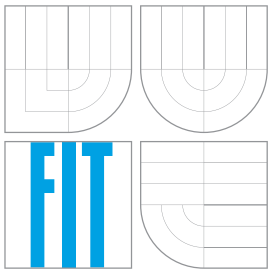
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INTEGRACE SOFTWAROVÝCH APLIKACÍ

INTEGRATING SOFTWARE APPLICATIONS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAROSLAV STROUHAL

VEDOUCÍ PRÁCE
SUPERVISOR

doc. RNDr. JITKA KRESLÍKOVÁ, CSc.

BRNO 2011

Abstrakt

Cílem této diplomové práce je: zpracovat problematiku integrace softwarových řešení, a navrhnout integraci takovýchto řešení, používaných v reálné firmě, do tamního informačního systému s využitím existujících datových formátů a rozhraní. Navrženou integraci implementovat a zhonotit výsledky práce.

Abstract

The aim of this master's thesis: processing problem of intregration software resolution and propose the integration such the resolution using in real company, implemented into local information system with using existing data formats and interface. Proposal integration must implement and evaluate resolution of work.

Klíčová slova

Databáze, OLAP, XML, SOAP, Webové služby, Integrace, Synchronizace, JAVA, JDBC, Datové formáty

Keywords

Database, OLAP, XML, SOAP, Web services, Integration, Synchronization, JAVA, JDBC, Data formats

Citace

Jaroslav Strouhal: Integrace softwarových aplikací, diplomová práce, Brno, FIT VUT v Brně, 2011

Integrace softwarových aplikací

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením paní doc. RNDr. Jitky Kreslíkové, CSc.

.....
Jaroslav Strouhal
25. května 2011

Poděkování

Chtěl bych poděkovat paní doc. RNDr. Jitce Kreslíkové, CSc. za odborné vedení a cenné rady při řešení diplomové práce.

Obsah

Kapitola 1

Úvod

Problematika integrace softwarových aplikací, kterou se zabývá tato diplomová práce, je poměrně aktuální. Mnoho společností používá různorodý software na různá odvětví své činnosti, ať už se jedná o účetnictví, řízení lidských zdrojů, řízení vztahů se zákazníky, různorodé docházkové či zabezpečovací systémy, skladové hospodářství nebo výrobní software.

Společnost pak může disponovat velkým množstvím informací, které se v mnohých případech mohou duplikovat. Navíc schází ucelený pohled na informace, které spolu souvisí, ale jsou obsaženy v různých softwarových řešeních. Představme si modelový příklad: zaměstnanec společnosti se vyskytuje v mzdové evidenci, docházkovém systému a gps kontrole firemních aut. Zaměstnancovy údaje se duplicitně vyskytují ve třech různých datových zdrojích a navíc neexistuje možnost celkového náhledu na pracovníka. Pokud by společnost chtěla zajistit hiearcii vedení s tím, že každý vedoucí by měl možnost nahlížet jen na své podřízené, tak narážíme na další problém. Jednotlivá softwarová řešení takovýto model mohou podporovat, ale je tím pádem nutné v každém zvlášť udržovat aktuální data. Vzniká tím prostor pro výskyt chyb a systém je hůře udržovatelný v aktuálním stavu.

Jak je vidět na příkladu, tak i při použití pouze tří různých softwarových produktů může vzniknout řada problémů, které se nadále prohlubují v případě, že takovýchto produktů chceme použít více. Nicméně se obvykle společnosti bez výskytu více druhů softwarového řešení neobejdou a to z důvodu, že rozsáhlá softwarová řešení, která by pokrývala celou paletu firemních požadavků a potřeb, jsou většinou poměrně nákladná na pořízení a údržbu.

Stručný obsah kapitol

Tato diplomová práce stučně popisuje v kapitole 2 datové formáty, rozhraní a úložiště dat, které jsou především využívány v rámci společnosti, pro kterou jsou výsledky této práce určeny.

Kapitola 3 analyzuje a popisuje aktuální stav softwarového vybavení ve společnosti. Softwarové vybavení je v této kapitole rozděleno podle jeho druhu do jednotlivých podkapitol.

Následující kapitola, tedy kapitola 4, uvádí informační požadavky jednotlivých uživatelských skupin v rámci informačního sytému společnosti. Popisuje způsob, jakým jsou uživatelé ke skupinám přiřazováni, jaké existují uživatelské skupiny a diagram užití, znázorňující potřeby jednotlivých skupin.

Kapitola 5 uvádí, jakým způsobem jsou ve společnosti zpracovávána data, kdo za ně nese zodpovědnost a návaznosti akcí jednotlivých uživatelů z hlediska zpracování těchto dat.

Strategie rozvoje informačního systému je popsána v kapitole 6. Změny vnitřní politiky společnosti bývají poměrně rychlé a dynamické, takže se projevují i ve strategii rozvoje informačního systému, proto strategie popsaná v této kapitole bude pravděpodobně za nedlouho zastaralá.

