křížového prodeje (cross-selling) nebo vývoji nových produktů.

### datový sklad (DWH, *Data Warehouse*)

jedna či více propojených databází

### dCRM (*Distributed CRM*, distribuované CRM)

pojem zavedený v této práci pro označení studie CRM vycházející z myšlenek SNA

DI (*Data Interface*, datové rozhraní)

datová složka IS umožňující interakci se systémem přímým přístupem k datům

doménová oblast

Doménovou oblastí se míní určitá množina činností se společným rysem. Hranici mezi doménovými oblastmi pak tvoří vzdálenost mezi rysy. O doménových oblastech se hovoří zejména ve spojení s lidským konáním, kde jedné doméně odpovídá obor specializace. Není přitom vyloučeno, že se doménové oblasti překrývají či dokonce vnořují (záleží jen na míře specializace). Příkladem doménových oblastí může být účetnictví, zpracování textů, ale stejně tak i kompletní ekonomická agenda z pohledu manažera.

DOT (*Distributed Object Technology*, distribuovaná objektová technologie)

middleware zajišťující spolupráci distribuovaných objektů, mechanismus distribuce a komunikace objektů je součástí DOT

DWH

viz „datový sklad“

EAI (*Enterprise Application Integration*, integrační podniková platforma)

ERP (*Enterprise Resource Planning*)

Finančně orientovaný informační systém pro určení a plánování podnikových zdrojů potřebných k přijetí, zhotovení, dodání a zaúčtování zákaznického obchodního případu. Tyto systémy bývají považovány za jádro celého informačního systému, nabízejí komplexní pohled na oblast zdrojů podniku.

ETL (*Extraction, Transformation and Loading*, datová pumpa)

mechanismus přenesení dat z jednoho či více systémů do jiného; přenos probíhá ve třech fázích: získání a přečištění (extrakce), jejich úprava do formátu vhodného pro příjemce (transformace) a načtení do cílového systému

genetické algoritmy

Optimalizační techniky napodobující teorii evolučního vývoje v přírodě. Základní stavební prvky těchto algoritmů jsou selekce, křížení a mutace.

informační mediace

viz „sémantická transformace“

IP telefonie

viz VoIP

IS (informační systém)

KIS (komunikační infrastruktura)

Pojmem KIS v práci souhrně označují různé technologie (middleware, komunikační protokoly...), které zajišťují transport informace mezi odesílatelem a příjemcem.



## lineární regrese

statistická technika používaná k nalezení nejlepší lineární závislosti mezi cílovou proměnnou a množinou vysvětlujících proměnných

## logistická regrese

obdobu lineární regrese, ale pro kategoriální cílovou proměnnou (např. typ klienta, příznak ne/splácel u půjčky)

MI (*Machine Interface*, strojové rozhraní)

rozhraní pro použití funkcí aplikace strojem; rozhraní sestává z definovaných vstupů a výstupů, většinou v abstraktní strojově čitelné formě, např. XML; neobsahuje žádné grafické prvky; používá se i formou vzdáleného volání (viz RPC), zasíláním zpráv (MOM)

MOM (*Message Oriented Middleware*, middleware zasílající zprávy)

middleware zajišťující spolupráci entit (typicky služeb) prostřednictvím předávání zpráv

## neuronové síť

nelineární prediktivní model, který vznikl jako analogie k mozkovým neuronovým sítím živých organismů

NFS (*Network File System*)

distribuovaný síťový souborový systém, byl prvním větším nasazením mechanismu RPC

OLAP (*On-Line Analytical Processing*)

Způsob analýzy při kterém procházíme data a podle zjištěných informací volíme nové pohledy, vysvětlující a doplňující zjištěné hodnoty a trendy. Softwarové nástroje umožňující analýzu OLAP procházejí dimenzionálním modelem způsobem *slice'n'dice*, tj. provádí rozpad a sloučení řezů dat. Výsledky prezentují formou mnoha typů grafů a tabulek.

OLTP (*On-Line Transaction Processing*, zpracování transakcí v reálném čase)

Režim práce s databází, kdy se zpracování požadavků neodkládá, ale provádí se okamžitě, jakmile požadavek od uživatele přijde.

OOM (*Object-Oriented Middleware*, objektově orientovaný middleware)

označení pro obslužnou vrstvu v DOT, např. ORB z Corby

PLM (*Product Lifecycle Management*)REST (*Representational State Transfer*)

Jedna z implementací webových služeb založená na použití operátorů PUT, GET, POST a DELETE protokolu HTTP.

rozhodovací stromy (*Decision Trees*)

Grafická prezentace (připomínající obrácený strom) vztahů mezi závislou proměnnou a množinou nezávislých proměnných. Příkladem může být sledování vlivu věku, pohlaví, povolání a místa bydliště na bezproblémové splácení půjčky.

