

Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Martin Vojtek

*Prostředky podpory byznysu v moderních informačních
systémech*

Katedra softwarového inženýrství

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

Studijní program: *Informatika, softwarové systémy*

Predovšetkým ďakujem Prof. RNDr. Jaroslavovi Královi, DrSrc. za vedenie diplomovej práce, za jeho cenné rady a pripomienky. Ďalej by som rád poďakoval Pavlovi Köhlerovi a Vladimírovi Klementovi, spolumajiteľom spoločnosti Flynet za poskytnutie prístupu k informačnému systému Devis a konzultácie pri vývoji systému EPA. Nakoniec by som chcel poďakovať svojej manželke Lee a našej dcérke Eme za pochopenie a trpezlivosť.

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

V Praze dne 11.12.2007

Martin Vojtek

Obsah

1. ÚVOD	5
2. ZÁMER PRÁCE	6
2.1 ZÁMER	7
3. POJEM BUSINESS INTELLIGENCE.....	8
3.1 POHĽAD NA BI Z RÔZNYCH PERSPEKTÍV.....	8
3.1.1 Manažérsky pohľad	8
3.1.2 BI ako architektúra	8
3.1.3 BI ako proces.....	10
3.1.4 BI ako schopnosť.....	10
4. BI APLIKÁCIE.....	11
4.1 KOMPONENTY BI APLIKÁCIÍ.....	11
4.2 KATEGÓRIE A FUNKCIE BI APLIKÁCIÍ	12
5. CUSTOMIZOVANÉ SYSTÉMY	13
5.1 VÝHODY A NEVÝHODY CUSTOMIZOVANÝCH IS.....	13
5.2 VÝHODY A NEVÝHODY CUSTOMIZOVANÝCH BI APLIKÁCIÍ	14
6. MOŽNOSTI INTEGRÁCIE BI IS DO ERP	17
6.1 INTEGRÁCIA APLIKÁCIÍ.....	17
6.2 INTEGRÁCIA BI APLIKÁCIE VYVÍJANEJ OD ZAČIATKU	23
6.2.1 Integrácia vstupov	23
6.2.2 Integrácia výstupov.....	24
7. NÁVRH ROZŠÍRENIA IS.....	26
7.1 FÁZY NÁVRHU ROZŠÍRENIA IS O PRVKY BI.....	26
7.2 NÁVRH DÁTOVÉHO SKLADU.....	27
7.2.1 Multidimenzionálna analýza a modelovanie	28
7.2.2 Modelovanie dátového skladu.....	28
7.2.3 Riešenie zmien v dimenziách.....	29
7.3 NÁVRH ETL.....	30
7.4 NÁVRH ÚLOŽISKA METADÁT	32
7.4.1 Centralizované úložisko metadát.....	32
7.4.2 Decentralizované úložisko metadát.....	33
7.4.3 Distribuované riešenie založené na XML.....	34
7.4.4 Zvolené riešenie úložiska metadát.....	35
8. EPA.....	36
8.1 ZADANIE PROJEKTU	36
8.1.1 OLAP grafy	37
8.1.2 Porovnanie.....	37

8.2 PRÍNOS PROJEKTU	38
8.3 TECHNICKÁ ARCHITEKTÚRA	40
8.3.1 Mantis.....	42
8.3.2 Devis.....	42
8.3.3 BI platforma	42
8.3.4 ETL.....	43
8.3.5 Dátový sklad	44
8.3.6 Dolovanie dát.....	46
8.3.7 Aplikačný server.....	47
8.3.8 Podniková zbernica služieb	54
8.4 POUŽITÉ TECHNOLOGIE A SOFTVÉR	55
8.4.1 Mantis.....	56
8.4.2 Devis.....	56
8.4.3 Dátový sklad	56
8.4.4 Aplikačný server.....	56
8.4.5 Podniková zbernica služieb	57
8.4.6 Dolovanie dát.....	57
8.4.7 OLAP	58
8.4.8 Ostatné	58
8.5 INTEGRÁCIA DÁTOVÝCH ZDROJOV A SYSTÉMOV	59
8.6 PROBLÉMY PRI VÝVOJI.....	60
8.7 BUDÚCI VÝVOJ SYSTÉMU	61
8.8 NASADENIE SYSTÉMU	61
9. ZÁVER	62
ZOZNAM POUŽITÝCH SKRATIEK	63
LITERATÚRA	65
PRÍLOHA A	70
PRILOŽENÉ CD	70

Název práce: *Prostředky podpory byznysu v moderních informačních systémech*

Autor: *Martin Vojtek*

Katedra (ústav): *Katedra softwarového inženýrství*

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

e-mail vedoucího: *Jaroslav.Kral@mff.cuni.cz*

Abstrakt: *Byznys inteligence (BI) je schopnost najít a pochopit komplexní vztahy v byznysu, efektivně se adaptovat změnám prostředí (konkurence, trhu, trendům), učit se z dat. Integrace BI aplikací do existujících podnikových informačních systémů představuje komplexní proces. Tento proces může zahrnovat informačně, procesně, portálově nebo servisně orientovanou integraci. K integraci BI aplikací s podnikovými informačními systémy je možné použít také podnikovou sběrnici služeb. Při integraci takto komplexních systémů nabývá použití standardů velkého významu a výrazně přispívá k architektuře výsledného řešení.*

Implementaci integrace BI aplikace s podnikovým informačním systémem představuje aplikace EPA, která k integraci využívá integrační platformu Java EE, open source aplikační server JBoss, databázi Oracle a množství různých knihoven. Aplikace EPA integruje prvky BI aplikací, jako jsou OLAP, dolování dat, datové pumpy. K integraci datových zdrojů se využívá datový sklad implementovaný nad databází Oracle. Část aplikace prezentující grafy a interaktivní křížové tabulky je postavena na moderním integračním frameworku JBoss Seam, který integruje technologie, jako jsou například JBoss Rules, JSF, EJB3 a jBPM.

Klíčová slova: *byznys inteligence, informační systémy, integrace*

Title: *Business intelligence means in modern information systems*

Author: *Martin Vojtek*

Department: *Department of Software Engineering*

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

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

Abstract: *Business intelligence (BI) is the ability to find and understand complex relations in business, to effectively adapt to changes of environment (competition, market, trends), to learn from data. Integration of BI applications with existing enterprise information systems leads to complex process. This process can comprise information, process, portal and service oriented integration. For this type of integration, it is also possible to use ESB technology. When integrating such complex systems, usage of standards gains great importance and significantly contributes to the architecture of result solution.*

Implementation of integration of BI applications with enterprise information systems represents the EPA application, which is built on Java EE integration platform with usage of JBosss, open source application server, Oracle database, data mining tools and ETL. Data warehouse implemented on the top of Oracle database is used for the integration of data sources. Part of the application presenting graphs and interactive cross tables is built on JBoss Seam, modern integration framework, which integrates technologies like JBoss Rules, JSF, EJB3 and jBPM.

Keywords: *business intelligence, information systems, integration*

1. Úvod

V posledných desaťročiach organizácie, spoločnosti a podniky zhromažďovali dáta vo svojich informačných systémoch. Sú to napríklad dáta o zamestnancoch, o zákazníkoch, o obchodných transakciách. Značná časť majetku spoločnosti leží práve v týchto dátach. Informácie v nich skryté je možné využiť k zvýšeniu profitu, ako aj k udržaniu kroku s konkurenciou. Business Intelligence (BI) aplikácie predstavujú nástroje určené k odhaľovaniu, prezentácii a využitiu týchto dát. Aby mohli BI aplikácie získavať dáta z podnikových informačných systémov, je nutné ich určitým spôsobom s týmito systémami integrovať.

Integrácia BI aplikácií s podnikovými informačnými systémami je komplexný proces, ktorého zložitosť závisí lineárne na požiadavkách výstupov a častokrát exponenciálne na množstve dátových zdrojov, respektíve počte informačných systémov reprezentujúcich podnikový informačný systém.

Do množiny BI aplikácií patrí veľké množstvo analytického softvéru, ako aj rôznych špecializovaných nástrojov. Medzi najbežnejšie a najpoužívanejšie je možné zaradiť Online Analytical Processing (OLAP), nástroje dolovania dát, dátové pumpy a dátový sklad. So vznikom štandardov, ako je architektúra Java 2 Enterprise Edition (J2EE) konektorov, Java Data Mining API a ďalších, sa proces integrácie BI aplikácií s Enterprise Resource Planning (ERP) systémami postupne štandardizuje. Predmetom nasledujúcich kapitol je analýza možností integrácie prvkov BI aplikácií do ERP, ako aj návrh a implementácia takejto integrácie.

Aplikácia EPA, ktorá je popísaná v tejto práci, implementuje návrh integrácie dvoch informačných systémov predstavujúcich podnikový informačný systém so štandardnými typmi BI aplikácií. K integrácii je využitá celá paleta existujúcich technológií, pričom základným kameňom architektúry tohto systému je platforma Java EE.

2. Zámer práce

K popisu zámeru tejto diplomovej práce je nutné definovať základné pojmy, aby tento zámer bol jasný a začlenený do širšieho kontextu. V prvom rade definujeme pojem ERP (Enterprise Resource Planning System), teda podnikový informačný systém.

Informačný systém (IS) je množina hardvér, softvér, a telekomunikačných komponentov, podporovaná ľuďmi a procedúrami, ktorej účelom je spracovávať dáta a meniť ich na užitočné informácie [1].

Manažérsky informačný systém (MIS) je informačný systém používaný pre plánovanie, kontrolu, rozhodovanie alebo riešenie problémov [1].

Podnikový informačný systém (ERP) je informačný systém, ktorý podporuje rôzne aktivity pre rôzne oddelenia, pomáhajúci manažérom s plánovaním a chodom rôznych, nezávislých funkcií [1].

ERP budeme chápať ako systém podporujúci všetky alebo aspoň väčšinu obchodných (biznis) aktivít organizácie alebo podniku, pozostávajúci z viacerých nezávislých, nejakým spôsobom integrovaných informačných systémov.

Ďalším dôležitým pojmom je integrácia. Pod pojmom integrácia budeme rozumieť integráciu aplikácií.

Z biznis pohľadu je *integrácia* strategická aktivita a množina technológií, ktorá umožňuje organizácii fungovať efektívnejšie, a v mnohých prípadoch poskytuje značnú konkurenčnú výhodu [2].

Z technického pohľadu je *integrácia* strategický prístup spojenia viacerých informačných systémov na úrovni služieb a informácií, ktorý podporuje schopnosť výmeny informácií a pôsobenie na procesy v reálnom čase [2].

Customizácia je proces prispôsobenia kupovaného informačného systému potrebám a požiadavkám užívateľa [3].

Pojmom *business intelligence (BI)* sa budeme hlbšie zaoberať v nasledujúcej kapitole.

2.1 Zámer

1. Analyzovať možnosti integrácie prostriedkov efektívnej podpory manažmentu a obchodu (BI) do ERP.
2. Analyzovať cesty využívania BI v ERP.
3. Samotný pojem business intelligence.
4. Metódy použiteľné k podpore biznisu vo všeobecne dostupných komerčných softvérových produktoch a cesty ich využitia či integrácie, výhody a nevýhody oproti možnostiam v customizovaných systémoch.
5. Návrh implementácie a implementácia rozšírenia informačného systému o prostriedky podpory biznisu.

3. Pojem Business Intelligence

Termín *business intelligence* zaviedol v roku 1989 Howard J. Dresner [11], analytik spoločnosti Gartner Group, ktorý ho popísal ako „sadu konceptov a metód určených pre skvalitnenie rozhodnutí firmy“. Vyzdvihuje tu význam dátovej analýzy, nástrojov na tvorbu reportov, nástrojov na tvorbu dotazov (query tools), ktoré sprevádzajú užívateľa množstvom dát a pomáhajú mu so syntézou hodnotných a užitočných informácií.

Pojem business intelligence je v súčasnej dobe hojne používaný. Trh je doslova zaplavený aplikáciami a nástrojmi, ktoré o sebe tvrdia, že spadajú do kategórie business intelligence. Existuje mnoho veľkých nadnárodných spoločností, ktoré venujú nemalé prostriedky do vývoja BI nástrojov a konkurencia v tejto oblasti je skutočne enormná. Aj keď už existuje množstvo takýchto nástrojov, pojem business intelligence nie je pevne vymedzený a neexistuje jediná správna definícia. Prakticky ľubovoľne zvolený zdroj, ako sú rôzne odborné články alebo knihy o business intelligence, predstavuje svoju vlastnú definíciu. Tieto definície sa v rôznych ohľadoch prekrývajú a tiež existujú rôzne pohľady, ako je napríklad pohľad zo strany manažéra, alebo technický pohľad zo strany softvérového architekta. Ďalšie rozdelenie je z hľadiska vzťahu entity, respektíve role k BI. Definícia z pohľadu organizácie (podniku) kombinuje definície z pohľadov jednotlivých rolí užívateľov - manažér, programátor ...

3.1 Pohľad na BI z rôznych perspektív

3.1.1 Manažérsky pohľad

Business intelligence je konzistentná, metodická transformácia z ľubovoľných a všetkých dátových zdrojov do nových foriem, ktorá poskytuje biznisom riadené informácie orientované na výsledok [4].

3.1.2 BI ako architektúra

BI nie je ani produkt, ani systém. Je to architektúra a kolekcia integrovaných operatívnych aplikácií, aplikácií na podporu rozhodovania a databáz, ktorá poskytuje biznis komunitě jednoduchý prístup k biznis dátam [5].

Doposiaľ uvedené definície nehovoria nič o tom, aké aplikácie sa považujú za aplikácie BI. Aplikácie BI je v súčasnej dobe možné kategorizovať podľa aktivít, ktoré vykonávajú. Medzi tieto aktivity patria mimo iných nasledujúce [5]:

- Multidimenzionálna analýza, napríklad OLAP (online analytical processing).
- Analýza webových kliknutí (click-stream analysis).
- Dolovanie dát.
- Predpovede.
- Biznis analýza.
- Príprava objektívnych výsledkových lístkov (balanced scorecards).
- Vizualizácia.
- Dotazovanie, reporty a grafy.
- Manažment vedomostí.
- Analýza na základe geografického priestoru.
- Implementácia podnikového portálu.
- Dolovanie dát z textov, hlasu.
- Digitálny prístup k elektronickej nástenke.
- Transformácia produkčných, zdrojových systémov.
- Zaistenie kvality dát.
- Správa metadát.
- Monitorovanie.
- Podpora rozhodovania.

Aplikácia implementovaná v tejto diplomovej práci spadá do viacerých kategórií. Sú to predovšetkým OLAP, dolovanie dát, predpovede a monitorovanie.

Aplikácie, ktoré implementujú tieto aktivity zákonite nemôžu existovať izolovane, ale musia čerpať dáta z určitých zdrojov. Je samozrejme možné čerpať dáta v neupravenom stave z databáz a iných nespracovaných zdrojov, ako sú napr. texty, ale oveľa väčšiu hodnotu majú informácie získané z dát spracovaných analytickými nástrojmi. Tieto dáta sa zvyčajne ukladajú v nejakom type dátového skladu (predmetovo-orientované, integrované, časovo-premenlivé, ...), a to na rôznych úrovniach podnikovej hierarchie (oddelenie, pobočka, ...).

Vo svetle aktivít BI aplikácií je možné zovšeobecniť procesy, ktoré v týchto aplikáciách prebiehajú. Lubovoľná BI aplikácia nejakým spôsobom spracováva dáta alebo prezentuje informácie zo spracovaných dát, alebo oboje. To, čo sa uskutoční na základe týchto informácií už do aplikácií BI nepatrí. BI aplikácie v podstate implementujú proces tlmočenia surových dát do formy informácií dávajúcich zmysel v určitom konkrétnom kontexte. BI aplikácie sa potom delia podľa vstupov, výstupov a samotného algoritmu tohto procesu. U BI aplikácií je špecifické, že užívateľ takejto aplikácie musí poznať a brať na vedomie nielen dáta a kvalitu dát, z ktorých sa dané výstupy vytvárajú, ale aj algoritmus, akým sa k daným výstupom dospelo. Je realitou, že dve rôzne konkurenčné BI aplikácie dávajú protichodné výsledky na základe

rovnakých vstupných dát (s rovnakou kvalitou dát). Rozdiel je práve v algoritme, ktorým sa aplikácia dopracuje k výsledkom.

3.1.3 BI ako proces

BI je proces zvýšenia konkurenčnej výhody biznisu inteligentným využitím dát pri tvorbe rozhodnutí [66]. Tento proces pozostáva z piatich fáz:

1. Získanie dát – Extrahovanie dát z elektronických textov, databáz, obrázkov, mediálnych súborov a webových stránok.
2. Dátová analýza – Syntetizácia použiteľných znalostí zo zozbieraných dát použitím algoritmov dolovania dát, algoritmov porozumenia textom a analyzovania obrázkov.
3. Uvedomenie situácie – Spojenie užitočných faktov a odvodení a odfiltrovanie irelevantných informácií.
4. Analýza rizík – Identifikovanie racionálnych rozhodnutí alebo smeru akcií založených na očakávaní rizika.
5. Podpora rozhodovania – Použitie polo automatizovaného softvéru k identifikácii dobrých rozhodnutí a stratégií.

3.1.4 BI ako schopnosť

Pojem BI nepredstavuje kategóriu softvéru. BI aplikácie budeme chápať ako softvér na podporu BI. Pojem BI budeme naďalej chápať v zmysle nasledujúcej definície:

Business intelligence (BI) je schopnosť nájsť a pochopiť komplexné vzťahy v biznise, efektívne sa adaptovať zmenám prostredia (konkurencii, trhu, trendom), učiť sa z dát. BI aplikácie sú aplikácie, ktoré z dát z rôznych zdrojov dokážu získať, objaviť a prezentovať informácie, ktoré pomáhajú s hľadaním a jednoduchším pochopením týchto vzťahov, uľahčujú adaptáciu na tieto zmeny, ako aj predpovede týchto zmien.

4. BI aplikácie

BI aplikácie sú softvérové produkty určené takmer výhradne k podpore biznisu. V súčasnej dobe existuje na poli BI viacero silných spoločností, ktoré poskytujú širokú paletu kvalitných nástrojov použiteľných k podpore biznisu. Podľa uskutočneného prieskumu portálu informationweek z marca 2007 medzi preferovaných predajcov BI aplikácií patria Oracle (40%), Microsoft – špecifické produkty (38%), Business Objects (32%), Cognos (30%), IBM – špecifické produkty (29%), SAP (20%) a Hyperion (19%) [64]. Percentá v zátvorkách vyjadrujú počet respondentov preferujúcich danú platformu. Respondenti mali možnosť uviesť viacero preferovaných platforiem.

4.1 Komponenty BI aplikácií

BI aplikácia je vždy kombináciou najrôznejších komponentov, pričom konkrétna podoba výsledného riešenia, rozsah celého systému a zložitosť sa líšia v závislosti na faktoroch vyplývajúcich z potrieb daného podniku alebo inštitúcie. Neexistuje teda štandardné alebo prevažujúce riešenie. V najširšom kontexte medzi komponenty BI aplikácie zahŕňame [10]:

- Produkčné a zdrojové systémy, väčšinou transakčného charakteru, vytvárajúce z pohľadu BI zdrojové databázy (tieto systémy nie sú primárnymi komponentmi riešenia BI, len zdrojom dát).
- Dočasné úložiská dát (DSA) pre okamžité uloženie dát a ich úpravy pred transformáciou do dátového skladu.
- Operatívne úložiská dát (ODS), s aktuálnou snímkou dát, určené pre operatívne dotazy.
- ETL, dátové pumpy zaisťujúce výbery, transformácie a ukladanie dát medzi produkčnými databázami, DSA, ODS, dátovým skladom a dátovými trhoviskami.
- Dátové sklady, väčšinou celopodnikového charakteru.
- Integrované nástroje (EAI) pre on-line, okamžité aktualizácie dátového skladu, umožňujúce tzv. „dátové sklady v reálnom čase“.
- Dátové trhoviská orientované na vybrané útvary (divízie apod.) alebo oblasti riadenia firmy.
- OLAP databázy zahŕňajúce jednu alebo viacero OLAP kociek charakteristických agregovanými údajmi a krátkou dobou odozvy.
- Nástroje pre ohlasovanie, pre generovanie výstupných prehľadov, správ a podobne.
- Manažérske aplikácie, užívateľské aplikácie nad dátovými skladmi alebo OLAP kockami.