Kapitola 7 popisuje teorii a způsoby realizece integrace a samotný návrh integrace jednotlivých softwarových prostředků pro společnost. Každý návrh obsahuje popis jakým způsobem by měla být integrace realizována a co se od této integrace očekává, resp. co by mělo být jejím výsledkem.

Existující integrační nástroje, implementace vlastního integračního nástroje a realizované jednotlivé integrace jsou popsány v kapitole 8. V popisu implementovaných integrací je popsáno, jakým způsobem je integrace provedena, popř. zdůvodnění, z jakého důvodu nebyla integrace realizována.

Kapitola 9 obsahuje jednoduchou případovou studii, na které je demonstrována činnost nově implementovaného integračního nástroje mimo prostředí společnosti. Konfigurace pro jednotlivé integrace nelze mimo společnost testovat, resp. je vyžadována přítomnost na lokální síti, či virtuální síti (VPN).

Poslední kapitola 10 obsahuje zhodnocení výsledků této diplomové práce. Naznačuje jakým směrem by bylo možné projekt do budoucna rozvíjet.

Návaznost na semestální projekt

Tato diplomová práce navazuje na semestrální projekt a přebírá z něj kapitoly: Datové formáty a rozhraní, Analýza současného stavu softwarového vybavení, Informační požadavky jednotlivých skupin uživatelů, Způsoby zpracování dat, Strategie rozvoje informačního systému a Návrh integrace softwarových prostředků do jednoho funkčního celku. Všechny jmenované kapitoly prošly v průběhu vytváření této diplomové práce menšími změnami (např. kvůli změně firemní politiky) a úpravami.

Výrázné změny proběhly pouze v kapitole Informační požadavky jednotlivých skupin uživatelů. Ukázalo se, že kapitola v rámci semestrálního projektu přesně nekorespondovala s aktuálním stavem uživatelů.

Kapitola 2

Datové formáty a rozhraní

Možností jak řešit problematiku ukládání dat a přístupu k nim je mnoho. Cílem kapitoly je nastínit některé možnosti ukládání a přístupu, kde některé z těchto možností jsou dále použity v této diplomové práci.

Kapitola je rozdělena na tři logické části:

- Datové formáty,
- Datová rozhraní,
- Datová úložiště,

2.1 Datové formáty

Datové formáty standardizují ukládání dat stejného typu, aby s nimi bylo možné dále pracovat. Některé formáty mohou být ve způsobu ukládání benevolentní a naopak některé mohou být velice striktní.

Následuje příklad datových formátů:

- XML,
- PDF,
- CSV,
- XLS,
- EDW.

2.1.1 XML

XML je otevřeným datovým formátem, který je jedním z nejvýznamnějších počinů ve vývoji syntaxe pro popis dokumentů v historii. Jedná se o rozšiřitelný značkovací jazyk (z anglického Extensible Markup Language), je standardem schváleným konsorciem W3C pro značkování dokumentů, též nazýván jako meta-značkovací jazyk pro textové dokumenty.

Data jsou v dokumentech XML obsažena ve formě textových řetězců, jež jsou uzavřeny do textových značek, které tato data popisují. Není tudíž vhodné vkládat přímo binární data. Mohlo by se stát, že binární data poruší strukturu XML. Tento problém se řeší použitím vhodného kódování binárních dat — např. base64.

Tento formát je natolik flexibilní, že jej lze dále upravit pro tak různorodé oblasti, jako jsou webové stránky, elektronická výměna dat, vektorová grafika, genealogie, aplikace katastrů nemovitostí, serializace objeků, vzdálená volání procedur či systémy hlasové pošty. [10]

Následují příklady použití XML jako datových formátů:

SVG

Scallable Vector Graphics, což je standard přijatý konsorciem W3C, určený pro zápis výkresů ve 2D vektorové grafice pomocí XML. [10] Umožňuje popisovat 3 základní typy objektů: vektorovou grafiku, obrázky a texty. Pomocí něj je možné popisovat interaktivní a dynamické grafické objekty. [22]

MathML

Matematický značkovací jazyk (Mathematical Markup Language) je standardní XML-aplikace, přijatá konsorciem W3C, který je určen pro vkládání matematických rovnic do webových stránek a jiných dokumentů. [10]

CML

Chemický značkovací jazyk (Chemical Markup Language) byl jednou z prvních aplikací XML. Lze jej použít pro popis vzorců v chemii, fyzice pevné fáze, molekulární biologii a jiných molekulárních vědách. [10]

RDF

Resource Definition Framework je standardní aplikace přijatá konsorciem W3C, která je určena pro popis zdrojů a prostředků s konkrétním zaměřením na ty popisné údaje, které jsou většinou součástí katalogizačních lístků v knihovnách. [10]

DocBook

Jazyk určený především pro tvorbu technické dokumentace. Postupně se stále vyvíjí a dnes je možné ho použít obecně pro psaní článků a knih.