SCM (*Supply Chain Management*)

je procesně orientované řízení logistické sítě prostřednictvím plánování a řízení toků materiálu, informací a financí na základě IT

## sémantická transformace

převod zprávy mezi dvěma syntakticky různými formáty takový, že je zachována sémantika původní zprávy; např. převod zprávy „jméno;příjmení“ na „příjmení;jméno“

SFA (*Sales Force Automation*, automatizace obchodních procesů)

## shlukové algoritmy

Skupina algoritmů určených pro identifikaci skupin podobných případů (např. zákazníků, smluv, účtů). Mezi typické představitele těchto algoritmů lze zařadit algoritmus K-Means nebo speciální typ neuronových sítí (Kohonenovy neuronové sítě).

SOA (*Service Oriented Architecture*, architektura orientovaná na služby)

viz kapitola 6 „Technologie a SNA“

SOAP (*Simple Object Access Protocol*)

Protokol, který se využívá ke komunikaci s XML službami. Parametry dotazu jsou strukturovány v XML a zaslány webové službě pomocí protokolu HTTP.

SRM (*Supplier Relationship Management*, řízení dodavatelských vztahů)SSL (*Secure Socket Layer*)

protokol pro zabezpečení komunikace po Internetu; podporován většinou prohlížeč

UI (*User Interface*, uživatelské rozhraní)

rozhraní uzpůsobené pro vyvolání funkcí IS, aplikace člověkem; obsluha pomocí HID, např. klávesnice, myši...

VoIP (*Voice over Internet Protocol*)

Způsob přenosu hlasu po Internetu. Hlas je zkomprimovaný do objemově malých datových souborů, tzv. paketů, které jsou v koncové destinaci opět dekomprimovány na hlas.

WSDL (*Web Services Description Language*)

obdoba IDL známého z DOT, kódovaný v XML

## zdeděný systém

O zděném systému se hovoří v souvislosti se zaváděním nového IS do organizace. Z pohledu zaváděného systému se všechny používané informační systémy, které nebudou nahrazeny zaváděným systémem, označují jako zděné.

## Seznam zdrojů

- [1] A. B. Andersson. *CRM aneb začněte pečovat o své zákazníky dříve, než vám utečou ke konkurenci* [online]. [cit. 19.2.2005]. Dostupné z: <http://www.abandersson.com/index.php?ID=clanky#CRM1>
- [2] *Adastra: Slovník pojmů* [online]. [cit. 20.3.2005]. Dostupné z: <http://www.adastra.cz/default.asp?menu=technologie/&page=slovník.cz.html>
- [3] ANTLOVÁ, Klára. *Rozšíření ERP za hranice podniku* [online]. Liberec, Technická Univerzita Liberec, 2003. [cit. 15.3.2005]. Dostupné z: [http://militedu.vlib.cz/autori/klara\\_antlova/MIS/mis\\_crm.ppt](http://militedu.vlib.cz/autori/klara_antlova/MIS/mis_crm.ppt)
- [4] *Basic Profile Version 1.1, Final Material* [online]. Web Services Interoperability Organization (WS-I). Dostupné z: <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>
- [5] BRUIJN, J., BUSSLER, Chris, FENSEL, Dieter, KRUMMENACHER, Reto, LAUSEN, Holger, MORAN, Matthew, ISTRANG, Thomas, ZAREMBA Michal. *Semantic Web Services Roadmap* [online]. DERI Gaway, DERI Innsbruck : 2005. Dostupné z: <http://deri.at/publications/presentations/document/WSMLXRoadmap.ppt>
- [6] *Business Process Modeling Language 1.0* [online]. Business Process Management Initiative. Dostupné z: <http://www.bpmi.org/BPM1.htm>
- [7] CORBA [online]. Dostupné z: <http://www.corba.org>
- [8] CORMELA-DORDA, Santiago. *Component Object Model (COM), DCOM and Related Capabilities* [online]. Software Engineering Institute, Carnegie Mellon University, 2004. [cit. 25.7.2005]. Dostupné z: <http://www.sei.cmu.edu/str/descriptions/combody.htm>
- [9] COVER, Robin. *Standardized Generalized Markup Language (SGML)* [online]. [cit. 13. 5. 2006]. Dostupné z: <http://xml.coverpages.org/sgml.htm>
- [10] *CRM je cestou k poznání a udržení zákazníka* [online]. Česká agentura na podporu obchodu CzechTrade, 2003. [cit. 8.8.2005]. Dostupné z: <http://www.czechtrade.cz/global.xml?prone=diarforma+ce+pro+vas/tiskove+zpravy.xml&id=tiskova+zprava=159>
- [11] ČÁBELA, Miroslav. *Jedno CRM nestačí* [online]. [ČR : Adastra, 1999]. [cit. 17.2.2005]. Dostupné z: <http://www.adastra.cz/downloads/ppt/SCFITP+Cabela+Jedno+CRM+nesti+aci.ppt>