- Nástroje pre dolovanie dát a realizáciu zložitých a náročných štruktúrovaných analýz.
- Nástroje pre zaistenie kvality dát.
- Nástroje pre správu metadát.
- Systémy pre podporu rozhodovania (DSS) a expertné systémy (ES).

4.2 Kategórie a funkcie BI aplikácií

Podrobné a kompletne vymenovanie kategórií a funkcií BI aplikácií presahuje rámec tejto diplomovej práce. Obsiahle vymenovanie kategórií a funkcií súčasných BI aplikácií je dostupné na stránkach wikipédie [65]. Ďalším rozsiahlym zdrojom je zoznam kategórií tzv. bielych kníh (white papers) v oblasti BI podľa portálu InformationWeek [6].

5. Customizované systémy

5.1 Výhody a nevýhody customizovaných IS

Vývoj a nasadenie IS má radu špecifických rysov. Obvykle nie je možné IS kúpiť a bez podstatnejších zmien používať tak ako napr. Excel. IS je často nutné vyvíjať od začiatku. V prípade predpripravených IS ako tomu je napr. u Oracle, je potom ešte nutné tieto aplikácie upraviť – proces tzv. customizácie. Rozhodnutie vyvíjať IS od začiatku alebo kúpiť customizovaný IS je veľmi komplikované a v mnohých prípadoch sa môže stať príčinou neúspechu IS.

Vývoj IS je zdĺhavá a riskantná záležitosť, ktorá vo väčšine prípadov znamená nutnosť vynaloženia enormných finančných prostriedkov. Z tohto pohľadu je pre zákazníka jednoduchšie – istota väčšej záruky - kúpiť už predpripravený IS a len ho upraviť - customizovať – tak, aby v čo najväčšej miere vyhovoval jeho potrebám. Vlastné prispôbenie požiadavkám zákazníka sa uskutočňuje parametrizáciou – nastavením určitých parametrov systému implementovaných dodávateľom. Medzi týmito parametrami môžu byť napríklad rôzne lokálne nastavenia ako je jazyk, mena, rozloženie obrazoviek (napr. arabské štáty verzus európske štáty). Za parametre môžeme považovať tiež zapnutie/vypnutie určitých funkčností systému. Do procesu customizácie môže spadať aj vývoj určitej funkčnosti závislej na prostredí zákazníka, aj keď tento proces už úzko súvisí s procesom integrácie. Modernejším spôsobom customizácie je proces vymodelovania a následného vygenerovania systému na základe modelu alebo vopred špecifikovaných vstupných parametrov. Nástroje, ktoré implementujú tento proces sa označujú súhrnným názvom ako CASE systémy. Na prvý pohľad nie je zrejмый rozdiel v cene medzi IS vytvoreným od začiatku a customizovaným softvérom. Mohlo by sa zdať, že cena customizovaného IS je menšia, ako cena IS vytvoreného od začiatku. Vo väčšine prípadov opak je pravdou. Nejde len o priame finančné investície do kúpeného IS, ale ide v podstate o princíp prispôbenia sa firmy produktu, namiesto prispôbenia sa produktu na mieru firme. Customizovaný IS má častokrát funkcie, ktoré daná firma nepotrebuje, nemá niektoré časti, ktoré potrebuje a podobne. Pričom IS vyvíjaný od začiatku sa po celý čas vyvíja so zámerom uspokojenia určitých konkrétnych potrieb firmy a rešpektuje organizačnú a spoločenskú štruktúru danej firmy. Na druhej strane ak cieľom zavedenia IS je získať konkurenčnú výhodu, tak je zrejмый, že kúpením predpripraveného IS a následným procesom customizácie sa táto konkurenčná výhoda stráca, keďže aj konkurencia má tú istú možnosť kúpy daného IS. Životný cyklus IS sa skladá z viacerých častí ako len vývoj. Jednou z týchto častí je aj prevádzka a údržba IS. Táto fáza je zvyčajne zabezpečovaná dodávateľom IS (nemusí to byť pravidlom) a tento si môže (a v drvivej väčšine prípadov to tak je) účtovať nemalé poplatky za túto činnosť. Je zrejмый, že IS, ktorých prevádzku plánujeme na dlhšie obdobie, majú väčšiu pomernú časť fázy prevádzky a údržby z celkového životného cyklu. Podrobný popis prácností jednotlivých etáp pri vývoji a pri customizácii je možné nájsť v publikácii Informačné Systémy [3].

Vývoj od počiatku ako aj proces customizácie má svoje výhody a nevýhody. Nedá sa povedať, ktorý prístup je lepší alebo horší. Rozhodnutie pre vývoj od počiatku alebo customizáciu závisí na kontexte toho konkrétneho projektu, respektíve cieľa z pohľadu biznisu. Podrobnejší popis výhod a nevýhod customizovaných informačných systémov je možné nájsť v publikácii Informačné Systémy [3].

5.2 Výhody a nevýhody customizovaných BI aplikácií

BI aplikácie sú špecifickým typom informačných systémov. Ako na také, sa na nich vzťahujú už uvedené výhody a nevýhody customizovaných IS, ale niektoré sa na tieto aplikácie vzťahujú v menšom meradle a naopak niektoré body vystupujú do popredia.

BI aplikácie sú silne zamerané na strategické ciele, aj keď v súčasnosti je trend posúvať ich vplyv a využitie čím ďalej tým viac aj do taktickej oblasti operatívy. Ďalším špecifikom je malý počet užívateľov na najvyšších úrovniach organizačnej štruktúry. Trend je rozširovať počet užívateľov BI aplikácií, a to aj na nižšie úrovne organizačnej hierarchie. Stanovenie cieľov a špecifikácia požiadaviek pre BI aplikácie je komplikovaným procesom. Vyplýva to už z podstaty BI aplikácií.

Ako príklad BI aplikácie vezmeme systém na podporu rozhodovania (decision support system - DSS). Klasický vývojový cyklus vodopád predpokladá, že požiadavky na vyvíjaný systém sú známe alebo je možné ich poznať už na začiatku návrhu, alebo že prinajmenšom je možné tieto požiadavky objaviť. Lenže vo svete analytika systému na podporu rozhodovania, požiadavky sú tou poslednou vecou, ktorá má byť objavená v priebehu vývojového životného cyklu. Analytik DSS začína s existujúcimi požiadavkami, ale zmena k novým požiadavkám je takmer nemožná. Jadrom väčšiny BI aplikácií je dátový sklad, ktorý integruje dáta z rôznych zdrojov. Je to zrejme tá najdôležitejšia časť celého systému, pretože v prípade, že dátový sklad obsahuje chybné dáta, tiež aj informácie odvodené a prezentované na základe týchto dát musia byť zákonite chybné. V takomto prípade je lepšie nemať žiaden dátový sklad, ako mať dátový sklad s úplne mylnými dátami. Samozrejme aj reálne dátové sklady obsahujú isté percento chybných dát, ale toto percento musí byť v nejakých rozumných medziach, aby neznehodnotilo celý dátový sklad. Otázka chybných dát sa rieši v rámci procesov zabezpečujúcich kvalitu dát. S vývojom dátového skladu je spojený veľmi odlišný vývojový cyklus.

Operačné dáta sú zvyčajne aplikačne orientované a v dôsledku neintegrovane, pričom dáta v dátovom sklade musia byť integrované. Existujú veľké rozdiely medzi dátami a ich spracovaním na úrovni operatívy a na úrovni dátového skladu. Vývoj dátového skladu sa odvíja od úplne iného vývojového cyklu ako je klasický vodopád, a to v podstate opačného cyklu. Klasický vodopád je založený na požiadavkách. Po spracovaní požiadaviek nasledujú fázy návrhu a vývoja. Vývojový cyklus DSS je takmer presne opačný [9]. Cyklus začína dátami. Získané dáta sa následne integrujú

a potom testujú, aby bolo zrejmé, aké tendencie tieto dáta obsahujú. V ďalšej fáze sa proti daným dátam napíšu programy. Výsledky programov sa analyzujú a nakoniec sú pochopené požiadavky. V okamžiku pochopenia požiadaviek sa vyladzuje návrh systému a cyklus štartuje znova od začiatku pre inú množinu dát. V dôsledku opätovného reštartovania vývojového cyklu pre rôzne typy dát, je tento vývojový cyklus typom špirálového modelu vývoja. Tento vývojový cyklus je založený na dátach, pričom vývojový cyklus vodopád je založený na požiadavkách. Snažiť sa použiť nevhodné nástroje a techniky vývoja môže vyústiť jedine k mrhaniu prostriedkov a zmätku. Napríklad svetu nástrojov pod menom CASE (Computer Aided Software Engineering) dominuje analýza založená na požiadavkách. Snažiť sa použiť nástroje CASE na vývoj dátového skladu alebo DSS systémov, nie je veľmi rozumné.

Už z podstaty veci BI aplikácie nemôžu byť distribuované ako celok. Na druhej strane môžu byť predávané samostatne jednotlivé menšie aplikácie striktne určené k nejakému účelu. Príkladom takéhoto softvéru môže byť aplikácia zobrazujúca grafy nad určitým dátovým zdrojom. Predpokladom takéhoto softvéru je dodržovanie štandardu na formát dátového zdroja respektíve konektora (napríklad XMLA [67]).

Tabuľka 5.2 sumarizuje rozdiely medzi BI aplikáciami a klasickými IS:

Klasické IS	BI aplikácie
1. veľký počet užívateľov	malý počet užívateľov
2. užívatelia naprieč celou organizačnou štruktúrou	typickým užívateľom je manažment
3. skôr taktické zameranie	strategické zameranie
4. skôr operačné dáta	dáta dátového skladu
5. vývoj hnaný požiadavkami	vývoj hnaný dátami
6. princíp vývoja metódou vodopád	v prípade aplikácií obsahujúcich dátový sklad opačný vývojový cyklus ako vodopád

Tabuľka 5.2: porovnanie IS a BI aplikácie

Súčasný trend je znižovať rozdiely medzi BI aplikáciami a klasickými IS. Prvé tri rozdiely sa postupne stierajú, a to smerom od BI aplikácií k IS. Typicky sú BI aplikácie produktom integrujúcim množstvo menších aplikácií a v prípade bodu 6, sú len niektoré aplikácie vyvíjané opačným vývojovým cyklom ako je vodopád – typicky dátový sklad. Niektoré aplikácie ako je podporný softvér – ETL (Extract, Transform, Load) – softvér na extrahovanie dát z dátových zdrojov, ich čistenie a transformovanie do jednotného integrovaného dátového skladu, je vyvíjaný klasickými metódami vývoja a súčasne sa považuje za súčasť BI aplikácií. To znamená, že jednotlivé aplikácie súhrnného BI IS sú vyvíjané rôznymi metodológiami. V praxi je výhodné niektoré z týchto aplikácií kúpiť už ako hotové produkty a zvyšné časti vyvíjať od začiatku

pomocou moderných vývojových postupov a technológií, ako je objektovo orientované programovanie a komponentová architektúra. Vývoj by mal prebiehať nad nejakou modernou integračnou platformou ako je napríklad Java EE a výsledným produktom by mali byť integrované jednotlivé aplikácie do väčšieho celku, ktorý už je možné nazvať BI IS – business intelligence informačný systém.

Nie všetky BI aplikácie je možné zaškatuľkovať pod termín IS. Niektoré špecifické systémy, ako sú expertné systémy vystupujú väčšinou ako samostatné jednotky a len v kontexte ďalších aplikácií spojených perifériami alebo sieťou, ich môžeme považovať za súčasť IS. Takýto systém môže mať prístup k externej databáze obsahujúcej bázu dát a výsledky takejto aplikácie (napr. zdravotného zariadenia na diagnostiku) môžu byť napojené na nejakú databázu pacientov a podobne.

6. Možnosti integrácie BI IS do ERP

6.1 Integrácia aplikácií

Na ERP je možné pozerat' ako na veľký podnikový systém pozostávajúci z množstva integrovaných menších informačných systémov. Jedným z cieľov tejto diplomovej práce je analyzovať možnosti integrácie špecifického IS, BI IS, do ERP. Podľa Linthicum [2] existujú dva rôzne pohľady na integráciu. Prvým je biznis pohľad, ktorý hovorí, že *integrácia aplikácií je strategická aktivita a množina technológií, ktorá umožňuje organizácii fungovať omnoho efektívnejšie a v mnohých prípadoch poskytuje značnú konkurenčnú výhodu*. Druhým pohľadom je abstraktný technický pohľad, ktorý hovorí, že *integrácia aplikácií je strategický prístup spojenia mnohých informačných systémov na úrovni služieb a informácií, podporujúci schopnosť výmeny informácií a ovplyvňujúci procesy v reálnom čase*. Pri integrácii nesmieme zabúdať na biznis pohľad. Nie je rozumné integrovať aplikácie, ktoré je síce možné integrovať, ale ich integrácia neprinesie žiaden úžitok. Zbytočné integrovanie aplikácií nutne povedie k stále väčšej komplikovanosti výsledného integračného riešenia so zbytočným plytvaním zdrojov a v konečnom dôsledku to môže ohroziť celý projekt ako taký. Čím väčšie množstvo aplikácií integrujeme, tým väčšia je potom údržba tejto integrácie a komplikovanosť riešenia spočíva napríklad aj s každou novou verziou jednotlivých integrovaných aplikácií. Abstraktný technický pohľad už naznačuje, akým spôsobom je možné aplikácie integrovať a čo by malo byť výsledkom takejto integrácie.

Rozlišujeme dve základné formy integrácie [2]:

1. Integrácia aplikácií v rámci podniku (Enterprise Application Integration – EAI).
2. Integrácia aplikácií medzi podnikmi (Business-to-Business Application Integration – B2B).

Keďže aplikácia BI má predstavovať strategickú výhodu v biznise, budeme sa vo vzťahu integrácie aplikácie BI do ERP, zaoberať formou EAI. Existujú samozrejme aj firmy, ktorých oblasťou podnikania je poskytovanie BI ako externej služby, ale tento koncept je v praxi tak trochu exotický, pretože je buď veľmi náročný alebo len veľmi ťažko uskutočniteľný – BI potrebuje množstvo interných dát, ktoré sa považujú za bohatstvo organizácie. Tu nastáva problém bezpečnosti, keďže dáta patria medzi najcennejšie hodnoty organizácie. Ďalším problémom B2B v oblasti BI je, že aplikácie BI sú až príliš špecifické, a tým pádom je každá komunikácia medzi týmito aplikáciami novým komplikovaným problémom. V súčasnosti existuje viacero štandardov, ktoré prispievajú k prekonaniu tohto procesu, ako je napr. ebXML [50] alebo architektúra webových služieb [51]. Najnovším počínom v oblasti B2B je takzvaný mash-up (tiež mixovanie) aplikácií.

Mash-up je webová stránka alebo aplikácia, ktorá kombinuje obsah z viacej ako jedného zdroja a dáva tak možnosť súčasného využitia rôznych typov informácií skombinovaných dohromady. Jedným z najpopulárnejších zdrojov pre mash-up aplikácie sú napr. GoogleMaps, YouTube alebo Flickr. Spoločnosti ako Google ponúkajú svoje aplikačné rozhrania pre programovanie aplikácií (tzv. API), využiteľné programátormi pri tvorbe Web 2.0 aplikácií (vrátane mash-up) [52, 83].

Každá integrácia aplikácií vyžaduje svoj vlastný osobitý prístup. Je možné vymedziť všeobecné kategórie prístupov v EAI:

1. Informačne orientované (IOAI).
2. Integrácia orientovaná na biznis procesy (BPOAI).
3. Orientované na služby (SOAI).
4. Portálovo orientované (POAI).
5. Podniková zbernica služieb (ESB – Enterprise Service Bus).

Špecialisti zaoberajúci sa informačne orientovanou integráciou odporúčajú integráciu na úrovni databáz alebo nejakého proprietárneho API. Spomedzi informačne orientovanej integrácie existuje viacero prístupov. Tieto je možné rozdeliť do troch skupín:

- Replikácia dát – prenášanie dát medzi databázami.
- Federácia dát – integrácia viacerých databáz do jedinej unifikovanej virtuálnej databázy.
- Spracovanie rozhraní – integrácia uzavretých aplikácií pomocou dobre definovaného rozhrania. Ako príklad posluží integrácia ERP systémov SAP, Peoplesoft a Oracle.

Integrácia pomocou biznis procesov pozostáva z vytvorenia novej vrstvy jednoducho definovaných a centrálnie spravovaných procesov, zapúzdrujúcich jednotlivé už existujúce procesy a informácie obsiahnuté v samotných aplikáciách, ktoré sa snažíme integrovať.

Servisne orientovaná integrácia umožňuje aplikáciám zdieľať bežnú biznis logiku a metódy. Toto je umožnené definovaním metód, ktoré môžu byť zdieľané a tým pádom integrované, alebo vytvorením infraštruktúry, ktorá by umožňovala zdieľanie metód v podobe webových služieb. Servisne orientovaná integrácia aplikácií poskytuje mechanizmus na vytvorenie kompozitných aplikácií ovplyvňujúcich služby nachádzajúce sa v mnohých vzdialených aplikáciách.

Portálovo orientovaná integrácia umožňuje vidieť viacero navzájom rôznych systémov cez jednotné užívateľské rozhranie.

ESB (Enterprise Service Bus – Podniková zbernica služieb) [54] je najnovšia generácia EAI (Enterprise Application Integration). ESB neimplementuje SOA [68], ale poskytuje vlastnosti, s využitím ktorých je možné implementovať SOA. V súčasnosti existuje konsenzus, že ESB je súčasťou architektúry SOA. SOA nie je technológia ani produkt, je to štýl návrhu s mnohými aspektmi (týkajúcimi sa architektúry, metodológie a ďalších), ktoré nie sú spojené s aktuálnou technológiou. ESB rieši problém mapovania abstraktnej SOA na konkrétnu implementáciu. Medzi kľúčové funkcie ESB je možné zaradiť nasledujúce [54]:

- Vyvolanie (Invocation).
 - Podpora synchrónnych a asynchrónnych komunikačných protokolov.
- Smerovanie.
 - Adresovateľnosť, smerovanie na základe obsahu.
- Sprostredkovanie.
 - Adaptéry, preklad protokolov, transformácie a preklad dát.
- Inštrumentácia procesov.
 - Definícia biznis procesov.
- Komplexné spracovanie udalostí.
 - Interpretácia udalostí, vzájomný vzťah udalostí, porovnanie vzorov, publish/subscribe.
- Kvalita služieb.
 - Bezpečnosť, spoľahlivý prenos, transakcie.
- Manažment.
 - Monitorovanie (aj biznis procesov), audit, logovanie, meranie, administrátorské rozhranie.

ESB je možné použiť k integrácii samotných BI komponentov, ako aj k integrácii výsledného BI riešenia s ERP systémom. Príklad takejto integrácie predstavuje projekt SpagoBI [45], ktorý sa snaží vytvoriť integračnú platformu založenú na integrácii open source BI projektov a ďalších open source projektov, ako je Drools a Lucene.

Tabuľka 6.1 sumarizuje výhody a nevýhody jednotlivých prístupov [2].