- [12] ČÁBELA, Miroslav. *Komplexní CRM - optimální přístup k zákazníkům*. In: IT System 11/2001. [ČR : Adastra, 2001]. [cit. 10.3.2005]. Dostupné z: [http://www.adastra.cz/default.asp?menu=events/&page=adastra\\_v\\_tisku\\_cz.asp&ID=32](http://www.adastra.cz/default.asp?menu=events/&page=adastra_v_tisku_cz.asp&ID=32)
- [13] ČÁBELA, Miroslav. *Máte systém CRM? Nebo jen drahou kartotéku?* [online]. In: COMPUTERWORLD 33/2001. [cit. 5.3.2005]. Dostupné z: [http://www.adastra.cz/default.asp?menu=event\\_s/&page=adastra\\_v\\_tisku\\_cz.asp&ID=30](http://www.adastra.cz/default.asp?menu=event_s/&page=adastra_v_tisku_cz.asp&ID=30).
- [14] Česká společnost pro systémovou integraci. *CRM – Řízení vztahů se zákazníky*. [cit. 17.2.2005]. Dostupné z: <http://www.cssi.cz/oblastindex.asp?oblast=030301D>
- [15] DCOM [online]. Microsoft Dostupné z [http://msdn.microsoft.com/library:/default.asp?url=/library/en-us/dndcom/html/msdn\\_dcomtec.asp](http://msdn.microsoft.com/library:/default.asp?url=/library/en-us/dndcom/html/msdn_dcomtec.asp)
- [16] DOHNAL, Jan. *Řízení vztahů se zákazníky: Procesy, pracovníci, technologie*. Praha : Grada, 2002. 164s. ISBN 80-247-0401-3.
- [17] DOHNAL Jan, KUČERA Miroslav. *Úvod do CRM v informační společnosti*. Praha : Vysoká škola ekonomická v Praze, 2001. ISBN 80-245-0139-2.
- [18] *EntireX DCOM for Linux* [online]. Software G. Dostupné z [http://www1.softwar eag.com/Corporate/products/entirex/downloads/download\\_exxdcom.asp](http://www1.softwar eag.com/Corporate/products/entirex/downloads/download_exxdcom.asp)
- [19] ERL, Thomas. *Service-oriented Architecture: Concept, Technology, and Design*. US, Indiana, Crawfordsville: 2005. ISBN 0-13-185858-0.
- [20] FERGUSON, Renee Boucher. *New Tools Aid Modeling Processes* [online]. eWeek [cit. 8.8.2005]. Dostupné z: <http://www.parnold.com/articles/handysoft/sevee k.htmfD>
- [21] Fiorano Software, Inc. *Demystifying Enterprise Service Bus Technology* [online]. USA, Los Gatos, 2003. Dostupné z: [http://www.fiorano.com/wltpapers/fiorano\\_esb.pdfDfD](http://www.fiorano.com/wltpapers/fiorano_esb.pdfDfD)
- [22] *Getting on The Bus, An Interview with Dave CiappelE* [online]. Sonic Software. [cit. 3.8.2005]. Dostupné z: [http://www.sonicsoftware.com/sdutiors/learningcenter/books/enterprise\\_service\\_bus/getting\\_on\\_the\\_bus/index.spm](http://www.sonicsoftware.com/sdutiors/learningcenter/books/enterprise_service_bus/getting_on_the_bus/index.spm)
- [23] CHUNG, E. P, HUANG, Y., YAJNIK, S., LIANG, D., SHIH, J. C., WANG, C., WANG, Y. *DCOM and CORBA Side by Side, Step by Step, and Layer by Layer* [online]. 1997. [cit. 24.7.2005]. Dostupné z: <http://www.id.liu.se/~TDT S43/labs/S.pdf u>
- [24] *J-Integra*. Intrinsic Software International, Inc. Dostupné z <http://js-integra.intrinsic.com/>