Prístup	Výhody	Nevýhody
replikácia dát	<ul style="list-style-type: none"> ● jednoduchosť ● nízka cena ● jednoduchá implementácia ● existujúce lacné riešenia 	<ul style="list-style-type: none"> ● nevhodné, ak je potrebné spojiť dáta s metódami alebo ak metódy zdieľajú dáta; v takom prípade je vhodné zvážiť integráciu orientovanú na služby ● môže vyžadovať zmeny zdrojových a cieľových systémov
federácia dát	<ul style="list-style-type: none"> ● umožňuje spojiť rôzne typy dát do jednotného modelu, ktorý podporuje výmenu informácií ● umožňuje prístup k existujúcim databázam cez jednotné, dobre definované rozhranie ● nevyžaduje zmeny zdrojových a cieľových systémov 	<ul style="list-style-type: none"> ● riešenie zvládnuté len nedávno – je potrebný middleware

Tabuľka 6.1: Prístupy podnikovej integrácie aplikácií

spracovanie rozhraní	<ul style="list-style-type: none"> ● efektívna integrácia mnohých rozdielnych typov aplikácií ● ak už sú implementované adaptéry integrovaných aplikácií, je možné integráciu uskutočniť i behom niekoľkých dní 	<ul style="list-style-type: none"> ● malý ohľad na biznis logiku a metódy medzi zdrojovými a cieľovými systémami; ak je toto považované za problém, je vhodné zvážiť integráciu orientovanú na služby
integrácia orientovaná na biznis procesy	<ul style="list-style-type: none"> ● vizualizácia procesov ● monitorovanie stavu procesov ● možnosť redefinovania procesu kedykoľvek v čase (real-time) ● možnosť merania výkonnosti procesov ● logika procesu je oddelená od aplikačnej logiky 	<ul style="list-style-type: none"> ● adresuje len procesný tok a integráciu
integrácia orientovaná na služby	<ul style="list-style-type: none"> ● znovu použiteľnosť 	<ul style="list-style-type: none"> ● finančne náročná ● vyžaduje zmeny zdrojových a cieľových systémov

Tabuľka 6.1: Prístupy podnikovej integrácie aplikácií pokračovanie

portálovo orientovaná integrácia	<ul style="list-style-type: none"> • nevyžaduje zmeny zdrojových a cieľových systémov 	<ul style="list-style-type: none"> • nie je zameraná na výmenu informácií v reálnom čase • vyžaduje asistenciu koncového užívateľa
podniková zbernica služieb	<ul style="list-style-type: none"> • nevyžaduje zmeny zdrojových systémov • pomocou podnikovej zbernice služieb je možné implementovať SOA 	<ul style="list-style-type: none"> • závislá na dodávateľovi – toto nie je problém v prípade open source projektov (Mule, ServiceMix) • extra vrstva pre preklad správ na rozdiel od tradičných riešení • nový prístup, málo skúseností

Tabuľka 6.1: Prístupy podnikovej integrácie aplikácií pokračovanie

Typická BI aplikácia vyžaduje dáta z mnohých zdrojov. Čím viac takýchto dátových zdrojov aplikácia vyžaduje, tým komplikovanejšia je jej implementácia a udržanie jednotného pohľadu na dáta. S touto situáciou sa dokáže najlepšie vysporiadať dátový sklad, ktorý predstavuje jednotný pohľad na dáta. Tieto dáta z rôznych zdrojov majú rôzne formáty a musia prejsť najprv procesom mnohých úprav ako je čistenie, transformácia a extrahovanie zo starých aplikácií s proprietárnym formátom dát a až potom sú pripravené na vloženie do dátového skladu. Tieto činnosti zabezpečujú tzv. ETL aplikácie. Dátovej časti integrácie odpovedá prístup IOAI, konkrétne replikácia dát a spracovanie rozhraní pre niektoré už existujúce ERP systémy. Federácia dát sa javí ako čisté, ale na druhej strane komplikované riešenie s neistým výsledkom, a to hlavne kvôli problémom s reprezentáciou dát a neexistenciou jednotného formátu dát. To môže spôsobovať značné problémy s ďalším spracovaním, ako aj výkonnostné problémy aplikácie postavenej nad takouto virtuálnou databázou. V prípade IOAI nám nevadia zmeny cieľových systémov, keďže tieto predstavujú dátový sklad, ktorý je súčasťou výslednej aplikácie. Zmeny zdrojových systémov môžu predstavovať určitý problém, ale v súčasnosti existuje viacero riešení pre rôzne takéto prípady.

ERP systémy, ako z uvedenej definície vyplýva, pozostávajú z množstva integrovaných informačných systémov. Informačný systém môže predstavovať aj dokument napísaný v tabuľkovom procesore. Každý ERP systém je jedinečný a rôzne ERP systémy sa líšia prakticky vo všetkých možných parametroch, ako je napríklad

počet jednotlivých IS, metódy integrácie jednotlivých IS, jednotlivé architektúry a implementačné platformy IS, ako aj prostredie v ktorom dané aplikácie bežia. Veľmi dôležitým parametrom je tiež množstvo dát, ich kvalita, čo zahŕňa okrem iného aj úplnosť dát a dostupnosť - umiestnenie. Jadrom väčšiny BI aplikácií je analýza veľkého množstva dát, ich premena v užitočné informácie a nakoniec ich prezentácia v určitej forme. Teda prvým krokom v integrácii BI aplikácie do ERP je získať tieto dáta z jednotlivých systémov a transformovať ich do štruktúry použiteľnej pre ďalšie spracovanie.

Pri analýze možností integrácie BI IS do ERP, je nutné rozlišovať dva základné prípady:

1. Integrácia hotovej BI aplikácie a jej customizácia.
2. Integrácia BI aplikácie vyvíjanej od začiatku.

Pri integrácii už hotovej BI aplikácie sú k dispozícii presne určené možnosti integrácie samotným producentom danej aplikácie, a teda v tomto prípade je cesta integrácie už konkrétne vytýčená. Zvyčajne takéto aplikácie požadujú nejakú konkrétnu štruktúru dát a poskytujú API použiteľné pre implementáciu integračného mostu medzi existujúcimi informačnými systémami a BI aplikáciou, ako aj adaptéry pre integráciu s najrozšírejšími ERP riešeniami (napr. SAP).

6.2 Integrácia BI aplikácie vyvíjanej od začiatku

V prípade vývoja BI aplikácie od začiatku je k dispozícii podstatne viacej možností ako v prípade integrácie hotovej aplikácie. Integrácia BI aplikácie je možná na viacerých úrovniach:

- Integrácia vstupov BI aplikácie.
- Integrácia výstupov BI aplikácie.
- Integrácia procesov BI aplikácie.

6.2.1 Integrácia vstupov

Najstarším a najpoužívanejším riešením integrácie vstupných dát je IOAI. Predstaviteľom nástrojov IOAI sú ETL nástroje, ktoré dokážu integrovať dáta prakticky z ľubovoľného zdroja, či už to sú databázy alebo napríklad dokumenty vytvorené pomocou tabuľkového procesora. Z niektorých systémov, kde nie je možný prístup k dátam uloženým v databáze, je umožnené tieto dáta získať napríklad pomocou webových služieb, teda SOAI. Toto riešenie nie je až také obvyklé [2] ako IOAI, ale je v súčasnosti už technicky zvládnuté a existujú aj praktické skúsenosti. SOAI má radu

nevýhod, ako je nutnosť zmeny zdrojových a cieľových systémov, vysoká cena, ako aj pomerná novosť celej technológie. Typickými problémami takýchto riešení je bezpečnosť, výkonnosť a spoľahlivosť. Niektoré z týchto problémov sa snaží riešiť prístup podnikovej zbernice služieb.

V prípade projektu implementovaného v rámci tejto diplomovej práce je integrácia vstupných dát riešená pomocou ETL nástroja čo znamená informačne orientovanú integráciu. V pravidelných intervaloch prebieha integrácia vstupných dát do dátového skladu. Použitie ETL je bližšie popísané v kapitole 8.3.4.

6.2.2 Integrácia výstupov

Výstupy procesov a algoritmov BI aplikácií predstavujú:

- Desktopové rozhranie.
- Webové rozhranie.
- Reporty (Word, Excel, PDF, ...).
- Grafy.
- Štruktúrované dáta (napríklad v podobe tabuliek alebo v dopredu dohodnutej štruktúre).

Z týchto možností je zaujímavá integrácia webového rozhrania BI aplikácie do existujúceho rozhrania ERP systému. Predpokladajme, že ERP systém, do ktorého chceme integrovať nový BI IS, už má nejaké webové rozhranie, či už vo forme samostatných webových rozhraní jednotlivých informačných systémov alebo nejakej formy portálového riešenia. Pri integrácii jednej webovej aplikácie do druhej, je nutné vyriešiť viacero problémov. Medzi najvýznamnejšie patria [98]:

- Vzťahy medzi aplikáciami.
- Komunikácia medzi aplikáciami.
- Manažment užívateľov a skupín.
- Manažment zasadania (session management).
- Autentifikácia.
- Kontrolovanie prístupu.
- Manažment práv.
- Vizualna integrácia.
- Vyhľadávanie.
- Konzistentné prostredie (lokalizácia, kódovanie súborov, ...).

Je to mnoho bodov, ale pri integrácii viacerých webových aplikácií je nutné si rozmyslieť a zvoliť tie správne postupy vhodné pre to konkrétne riešenie. Pri integrácii sa môžeme riadiť jedným z viacerých princípov – vytvoriť API a kód spájajúci dané aplikácie pri zachovaní samostatnosti týchto aplikácií, spojiť tieto aplikácie do jednej, udržiavať aplikácie v synchrónnom stave, a to buď v každom okamihu alebo dávkovo, alebo riadiť sa princípom minimálneho kódu – čím menej integračného kódu, tým menej času zaberie jeho napísanie a tým menej času bude potreba na jeho udržiavanie. Vzťahom medzi dvoma aplikáciami sa rozumie buď rovnocenný alebo nerovnocenný vzťah. Nerovnocenný vzťah sa tiež označuje ako master-slave vzťah. Rovnocenný vzťah znamená, že obidve aplikácie sa navzájom ovplyvňujú pomocou príkazov (napríklad správ). Nerovnocenný zase znamená, že jedna aplikácia generuje príkazy pre druhú, ktorá ich len prijíma a plní. Jedná sa teda o jednosmernú komunikáciu. Komunikácia medzi dvoma aplikáciami môže byť synchrónna, asynchrónna, uskutočnená nejakým protokolom (napr. HTML, FTP), pomocou správ, udalostí, cez middleware [53] a mnohými ďalšími spôsobmi. Manažment užívateľov a skupín môže byť uskutočnený cez spoločnú databázu alebo cez samostatné riešenia pre každú aplikáciu, pričom tieto dáta musia byť nejakým spôsobom synchronizované. Pre session manažment platí to isté ako pre manažment užívateľov a skupín, teda že môže byť uskutočnený cez nejaký spoločný mechanizmus alebo samostatne – tu nastupuje synchronizácia (napr. session timeouty). Štandardným spôsobom autentifikácie vo viacerých systémoch je metóda single-sign-on. Teda, že užívateľ sa autentifikuje len raz a nemusí sa autentifikovať do každého systému zvlášť. Vizualná integrácia znamená spôsob, akým budú jednotlivé aplikácie do seba vizuálne zapadať, aké šablóny používať. Cieľom je, aby v konečnom dôsledku užívateľ pri prechode z jedného systému do druhého (napr. cez webový odkaz) nepostrehol, že komunikuje s iným systémom (transparentný prechod). Manažment práv je veľmi dôležitý aspekt. Existuje viacero prístupov. Asi najrozšírenejšie sú nasledujúce spôsoby – samostatné riešenie v každej aplikácii, spoločné riešenie v jednej z aplikácií – druhá sa dotazuje, a napokon riešenie pomocou nejakej externej aplikácie spravujúcej práva. Kontrolovaním prístupu sa myslí to, či má daná aplikácia bežať za firewallom, v intranete a podobne. Udržiavať konzistentné prostredie naprieč aplikáciami je vskutku náročná úloha. Ako príklad prostredia, ktoré je potrebné udržiavať uvediem nastavenie jazyka. Keď užívateľ prechádza z jednej aplikácie do druhej, malo by byť toto nastavenie zohľadňované.

V prípade projektu implementovaného v rámci tejto diplomovej práce bola uskutočnená integrácia dvoch webových aplikácií. Tieto aplikácie (a ich integrácia) sú bližšie popísané v kapitole 8.3.7.

7. Návrh rozšírenia IS

Z predchádzajúcej kapitoly vyplýva, že rozšírenie IS o prostriedky podpory biznisu je možné realizovať vo viacerých krokoch. Prvým krokom je nutnosť získania dát pre BI aplikáciu – integrácia dát zdrojových systémov do dátového skladu BI aplikácie. Druhým krokom je integrácia výsledkov BI aplikácie (grafy, analýzy, varovania, ...) do existujúceho informačného systému, konkrétne ERP (množina IS).

V tejto kapitole rozoberieme všeobecne možnosti, ktoré sú k dispozícii pri návrhu jednotlivých rozšírení IS o prvky BI, pričom v nasledujúcej kapitole sa budeme zaoberať konkrétnym zadaním, návrhom rozšírenia IS o prvky BI a realizáciou tohto rozšírenia.

7.1 Fázy návrhu rozšírenia IS o prvky BI

Pri návrhu rozšírenia IS o prvky BI je možné postupovať viacerými metodológiami. Samotné spoločnosti zaoberajúce sa implementáciou BI sa buď riadia nejakou štandardnou metodológiou ako je vodopád, špirála, RUP, alebo si vyvinuli nejakú vlastnú metodológiu na základe vlastných skúseností. Zrejme nemá zmysel popisovať jednotlivé štandardné metodológie, keďže sú všeobecne známe. Bližšie informácie o jednotlivých metodológiách sú dostupné napríklad na stránkach wikipédie [12]. Rozšírenie IS o prvky BI je vhodné realizovať v nasledujúcich fázach:

- Fáza zdôvodňovania oprávnenosti projektu.
 - Hodnotenie biznis prípadu – cena verzus zisk (ROI).
- Fáza plánovania.
 - Vyhodnotenie existujúcej infraštruktúry IS – technická (softvér, hardvér, ...) aj netechnická (štandardy, metodológie, ...) infraštruktúra.
 - Plánovanie projektu vo forme zdrojov – čas, peniaze, ľudské zdroje.
- Fáza biznis analýzy.
 - Definícia projektových požiadaviek.
 - Dátová analýza (napr. analýza súčasnej kvality dát) – veľmi časovo náročné.
 - Vytvorenie aplikačného prototypu – prostriedok systémovej analýzy – umožňuje vidieť potenciál a limity technológie, často vedie k vyjasneniu funkčných požiadaviek.
 - Analýza úložiska metadát – určenie požiadaviek na metadáta (biznis a technické metadáta) a ich zdokumentovanie.
- Fáza návrhu.

- Návrh dátového skladu – musí zodpovedať požiadavkám na prístup k informáciám.
- Návrh ETL – ETL proces je často veľmi komplikovaný, často z dôsledku malej kvality zdrojových dát.
- Návrh úložiska metadát – návrh musí spĺňať požiadavky na logický model metadát, môže byť založený na ER vzťahoch alebo objektovo orientovaný.
- Fáza implementácie.
 - Implementácia ETL – aj pri zakúpení ETL nástrojov je často nutné implementovať niektoré časti ETL procesu.
 - Implementácia prvkov BI – zvyčajne pokryté existujúcimi BI aplikáciami.
 - Dolovanie dát – skutočná hodnota dát môže byť objavená len nástrojmi dolovania dát – zvyčajne pokryté existujúcimi BI aplikáciami.
 - Implementácia úložiska metadát – ak nepadne rozhodnutie na využitie hotového riešenia, je nutná implementácia vlastného riešenia.
- Fáza nasadenia.
 - Školenie užívateľov.
 - Údržba integračnej časti BI aplikácie – udržovanie databáz, realizácia dávkových ETL procedúr, monitorovanie výkonnosti a vyladovanie výkonu.
 - Vytváranie releasov – vylepšovanie aplikácie, upravovanie funkcionalít na základe spätnej väzby od užívateľov.

7.2 Návrh dátového skladu

Podrobnejšie sa budeme zaoberať fázami návrhu a implementácie. Pre úplnosť sa okrajovo zmienime o návrhu dátového skladu. K popisu návrhu dátového skladu je nutné priblížiť pojem dimenzie. Dimenzia definuje vzťahy typu rodič-potomok medzi množinami stĺpcov, pričom všetky stĺpce z danej množiny stĺpcov patria do rovnakej tabuľky. Príklad dimenzie predstavuje dimenzia čas, kde vzťahy rodič-potomok predstavujú napríklad vzťahy rok-mesiac, mesiac-týždeň. Návrh dátového skladu pozostáva z troch základných činností [11]:

- Multidimenzionálna analýza a modelovanie.
 - Analýza návrhu obsahu riešenia na základe určenia sledovaných ukazovateľov dimenzií a ich vzájomných väzieb.

- Modelovanie dátového skladu.
 - Vytvorenie logického modelu založeného na bežných princípoch dátového modelovania, ale s rešpektovaním charakteru databázových schém využívaných v dátových skladoch a trhoviskách (STAR, SNOWFLAKE).
- Riešenie zmien v dimenziách.
 - Čisto špecifická záležitosť u dátových skladov a trhovísk spojená s faktorom času – riešenie porovnateľnosti výsledkov aplikácií pri rešpektovaní zmien v obsahu dimenzií.

7.2.1 Multidimenzionálna analýza a modelovanie

Multidimenzionálne modelovanie je jadrom modelovania a návrhu pri riešení úloh BI. Po tom, čo sú zhromaždené užívateľské požiadavky na dátový sklad a zmapované zdrojové systémy, je možné pristúpiť k návrhu dimenzionálneho modelu. Multidimenzionálne modelovanie definuje všetky dimenzie, ich obsah, vrátane vnútornej hierarchie prvkov, a čiastočnej charakteristiky jednotlivých dimenzií. Tiež určuje sústavu sledovaných ukazovateľov a špecifikuje väzby medzi ukazovateľmi a odpovedajúcimi dimenziami.

Je možné vymedziť určitý všeobecný postup dimenzionálneho modelovania:

1. Návrh dimenzií, riešenie faktoru času.
2. Návrh ukazovateľov – často sú označované ako funkcie.
3. Riešenie väzieb medzi ukazovateľmi a dimenziami.
4. Premietnutie riešenia do návrhu tabuliek dimenzií, návrhu tabuliek faktov, návrhu schém (SNOWFLAKE, STAR) – návrh schém nie je nutný v prípade MOLAP, ktoré sú v proprietárnom formáte.

Dimenzionálne modelovanie prináša možnosť prezentovať dáta na ľubovoľnej úrovni agregácie, kombinovať jednotlivé dimenzie a tým prinášať nové uhly pohľadu na dáta a aplikovať najrôznejšie aritmetické a množinové operácie, agregáčn é i štatistické funkcie.

7.2.2 Modelovanie dátového skladu

Modelovanie dátových skladov a trhovísk je založené na nasledujúcich typoch tabuliek vo vzťahu k OLAP kockám:

- Tabuľka faktov.
 - Reprezentuje jednotlivé ukazovatele a ich súradnice - bunky OLAP kocky.
- Tabuľky dimenzií.
 - Reprezentujú dimenzie, ich prvky a ďalšie atribúty - hrany OLAP kocky.

Riešenie vzťahov medzi dimenziami a faktovou tabuľkou môže byť založené na dvoch základných schémach – STAR a SNOWFLAKE. Podrobný popis týchto schém je možné nájsť napríklad v publikáciách Inmon [9] a Kimball [13].

7.2.3 Riešenie zmien v dimenziách

Jedným z problémov, ktorý sa musí v BI riešiť, sú zmeny v dimenzionálnych tabuľkách v priebehu životného cyklu aplikácie. Tieto zmeny sú označované termínom SCD, tzv. Slowly Changing Dimensions.

SCD podľa Vlčka [14] predstavuje zmeny v štruktúre a v prvkoch dimenzií, teda u číselníkov, v čase. Pri riešení úloh BI existuje reálna potreba zachovať konzistenciu dát z časového hľadiska. V tomto kontexte môže dôjsť ku nasledujúcim zmenám prvkov v dimenziách i v hierarchii dimenzií [11]:

- Pridanie nových prvkov.
- Zrušenie prvkov.
- Zmeny v názvoch prvkov.
- Zmeny v štruktúre prvkov, napríklad zatriedenie do vyšších skupín.

Históriu v dátach je možné uchovávať dvoma spôsobmi:

- V rámci tabuľky faktov.
 - Atribút, ktorý má byť zmenený, sa presunie z tabuľky dimenzie do tabuľky faktov.
- Historickými číselníkmi.
 - Číselník obsahuje atribúty určené k sledovaniu histórie – CONST_ID, VAR_ID, DATUM_OD, DATUM_DO. Teda identifikátor, ktorý sa s časom nemení – CONST_ID, identifikátor, ktorý sa mení s časom – VAR_ID, ktorý predstavuje alternatívny primárny kľúč spolu s atribútmi platnosti záznamu DATUM_OD a DATUM_DO.

7.3 Návrh ETL

Návrh ETL obsahuje najmä:

- Návrh algoritmov pre transformácie zdrojových dát do cieľových databáz.
- Integráciu dát z viacerých zdrojov a vyriešenie rozporu medzi nimi tak, aby bolo možné zaručiť presnosť výstupných analýz a reportov, ktoré sú z nich vytvárané.
- Kontrolu extrahovaných dát, aby bolo možné zaručiť presnosť výstupných analýz a reportov.
- Opravu nekonzistentných dát [13, 81].
- Vytvorenie celoorganizačných dátových štandardov po stránke sémantickej tak i syntaktickej.
- Uplatnenie štandardov alebo odborných postupov a testov kvality aj na externé dáta.

Významnou súčasťou riešenia ETL sú metadáta. Metadáta u každého transformovaného záznamu poskytujú informácie o jeho význame, obsahu a ďalších charakteristikách. Použitím metadát sa získavajú dostupné informácie o zdrojových systémoch, a pre návrh transformácií zo zdroja do dátového skladu je ich potom možné efektívne využiť.

Pri návrhu ETL je nutné vyriešiť viacero problémov. Týmito problémami sú najčastejšie záležitosti kvality vstupných dát [11]:

- Rôzne dátové zdroje a transakčné systémy obsahujú rôzne číselníky predstavujúce rovnakú doménu.
- Neexistujú alebo nie sú definované primárne kľúče.
- Neúplné dáta – NULLy na dôležitých miestach.
- Dáta nie sú v 3. normálnej forme.
- Jeden stĺpec v tabuľke zahŕňa viacej atribútov.
- Je problematické vytvárať spojenia nad tabuľkami, podmienky spojenia sú príliš zložité a vedú k nesprávnym výsledkom, respektíve agregáciám.
- Konverzia znakových sád.

Z dôvodu týchto problémov väčšinou nie je reálne použitie hromadného spracovania pomocou SQL, ale je nutné implementovať transformačné procedúry pomocou nástrojov ETL, skriptov ako je PERL, alebo nejakého 4GL jazyka. Tieto transformačné procedúry prebiehajú zvyčajne vo viacerých krokoch:

- Samotné skopírovanie zdrojových dát do prostredia dátového skladu.
- Transformácie dát do previazaných tabuliek dátového skladu s rešpektovanou referenčnou a doménovou integritou.
- Vytváranie agregácií a ďalších kalkulácií dát pre účely náročnejších analýz.

V rámci ETL návrhu je ďalej vhodné definovať nasledujúce funkcie:

- Premietanie zmien v dimenziách.
- Uchovávanie histórie uskutočnených zmien v prvkoch a štruktúre dimenzií.
- Normalizáciu dát, teda transformácie podľa princípov SNOWFLAKE schém alebo denormalizáciu dát podľa STAR schém.
- Zaistenie konzistencie dát v tabuľkách dimenzií.
- Zaistenie inkrementálneho dopĺňovania časových snímok dát do databázy.
- Riadenie zmien v tabuľkách faktov.
- Zaistenie referenčnej integrity faktovej tabuľky s tabuľkami dimenzií.

Je nutné definovať primárne kľúče vo všetkých zdrojových tabuľkách, teda aj tam, kde chýbajú [11]. Tiež je nutné definovať kompletnú doménovú a referenčnú integritu a riešiť históriu dát, najmä históriu tabuliek dimenzií. Samozrejmosťou musí byť protokolovanie záznamov o všetkých zmenách vo forme metadát a dokumentov. Ďalšie informácie ohľadom návrhu ETL je možné čerpať z publikácií Inmon a Kimball [9, 13].

V prípade systému implementovaného v rámci tejto diplomovej práce je transformácia dát riešená pomocou softvéru Oracle Warehouse Builder [81]. Návrh dátových púmp je realizovaný v grafickom prostredí. Všetky zmeny v návrhu sú uložené v podobe metadát. Metadáta sú uložené v špeciálnom úložisku metadát v databáze Oracle. Tieto metadáta predstavuje XML dokument. Nástroj Oracle Warehouse Builder obsahuje podporu pre verzovanie metadát, porovnanie verzií a vytváranie snapshotov, čo sú obrazy metadát v danom čase. Na základe vytvorených metadát sú vygenerované optimalizované skripty pre transformáciu zdrojových dát do dátového skladu v požadovanom formáte.

7.4 Návrh úložiska metadát

Metadáta sú významnou súčasťou integračného riešenia, pretože databázové systémy a množstvo ďalších nástrojov bez nich nedokáže plniť svoju funkciu. Metadáta sú súčasťou jazyka databáz. Dávajú inštrukcie databázovým systémom, aký typ dátovej štruktúry vytvoriť, ETL nástrojom, aké dáta transformovať, OLAP nástrojom dávajú inštrukcie o tom, akým spôsobom agregovať dáta. Rôzne metadáta sú uložené v rozličných nástrojoch, a žiaden nástroj nie je navrhnutý tak, aby uložil všetky možné metadáta z ostatných nástrojov. Je veľmi náročné udržiavať metadáta roztrúsené v rôznych aplikáciách. Riešením tohto problému je vybudovať úložisko metadát na úrovni celej organizácie. Toho je možné dosiahnuť tromi nasledujúcimi spôsobmi:

1. Jedna centralizovaná databáza predstavujúca úložisko metadát.
2. Fyzicky distribuované metadáta vo viacerých databázach s jedným centrálnym metamodelom.
3. Distribuované riešenie založené na XML, kde metadáta sú uložené na ich pôvodnom mieste (v rôznych nástrojoch ako ETL – napríklad Oracle Warehouse Builder, CASE - napríklad Enterprise Architect, ...).

7.4.1 Centralizované úložisko metadát

Toto riešenie je najpopulárnejšie a najjednoduchšie na implementáciu, keďže tu figuruje len jedna databáza, či už relačná alebo objektovo orientovaná, a len jedna aplikácia, ktorú je nutné udržiavať. Centralizované úložisko metadát môže byť buď implementované od začiatku, alebo kúpené ako licencovaný softvér. V nasledujúcich sekciách sú popísané výhody a nevýhody jednotlivých možností.

Implementácia od začiatku – výhody

- Je možné navrhnuť databázu, ktorá obsahuje všetky požiadavky na metadáta.
- Front end a prístup k rozhraniam nástrojov (ETL, OLAP, ...) je navrhnutý s ohľadom na požiadavky.
- Reporty, ako aj pomocné funkcie sú navrhnuté presne tak, ako to je potrebné.

Implementácia od začiatku – nevýhody

- Na udržiavanie úložiska sú potrebné ľudské zdroje.
- Front end prístup k rozhraniam musí byť implementovaný a potom udržiavaný, čo je časovo náročné.
- Sú potrebné nové releasy – aby pokryli nové požiadavky na metadáta.

Licencované úložisko – výhody

- Ušetrený čas za vývoj a návrh úložiska.
- Menej riskantné ako vlastný vývoj.

Licencované úložisko – nevýhody

- Nie je šité na mieru – niektoré funkcie sú nadbytočné, niektoré funkcie chýbajú.
- Nutné zaškolenie pracovníkov.

7.4.2 Decentralizované úložisko metadát

Decentralizované úložisko metadát ukladá metadáta vo viacerých databázach na viacerých miestach. Často využívané metadáta je vhodné replikovať pozdĺž viacerých databáz, ale je nutné udržať tieto metadáta v konzistentnom stave. K samotnému úložisku sa potom pristupuje cez bránu, ktorá smeruje požiadavky do jednotlivých databáz. V nasledujúcich sekciách sú popísané výhody a nevýhody jednotlivých možností.

Výhody

- Rôzni vlastníci môžu oddelene udržiavať a spravovať svoje vlastné množiny metadát.
- Jednotlivé databázy sú menšie a jednoduchšie na používanie, pretože takéto databázy obsahujú len metadáta tých komponentov, ktoré sú zaujímavé len pre určitú špecifickú skupinu pracovníkov.
- Každá databáza môže mať svoj vlastný metamodel.
- Reporty môžu byť upravené individuálne pre jednotlivé databázy.
- Brána poskytuje názov a umiestnenie metadát transparentne pre osoby, ktoré k nim potrebujú prístup.

Nevýhody

- Kontrola duplicit pozdĺž viacerých databáz a udržiavanie metadát v konzistentnom stave je zložité.

- Udržiavanie a spravovanie viacerých databáz na viacerých platformách vyžaduje viac času. Taktiež tu môžu nastať problémy s prechodom na nové verzie databáz.
- Nutnosť zvýšenej komunikácie medzi správcami databáz. Nutnosť udržiavania metamodelu integráciou metamodelov jednotlivých databáz.
- Architektúra celého riešenia je viac komplikovaná a krivka učenia pre používanie viacerých databáz môže byť vysoká.

7.4.3 Distribuované riešenie založené na XML

Toto riešenie posúva koncept úložiska metadát zase o kúsok ďalej. Na rozdiel od predchádzajúceho riešenia, metadáta zostávajú na svojom originálnom mieste v slovníkoch rôznych nástrojov. Brána XML vystupuje ako adresár jednotlivých umiestnení nástrojov. Tým sa odstráni potreba dvojitého spravovania metadát, ako aj spravovanie oddelených databáz metadát. V nasledujúcich sekciách sú popísané výhody a nevýhody jednotlivých možností.

Výhody

- XML tagy umožňujú prístup k metadátam naprieč rôznymi typmi dátových úložísk cez štandardizovanú kategorizáciu a označkovanie (tagging) komponentov metadát.
- Metadáta nemusia byť duplikované a prenášané z ich originálnych zdrojov.
- Brána poskytuje transparentný prístup k metadátam.
- Štandardné webové prehľadávače, ako napríklad Google, by mali byť schopné nájsť metadáta uložené na ľubovoľnom mieste.
- Metadáta a biznis dáta môžu byť prenášané simultánne.

Nevýhody

- Prvotné označenie (tagging) všetkých metadát XML tagmi je manuálny a zdĺhavý proces.
- Model metadát musí byť vytvorený ako mapovanie množstva rôznych typov úložísk metadát.
- Aby nejaký databázový produkt alebo nástroj umožňoval jednoduchý prístup k metadátam, musel by podporovať nejaký štandard pre metadáta. Štandardy metadát sú vyvíjané napríklad skupinami Meta Data Coalition [55] a Object Management Group [56].
- Nie všetky databázy a nástroje podporujú XML.

7.4.4 Zvolené riešenie úložiska metadát

Metadáta využívané systémom, ktorý je implementovaný v tejto diplomovej práci sú uložené v databáze Oracle. Tieto metadáta pochádzajú z rôznych nástrojov spoločnosti Oracle ako je Oracle Warehouse Builder, Oracle Analytic Manager, Oracle JDeveloper a Oracle BI Beans [71]. Metadáta sú uložené v úložiskách metadát dodávaných spolu s daným softvérom. Prístup k týmto metadátam je možný buď priamo prostredníctvom databázy, špeciálneho API alebo s využitím špeciálneho softvéru ako je Repository Manager, ktorý je dodávaný s Oracle Warehouse Builder [71]. Verzovanie metadát je možné riešiť buď na úrovni databázy alebo pomocou už spomínaného softvéru. Veľmi príjemnou vlastnosťou týchto metadát je export do XML. V tomto prípade je možné spravovať a verzovať metadáta na úrovni XML dokumentov.

8. EPA

Táto kapitola obsahuje konkrétne požiadavky na rozšírenie už existujúceho informačného systému o prvky BI aplikácií a návrh technického riešenia takéhoto prípadu. Nasledujúce zadanie sa vzťahuje na projekt analýzy a monitorovania výkonnosti zamestnancov pre malú IT firmu, Flynet s.r.o. V ďalšom texte sa budeme odkazovať na daný projekt skratkou EPA (Employee Performance Analysis).

8.1 Zadanie projektu

Cieľom je rozšíriť podnikový informačný systém zložený z viacerých informačných systémov o prvky podpory biznisu. Medzi tieto informačné systémy patrí systém na sledovanie chýb Mantis [17] a systém pre dochádzku zamestnancov Devis [77], vyvinutý spoločnosťou Flynet. Rozšírený systém bude poskytovať podporu pre analýzu výkonnosti zamestnancov. Tento systém bude integrovať dáta zo systémov Mantis a Devis. Výsledný integrovaný systém bude obsahovať údaje o tom, ktorý zamestnanec bol kedy v práci a akú činnosť v danom čase vykonával (napr. tvorba dokumentácie, testovanie, implementácia). Súčasťou systému bude aj trasovanie hlásení problémov (HP, bugov), teda bude možné dohľadať, koľko HP za aké obdobie bolo znovu otvorených.

Účelom tohto softvéru nie je bič na zamestnancov, ale objasnenie a sledovanie stavu projektu, ako aj možnosť určitých predpovedí vývoja úloh do budúcnosti a následná možnosť na tento stav reagovať. Projekt prinesie pozitíva nielen pre zamestnávateľa, ale aj pre zamestnancov. Z lepšieho sledovania a predpovedí prácnosti práce budú zamestnanci dostávať rozumné množstvo práce a v dôsledku toho nebudú pracovať pod vysokým tlakom termínov. Z lepšieho plánovania a monitorovania tak vyplýva menej stresu pre zamestnancov a v nadväznosti na to aj menej chýb zavlečených v priebehu vývoja. Toto všetko prinesie lepšie možnosti riadenia projektov, ako aj zvýšenie kvality vyvíjaného softvéru. To bude mať za následok zvýšenie spokojnosti zákazníkov spoločnosti a napokon zamestnávateľa.

Systém bude poskytovať prostriedky na zistenie výkyvov výkonnosti zamestnancov, porovnanie zamestnancov v rôznych ukazovateľoch. Dôraz sa kladie na úlohy dané zamestnancom (prípadne opravy HP). Každá takáto úloha bude mať daný odhad prácnosti. Ku každej úlohe sa bude evidovať reálna miera prácnosti (čas strávená riešením úlohy konkrétnym zamestnancom). Úlohy budú rozdelené podľa typu, pre prípadné ďalšie analýzy. Hlavným cieľom systému je analýza plánovanej a reálnej prácnosti jednotlivých úloh.

Systém bude poskytovať prostriedky na podporu evidencie a zlepšenia pomeru predpovedanej a reálnej prácnosti úloh. Ďalším bodom je analýza procesov, ktoré

zamestnanci v priebehu vývoja vykonávajú a ich vplyv na prácnosť. Cieľom je zistiť, v ktorej časti ktorého procesu zamestnanec trávi najviac času, čo dáva priestor pre zefektívnenie daného procesu.

Systém bude obsahovať prostriedky pre dolovanie dát a teda pre zisťovanie vzorov, trendov a závislostí vo výkonnosti zamestnancov. Príkladom môže byť efektivita zamestnancov v závislosti na mieste výkonu práce alebo pokles/vzrast efektivity v závislosti na dovolenke, začiatok a koniec týždňa a podobne. V nasledujúcich podkapitolách rozoberieme základné funkcie rozširujúcich BI prvkov.

8.1.1 OLAP grafy

V nasledujúcich bodoch budeme v prípade HP uvažovať aj novú funkcionálnosť.

- Počet vyriešených HP vybraného zamestnanca za vybrané obdobie.
- Počet znovuo tvorených HP (teda HP vyriešených týmto zamestnancom) vybraného zamestnanca za vybrané obdobie.
- Celkový počet vyriešených HP vybraného zamestnanca.
- Celkový počet znovuo tvorených HP vybraného zamestnanca.
- Percentuálny prehľad činností vybraného zamestnanca za vybrané obdobie. Príklad: V máji 2006 Peter Novák venoval 10% času dokumentácii, 30% implementácii a 60% testovaniu.
- Percentuálny prehľad počtu riešených HP podľa náročnosti a ich vplyv na čas riešenia pre vybraného zamestnanca za vybrané obdobie.
- Počet vyriešených a nevyriešených HP v odhadovanom termíne vybraného zamestnanca za vybrané obdobie.
- Počet vyriešených HP podľa prekročeného času riešenia pre všetkých zamestnancov.

8.1.2 Porovnanie

Rovnako ako v predchádzajúcom texte, budeme v prípade HP uvažovať aj novú funkcionálnosť.

- Porovnanie vyriešených HP vybraných alebo všetkých zamestnancov za vybrané obdobie.
- Porovnanie znovuo tvorených HP vybraných alebo všetkých zamestnancov za vybrané obdobie.

- Porovnanie celkového počtu vyriešených HP vybraných alebo všetkých zamestnancov za vybrané obdobie.
- Porovnanie celkového počtu znovuotvorených HP vybraných alebo všetkých zamestnancov za vybrané obdobie.
- Porovnanie činností v percentách vybraných alebo všetkých zamestnancov za vybrané obdobie.
- Porovnanie počtu vyriešených HP (aj v percentách) a času ich riešenia pre vybraných alebo všetkých zamestnancov za vybrané obdobie.
- Porovnanie počtu vyriešených a nevyriešených HP v odhadovanom termíne vybraných alebo všetkých zamestnancov za vybrané obdobie.
- Porovnanie počtu vyriešených HP podľa prekročeného času riešenia pre vybraných alebo všetkých zamestnancov za vybrané obdobie.

Jednou z podmienok tohto projektu sú nízke náklady na technológie a nástroje, s výnimkou niektorých technológií spoločnosti Oracle. Tým pádom do úvahy spadajú open source nástroje a vybrané softvérové produkty spoločnosti Oracle. Keďže firma Flynet je malá firma, nemá koncept oddelení ako veľké firmy a tým pádom sa diskusia na téma dátové trhoviská verzus jeden dátový sklad redukuje len na dátový sklad. Teda riešenie BI sa bude sústreďovať okolo jediného centrálného dátového skladu.

8.2 Prínos projektu

Motiváciou projektu EPA sú problémy s vývojom predchádzajúcich a súčasných projektov spoločnosti Flynet, kde významné percento projektov prekročilo plánovanú prácnosť. Prekročenie prácnosti predstavuje veľký problém pri predom dohodnutej prácnosti projektu so zákazníkom. Systém EPA je výsledkom snahy zlepšiť vývojový proces v spoločnosti Flynet a má viesť k väčšej spokojnosti zákazníkov, ale aj zamestnancov spoločnosti Flynet. Riešenie systému EPA bolo zvolené na základe špecifického prostredia spoločnosti Flynet, kde nie je vhodné použitie súčasných riešení existujúcich na trhu (príkladom takéhoto riešenia je napríklad Performix Performance Manager [88], ktorý využíva v USA približne 120 firiem s obratom väčším ako miliarda dolárov) a to buď z dôvodu ceny alebo z dôvodu nesúlady s potrebami spoločnosti Flynet.

Na základe analýzy súčasnej výkonnosti zamestnancov je možné vysledovať závislosti v dátach, ako aj zmenu kvality a výkonnosti vybraných zamestnancov za vybrané obdobie a na tieto zmeny následne reagovať. Kvalitu práce zamestnanca – programátora je možné objektívne merať na základe metrík. Jednou z takýchto metrík je napríklad počet zistených, respektíve nahlásených chýb v kóde, ktorý vytvoril daný zamestnanec. Efektivitu práce je možné merať na základe priemerného času alebo mediánu stráveného na úlohe danej náročnosti so zachovaním určitej kvality. Problémom takýchto meraní je subjektivita niektorých vstupných parametrov. Príkladom takéhoto parametra je náročnosť úlohy. Náročnosť úlohy je možné čiastočne objektivizovať na základe systematizácie a klasifikácie atribútov úlohy. Napríklad

náročnosť úlohy je možné klasifikovať počtom java tried, ktoré bude nutné zmeniť alebo vytvoriť, technológiou, ktorá bude na úlohu použitá, prípadne či je nutné implementovať nový algoritmus alebo sa použije už existujúce riešenie a ďalšími parametrami. Náročnosť by sa následne stanovovala automaticky na základe takýchto parametrov konkrétnym algoritmom. Predpokladom analýzy je dostatočne veľké množstvo vstupných dát ako aj ich kvalita, čo je závislé na prístupe jednotlivých členov projektu a hlavne vedúcich projektov a manažérov zodpovedných za vedenie projektov.

Zbieraním projektových dát a ich následnou analýzou formou dolovania dát je možné odhaliť niektoré skryté závislosti medzi dátami a aspoň čiastočne vysledovať príčiny poklesu, respektíve rastu produktivity alebo kvality práce jednotlivých zamestnancov. Na základe takto zistených skutočností je následne možné reagovať obmedzením alebo eliminovaním jednotlivých faktorov alebo naopak posilnením faktorov zvyšujúcich produktivitu práce.

Primárne ciele tohto projektu sú nasledujúce:

1. Zvýšenie produktivity práce zamestnancov.
2. Systematické a jednoduché monitorovanie efektivity práce programátorov.
3. Systematické ohodnotenie náročnosti úlohy a objektívna predpoveď prácnosti danej úlohy.
4. Zvýšenie obchodnej hodnoty systému Devis.