- [25] KOPECKÝ, Jacek. *SOAP – konečně správné pH pro váš softwaru?* [online]. [cit. 27.7.2005]. Dostupné z: <http://www.zive.cz/hf/programovani/AR.asp?ARI=100819>
- [26] KOSEK, Jiří. *Inteligentní podpora navigace na WWW s využitím XML, Diplomová práce : Kapitola 4. Komunikační infrastruktura* [online]. [cit. 19.7.2005]. Dostupné z: <http://www.kosek.cz/dp/online/ltmh/d04s02.ltmh>
- [27] KRÁL, Jaroslav. *Informační systémy*. Veletiny: Science, 1998. 356s. ISBN 80-86083-00-04.
- [28] KRÁL, Jaroslav. *Orientace na služby – klíčové paradigma současného softwaru* [online]. Praha, Univerzita Karlova, Matematicko-fyzikální fakulta, Katedra softwarového inženýrství, 2004. Dostupné z: <http://www.datakon.cz/datakon04/d024itkral.pdf>
- [29] KRIŽKO, Ivo. *Je CRM módou, nebo nutností?* [online]. In: IT SYSTEM 11/2003. [cit. 17.2.2005]. Dostupné z: <http://www.systemonline.cz/sit/prahedy/systemu/crm/d0401ness.htm>
- [30] LEWIS, Grace, MORRIS, Edwin, SMITH, Dennis, WRAGE, Lutz. *Service-Oriented Architectures as an Interoperability Mechanism* [online]. 2004. [cit. 27.7.2005]. Dostupné z: <http://www.sei.cmu.edu/research-at-sei/columns/eye-on-integration/eye-on-integration.htm>
- [31] LEYMANN, Frank, CURBERA Francisco, FERGUSON Donald, STOREY Tony, WEERAWARANA Sanjiva. *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More*. Indiana, Indianapolis : Prentice Hall PTR, 2005. ISBN 0131488740. Kapitola *Web Services: A Realization of SOA*. Dostupné z: <http://www.phptr.com/articles/article.asp?p=378135&l=1>
- [32] *Message-Oriented Middleware* [online]. IEEE Distributed Systems Online. ISSN 1541-4922. [cit. 29.7.2005]. Dostupné z: [http://dsnlire.computer.org/portal/site/dsnlire/mnuitem/d3d9924ae1b0dcd82ccc6716bbe36ec/index.jsp?&pName=dso\\_level1&path=dsonline/topics/middleware&fle=itromom.xml&sl=article.xml](http://dsnlire.computer.org/portal/site/dsnlire/mnuitem/d3d9924ae1b0dcd82ccc6716bbe36ec/index.jsp?&pName=dso_level1&path=dsonline/topics/middleware&fle=itromom.xml&sl=article.xml)
- [33] *Mule – Universal Message Objects* [online]. [cit. 9.8.2005]. Dostupné z: <http://mulecodehaus.org>
- [34] PETERKA, Jiří. *Archiv článků a přednáček Jiřího Peterky : RPC II*. [online]. [cit. 19.7.2005]. Dostupné z: <http://www.archiv.cz/d4/a01c110.php3>
- [35] QUIN, Liam. *The Extensible Stylesheet Language Family (XSL)* [online]. Dostupné z: <http://www.w3.org/Style/XSL/>
- [36] RHODES, Kates. *XML-RPC vs. SOAP* [online]. [cit. 22.7.2005]. Dostupné z: [http://weblog.makumi.org/witirgs/xml\\_rpc\\_vs\\_soap.htm](http://weblog.makumi.org/witirgs/xml_rpc_vs_soap.htm)



- [37] RUD, Olivia Parr. *Data mining : praktický průvodce dolováním dat pro efektivní prodej, cílený marketing a podporu zákazníků (CRM)*. Praha : Computer Press, 2001. 329 s. ISBN 80-7226-577-6.
- [38] SAP. *SAP AG – SAP Web Application Server in Switchover Enviroments : Release 6.40* [online]. [cit. 4.7.2005]. Dostupné z: <http://service.sap.com/ha5> → Media Library → Documentation → Switchover
- [39] SAP. *SAP Česká republika* [online]. [cit. 9.7.2005]. Dostupné z: <http://www.sap.com/cz/company/index.aspx>
- [40] SAP. *SAP Česká republika – SAP NetWeaver* [online]. [cit. 10.7.2005]. Dostupné z: <http://www.sap.com/cz/solutions/netweaver/index.aspx>
- [41] SAP. *SAP Global* [online]. [cit. 9.7.2005]. Dostupné z: <http://www.sap.com/company/index.aspx>
- [42] SAP. *Switchover Software for High Availability* [online]. Dostupné z: <http://help.sap.com/saphelpnw04/helpdata/en/08/544e44ae611d1894f000e829fbd/france.set.ltr>
- [43] SHÜTTE, Alois. *Verteilte Systeme: XML RPC* [online]. Fachhochschule Darmstadt, Fachbereich Informatik. [cit. 21.7.2005]. Dostupné z: [http://www.fb.fhm-darmstadt.de/~schuette/Vorlesungen/VerteilteSysteme/Skript/4\\_Objektdienste/XMLRPC/XML-RPC.pdf](http://www.fb.fhm-darmstadt.de/~schuette/Vorlesungen/VerteilteSysteme/Skript/4_Objektdienste/XMLRPC/XML-RPC.pdf)
- [44] *Slovník pojmů* [online]. K2amitec. Dostupné z: <http://www.k2amitec.cz/slovník.htm>
- [45] *SOA for the Enterprise: Enterprise Service Bus (ESB)* [online]. Sonic Software. [cit. 5.8.2005]. Dostupné z: [http://www.sonicsoftware.com/solutions/soa\\_enterprise/enterprise\\_service\\_bus/index.ssp](http://www.sonicsoftware.com/solutions/soa_enterprise/enterprise_service_bus/index.ssp)
- [46] *SOAP Basics, chapter 3. SOAP Messages* [online]. SoapUser.com. [cit. 23.7.2005]. Dostupné z: <http://www.soapuser.com/basics3.ltr>
- [47] *SOAP Tutorial* [online]. [cit. 23.7.2005]. Dostupné z: <http://www.vschool.com/soap/default.asp>
- [48] *SOAP Version 1.12* [online]. W3C Recommendation, 2003. Dostupné z: <http://www.w3.org/TR/soap12/o>
- [49] SOMMERGUT, Wolfgang. *Alternativní cesty k webovým službám* [online]. In: Computerworld 3/2004. Praha : IDG Czech, 2004. ISSN: 1210-9924. [cit. 6.8.2005]. Dostupné z: [http://www.cw.cz/cw\\_rsf/ID/123F7E3FCDC05BBAC1256E9400332DE9?OpenDocument&cast=3](http://www.cw.cz/cw_rsf/ID/123F7E3FCDC05BBAC1256E9400332DE9?OpenDocument&cast=3)
- [50] *Sonic ESB®, Overview* [online]. Dostupné z: [http://www.sonicsoftware.com/products/sonic\\_esb/index.ssp](http://www.sonicsoftware.com/products/sonic_esb/index.ssp)