Vedľajšie ciele tohto projektu sú nasledujúce:

1. Získanie praktických skúseností s Business Intelligence s využitím moderných technológií.
2. Vytvorenie referenčného projektu využívajúceho prvky Business Intelligence pre účely prezentácie možností tejto technológie zákazníkom za účelom vývoja ďalších projektov využívajúcich podobné technológie.

Základným ukazovateľom prínosu projektu je jeho prínos vyčíslený v peniazoch. Nasledujúce údaje sú založené na údajoch a odhadoch spolumajiteľa firmy Flynet, Pavla Köhlera. Predpokladajme, že sa po nasadení projektu do prevádzky produktivita práce zvýši o 10% a že sa predpoveď prácnosti danej úlohy zlepší o 50%. Za predpokladu projektu, ktorého prácnosť bola pred nasadením systému EPA plánovaná na 100 človekodní, ale reálna prácnosť je 120 človekodní bez využitia systému EPA, bude následne s využitím systému EPA táto prácnosť predpovedaná na 110 človekodní. Vzhľadom k zvýšeniu produktivity práce zamestnancov sa reálna prácnosť zmení na 108 človekodní. Za predpokladu hodinových nákladov na zamestnanca 500 Sk, je cena projektu vyčíslená v tabuľke 8.2.

	Bez použitia systému EPA	S použitím systému EPA
Predpokladaná cena projektu	1 200 000 Sk	1 320 000 Sk
Skutočné náklady	1 440 000 Sk	1 296 000 Sk
Rozpor	-240 000 Sk	24 000 Sk

Tabuľka 8.2: Porovnanie nákladov na projekt s využitím systému EPA

Z tabuľky 8.2 vyplýva, že za uvedených predpokladov je prínos systému EPA pri vývoji stredne veľkého projektu 264 000 Sk. Náklady na údržbu projektu závisia na počte ľudskohodín strávených údržbou systému. Po odladení chýb systém EPA nebude náročný na údržbu. Za predpokladu 8 ľudskohodín strávených mesačne na údržbe, sú mesačné náklady na prevádzku systému 4 000 Sk. Systém EPA bude vyvíjaný pre ďalších zákazníkov a budú sa na neho uplatňovať vývojárske licencie na produkty spoločnosti Oracle, čo znamená minimálne licenčné náklady. Systém bude bežať na open source operačnom systéme, ďalšie použité knižnice sú distribuované pod licenciou LGPL (produkty Red Hat Middleware) alebo Apache License (produkty Apache Software Foundation) [85, 86]. Náklady na hardvér vybavenie predstavujú použitie staršieho servera s minimálne 1GB RAM.

8.3 Technická architektúra

Aby bolo možné analyzovať dáta o zamestnancoch, je potrebné vytvoriť jeden integrovaný pohľad na dáta a nad týmto pohľadom vytvárať potrebné analýzy. Integrovaný pohľad na dáta je možné dosiahnuť vytvorením dátového skladu. Tento dátový sklad sa bude plniť dátovými pumpami. K jednoduchším analýzám je vhodná technológia OLAP. K tým zložitejším je možné využiť technológiu dolovania dát. Výsledky analýz je vhodné prezentovať v podobe tabuliek a grafov. Medzi základné požiadavky moderného analytického softvéru nesporne patrí online prístup k analýzám prostredníctvom interaktívneho webového rozhrania. Štandardný spôsob integrácie spomínaných aplikácií predstavuje aplikačný server, ktorý spolu s webovým serverom poskytuje možnosti online prezentácie dynamických analýz. Na nasledujúcom obrázku je zobrazený diagram nasadenia, ktorý zobrazuje jednotlivé uzly výsledného podnikového informačného systému. Výsledná aplikácia sa skladá z nasledujúcich systémov:

- Dochádzkový systém Devis, ktorý plní mimo svojej pôvodnej funkcie aj funkciu poskytovateľa zdrojových dát pre BI komponenty.
- Systém na evidenciu chýb Mantis, ktorý tiež predstavuje zdroj dát pre BI komponenty.
- Databáza systému Mantis, ktorá obsahuje evidenciu chýb a úloh projektov.
- Databáza systému Devis, ktorá obsahuje dáta o dochádzke všetkých zamestnancov.

- Dátový sklad Oracle 10g, ktorý integruje dáta zo zdrojových IS a poskytuje dáta pre BI komponenty ako sú OLAP, dolovanie dát a úložisko dát pre webové rozhranie BI aplikácie.
- Nástroj ETL Oracle Warehouse Builder, ktorý v pravidelných intervaloch plní dátový sklad transakčnými dátami, ktoré medzičasom vznikli v systémoch Devis a Mantis.
- Aplikačný server JBoss 4.2, ktorý hostí aplikačnú logiku prístupu k dátam a externým systémom a webovú aplikáciu poskytujúcu webové rozhranie.

Na diagrame 8.3 je zobrazená výsledná architektúra aplikácie EPA pozostávajúca z už spomínaných subsystémov.

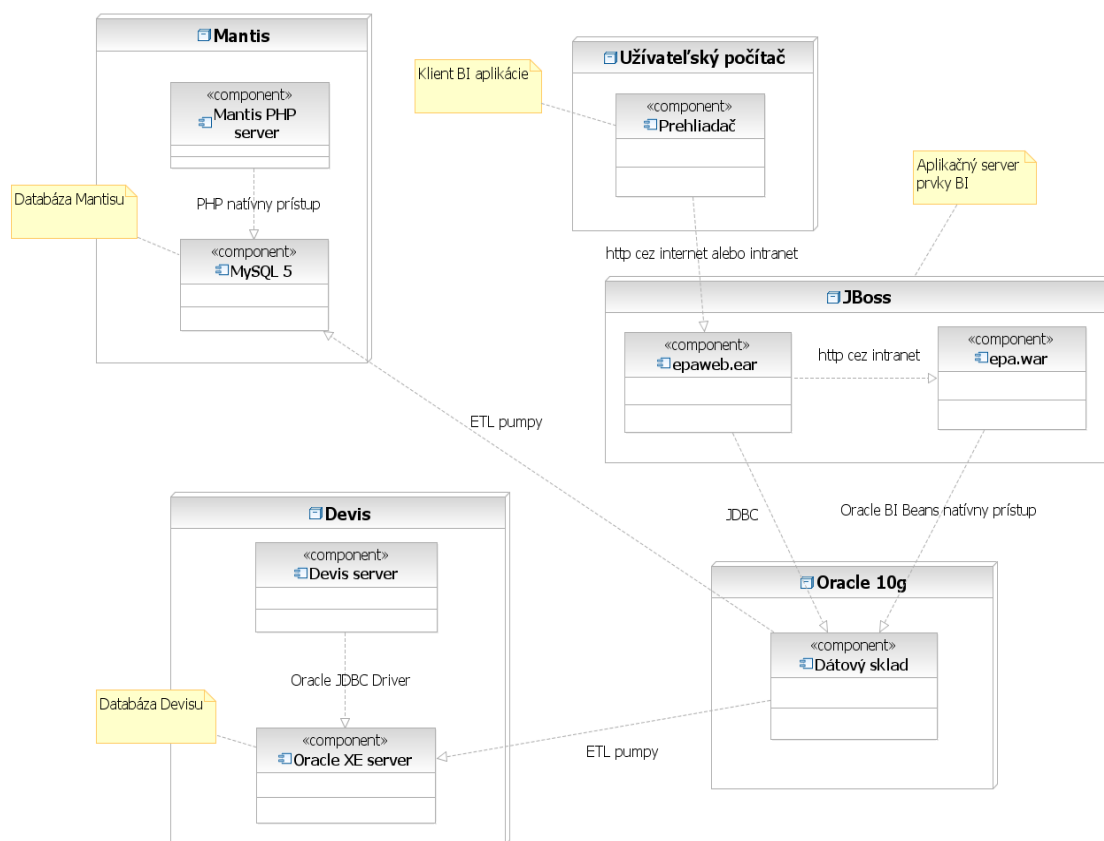


Diagram 8.3: Diagram nasadenia ERP

Systémy Devis a Mantis nie sú priamou súčasťou výsledného riešenia. Databázy týchto systémov vystupujú ako zdroje dát pre dátový sklad, nad ktorým prebiehajú analytické procesy. Oracle 10g, ktorý plní funkciu dátového skladu, obsahuje samotné integrované dáta aplikácie. Dátový sklad je plnený skriptami generovanými ETL nástrojom Oracle Warehouse Builder. Samotné procesy plnenia dátového skladu sú pravidelne spúšťané plánovačom Oracle Scheduler. Jadrom celého systému je aplikačný server JBoss, ktorý poskytuje behové prostredie pre webovú aplikáciu komunikujúcu s užívateľom prostredníctvom prehliadača. Táto viacvrstvová webová aplikácia prezentuje výsledky analýz algoritmov dolovania dát, ako aj OLAP grafy a krížové tabuľky. Integráciu webovej aplikácie a dát uložených v databáze a v dátovom sklade zabezpečuje integračný framework JBoss Seam [25], ktorý integruje štandardné

technológie ako je EJB 3 [26], JSF [29] a mnoho ďalších. Komunikácia s enginom na dolovanie dát je realizovaná pomocou webových služieb, štandardu JSR 73 [43] a podnikovej zbernice služieb Mule [74].

8.3.1 Mantis

Mantis je webový systém na evidenciu a sledovanie objavených chýb (softvéru, procesov, funkčnosti, ...), ktorý je v súčasnosti vo vývoji a nie je považovaný za produkčne stabilný. Je implementovaný v skriptovacom jazyku PHP [45] a vyžaduje MySQL [46] databázu a webový server. Mantis je možné nainštalovať na Windows, MacOS, OS/2, a variácie operačného systému Unix. Ako klient môže vystupovať ľubovoľný webový prehliadač. Tento systém je uvoľnený pod podmienkami licencie GNU General Public License (GPL) [70].

8.3.2 Devis

Devis je systém pre dochádzku zamestnancov. Je implementovaný v ASP.NET [69] a vyžaduje databázu Oracle 10g [71], respektíve Oracle XE [71]. Tento softvér je majetkom spoločnosti Flynet s r.o.

8.3.3 BI platforma

Pri výbere technológií, ktoré sa použijú na implementáciu prvkov BI je k dispozícii viacero technických riešení, komerčných i z ponuky open source projektov. Ako najperspektívnejší kandidáti s ohľadom na cenu a škálu funkcií sa javia open source projekt Pentaho [89] a komerčné produkty spoločnosti Oracle [71]. Architektúra platformy Pentaho pozostáva z prezentačnej vrstvy, reportovania, analýz, násteniek, manažmentu procesov, BI platformy a vrstvy dátovej a aplikačnej integrácie. Pentaho BI integruje open source projekty potrebné k implementácii kompletného BI riešenia. Produkty open source sú veľmi lákavé hlavne z pohľadu ceny. Nevýhodou je málo dostupnej dokumentácie a málo skúseností s touto platformou, keďže sa jedná o integráciu veľkého počtu open source projektov. V prípade OLAP riešenia je k dispozícii len ROLAP architektúra v podobe projektu Mondrian. Pre úplnosť, projekt Pentaho obsahuje nasledujúce open source projekty [89]:

- Mondrian - Open Source OLAP Server.
- JFreeReport - Open Source Reporting.
- Kettle - Open Source Data Integration (E.T.T.L.).
- Pentaho - Open Source BI Suite.
- Weka - Open Source Data Mining.

Na rozdiel od platformy Pentaho, je Oracle BI Suite a Oracle 10g server [71] licencovaný softvér. K danému systému je dostupné veľké množstvo dokumentácie čítajúce niekoľko desiatok tisíc strán, množstvo literatúry a webových serverov zaoberajúcich sa týmito produktami. Platforma Oracle je takisto preferovanou platformou firmy Flynet, ktorá túto platformu využíva pre implementáciu IS pre svojich zákazníkov. Výhodou riešenia Oracle je veľmi úzke napojenie na najlepšiu súčasnú databázu Oracle 10g, ako aj samotné OLAP funkcie daného servera. Jedinou zjavnou nevýhodou riešenia Oracle je jeho vysoká cena. Oracle poskytuje nástroje pre tvorbu kompletných BI aplikácií, ako aj už hotové predpripravené aplikácie pre konkrétne špecifické oblasti, ako je napríklad manažment dodávateľského reťazca. Oracle ťaží predovšetkým zo svojho postavenia na trhu databáz, keďže databáza Oracle 10g je v súčasnosti v množstve ohľadov na špici medzi databázami. Databáza Oracle je základom pre Oracle OLAP a Oracle Data Mining (dolovanie dát) [71]. Z vymenovaných funkcií zozbieraných na internete nie je samozrejme možné posúdiť kvalitu jednotlivých riešení. Preto som sa rozhodol v nejakom krátkom časovom úseku porovnať obidva konkurenčné produkty. Jedným z hlavných komponentov BI riešenia sú ETL nástroje. V nasledujúcich podkapitolách bližšie rozoberieme vlastnosti jednotlivých komponentov týchto dvoch BI platforiem.

8.3.4 ETL

Oracle Warehouse Builder (OWB) [81] je kvalitným komerčným nástrojom optimalizovaným pre plnenie dátového skladu realizovaného prostredníctvom databázy Oracle 10g. V nasledujúcich bodoch je stručné porovnanie nástroja Kettle platformy Pentaho a nástroja OWB spoločnosti Oracle.

- Riešenie ETL v Pentaho.
 - Nástroj Kettle.
 - Prístup k databázam pomocou JDBC.
 - Menej možností ako v Oracle.
 - Jednoduchá inštalácia, konfigurácia i zvládnutie nástroja.
 - Málo dokumentácie.
 - Neočakávané chyby programu, nutnosť workaroundov.
 - Prístup len cez javu, vygenerovaný kód nie je optimalizovaný pre žiadnu platformu.
 - Podpora databáz, ku ktorým je k dispozícii jdbc driver.
 - Nutnosť externého scheduleru pre pravidelné spúšťanie procesov.

- Riešenie ETL v Oracle.
 - Nástroj Oracle Warehouse Builder.
 - Prístup k databázam pomocou ODBC – nutnosť konfigurácie v OS.

- Vygenerovaný kód optimalizovaný pre Oracle.
- Zložitejšia inštalácia, konfigurácia a zvládnutie nástroja.
- Rozsiahla dokumentácia.
- Bez prístupu na metalink chyby v nástroji (napr. nezobrazovanie niektorých okien po istom čase).
- Podpora databáz, ku ktorým je k dispozícii odbc driver.
- Napojenie na Oracle workflow a Oracle scheduler.
- Skriptovací jazyk – rozšírenie Tcl/Tk.

Z jednotlivých vlastností týchto produktov plynú nasledujúce dôsledky:

- Ani jeden nástroj nie je bez chyby.
- Základné ETL procedúry je možné uskutočniť v oboch nástrojoch.
- Produktivita s Oracle Warehouse Builder stúpa po dlhšom čase ako u Kettle, ale od určitého času je táto produktivita oveľa vyššia.
- V množstve funkcií Kettle nemôže OWB konkurovať.
- Ak je hlavným kritériom cena, Kettle je silným favoritom.
- Ak už vlastníme licenciu na databázu Oracle 10g Enterprise Edition, OWB je tá správna voľba.

V prípade projektu EPA sú dátové pumpy namapované pomocou nástroja Oracle Warehouse Builder (OWB) [81] a prelievajú vybrané dáta zo zdrojových databáz systémov Mantis a Devis do dátového skladu. Mapovanie dátových zdrojov je uložené v podobe metadát v databáze dátového skladu Oracle 10g. Tieto metadáta je možné ďalej upravovať v nástroji Oracle Warehouse Builder. Z vytvorených mapovaní, ktoré sú interne uložené v jazyku XML, vygeneruje OWB optimalizované skripty (na základe nastavení) v podobe PL/SQL procedúr a balíčkov. Tieto skripty sú tiež uložené a skompilované v dátovom sklade. Po vytvorení mapovania dátových zdrojov na štruktúry dátového skladu sa v OWB vytvorí workflow plnenia dátového skladu, ktorá zreťazí jednotlivé ETL pumpy do logického sledu. Táto workflow je reprezentovaná XML súborom, ktorý je uložený v dátovom sklade. V daných časových intervaloch je workflow automaticky spúšťaná nástrojom Oracle Scheduler [71] a dochádza k aktualizácii dátového skladu. Podrobnejší popis integrácie zdrojových dát je popísaný v sekcii 8.5.

8.3.5 Dátový sklad

Dôležitou časťou riešenia je dátový sklad, ktorý obsahuje samotné integrované dáta aplikácie. Preferovanou databázou Pentaho je MySQL, pričom samotná databáza nie je súčasťou platformy Pentaho. Ani najnovšia verzia 5 databázy MySQL neposkytuje vlastnosti plnohodnotnej OLAP databázy. Podľa Koflera [18] je *MySQL*

v skutku veľmi rýchla databáza, ale veľmi limitovaná v použiteľnosti pre real-time aplikácie, a neposkytuje žiadne OLAP funkcie. MySQL podporuje, od verzie 5.0, uložené procedúry a trigger, ale tieto funkcie ešte plne nedospeli (toto sa týka predovšetkým triggerov) a ešte stále nemajú tú samú stabilitu a množstvo funkcií ponúkaných komerčnými databázovými systémami.

V prípade aplikácie EPA, dáta v dátovom sklade predstavujú jednotný pohľad na integrované dáta jednotlivých informačných systémov Devis a Mantis. Dátový sklad je zložený z databázových tabuliek, pohľadov a multidimenzionálneho úložiska. Pohľady a tabuľky sa logicky členia na staging tabuľky, pohľady dimenzií a faktové tabuľky. Staging tabuľky sú tabuľky pre dáta zo zdrojovej databázy v pôvodnej podobe. V dátovom sklade sa nenachádzajú len dáta pochádzajúce zo zdrojových databáz. Príkladom takýchto dát je tabuľka dimenzie Time. Tabuľky dimenzií sú pomocou nástroja Oracle Analytic Workspace Manager 10g namapované na dimenzie v multidimenzionálnom úložisku. Faktové tabuľky spolu s dimenziami sú namapované na OLAP kocky (tiež uložené v multidimenzionálnom úložisku). Multidimenzionálne OLAP objekty sú uložené v natívnej binárnej forme, ktorá umožňuje rýchly prístup k objektom.

Do problematiky dátového skladu spadá i dátová časť funkčnosti OLAP. Existujú v zásade dva prístupy k riešeniu OLAP databáz. A to relačný (ROLAP) a multidimenzionálny prístup (MOLAP), prípadne kombinácia oboch prístupov (HOLAP). Hlavné rozdiely medzi týmito prístupmi sú nasledujúce:

- Relačná databáza.
 - Poskytuje dlhšiu dobu odozvy.
 - Pri realizácii analytických aplikácií neposkytuje vysokú flexibilitu, vývoj aplikácií je zložitejší.
 - Je vhodná pre veľké objemy dát.
 - Poskytuje lepšiu možnosť riešenia problému zmien v dimenziách.
 - Aktualizácia agregovaných hodnôt je zložitejšia.

- Multidimenzionálna databáza.
 - Poskytuje výraznejšie kratšiu dobu odozvy.
 - Poskytuje vysokú pružnosť pri riešení analytických aplikácií, vývoj aplikácií je efektívnejší.
 - Nie je náročná na objem dát, pracuje s dátami na nízkej úrovni detailu.
 - Nie je vhodná pre extrémne veľké objemy dát.
 - Riešenie problému zmien v dimenziách je zložitejšie.
 - Aktualizácia agregovaných hodnôt je podstatne jednoduchšia.

V prípade platformy Pentaho je k dispozícii produkt Mondrian, ktorý je predstaviteľom ROLAP riešenia s použitím MDX [102] ako dotazovacieho jazyka. BI suite od firmy Oracle poskytuje nástroje podporujúce obidva prístupy, ROLAP i MOLAP. Keďže súčasťou riešenia je webový portál predstavujúci rozhranie medzi užívateľom a BI aplikáciou v podobe grafov a tabuliek, je vhodnejšie použitie multidimenzionálnej databázy, a to z dôvodu výraznejšie kratšej odozvy. Keďže sa nepredpokladá, že bude v prípade aplikácie EPA dátový sklad obsahovať veľké objemy dát, je vhodné použiť multidimenzionálnu databázu. Medzi produktom Mondrian a Oracle Warehouse Builder je výrazný rozdiel, čo sa týka funkčnosti a výkonnosti. Z pohľadu ceny je Mondrian jasným favoritom. Na základe uvedených skutočností sa na implementáciu funkčnosti OLAP použila multidimenzionálna databáza v prostredí Oracle 10g.