- [51] SRINIVASAN, R. *RFC 1831 – Remote Procedure Call Protocol Specification, c Verison 2* [online]. 1995, August. Sun Microsystems. Dostupné z: [http://www. faqs. org/rfcs/rfc1831. ht m s](http://www.faqs.org/rfcs/rfc1831.htm)
- [52] Sun Microsystems. *RFC 1014 – XDR: External Data Representation Standard* [online]. 1987, June. Dostupné z: <http://www. faqs. org/rfcs/rfc1014. ht m>
- [53] *Sun Remote Procedure Call Mechanism* [online]. Worcester Polytechnic Institute. Dostupné z: <http://www. cs. wpi. edu/~cs4513/b01/week4 - sunrpc/week4-sunrpc.html>
- [54] THOMAS, Nigel, DAVIES, Robert. *ESB Technology & Inovation: Extending Web services with asynchronous message delivery and intelligent routing* [online]. XML Journal, 2004. [cit. 5.8.2005]. Dostupné z: <http://xml. sysi- con.com/read/44672.htm>
- [55] WALLNAU, K., MORRIS, E., FEILER, P., EARL, A., LITVAK, E. *Engineering Component-Based Systems with Distributed Object Technology* [online]. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, USA. [cit. 25.7.2005]. Dostupné z: [http://www. sei. cmu. edu/staff/kw/wall96. pdf](http://www. sei. cmu. edu/staff/kw/wall96.pdf)
- [56] *Web Service Modeling Ontology* [online]. [cit. 9.8.2005]. Dostupné z: <http://www. wsmo. org>
- [57] *Web Services Business Process Execution Language (WS-BPEL)* [online]. Organization for the Advancement of Structured Information Standards. Dostupné z: [http://www. oasis- open.org/committees/tc\\_home. php?wg\\_abbrev=ws\\_bpel](http://www. oasis- open.org/committees/tc_home. php?wg_abbrev=ws_bpel)
- [58] WELSH, Matt, CULLER, David, BREWER, Eric. *SEDA: An Architecture for Well-Conditioned, Scalable Internet Services* [online]. California, Berkley : University of California, Computer Science Division, 2001. Dostupné z: <http://www. eecs. harvard. edu/~mwl/papers/sedaD-sosp01.pdf>
- [59] WESSLING, Harry. *Aktivní vztah k zákazníkům pomocí CRM : strategie, praktické příklady a scénáře*. Praha : Grada, 2003. 192s. ISBN 80-247-0569-9.
- [60] WINNER, Dave. *Dave's History of SOAP* [online]. UserLand Software, 1999. Dostupné z: [http://www. xml. rpc. com/stories/storyReader\\$555](http://www. xml. rpc. com/stories/storyReader$555)
- [61] *XML-RPC Homepage* [online]. Dostupné z: <http://www. xml. rpc. com>

## Seznamy obrázků, grafů a tabulek

### Seznam obrázků

<b>OBR. 1</b> <i>MAGICKÁ OBRAZOVKA</i> .....	10
<b>OBR. 2</b> <i>APLIKAČNÍ ARCHITEKTURA</i> .....	12
<b>OBR. 3</b> <i>OBECNÁ STRUKTURA INFORMAČNÍHO SYSTÉMU</i> .....	28
<b>OBR. 4</b> <i>KOMPOZICE INFORMAČNÍCH SYSTÉMŮ</i> .....	30
<b>OBR. 5</b> <i>PŘÍMÁ ADAPTACE</i> .....	33
<b>OBR. 6</b> <i>ADAPTACE S VYUŽITÍM UNIVERZÁLNÍHO MEZIFORMÁTU</i> .....	34
<b>OBR. 7</b> <i>ÚPLNÉ PROPOJENÍ NEZÁVISLÝCH APLIKACÍ PŘÍMOU ADAPTACÍ.</i> .....	34
<b>OBR. 8</b> <i>ÚPLNĚ PROPOJENÍ NEZÁVISLÝCH APLIKACÍ NEPŘÍMOU ADAPTACÍ.</i> .....	35
<b>OBR. 9</b> <i>UNIVERZÁLNÍ ADAPTÉR PRO INFORMAČNÍ MEDIACI</i> .....	44
<b>OBR. 10</b> <i>SCHÉMA SONIC ESB</i> .....	53