8.3.6 Dolovanie dát

Dátový sklad a relačné databázy sú primárnym zdrojom pre dolovanie dát. Historicky, nástroje dolovania dát boli zamerané na dáta obsiahnuté v súboroch s plochou štruktúrou. Asi najväčším nedostatkom súborov s plochou štruktúrou je ich zložitá údržba a správa. Naproti tomu, databázové súbory poskytujú funkcie dotazovania sa, bezpečnosti, garancie korektnosti, kontroly prístupu a množstvo ďalších funkcií. Prístup ku viacerým komerčným databázam sa považuje za štandard súčasných nástrojov na dolovanie dát. Návrh dátového skladu s ohľadom na požiadavky pokročilých analytických funkcií dolovania dát môže výrazne zvýšiť hodnotu dátového skladu a zároveň zjednodušiť proces extrahovania informácií z dát.

Nástroje dolovania dát poskytujú viacero možností integrácie s ERP a biznis procesmi. Pre diskusiu týchto možností je nutné priblížiť architektúru samotných nástrojov dolovania dát. Nástroje dolovania dát typicky pozostávajú zo štyroch častí [90]:

- Grafické užívateľské rozhranie (GUI) pre interaktívne dolovanie dát.
- Aplikačné rozhranie (API) pre automatické dolovanie dát.
- Engine dolovania dát (DME) predstavujúci prostredie behu algoritmov dolovania dát.
- Úložisko objektov (MOR), ktoré boli vytvorené v priebehu procesu dolovania dát.

Niektoré nástroje umožňujú GUI prístup priamo do DME, iné tento prístup umožňujú len internému API.

DME, ktorý prístupuje ku vzdialeným dátam môže vystupovať ako samostatný server, proces operačného systému alebo ako množina procesov, ktorá načíta dáta zo súborov alebo extrahuje dáta z databáz. Ako úložisko objektov sa väčšinou využíva relačná databáza, prípadne súbory XML. Niektoré nástroje dolovania dát majú snahu umiestniť DME čo najbližšie k databáze. Toto prináša viacero výhod, ako je zvýšenie výkonu, bezpečnosti alebo zjednodušenie správy. Jedným z takýchto nástrojov je aj

produkt od firmy Oracle, ktorý je súčasťou databázy Oracle 10g. Tu existuje viacero ciest prístupu k DME, ako je PL/SQL API [71], JDM API [43] a Oracle Data Miner [71]. PL/SQL API je špecifické aplikačné rozhranie databázy Oracle, JDM API je java aplikačné rozhranie implementujúce štandard JSR 73. JDM API predstavuje aplikačné rozhranie umožňujúce automatizovať procesy dolovania dát, ako aj implementovať riešenie nezávislé na implementácii. Je napríklad možné zameniť implementáciu Oracle za implementáciu od spoločnosti KXEN [91], ktorá poskytuje referenčnú implementáciu štandardu JDM 1.1. Na základe prezentovaných skutočností je výhodné pre automatické dolovanie dát využiť štandard JDM 1.1 a implementáciu tohto štandardu od Oracle, ktorá spája výhody štandardu so všetkými výhodami databázy Oracle 10g. Pre interaktívne dolovanie dát je stále možné využiť už existujúcu desktopovú aplikáciu Oracle Data Miner.

V prípade aplikácie EPA je dolovanie dát realizované ako samostatný projekt odmlogic.ear, ktorý je nasadený na aplikačnom serveri. Po prijatí správy do vstupnej fronty túto správu vyberie Message Driven Bean, ktorá danú správu spracuje. Spracovanie správy predstavuje spustenie procesu dolovania dát, ktorého účelom je odhaliť súvislosti medzi dátami v danej databázovej tabuľke. Tento projekt bude v budúcnosti rozšírený o ďalšie procesy na základe požiadaviek manažmentu.

8.3.7 Aplikačný server

Aplikačný server je strednou vrstvou (middle tier) takzvanej viacvrstvovej architektúry. U viacvrstvovej architektúry je aplikačná biznis logika vykonávaná v strednej vrstve a je prevádzkovaná práve na aplikačnom serveri. Klient môže byť vybavený len webovým prehliadačom alebo klientskou aplikáciou komunikujúcou so serverovou časťou na aplikačnom serveri. Táto komunikácia potom funguje na jednoduchom princípe požiadavok-odpoveď. Aplikačný server potom okrem výkonných funkcií biznis logiky uskutočňuje komunikáciu s databázovými servermi.

Za veľkú výhodu aplikačných serverov je považovaná schopnosť integrovať aj veľmi zložitú heterogénne prostredia pri využití viacvrstvových architektúr, predovšetkým vďaka podpore viacerých otvorených štandardov. Aplikačné servery tiež podporujú prístup k rôznym dátovým zdrojom ako sú databázové platformy a legacy systémy podporujúce štandard architektúry J2EE konektorov JSR 112. J2EE aplikačné servery tiež podporujú štandard JDBC umožňujúci pripojenie k databázam poskytujúcim JDBC driver. Drvivá väčšina aplikačných serverov vychádza z podpory štandardu Java 2 Platform Enterprise Edition (J2EE), obvykle sa jedná o verziu J2EE 1.4, ale objavuje sa už aj podpora verzie Java EE 1.5. Aplikačné servery založené na J2EE dnes predstavujú dôkaz oprávnenosti jednej zo základných vízií technológie Java – prenositeľnosti.

Z podpory J2EE vychádza možnosť využívania ďalších vlastností, za všetky uvedme napríklad dostupnosť rôznych kontajnerov (EJB, webových...) či dostupnosť služieb so štandardizovanými rozhraniami (asynchrónne zasielanie správ podľa štandardu JMS, spracovanie XML podľa štandardu JAXP...). U všetkých hlavných

serverov sa ale nejedná o jediný podporovaný štandard, medzi ďalšie patria napríklad vlastnosti súvisiace s XML a webovými službami (SOAP, WSDL, UDDI...), s otvorenosťou platformy (CORBA, dátové rozhrania...) a bezpečnosťou (XACML, WSS, SSL, X.509...).

Alternatívou k aplikačným serverom sú aplikačné frameworky podporujúce vybrané otvorené štandardy. Lídrom v oblasti Java/Java EE aplikačných frameworkov je Spring. Tento framework vznikol predovšetkým ako odozva na komplikovanosť vývoja aplikácií nasadených na aplikačnom serveri. Zjednodušenie pri vývoji je dosiahnuté odklonom od štandardov ako je napríklad EJB a ponúknutie alternatívnej technológie.

Medzi najdôležitejšie technológie použité k implementácii aplikácie EPA nasadenej na aplikačnom serveri patria JBoss Seam [25] a Oracle BI Beans [71]. Tieto technológie poskytujú veľké množstvo funkcií. Tie najhlavnejšie rozoberieme v nasledujúcich odstavcoch.

V rámci prezentačnej vrstvy poskytuje Oracle produkt Oracle BI Beans, čo sú de facto JSP tagy určené špeciálne pre BI aplikácie predstavujúce rôzne grafy a krížové tabuľky. Výhodou BI Beans je, že samotné grafy môžu byť uložené v databáze, pričom grafy sú stále aktuálne, teda reflektujú najaktuálnejšie dáta prítomné v Oracle Analytic Workspace, čo je len iný názov pre multidimenzionálne úložisko dát. Oracle BI Beans je množina JavaBeans [92], ktorá poskytuje komponenty pre vývoj OLAP riešení. Oracle BI Beans nepredstavujú hotové riešenie, ale len stavebné prvky, z ktorých je možné postaviť OLAP riešenie zodpovedajúce konkrétnym potrebám určitého projektu. Technológia Oracle BI Beans má nasledujúce významné vlastnosti:

- Integrácia s java architektúrou – Oracle BI Beans plne vyhovujú štandardu J2EE a môžu byť nasadené na rôznych aplikačných serveroch ako sú napríklad BEA Weblogic, Oracle Application Server a JBoss.
- Predpripravené komponenty (JSP tagy) pre OLAP aplikácie s možnosťou customizácie.
- Pokročilé analytické prvky – formulovanie otázok a prezentácia odpovedí, vytváranie indikátorov, analýza časových rád.
- Prezentačné komponenty ako sú grafy (viac ako 70 typov) a krížové tabuľky.
- Centrálné úložisko BI Beans Catalog používajúce XML pre ukladanie grafov a krížových tabuliek v stromovej štruktúre umiestnené buď v databáze alebo v lokálnom adresári.
- Bezpečnosť.
- Exportovanie grafov do html, txt, xls.
- Prístup k živým dátam – grafy a krížové tabuľky sú objekty reflektujúce aktuálne dáta.
- Možnosť nasadenia v podobe desktopovej aplikácie.

Keďže Oracle BI Beans sú JSP tagy, potrebujú pre svoj beh prostredie nejakého servlet kontajnera [93]. Tu nastáva otázka, aký kontajner použiť. Oracle poskytuje svoj vlastný aplikačný server Oracle AS obsahujúci servlet kontajner presne pre takéto účely. Vo svete open source existuje množstvo kvalitných servlet kontajnerov, webových a aplikačných serverov. Servlet kontajner je v zásade časťou webového servera, aj keď môže bežať ako samostatný proces. V tomto zmysle sa servletové kontajnery klasifikujú do nasledujúcich 3 typov [94]:

- Samostatné – Servlet kontajnery tohto typu sú typicky webové servery napísané v jave, kde dva moduly – hlavný webový server a servlet kontajner – sú nedeliteľnou súčasťou jedného programu. Príkladom takéhoto riešenia je Tomcat [21].
- Vo vnútri procesu – Hlavný webový server a servlet kontajner sú oddelené programy, ale servlet kontajner beží v adresnom priestore hlavného webového servera ako plugin. Príkladom takéhoto riešenia je Tomcat, ktorý beží v rámci webového Apache servera [22]. V tomto prípade Apache obsluhuje statické stránky a Tomcat obsluhuje dynamické JSP stránky (.jsp) alebo dokumenty (.jspx).
- Mimo procesu - Hlavný webový server a servlet kontajner sú oddelené programy. Webový server a servlet kontajner bežia v rámci samostatných procesov. Pre komunikáciu so servlet kontajnerom, využíva webový server plugin, ktorý zvyčajne poskytuje dodávateľ servlet kontajnera. Príkladom takéhoto riešenia je Tomcat bežiaci v samostatnom procese, ktorý prijíma požiadavky od Apache servera, pričom komunikáciu zabezpečuje mod_jk plugin.

Každý z takýchto prístupov má očividne svoje plusy a zápory, ktoré na tomto mieste nebudeme ďalej rozvádzať. Podniková aplikácia EPA pre svoj beh využíva štandardné riešenie v podobe aplikačného servera JBoss [24], ktorý obsahuje Tomcat ako webový server. JBoss je najrozšírenejší aplikačný server s množstvom pokročilých funkcií. Jeho nespornou výhodou je prijateľná cena a licencia LGPL (GNU Lesser General Public License [85]).

JBoss Seam je integračný framework, ktorý integruje technológie ako Java EE 1.5, JSF a EJB 3.0 [15]. Seam integruje viaceré open source ajax knižnice založené na JSF, ako sú Richfaces [80] a Ajax4jsf [62] so svojim enginom zabezpečujúcim riadenie stavu a konkurenčný beh (súbežnosť). EJB 3.0 predstavuje štandardný komponentový model pre transakčné biznis komponenty. Webové a aplikačné frameworky založené na jave, ktoré boli navrhnuté ešte pred príchodom EJB 3.0 obsahujú len malú podporu pre tento nový komponentový model a je nutné písať vlastný integračný kód, a v mnohých prípadoch vyžadujú použitie architektúry so striktným oddelením vrstiev, čo nemusí vyhovovať každej aplikácii. Seam je už od počiatku navrhnutý pre použitie EJB 3.0 a každý komponent v Seam môže byť komponentom EJB. Seam nevyžaduje použitie konkrétnych vrstiev, a dokonca nevyžaduje použitie EJB 3.0, ale poskytuje podporu aj pre iné alternatívy, ako je napríklad Hibernate [30]. JSF vytvára ekosystém komponentových knižníc a v súčasnosti patrí medzi najvyspelejšie webové frameworky v jave. JSF ako štandard je v súčasnosti podporovaný prakticky všetkými najbežnejšími

java IDE, ako je Eclipse, Netbeans, Oracle JDeveloper a to buď v podobe pluginov, alebo ako súčasť IDE. JSF vo svojej pôvodnej podobe hojne využíva XML, čo môže viesť k mnohým problémom v priebehu vývoja. Seam nahrádza problémové XML použitím anotácií, čím dochádza k redukcii kódu. Seam takisto prináša k JSF nové prvky, ako sú operácie v rámci viacerých okien, workspace manažment, validácia vychádzajúca zo samotného modelu, tok stránok založený na jBPM [31], internacionalizáciu a caching častí stránok. Seam prináša možnosť pridávania JSF stránok do obľúbených položiek prehliadača, pričom nemožnosť pridávania JSF stránok do obľúbených položiek je jednou z najviac kritizovaných vlastností technológie JSF. Technológia jBPM je riešením pre komplexné interakcie medzi užívateľmi a podnikovými aplikáciami. Technológia jBPM uľahčuje implementáciu a optimalizáciu biznis procesov. Seam integruje jBPM a umožňuje jednoduché využitie manažmentu úloh a toku stránok (pageflow). Využíva anotácie vo všetkých úrovniach aplikácie a prináša svoj vlastný persistence manažment, ktorý je vhodný práve pre webové aplikácie. Ďalšou pokročilou funkčnosťou je integrácia s JBoss Rules [32] v oblasti bezpečnosti pre deklaratívne programovanie pravidiel. Snáď najväčším plusom JBoss Seam je jeho prístup ku štandardom a to, že tento framework je k dispozícii ako open source projekt. Pekne spracované porovnanie niektorých webových frameworkov, ako sú JSF, Spring, Struts, Tapestry a WebWork, je možné nájsť v prezentácii od Matt Raible [95].

Pri vývoji webovej časti aplikácie sa používajú facelety [38] namiesto JSP všade, kde to len je možné. Facelety sú odporúčané samotnými autormi JBoss Seam [25], keďže lepšie zapadajú do sveta JSF ako JSP. V nasledujúcich bodoch sú vypísané výhody faceletov [97]:

- Šablóny (ako Tiles [96]).
- Komponenty pre kompozíciu.
- Vlastné tagy.
- Funkcie výrazov.
- Užívateľsky príjemný vývoj stránok.
- Vytváranie komponentových knižníc.

JSP je jazyk na vytváranie šablón, ktorého výsledkom je servlet. Telo JSP je ekvivalentom service metódy servleta. JSF tagy sú v tomto prípade volania JSF komponentov na vyobrazenie vlastného obsahu daného stavu. Životný cyklus JSP je nezávislý od životného cyklu JSF, z čoho pramenia rôzne problémy. Na rozdiel od JSP, sú facelety jazyk na vytváranie šablón stvorený už od samého počiatku s myšlienkou životného cyklu JSF. Výsledkom nie je servlet, ale strom komponentov. Je možné skladať komponenty z ďalších komponentov. Facelety sa vyhýbajú potrebe písať vlastné tagy len pre to, aby bolo možné použiť JSF. Na rozdiel od JSP, facelety volajú komponenty JSF natívne. Nevýhodou faceletov je malá podpora zo strany IDE, ako aj to, že facelety podporujú až JSF 1.2, čo ale u nových projektov nie je prekážkou. V nasledujúcom texte priblížime aplikácie nasadené na aplikačnom serveri z pohľadu aplikácie EPA.

V prípade aplikácie EPA sú na aplikačnom serveri nasadené dve aplikácie: epaweb.ear a epa.war. Toto rozdelenie je z dôvodu použitia rôznych prístupov jednotlivých aplikácií a z dôvodu zvýšenia flexibility implementácie prezentačnej

OLAP vrstvy aplikácie. Aplikácia epa.war je vytvorená pomocou nástroja Oracle JDeveloper 10g [71] a obsahuje OLAP grafy vystavené ako jsp stránky. OLAP grafy sú implementované ako komponenty Oracle BI Beans [71]. Komponenty Oracle BI Beans prístupujú k multidimenzionálnej databáze natívne, čo predstavuje rýchlejší prístup k OLAP dátam ako v prípade relačného schéma. Na trhu existuje veľké množstvo OLAP nástrojov a hotových produktov od rôznych spoločností. Medzi najvýznamnejšie spoločnosti patria Microsoft, Business Objects, Cognos, IBM, SAP a Hyperion [99]. Výhodou produktu Oracle BI Beans je natívny prístup k OLAP objektom databáze Oracle a kompatibilita s architektúrou J2EE.

Aplikácia epaweb.ear integruje OLAP prvky a prezentuje ich v užívateľsky prístupnej forme. Poskytuje možnosti spustenia biznis procesov. Na obrázku 8.3.5 je zachytené zobrazenie procesov v aplikácii epaweb. Externé biznis procesy tretích systémov napojených na podnikovú zbernicu služieb momentálne predstavuje proces dolovania dát (konkrétne proces skúmania asociácií v dátach). Do budúca sa počíta s rozšírením ponuky procesov na základe konkrétnych potrieb manažmentu. Tiež umožňuje nastaviť odpovedajúce mapovanie užívateľov zdrojových systémov. Ku každej úlohe je prostredníctvom užívateľského rozhrania možné zadať predpokladaný čas potrebný na vyriešenie úlohy. Neskôr po uzatvorení danej úlohy je táto úloha zaradená do štatistík spolu s vopred odhadovaným časom riešenia.

Vytvoření procesu

* Popis Association Process B

Typ procesu Association

Vytvořit proces

Otevřené procesy

Popis	Cítnel	Vyvořeno	Začátek	Priorita	Den splatnosti	Následující 1
proces C	vojto	11.11.2007 9:53		3		Soustř. Ukončit
proces D	vojto	11.11.2007 9:53		3		Soustř. Ukončit
proces E	vojto	11.11.2007 9:53		3		Soustř. Ukončit
proces G	vojto	11.11.2007 9:53		3		Soustř. Ukončit
proces H	vojto	11.11.2007 9:55		3		Soustř. Ukončit
proces I	vojto	11.11.2007 9:55		3		Soustř. Ukončit
proces J	vojto	11.11.2007 9:55		3		Soustř. Ukončit
proces K	vojto	11.11.2007 9:56		3		Soustř. Ukončit
proces L	vojto	11.11.2007 9:56		3		Soustř. Ukončit
Association Process	vojto	11.11.2007 9:56		3		Soustř. Ukončit

Ukončené procesy

Popis	Cítnel	Vyvořeno	Začátek	Konec	Priorita	Den splatnosti	Výsledek
proces A	vojto	11.11.2007 9:50	11.11.2007 9:50	11.11.2007 9:51	3		Výsledek
proces B	vojto	11.11.2007 9:51	11.11.2007 9:51	11.11.2007 9:52			Výsledek
proces F	vojto	11.11.2007 9:53		11.11.2007 9:56	3		Výsledek

Process ID 12

Ended by user.