### Seznam grafů

<b>GRAF 1</b> <i>ROZVRSTVENÍ SYSTÉMŮ PODLE VELIKOSTI CÍLOVÉHO PODNIKU</i> .....	20
<b>GRAF 2</b> <i>ROZDĚLENÍ CRM SYSTÉMŮ PODLE NEJKRATŠÍHO ČASU POTŘEBNÉHO K NASAZENÍ SYSTÉMU</i> .....	21
<b>GRAF 3</b> <i>PODPOROVANÉ DATOVÉ SKLADY</i> .....	22
<b>GRAF 4</b> <i>CRM SYSTÉMY PODLE ARCHITEKTURY</i> .....	23

### Seznam tabulek

<b>TAB. 1</b> <i>ROZDĚLENÍ SYSTÉMŮ PODLE VELIKOSTI CÍLOVÉHO PODNIKU</i> .....	20
---	----

## Příloha A

### Struktura přehledu CRM systémů na českém trhu

#### PRODUKT

- Název produktu
- Verze
- Datum zápisu údaje
- Datum aktualizace údaje

#### DODAVATEL

- Jméno výrobce
- WWW výrobce
- Název dodavatele v ČR
- Adresa dodavatele
- Telefon
- E-mail
- WWW dodavatele
- Obrat společnosti v roce 2000 v mil. Kč
- Obrat společnosti v roce 2001 v mil. Kč
- Obrat společnosti v roce 2002 v mil. Kč
- Obrat společnosti v roce 2003 v mil. Kč
- Obrat společnosti v roce 2004 v mil. Kč
- Rok uvedení produktu na trh
- Rok uvedení aktuální verze
- Počet konzultantů produktu v ČR
- Roky zkušeností s implementací CRM
- Další dodavatelé nebo implementační partneři

#### FUNKČNOST SYSTÉMU

##### CALL CENTRUM

- Podpora různých komunikačních kanálů se zákazníky (telefon, fax, internet, e-mail, SMS)
- On-line přepojení zákazníka na specializovaného operátora
- Poskytování všech relevantních informací o právě obsluhovaném zákazníkovi operátorům
- Sběr a vyhodnocování statistických údajů
- Podpora Dialed Number Identification Service (DNIS)
- Podpora Automated Number Identification (ANI)
- Podpora Interactive Voice Response systems (IVR)

##### MARKETING

- Podpora různých komunikačních kanálů se zákazníky (telefon, fax, internet, e-mail, SMS)

- Podpora List Managementu (seznamy zákazníků, dodavatelů, jejich import a export)
- Automatizované plánování marketingové kampaně na základě analýz zákazníků
- Příprava rozpočtů plánovaných kampaní a sledování jejich plnění v průběhu kampaně
- Přesné zacílení kampaní na vybrané zákazníky nebo skupiny
- Realizace tržních testů (náhodný výběr zákazníků z cílové skupiny)
- Definování metrik, podílu a dalších ukazatelů a hodnocení úspěšnosti kampaně
- Realizace jedné kampaně několika komunikačními kanály najednou
- Sledování reakce zákazníků v reálném čase ze všech komunikačních kanálů
- Analýza průběžných výsledků probíhajících kampaní
- Operativní zasahování do provádění marketingové kampaně podle reakcí cílové skupiny
- Integrace s aplikacemi přímo souvisejícími se zákazníky (call centrum, servis, podpora, ...)
- Udržování a zpřístupňování báze informací o použitelných komunikačních kanálech
- Správa informací o zákaznících (vyloučení oslovení jednoho zákazníka dvakrát stejným sdělením)
- Hodnocení výsledků kampaně pomocí různých metrik a kritérií
- Podpora řízení a realizace propagačních akcí

*PRODEJ*

- Podpora různých komunikačních kanálů se zákazníky (telefon, fax, internet, e-mail, SMS)
- Přístup ke všem relevantním informacím o obchodním případě
- Profil zákazníka
- Dosavadní kontakty
- Vynaložené a naplánované náklady
- Řízení vývoje obchodního případu na základě stanovené metodiky
- Přístup k základním informacím (katalogy, ceny, konkurence, ...) přes web
- Automatizace korespondence a jejího vyřizování (hromadná korespondence, šablony dopisů, ...)
- Podpora týmového prodeje vícevrstevnými distribučními kanály
- Automatizace synchronizace dat centrálního serveru a mobilních obchodníků
- Product Configuration (vytváření nabídky přímo v místě prodeje)
- Předpovědi objemu prodeje
- Správa motivačního programu účastníků obchodního procesu

*SERVIS*

- Podpora různých komunikačních kanálů se zákazníky (telefon, fax, internet, e-mail, SMS)

- Případové analýzy
- Průchody rozhodovacími stromy
- Pomocné textové informace (rešerše technické dokumentace)
- Produktové specifikace
- FAQ databáze
- Řešení známých problémů z minulosti
- Automatické nasměrování požadavků zákazníka na příslušného řešitele
- Řízení garance doby vyřešení požadavku

#### *OPTIMALIZACE OBCHODNÍCH PROCESŮ*

- Zdokonalování prodejních a servisních procesů
- Vyhodnocování efektivity prodejních a servisních procesů
- Grafická vizualizace procesů a organizační struktury
- Simulace procesů a trasování procesů
- Měření efektu CRM systému

#### **SPECIALIZOVANÉ MODULY, FUNKCE NEBO NASTAVENÍ PRO RŮZNÁ ODVĚTVÍ**

- Obsahuje produkt spec. moduly
- Nasazení v odvětví (např. banky a finance, telekomunikace)

#### **DALŠÍ VLASTNOSTI SYSTÉMU**

- Pronájem systému po Internetu (ASP)
- Mobilní přístup

#### **ARCHITEKTURA A PLATFORMY**

- Architektura systému
- Možné platformy systému – operační systém
- Možné platformy systému – databáze

#### **UŽIVATELE PRODUKTU**

- Počet instalací produktu (počet zákazníků celosvětově)
- Počet instalací produktu v ČR

#### **PRO JAKOU VELIKOST PODNIKU JE PRODUKT URČEN**

- malé podniky (obrat do 250 mil. Kč ročně)
- středně velké podniky (obrat 250 mil. – 2 mld. Kč ročně)
- velké podniky (obrat nad 2 mld. Kč ročně)

#### **RŮZNÉ**

- Průměrná doba implementace u podniku střední velikosti
- Reference produktu v ČR
- Zajímavosti systému nespádající do žádné z výše uvedených kategorií

## Příloha B

Tato příloha obsahuje zhuštěný výtah z materiálů společnosti SAP o jejich produktu *SAP NetWeaver*. Výtah jsem sestavil na základě neveřejných materiálů, což je důvod, proč je zde malý počet odkazů na externí zdroje.

Příloha ukazuje, z čeho se skládá a jak vnitřně pracuje platforma založená na SOA.

### SAP NetWeaver

*SAP NetWeaver* je aplikační a integrační platforma. Je technologickou základnou pro balík firemních aplikací *ImySAP Business Suite*, pro kompozitní aplikace *SAP xApps* a pro jiná obecná nebo odvětvově orientovaná řešení společnosti SAP. *SAP NetWeaver* vytváří rovněž technologickou bázi nově navržené architektury společnosti SAP pojmenované *Enterprise Services Architecture*<sup>74</sup>, která slouží jako předloha pro řešení založená na využití webových služeb. *SAP NetWeaver* sjednocuje všechny integrační technologie do jediné platformy. Platforma *SAP NetWeaver* je postavena na bázi otevřených standardů, plně spolupracuje s jinými běžně používanými technologickými platformami, jako jsou Java 2 Platform, Enterprise Edition (J2EE), Microsoft .NET a IBM WebSphere.

*SAP NetWeaver* se skládá z několika technologických komponent. Protože *SAP NetWeaver* je funkčně velmi bohatý produkt, jehož popis je nad rámec této práce, zmíním jen nejzajímavější aspekty řešení. Propodrobnější popis doporučuji např. [40].

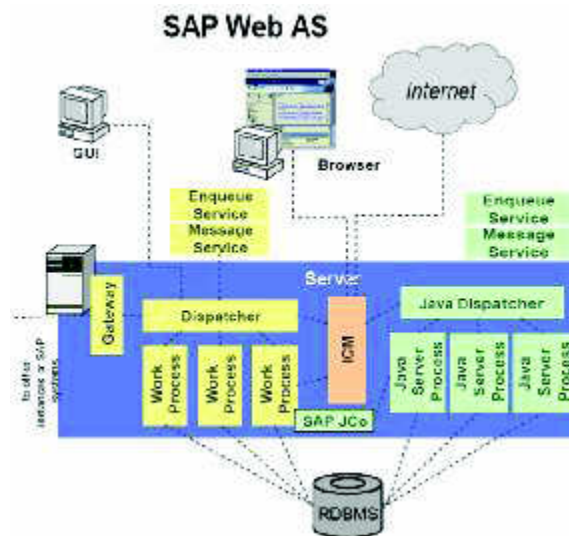
### SAP Web Application Server

Nejdůležitějším prvkem *SAP NetWeaveru* je *SAP Web Application Server* (dále jen *SAP Web AS*) poskytující základní rozhraní pro zpřístupnění funkcí jednotlivých poskytovaných aplikací na bázi standardu webových služeb (*Web Services*, viz dále). *SAP Web AS* je vývojová a provozní platforma založená na otevřených standardech, která vytváří provozní prostředí pro všechna řešení společnosti SAP a umožňuje vývoj s využitím klíčových technologií, jako jsou J2EE a ABAP. *SAP Web AS* se skládá z jedné nebo několika instancí běžících na jednom nebo více strojích. Složení jedné instance využívající všech modulů ilustruje následující obrázek:

---

<sup>74</sup> Jiné označení pro ESB, podnikovou specializaci SOA.