Obrázok 8.3.5: Zobrazenie procesov v aplikácii epaweb

Integrácia webových aplikácií epaweb.ear a epa.war v súvislosti s bodmi prezentovanými v kapitole 6.2 *Integrácia vstupov* je realizovaná nasledujúcim spôsobom:

- Vzťahy medzi aplikáciami – aplikácie epaweb.ear a epa.war vystupujú v nerovnocennom vzťahu, pričom aplikácia epaweb.ear predstavuje master aplikáciu. Aplikácia epa.war len prijíma požiadavky pomocou protokolu https od aplikácie epaweb.ear.
- Komunikácia medzi aplikáciami – komunikácia prebieha synchronne prostredníctvom zabezpečeného protokolu https.
- Manažment užívateľov a skupín – užívatelia a skupiny sú uložené v databázových tabuľkách v databáze Oracle. Prístup k týmto tabuľkám majú obidve aplikácie.
- Manažment zasadania (session management) – každá aplikácia má vlastnú session. Session v aplikácii epa.war sa vytvorí až prístupom z aplikácie epaweb.ear (po zalogovaní). Po odhlásení z aplikácie epaweb.ear, sa session v aplikácii epaweb.ear zneplatní a session v aplikácii epa.war po nastavenom čase samovoľne vyprší.
- Autentifikácia – Autentifikácia v aplikácii epaweb.ear je uskutočnená pomocou štandardu JAAS spolu s prístupom k databáze Oracle. Heslá sú chránené bezpečnosťou databázy Oracle a pomocou hash algoritmu sha1. Autentifikácia prebieha vždy na začiatku session. V aplikácii epa.war je autentifikácia riešená transparentne pomocou vlastného kódu a nie je nutné sa zvlášť do tejto aplikácie prihlasovať, čo je v súlade s princípom single sign-on.
- Kontrolovanie prístupu – len zalogovaný užívateľ má prístup k webovému obsahu.
- Manažment práv – práva sú momentálne spravované prostredníctvom databázy Oracle. Do budúca sa počíta s integráciou s nejakým portálovým riešením a správa práv bude riešená v rámci portálu.
- Vizuálna integrácia – aplikácia epa.war poskytuje grafy, ktoré sú transparentne zobrazené v rámci aplikácie epaweb.ear.
- Vyhľadávanie – nie je riešené.
- Konzistentné prostredie (lokalizácia, kódovanie súborov, ...) – v rámci grafov nie je lokalizácia momentálne riešená. Grafy sú koncipované v zásade ako nezávislé na jazyku, pričom lokalizáciu grafov je možné v budúcnosti docieľiť pomocou predávania parametra nastavenia v rámci aplikácie epaweb.ear a vytvorenia lokalizovaných popiskov pre OLAP dimenzie a OLAP kocky v rámci multidimenzionálnej OLAP databázy. Samotné Oracle BI Beans lokalizáciu podporujú, takže v budúcnosti nebude problém v prípade nutnosti lokalizovať aj samotné OLAP grafy.

8.3.8 Podniková zbernica služieb

Podniková zbernica služieb (ESB) poskytuje transparentný prístup k externým službám, či už ide o webové služby alebo služby vystavené prostredníctvom ďalších protokolov ako je JMS, FTP a podobne. V súčasnosti existuje na trhu viacero kvalitných ESB produktov. Medzi najvýznamnejšie patria IBM Websphere Enterprise Service Bus, BEA Aqualogic Service Bus, Mule, ServiceMix, JBoss ESB, FUSE a mnoho ďalších [99]. Komerčné ESB produkty sú dostupné za nemalé licenčné poplatky, a preto sa používajú predovšetkým na veľkých projektoch s dobrým financovaním. Príkladom takýchto projektov sú rôzne integračné projekty pre telekomunikačné spoločnosti, spoločnosti z oblasti finančného sektora a vládnych inštitúcií. Príkladom projektu realizovaného v Českej republike je Národný vízový informačný systém súvisiaci so vstupom Českej republiky do schengenského priestoru. V prípade open source ESB produktov odpadá problém licenčných poplatkov. Podľa InfoWorld Best of Open Source Software Awards [100] je najlepším open source ESB produktom Mule ESB [74]. Hlavnými argumentmi v prospech Mule ESB je počet funkcií, administrácia, ovládateľnosť, jednoduchosť inštalácie a nasadenia, možnosti prepojitelnosti (JMS, MQ/AQ, JBI, SOAP, REST, dátové fronty AS400 a ďalšie), silné stránky v oblasti transformácie dát, smerovania, autentifikácie a autorizácie. Mule ESB je tiež možné integrovať s ďalším open source ESB produktom ServiceMix [101]. Podnikovú zbernicu služieb Mule je možné prevádzkovať ako standalone alebo ju nasadiť napríklad na aplikačný server JBoss [24]. Na základe všetkých týchto skutočností sa Mule ESB javí ako tá správna voľba pre použitie v projekte EPA. V rámci projektu EPA sa Mule ESB používa k vyvolaniu externých služieb dolovania dát. Do budúca je možné vystaviť pomocou ESB aj iné služby, ktoré je potom možné napojiť na biznis procesy spúšťané pomocou užívateľského webového rozhrania z aplikácie epaweb. Pre vyvolanie služieb dolovania dát sa využíva asynchrónneho vyvolania pomocou softvérových nástrojov XFire [72] a JBoss WS [73]. Na diagrame 8.3.6 je zobrazená interakcia podnikovej zbernice služieb Mule so systémami epaweb, enginom na dolovanie dát a JBoss Messaging [75]. V prípade systému EPA, podniková zbernica služieb Mule [74] využíva JBoss Messaging [75] ako poskytovateľa JMS.

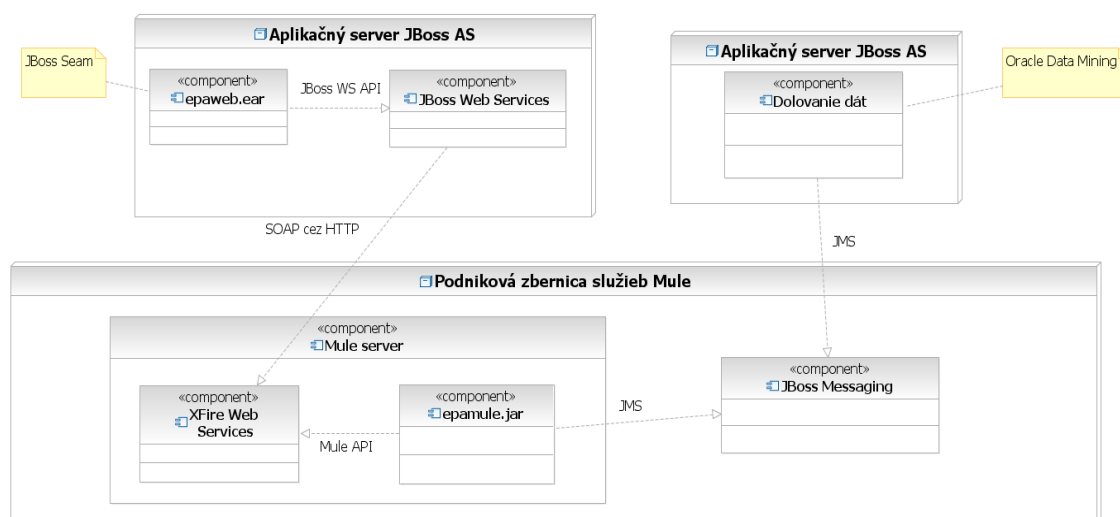


Diagram 8.3.6: Interakcia s podnikovou zbernicou služieb

Základná interakcia s podnikovou zbernicou služieb má nasledujúci priebeh. Cez webové rozhranie aplikácie epaweb.ear nasadenej na aplikačnom serveri JBoss sa vytvorí biznis proces typu dolovanie dát. Po vytvorení biznis procesu sa tiež z webového rozhrania tento proces spustí. Spustenie procesu znamená poslanie SOAP správy pomocou JBoss WS z aplikačného servera do podnikovej zbernice služieb. V podnikovej zbernici služieb Mule túto správu spracuje komponent webových služieb XFire. Tento komponent postúpi správu modulu epamule.jar, ktorý sa na základe typu správy rozhodne, ako túto správu spracovať. Keďže sa jedná o správu určenú pre engine dolovania dát, je SOAP správa prevedená na JMS správu a s využitím JBoss Messaging poslaná do perzistentnej fronty, z ktorej ju vyberie engine dolovania dát, ktorý následne uskutoční samotný proces dolovania. Po skončení tohto procesu je výsledok dolovania dát pomocou JMS poslaný späť do podnikovej zbernice služieb, odkiaľ je správa s výsledkom smerovaná do aplikácie epaweb.ear, ktorá ukončí biznis proces a priradí mu získaný výsledok. Následne sa vo webovom rozhraní aplikácie objaví tento biznis proces v tabuľke ukončených procesov, kde je možné zobrazit' výsledok procesu.

8.4 Použité technológie a softvér

V nasledujúcich podkapitolách sú vymenované najvýznamnejšie použité technológie a softvér. Tieto sú členené podľa subsystémov, v ktorých sa vyskytujú. Niektoré technológie sa používajú vo viacerých logických častiach projektu, ale sú uvedené len v jednej časti. Kategória ostatné predstavuje vývojové nástroje a softvér využitý pri tvorbe projektového portálu. Na vývoji systému EPA sa použilo približne 70 rôznych technológií a softvérových produktov [71 – produkty spoločnosti Oracle]. Rozsah projektu EPA vyjadrený v počte riadkov verzovaného kódu kategorizovaný podľa prípony je znázornený v tabuľke 8.4.

Prípona súboru	Počet riadkov
java	16786
sql	6699
pl	576
xhtml	974
jsp	5783
xml	6520
css	1166
tcl	25

Tabuľka 8.4: Rozsah projektu EPA vyjadrený v počte riadkov

8.4.1 Mantis

- Mantis – systém na evidenciu chýb [17], napísaný v PHP [45]
- MySQL 5 – databáza systému Mantis [18]

8.4.2 Devis

- Devis – dochádzkový systém [77], ktorý umožňuje zadávanie vykonanej práce zamestnancami. Táto práca je rozdelená podľa kategórii, miesta a projektu. Systém umožňuje export dát pre zodpovedajúcich vedúcich pracovníkov. Na základe dát sa vytvára výkaz práce, ktorý slúži ku stanoveniu mzdy.
- Oracle XE – databáza systému Devis [71]

8.4.3 Dátový sklad

- Oracle 10g Enterprise Edition – dátový sklad, ktorý integruje dáta zo zdrojových systémov, slúži ako úložisko metadát pre Oracle Warehouse Builder, Oracle Workflow, Oracle Scheduler, Oracle BI Beans a Oracle Analytic Workspace Manager [71].
- Oracle Warehouse Builder - nástroj ETL [71]
- Oracle Workflow – proces dátových púmp [71]
- Oracle Scheduler – spúšťanie dátových púmp [71]
- Oracle Heterogeneous Services – prepojenie databáz MySQL a Oracle [71]
- ODBC – prepojenie databáz MySQL a Oracle [78]
- PL/SQL – procedúry, testovacie dáta [71]
- OMB*Plus – import/export metadát z Oracle Warehouse Builder [63]

8.4.4 Aplikačný server

- JBoss AS 4.2 – aplikačný server, na ktorý sa nasadzuje aplikácia EPA [24]
- JBoss WS 2.0 – webové služby na strane aplikačného servera [73]
- JBoss Seam 2.0 – webový integračný framework [25]
- JBoss Messaging – perzistentné fronty a poskytovateľ JMS [75]
- EJB3 – vrstva perzistencie [26]
- JBoss RichFaces – jsf komponenty [80]

- Trinidad – jsf komponenty [61]
- Ajax4jsf – ajax jsf komponenty [62]
- Hibernate 3 – implementácia JPA, kritéria [30]
- JBoss jBPM – biznis procesy [31]
- JSF RI 1.2 – prezentačná vrstva [29]
- Facelets - JavaServer Faces View Definition Framework [38]
- CSS – kaskádové štýly pre témy [23]
- XHTML – webové stránky [27]
- JDBC – prepojenie databáz s Javou [20]
- J2SE 1.4 – implementácia javovej funkcie v Oracle [41]
- Java EE 5 – architektúra webovej aplikácie epaweb [15]
- J2EE 1.4 – architektúra webovej OLAP aplikácie epa [15]
- Apache Tomcat 5.5 – servlet kontajner [21]
- AJP Connector - komunikácia medzi Apache Tomcat a Apache 2 [60]
- Oscache – cache webového obsahu [19]
- Javascript – volanie funkcií [16]
- XML technológie (XML, DTD, XML Schema, XHTML) – konfiguračné súbory, prezentačná vrstva projektu epaweb [27]
- JAAS – autentifikácia a autorizácia [28]
- JBoss Rules – bezpečnosť webovej aplikácie [32]
- SSL – bezpečnosť [87]

8.4.5 Podniková zbernica služieb

- Mule – podniková zbernica služieb [74]
- XFire – webové služby použité na strane Mule [72]
- JMS – posielanie správ [8]
- JOTM – distribuované transakcie v Mule [33]
- Log4j – logovanie [34]
- Jakarta Commons – logovanie [35]
- Web Services – komunikácia medzi JBoss AS a Mule [51]

8.4.6 Dolovanie dát

- Oracle Data Mining – dolovanie dát [71]
- JSR 73 JDM API – štandardné API pre dolovanie dát [43]

8.4.7 OLAP

- Oracle OLAP – implementácia OLAP kociek [71]
- Oracle Analytic Workspace Manager – návrh a mapovanie dimenzií a OLAP kociek [71]
- Oracle Business Intelligence Beans 10g – jsp OLAP komponenty [71]
- Oracle JDeveloper 10g – vývoj OLAP webovej aplikácie [71]

8.4.8 Ostatné

- MySQL 5 – úložisko dát pre Wiki [18]
- SQL*Plus – spúšťanie databázových skriptov [71]
- Ubuntu 7 – testovací operačný systém [36]
- Windows XP SP2 – testovací operačný systém [37]
- Perl 5.8.8 – inicializácia testovacieho prostredia, spúšťanie skriptov [39]
- Red Hat Developer Studio 1.0 – vývoj aplikácie epaweb [40]
- Oracle SQL Developer – vývoj SQL dotazov [71]
- Ant – buildovanie projektu [42]
- Java SE 5 – na tejto architektúre je postavená webová aplikácia epaweb [41]
- Apache 2 – webový server pred Tomcat [22]
- Maven 2 – build JBoss WS ant tasku [57]
- Subclipse – plugin do Eclipse pre podporu SVN [48]
- EPIC – Eclipse Perl plugin [47]
- VMWare Player – prezentácia a testovanie [59]
- CVS – správa verzií [44]
- SVN – správa verzií [44]
- Mediawiki – projektový portál [49]
- PHP – konfigurácia Mediawiki [45]
- PhpBB Forum – fórum [79]

8.5 Integrácia dátových zdrojov a systémov

Systém EPA realizuje integráciu dátových zdrojov a viacerých systémov. Dátové zdroje aplikácie predstavujú databázy systémov Mantis a Devis. Tieto dáta sa dátovými pumpami integrujú do dátového skladu. Mapovanie medzi zdrojovými databázami a dátovým skladom je vytvorené pomocou nástroja Oracle Warehouse Builder. Dátový sklad predstavuje databáza Oracle 10g Enterprise Edition. Nástroj Oracle Warehouse Builder podporuje širokú paletu dátových zdrojov. Pre potreby systému EPA je využívané napojenie len na databázu Oracle 10g. Prenos dát medzi databázami Oracle je vysoko efektívny. Využívajú sa optimalizované interné knižnice databázy. Databáza systému Devis je postavená na Oracle 10g, takže prenos dát z tejto databázy do dátového skladu je vysoko efektívny. Pri tomto prenose je využívaná napríklad optimalizácia s využitím ROWID. Prenos dát z databázy Mantis do dátového skladu je realizovaný pomocou štandardu ODBC a Oracle Heterogeneous Services. Konkrétne je nutné nakonfigurovať v danom operačnom systéme dátový zdroj ODBC odkazujúci sa na databázu MySQL pomocou MySQL ODBC ovládačov. Takýto dátový zdroj je pomocou Oracle Heterogeneous Services vystavený ako služba podobne ako bežné služby Oracle. Prístup ku takejto službe je potom možný napríklad pomocou databázových linkov špecifických pre databázu Oracle. Integrácia databázy Mantis pozostáva z vytvorenia pohľadov v databáze Oracle nad vybranými tabuľkami z Mantisu. Takéto pohľady sú vytvorené pomocou databázového linku ukazujúceho na databázu Mantis. Takto vytvorené pohľady potom slúžia ako dátový zdroj pre Oracle Warehouse Builder, v ktorom sa následne namapujú na odpovedajúce tabuľky v dátovom sklade.

Tabuľky dátového skladu sú namapované na dimenzie a OLAP kocky multidimenzionálneho OLAP úložiska. Táto multidimenzionálna databáza je tvorená špeciálnymi binárnymi súborami. Prístup k takejto databáze je možný buď pomocou rozhrania PL/SQL, Java API alebo komponent Oracle BI Beans. K naplneným dimenziám a OLAP kockám je takisto možné pristupovať ako k relačným tabuľkám, ale tieto tabuľky je nutné najprv vytvoriť buď pomocou nástroja OLAP View Generator [71] alebo programovo pomocou PL/SQL. Oracle BI Beans predstavujú natívny prístup k multidimenzionálnej OLAP databáze pomocou komponentov Java Server Pages. Integrácia OLAP a systému EPA je realizovaná prostredníctvom komponentov Oracle BI Beans, ktoré predstavujú pokročilý, užívateľsky prívetivý nástroj pre prístup k multidimenzionálnemu úložisku implementovanému v databáze Oracle 10g. Výhodou týchto komponentov je natívny prístup a nasadenie vo webovom prostredí, čo znamená on-line prístup prostredníctvom intranetu alebo siete Internet.

Webová časť systému EPA (epaweb), je nasadená na aplikačnom serveri JBoss AS, ktorý predstavuje komplexný integračný nástroj. Táto aplikácia pracuje s dátami z databázy Mantis, databázy Devis, dátového skladu a svojej vlastnej databázy. Tiež využíva JMS zdroje zo systému JBoss Messaging. Tieto dátové zdroje sú spravované aplikačným serverom, ktorý poskytuje mimo iného správu spojení, prístup k dátovým zdrojom a distribuované transakcie. Prístup k databázam je uskutočnený prostredníctvom naviazania vrstiev JDBC, Hibernate a EJB 3. Integrácia webových komponentov je realizovaná prostredníctvom architektúry Java EE. Webová aplikácia

epaweb poskytuje podporu pre spúšťanie a monitorovanie biznis procesov. Integrácia biznis procesov je realizovaná pomocou integračného frameworku JBoss Seam.

Integráciu enginu dolovania dát zabezpečujú webové služby napojené na podnikovú zbernicu služieb. Výsledky dolovania dát sú posielané vo forme správ do perzistentnej fronty v JBoss Messaging. Tieto správy sú následne spracované v Message Driven Bean a výsledky algoritmov dolovania dát sú uložené v kontexte priradených biznis procesov.

8.6 Problémy pri vývoji

Pri vývoji systému EPA došlo k viacerým komplikáciám. Pôvodne mali byť prvky Business Intelligence implementované výhradne pomocou open source nástrojov. Problém s open source nástrojmi je ich pochybná kvalita, problematické licencie, nepravidelné alebo príliš rýchle releasy, malé množstvo dokumentácie, nízka úroveň podpory starších releasov a ďalšie. Pre potreby systému EPA sa k implementácii prvkov Business Intelligence využili komerčné nástroje spoločnosti Oracle, keďže táto je preferovanou platformou firmy Flynet a jej zákazníkov. K týmto nástrojom je dostupné veľké množstvo kvalitnej dokumentácie.

V priebehu integrácie dátových zdrojov do dátového skladu nastal problém pri mapovaní zdrojových tabuliek z MySQL databáze. V Oracle Warehouse Builder toto mapovanie nefunguje správne a vyskytujú sa pri tom nevysvetliteľné chyby. Namiesto mapovania je možné použiť databázový link z databázy Oracle do MySQL. Vývojový nástroj Oracle Warehouse Builder bez opráv dostupných v Oracle Metalink vykazuje relatívne vysokú mieru nestability.

Inštalácia produktov Oracle je relatívne komplikovaná v prostredí necertifikovaných operačných systémov. Systém EPA je vyvíjaný v testovacom prostredí operačného systému Ubuntu 7.

Počas vývoja webovej aplikácie epaweb nastal najväčší problém s nekompatibilitou knižníc jsf komponentov a s konfiguráciou jednotlivých integrovaných technológií. Je jednoduché nakonfigurovať každú technológiu samostatne, ale súčasná konfigurácia viacerých komponentov predstavovala problém. Takéto problémy sa vyriešili buď metódou pokus – omyl alebo prehľadávaním rôznych diskusných fór open source projektov. Problematický bol tiež prechod na nové verzie JBoss Seam a JBoss AS, ktorý častokrát znamenal prepísanie viacerých častí systému EPA.

Jednou z neuskutočnených častí integrácie systému EPA je integrácia s nejakým portálovým riešením. V rámci vyhodnocovania technológií vhodných pre systém EPA bolo niekoľko týždňov venovaných pokusu integrácie systému EPA s JBoss Portal [7]. Počas tejto integrácie sa vyskytlo niekoľko problémov. V súčasnosti prebieha hektický

vývoj na strane JBoss AS, JBoss Portal a JBoss Seam. Kombinácia týchto troch systémov ešte nie je produkčne stabilná. Posledná verzia JBoss Portal zmenila implementáciu Java Server Faces z MyFaces na JSF RI. V čase pokusov s JBoss Portal aplikácia EPA pracovala s implementáciou od MyFaces. S novou verziou JBoss Seam aplikácia EPA pracuje s implementáciou JSF RI 1.2. V čase odovzdávania diplomovej práce je podpora portálu zo strany JBoss Seam stále na experimentálnej úrovni a de facto nefunkčná. Jedinou možnosťou tak je v súčasnosti použitie iframe portletu, čo nie je skutočná integrácia. Stabilná podpora JBoss Portal zo strany JBoss Seam sa očakáva až v ďalšej verzii. Po vyriešení týchto problémov bude jednoduché integrovať aplikáciu EPA s JBoss Portal a využiť tak sadu voľne dostupných portletov distribuovaných s týmto portálom. Integračný framework JBoss Seam je veľmi sľubný, ale pomerne nový produkt, čo prinieslo nemalé množstvo problémov. Týmto problémom by sa zrejme dalo vyhnúť použitím už zavedeného frameworku Spring [84]. Silnou stránkou JBoss Seam je úzka integrácia s viacerými produktmi spoločnosti Red Hat Middleware.

8.7 Budúci vývoj systému

V sídle spoločnosti Flynet sa uskutočnila prezentácia, ktorá majiteľom spoločnosti predstavila výslednú podobu systému EPA. Na tejto prezentácii sa vytýčil smer budúceho vývoja aplikácie. S vývojom systému EPA sa plánuje aj do budúca. V krátkom časovom horizonte sa bude systém integrovať s novou verziou dátovej štruktúry dochádzkového systému Devis [77]. Niektoré časti systému EPA budú integrované s portálovým frameworkom Intrexx Xtreme [82]. Počíta sa s vývojom webového prostredia pre návrh dimenzií a OLAP kociek, ktoré by slúžilo ako náhrada za Oracle Analytic Workspace Manager 10g. Po nasadení do reálnej prevádzky sa bude systém upravovať na základe spätnej väzby od užívateľov systému.

8.8 Nasadenie systému

Nasadenie systému do reálnej prevádzky sa plánuje až po aktualizácii na najnovšiu verziu databázového schéma systému Devis. Táto verzia je v súčasnosti stále vo vývoji. Ešte pred nasadením do reálnej prevádzky sa uskutoční školenie programátorov firmy Flynet, v ktorom sa im priblíži systém EPA. Títo programátori budú mať potom na starosti budúci vývoj systému. V súčasnosti sa pre testovacie účely zvažuje využitie virtualizácie, aby mal každý programátor k dispozícii svoje vlastné testovacie prostredie. Ako softvér pre virtualizáciu je možné použiť open source softvér Xen [76] alebo VMWare Player, ktorý síce nie je open source, ale pre prezentačné účely je dostatočný.

9. Záver

V tejto diplomovej práci bola navrhnutá a implementovaná integrácia prvkov BI do podnikového informačného systému pozostávajúceho z viacerých informačných systémov. Jedným z hlavných prínosov je analýza možností integrácie prvkov BI aplikácií do už existujúceho ERP ako i popis výhod a nevýhod jednotlivých možností.

V rámci návrhu systému EPA s prvkami BI bola analyzovaná a následne použitá objemná množina moderných technológií použiteľných k integrácii heterogénnych komponentov za účelom dosiahnutia kompozitnej aplikácie. Jednou z najťažších častí implementácie tohto systému je integrácia jednotlivých špecializovaných technológií a to už ako v rámci Java EE aplikácie – dátová vrstva, integračná vrstva, vrstva biznis logiky, prezentačná vrstva, tak i v rámci samotných BI technológií, ako sú OLAP, dolovanie dát, ETL a dátový sklad. Integrácia je možná vo všetkých vrstvách informačného systému. Neexistuje jedno univerzálne integračné riešenie vhodné pre všetky situácie a prípady. Integrácia BI prvkov do ERP je komplikovaný proces, ktorý zvyčajne vyžaduje riešenie predstavujúce kombináciu jednotlivých integračných prístupov (na rôznych úrovniach aplikácie).

Základom integrácie jednotlivých komponentov systému EPA sú otvorené štandardy, najmä Java EE, viaceré JSR štandardy ako aj webové štandardy konzorcií W3C a OASIS [58].

Pri návrhu a vývoji systému EPA boli využité skúsenosti s implementáciou informačných systémov z prostredia štátnej správy i súkromného sektoru. Z týchto informačných systémov najvýznamnejšie sú informačný systém Českej správy sociálneho zabezpečenia, klientsky klub spoločnosti Volkswagen a Národný vízový informačný systém Českej republiky. Pri implementácii častí systému v jazyku Java boli využité vedomosti vyžadované certifikátom Sun Certified Java Programmer. Tento certifikát slúži k demonštrácii znalostí základnej syntaxe a štruktúr programovacieho jazyka Java nezávisle na implementácii.

Zoznam použitých skratiek

- API – Application Programming Interface
- B2B – Business-to-business
- BI – Business Intelligence
- BPOAI – Business Process Integration-Oriented Application Integration
- CASE – Computer Aided Software Engineering
- DME – Data Mining Engine
- DSA – Dočasné úložisko dát (Data Staging Area)
- DSS – Systém pre podporu rozhodovania (Decision Support System)
- EAI – Enterprise Application Integration
- EBXML – Electronic Business using eXtensible Markup Language
- EJB – Enterprise JavaBeans
- EPA – Employee Performance Analysis
- ERP – Enterprise Resource Planning
- ES – Expertný systém
- ESB – Podniková zbernica služieb (Enterprise Service Bus)
- ETL – Extract, Transform, and Load
- FTP – File Transfer Protocol
- GPL – General Public License
- GUI – Graphical User Interface
- HOLAP – Hybrid Online Analytical Processing
- HP – Hlásenie problému
- HTML – HyperText Markup Language
- IOAI – Information-Oriented Application Integration
- IS – Informačný systém (Information System)
- JAXP – Java API for XML Processing
- J2EE – Java 2 Platform, Enterprise Edition
- JDBC – Java Database Connectivity
- JDM – Java Data Mining
- JMS – Java Message Service
- JSF – Java Server Faces
- JSP – JavaServer Pages
- JSR – Java Specification Request

LGPL – GNU Lesser General Public License
MDX – Multidimensional Expressions
MIS – Manažérsky informačný systém (Management Information System)
MOLAP – Multidimensional Online Analytical Processing
MOR – Mining Object Repository
ODBC – Open Database Connectivity
ODS – Operatívne úložisko dát (Operating Data Store)
OLAP – Online Analytical Processing
OWB – Oracle Warehouse Builder
PDF – Portable Document Format
PL/SQL – Procedural Language/Structured Query Language
POAI – Portal-Oriented Application Integration
REST – Representational State Transfer
ROI – Return On Investment
ROLAP – Rational Online Analytical Processing
RUP – Rational Unified Process
SCD – Slowly Changing Dimensions
SOA – Architektúra orientovaná na služby (Service Oriented Architecture)
SOAI – Service-Oriented Application Integration
SOAP – Simple Object Access Protocol
SQL – Structured Query Language
SSL – Secure Sockets Layer
UDDI – Universal Description, Discovery, and Integration
WSDL – Web Service Definition Language
WSS – Web Services Security
XACML – Extensible Access Control Markup Language
XML – Extensible Markup Language
XMLA – XML for Analysis

Literatúra

- [1] Ozz E.: *Management Information Systems*, fifth edition, Thomson Course Technology, Canada, 2006.
- [2] Linthicum D. S.: *Next Generation Application Integration: From Simple Information to Web Services*, Addison-Wesley, 2003.
- [3] Král J.: *Informační systémy. Specifikace, realizace, provoz*, SCIENCE, 1998.
- [4] Biere M.: *Business Intelligence for the Enterprise*, IBM Press, 2003.
- [5] Moss L. T., Atre S.: *Business Intelligence Roadmap*, Addison-Wesley, 2003.
- [6] KnowledgeStorm Inc.: *Kategórie bielych kníh online autority v prinášaní správ z oblasti využitia technológií v biznise a technologických produktov*. <http://whitepaper.informationweek.com/cmpinformationweek/search/browse/1435/1435.jsp>, 2007.
- [7] Red Hat Middleware: *JBoss Portal*. <http://www.jboss.org/products/jbossportal>, 2007.
- [8] Sun Microsystems, Inc.: *Java Message Services*. <http://java.sun.com/products/jms/>, 2007.
- [9] Inmon H. W.: *Building the Data Warehouse*, fourth edition, Wiley, 2005.
- [10] Loshin D.: *Business Intelligence: The Savvy Manager's Guide*, Morgan Kaufmann, 2003.
- [11] Novotný O., Pour J., Slánský D.: *Business Intelligence. Jak využit bohatství ve vašich datech*, Grada Publishing, 2005.
- [12] Wikipedia: *Software Development Process*. http://en.wikipedia.org/wiki/Software_development_methodology, 2007.
- [13] Kimball R.: *The Data Warehouse Toolkit*, second edition, Wiley, 2002.
- [14] Vlček J.: *Řízení změn v dimenzích*, Systémová integrace, 2002.
- [15] Sun Microsystems, Inc.: *Java EE*. <http://java.sun.com/javaee/index.jsp>, 2007.
- [16] Jupitermedia Corporation: *Javascript*. <http://www.javascript.com/>, 2007.
- [17] Mantis team: *Mantis, web-based bugtracking system*. <http://www.mantisbt.org/>, 2007.
- [18] Kofler M.: *The definitive Guide to MySQL 5*, third edition, Apress, 2005.
- [19] OpenSymphony: *OSCache*. <http://www.opensymphony.com/oscache/>, 2007.
- [20] Sun Microsystems, Inc.: *Java SE Technologies*. <http://java.sun.com/javase/technologies/database/>, 2007.
- [21] The Apache Software Foundation: *Apache Tomcat Website*. <http://tomcat.apache.org/>, 2007.
- [22] The Apache Software Foundation: *Apache Http Server Project*. <http://httpd.apache.org/>, 2007.
- [23] W3C: *CSS*. <http://www.w3.org/Style/CSS/>, 2007.

- [24] JBoss.org: *JBoss Application Server*. <http://labs.jboss.com/jbossas/>, 2007.
- [25] Red Hat Middleware: *JBoss Seam*. <http://www.jboss.com/products/seam>, 2007.
- [26] Sun Microsystems, Inc.: *Enterprise JavaBeans Technology*.
<http://java.sun.com/products/ejb/>, 2007.
- [27] Refsnes Data: *XML Technologies*.
http://www.w3schools.com/xml/xml_technologies.asp, 2007.
- [28] Sun Microsystems, Inc.: *Java SE Security*.
<http://java.sun.com/javase/technologies/security/>, 2007.
- [29] Sun Microsystems, Inc.: *JavaServer Faces Technology*.
<http://java.sun.com/javaee/javaserverfaces/download.html>, 2007.
- [30] Red Hat Middleware: *Relational Persistence for Java and .NET*.
<http://www.hibernate.org/>, 2007.
- [31] Red Hat Middleware: *JBoss jBPM*. <http://www.jboss.com/products/jbpm>, 2007.
- [32] Red Hat Middleware: *JBoss Rules*. <http://www.jboss.com/products/rules>, 2007.
- [33] ObjectWeb Consortium: *JOTM*. <http://jotm.objectweb.org/>, 2007.
- [34] Apache Software Foundation: *Apache Logging Services Project*.
<http://logging.apache.org/>, 2007.
- [35] Apache Software Foundation: *Apache Commons*. <http://commons.apache.org/>, 2007.
- [36] Canonical Ltd.: *Ubuntu*. <http://www.ubuntu.com/>, 2007.
- [37] Microsoft Corporation: *Windows XP*.
<http://www.microsoft.com/windows/products/windowsxp/default.msp>, 2007.
- [38] CollabNet: *JavaServer Faces View Definition Framework*.
<https://facelets.dev.java.net/>, 2007.
- [39] The Perl Foundation: *The Perl Directory*. <http://www.perl.org/>, 2007.
- [40] Red Hat Middleware: *Red Hat Developer Studio*.
<http://www.redhat.com/developers/rhds/index.html>, 2007.
- [41] Sun Microsystems, Inc.: *Java SE*. <http://java.sun.com/javase/>, 2007.
- [42] Apache Software Foundation: *Apache Ant*. <http://ant.apache.org/>, 2007.
- [43] Java Community Process: *JSR 73: Data Mining API*.
<http://www.jcp.org/en/jsr/detail?id=73>, 2005.
- [44] Wikipedia: *Concurrent Versions System*.
http://en.wikipedia.org/wiki/Concurrent_Versions_System, 2007.
- [45] The PHP Group: *PHP*. <http://www.php.net>, 2007.
- [46] MySQL AB: *MySQL Database*. <http://www.mysql.com>, 2007.
- [47] SourceForge, Inc.: *EPIC*. <http://e-p-i-c.sourceforge.net/>, 2007.
- [48] CollabNet: *Subclipse*. <http://subclipse.tigris.org/>, 2007.
- [49] Mediawiki: *Mediawiki*. <http://www.mediawiki.org/wiki/MediaWiki>, 2007.
- [50] OASIS: *ebXML*. <http://www.ebxml.org/>, 2007

- [51] W3C: *Web Services Activity*. <http://www.w3.org/2002/ws/>, 2007.
- [52] Wikipedia: *Mashup (Web Application Hybrid)*.
[http://en.wikipedia.org/wiki/Mashup_\(web_application_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid)), 2007.
- [53] Wikipedia: *Middleware*. <http://en.wikipedia.org/wiki/Middleware>, 2007.
- [54] Wikipedia: *Enterprise Service Bus*.
http://en.wikipedia.org/wiki/Enterprise_service_bus, 2007.
- [55] MDNH Inc.: *Metadata Coalition*. <http://www.mdcinfo.com/>, 2007.
- [56] Object Management Group, Inc.: *The Object Management Group (OMG)*.
<http://www.omg.org/>, 2007.
- [57] Apache Software Foundation: *Maven*. <http://maven.apache.org/>, 2007.
- [58] OASIS: *OASIS Standards*. <http://www.oasis-open.org/specs/index.php>, 2007.
- [59] VMWare Inc.: *VMWare Player*. <http://www.vmware.com/products/player/>, 2007.
- [60] Apache Software Foundation: *The Apache Tomcat 5.5 Servlet/JSP Container*.
<http://tomcat.apache.org/tomcat-5.5-doc/connectors.html>, 2006.
- [61] Apache Software Foundation: *Apache Myfaces Trinidad*.
<http://myfaces.apache.org/trinidad/index.html>, 2007.
- [62] Red Hat Middleware: *JBoss Ajax4jsf*. <https://ajax4jsf.dev.java.net/>, 2007.
- [63] Oracle Corporation: *Introduction to OMB Plus*.
http://download.oracle.com/docs/cd/B28359_01/owb.111/b31279/omb_1intro.htm, 2007.
- [64] Weier M. H., Smith L.: *Many Companies Plan To Increase BI Spending*.
<http://www.informationweek.com/software/showArticle.jhtml?articleID=198001258>, 2007.
- [65] Wikipedia: *Business Intelligence*.
http://en.wikipedia.org/wiki/Business_intelligence, 2007.
- [66] Geoff Jarrad: *Business Intelligence*. <http://www.cmis.csiro.au/bi/what-is-BI.htm>, 2003.
- [67] Microsoft Corporation: *XML for Analysis*.
<http://technet.microsoft.com/en-us/library/ms187178.aspx>, 2007.
- [68] Wikipedia: *Service-Oriented Architecture*.
http://en.wikipedia.org/wiki/Service-oriented_architecture, 2007.
- [69] Microsoft Corporation: *The Official Microsoft ASP.NET 2.0 Site*. <http://asp.net/>, 2007.
- [70] Free Software Foundation: *GNU General Public License*.
<http://www.gnu.org/copyleft/gpl.html>, 2007.
- [71] Oracle Corporation: *The Official Oracle Site*. <http://www.oracle.com>, 2007.
- [72] Codehaus Foundation: *The Official XFire Site*. <http://xfire.codehaus.org/>, 2007.
- [73] Red Hat Middleware: *JBoss Web Services*. <http://labs.jboss.org/jbossws/>, 2007.
- [74] MuleSource Inc.: *Mule Open Source ESB*.
<http://mule.mulesource.org/display/MULE/Home>, 2007.

- [75] Red Hat Middleware: *JBoss Messaging*. <http://labs.jboss.com/jbossmessaging/>, 2007.
- [76] Citrix Systems, Inc.: *Citrix Xen Server*. <http://www.citrixserver.com/Pages/default.aspx>, 2007.
- [77] Flynet s.r.o: *Flynet systémy – Devis*. <http://www.a-z.cz/hlavni.htm?./CZ/Products/Flynet.htm>, 2007.
- [78] Wikipedia: *Open Database Connectivity*. <http://en.wikipedia.org/wiki/ODBC>, 2007.
- [79] SourceForge, Inc.: *PhpBB*. <http://sourceforge.net/projects/phpbb>, 2007.
- [80] Red Hat Middleware: *JBoss Richfaces*. <http://labs.jboss.com/jbossrichfaces/>, 2007.
- [81] Oracle: *Oracle Warehouse Builder*. http://download.oracle.com/docs/cd/B31080_01/doc/install.102/b28224/toc.htm, 2006.
- [82] United Planet GmbH. *Intrexx Xtreme The Portal Builder*. <http://www.intrexx.com/en/intrexx-xtreme>, 2007.
- [83] Francis Shanahan: *Amazon.com[®] Mashups*, Wiley Publishing, Inc., 2007.
- [84] Interface21: *Spring Framework*. <http://www.springframework.org/>, 2007.
- [85] GNU: *GNU Lesser General Public License*. <http://www.gnu.org/licenses/lgpl.html>, 2007.
- [86] Apache Software Foundation: *Licenses*. <http://www.apache.org/licenses/>, 2007.
- [87] Mozilla.org: *SSL/TLS*. <http://www.mozilla.org/projects/security/pki/nss/ssl/>, 2007.
- [88] NICE Systems: *Performix Performance Manager*. http://www.niceperformix.com/products/performance_manager.html, 2007.
- [89] Pentaho Corporation: *Open Source Business Intelligence*. <http://www.pentaho.com>, 2007.
- [90] Honrick M.F., Marcadé E., Venkayala S.: *Java Data Mining: Strategy, Standard, and Practice*, Morgan Kaufmann, 2007.
- [91] KXEN Inc.: *KJDM*. http://www.kxen.com/products/analytic_framework/kjdm.php, 2007.
- [92] Sun Microsystems, Inc.: *JavaBeans*. <http://java.sun.com/products/javabeans/>, 2002.
- [93] Java Community Process: JSR 154: *JavaTM Servlet Specification 2.5*. <http://jcp.org/aboutJava/communityprocess/final/jsr154/index.html>, 2003.
- [94] Deshmukh H., Malavia J., Scarpino M.: *SCWCD Exam Study Kit*, second edition, Manning Publications, 2005.
- [95] Raible M.: *Comparing Web Frameworks*. <https://equinox.dev.java.net/framework-comparison/WebFrameworks.pdf>, 2006.
- [96] Apache Software Foundation: *Struts Tiles*. <http://struts.apache.org/1.x/struts-tiles/index.html>, 2007.
- [97] Hightower R.: *Facelets fits JSF like a glove*. <http://www.ibm.com/developerworks/java/library/j-facelets/index.html>, 2006.

- [98] Nei development website: *Aspects of the integration of tow or more applications*.
http://dev.nei.ch/integration_aspects.pdf, 2005.
- [99] Google: *Google Search*. <http://www.google.com>, 2007.
- [100] InfoWorld: *InfoWorld Awards*. <http://www.infoworld.com/awards/>, 2007.
- [101] Apache Software Foundation: *Apache ServiceMix*.
<http://servicemix.apache.org/home.html>, 2007.
- [102] Microsoft Corporation: *MDX Reference*.
<http://msdn2.microsoft.com/en-us/library/ms145506.aspx>, 2007.

Príloha A

Priložené CD

Priložené CD obsahuje nasledujúce dáta:

doc/	dokumentácia k systému EPA
epa/	zdrojové súbory systému EPA
diplomová práca.pdf	táto diplomová práca v pdf formáte
diplomová práca.xps	táto diplomová práca v xps formáte