V každém informačním systému založeném na *SAP NetWeaveru* existuje právě jedna *centrální instance*. Centrální instancí se rozumí instance *SAP Web AS* společně s *Enqueue Service* a *Message Service*.

*Message Service* (služba zpráv) udržuje seznam dostupných plánovačů (*SAP Web Dispatcher*) a serverových procesů Java clusteru. Reprezentuje infrastrukturu pro výměnu menších dat mezi uzly informačního systému (instance *SAP Web AS*, jednotlivé služby v rámci systému i služby externí). Informuje také *SAP Web Dispatcher* o load-balancingu mezi jednotlivými instancemi.

*Enqueue Service* (služba řazení) spravuje logické databázové zámky, které jsou použity prováděnou aplikací v serverovém procesu (*Work Process*) a zajišťuje tak synchronizaci a bezpečnost operací napříč clusterem.

Každá instance obsahuje *SAP Web Dispatcher* (resp. *SAP Java Dispatcher* v případě Javy), jehož úkolem je rozdělovat příchozí požadavky mezi pracovní procesy instance. Na základě informací od *Message Service* zajišťuje load-balancing. Požadavky jsou obslouženy buď procesem stejné instance, ve které běží dispatcher, nebo jsou delegovány na jiné instance či systémy SAP prostřednictvím brány (*SAP Gateway*).

*ICM* (*Internet Communication Manager*) je implementován jako nezávislý proces spuštěný a monitorovaný dispatcherem. Jeho úkolem je zaručit SAP systému komunikaci s vnějším světem pomocí protokolů HTTP, HTTPS a SMTP. Může vystupovat v roli klienta i serveru v závislosti na typu požadavku. V roli server může zpracovat požadavky z Internetu, jejichž URL s názvem serveru a portu náleží danému *ICM*. *ICM* pak obslouží požadavek voláním odpovídajícího lokálního handleru. *ICM* proces používá vlákna pro rychlejší zpracování požadavků.

Dosud zmíněné komponenty se týkaly *ABAP*<sup>75</sup>. Vedle *ABAP* mohou běžet instance *SAP Web AS Java*. *SAP Web AS Java* je kompletní implementací standardu *Java*<sup>TM</sup>

<sup>75</sup> *ABAP* (*Advanced Business Application Programming*) je programovací jazyk používaný pro tisíce malých vložených programů nazývaných transakce, které společně vytváří *SAP* aplikaci.

2 Enterprise Edition, standardu vyvinutého speciálně pro deficienci prostředí pro běh podnikových aplikací vytvořených v Javě a pro poskytnutí infrastruktury požadované takovými aplikacemi<sup>76</sup>.

*SAP Web AS Java* má podobnou strukturu jako ABAP verze. Sdílí společně ICM, obsahuje vlastní *Java Dispatcher* a pracovní procesy označované jako *Java Server Process*. Oproti ABAP verzi obsahuje navíc *SAP Java Connector* (SAP Jco) zajišťující propojení ABAP a Java aplikací (komunikace probíhá pomocí protokolů SOAP a RFC), *Software Delivery Service* (SDM) sloužící k nahrání aplikací na server.

## SAP Exchange Infrastructure

*SAP Exchange Infrastructure* je robustní komunikační platforma založená na výměně zpráv formou XML. Skládá se z adaptérů zajišťujících konektivitu se všemi druhy aplikací. SAP poskytuje škálu adaptérů v rámci *SAP Exchange Infrastructure* pro integraci nejen existujících řešení SAP přes *IDoc* a *BAPI* rozhraní, ale také jiných systémů prostřednictvím souborů, zpráv nebo rozhraní webových služeb. Do frameworku lze zapojit i adaptéry systémů partnerských firem. Kromě toho lze se systémy, které nepodporují *SAP Exchange Infrastructure* lze použít i průmyslových standardů jako *RosettaNet*.

## SAP Composite Application Framework

*SAP Composite Application Framework* podporuje vytváření aplikací řízených modelem. Opírá se při tom o abstraktní objektovou vrstvu, která zpřístupňuje systémové repositáře a propojené systémy formou objektů. Při vytváření aplikací pak není potřeba zohledňovat, zda je funkce či operace zpřístupněná jako objekt abstraktní vrstvy objektů poskytována *SAP NetWeaverem* (např. webovou službou či přes *SAP Exchange Infrastructure*) nebo externím poskytovatelem. Proto lze v prostředí *SAP NetWeaveru* jednoduše vytvářet složené aplikace napříč dostupnými datovými zdroji.

---

<http://www.sap-basis-abap.com/sapabap.htm>

<sup>76</sup> Java2 Enterprise Edition standard definoval následující služby:

- Java Server Pages (JSP)
- Java Servlets
- Enterprise Java Beans (EJB)
- Java™ Message Service
- Java Mail™

Specifikace J2EE a souvisejících standardů je dostupná na [java.sun.com](http://java.sun.com).