2.1.2 PDF

PDF je zkratkou z anglického Portable Data Format, čili přenosný datový formát. Dokument PDF je věrným obrazem zdrojového dokumentu, přičemž zachovává fonty, obrázky i vzhled stánky. [6]

Tento formát je podporován na většině používaných platforem, od běžných stolních počítačů s různými operačními systémy, až po různá mobilní zařízení.

2.1.3 CSV

Formát pro ukládání tabulkových dat. Záleží na implementaci, ale základní myšlenkou formátu je, že každý řádek ve formátu SCV reprezentuje jeden řádek tabulky a data jsou odděleny separátory. Data mohou být v uvozovkách či apostrofech a jako separátor může být použit např. středník nebo čárka.

2.1.4 XLS

Dokument aplikace MS Excel (pro aktuální verzi MS Office je přípona těchto dokumentů xlsx), slouží pro ukládání dat tohoto tabulkového procesoru. Tento formát podporují i multiplatformní open sourcové aplikace (např. Open office), takže nejsou žádné výrazné problémy s kompatibilitou a přenositelností.

2.1.5 EDW

Formát pro ukládání dat zakázek programem Profi plast 3.4.1. Nejedná se o otevřený formát, obsahuje neznámé formátování dat. Tvar uložení dat připomína binární XML, resp. mezi XML značkami jsou uložena binární data.

2.2 Datová rozhraní

Datová rozhraní tvoří bránu mezi jednotlivými typy dat a tím umožňují datovou manipulaci nad různými datovými zdroji.

Mezi datová rozhraní patří:

- JDBC,
- Java RMI,
- Web service,
- SOAP.

2.2.1 JDBC

JDBC je aplikační rozhraní mezi java aplikacemi a relačními databázemi a dále poskytuje standardní knihovnu pro přístup k těmto databázím. Pomocí JDBC API je možné získat přístup k široké škále různých databází SQL, pomocí stejné syntaxe. Nesnaží se standardizovat syntaxi SQL, standardizuje pouze mechanismus pro připojení k databázím, syntaxi pro odesílání dotazů, provádění transakcí a datové struktury reprezentující výsledek. Takto definované rozhraní je obecné, ale k reálnému použití je potřeba zavést ovladač. Ovladač je část software, která umí komunikovat s databázovým serverem. Vše co je potřebné pro zavedení ovladače, je zavedení příslušné třídy. Statický inicializátor v samotné třídě vytvoří automaticky instanci ovladače a registruje ji pomocí správce ovladačů JDBC. Pro připojení k databázi se používá připojovací URL (angl. Connection string), ve které je nutno specifikovat umístění databáze, port a název databáze. Přesný formát se obvykle liší u jednotlivých databází a bývá definován v dokumentacích daných databází. [11]

2.2.2 Java RMI

RMI je zkratkou z anglického Remote Method Invocation, čili vzdálené volání metod. Jak již název napovídá, jedná se o distribuované volání metody z aplikace běžící na jednom virtuálním stroji aplikací běžící na jiném virtuálním stroji.

Koncept má výhodu v tom, že nezáleží na tom, jesli je metoda volána na vzdáleném počítači, anebo lokálně. Z pohledu programátora v tomto není žádný rozdíl.

Nevýhodou konceptu je fakt, že jej lze použít jen v rámci java aplikací.

2.2.3 Web service

Webové služby umožňují jednoduchou komunikaci mezi heterogenními systémy, protože komunikace je založena na platformově nezávislých technologiích (XML a HTTP). Aplikace si mezi sebou posílají zprávy XML, které přenášejí dotazy a odpovědi jednotlivých aplikací. [16]

Návrh aplikací založených na webových službách je moderní, s výhodou využitelný při integraci aplikace do složitějšího celku.

2.2.4 SOAP

SOAP je obálka okolo přenášených zpráv, používá XML syntaxi a je obvykle posílán pomocí HTTP, či HTTPS protokolu. SOAP ve speciálním jmeném prostoru definuje několik elementů, které slouží pro přenášení jednotlivých částí přenášených dat. Většina zpráv zasílaných pomocí SOAP obsahuje pouze elementy Envelope (obálka zprávy) a Body (kontejner pro přenos samotné zprávy). [16]

Protokol je platformově nezávislý, proto je vhodným prostředkem na propojování různorodých systémů. S výhodou jej lze využít např. k posílání zpráv ze systému na systém, či předávání dat.

2.3 Datová úložiště

Obecně datová úložiště slouží pro ukládání persistentních dat. Mezi datová úložiště mimo jiné patří:

- Databáze,
- OLAP.

2.3.1 Databáze

Co se databází týče, tak se jedná o téměř standardní úložiště persistentních dat a to minimálně pro aplikace, které pracují s větším množstvím dat, které se již např. nevyplatí parsovat z XML, ale zároveň se nejedná o obrovské množství dat (řádově stovky gigabytů a více). Optimální velikost databáze je podmíněna i typem a výrobcem.

Nejrozšířenějším typem databázových systémů jsou relační databáze. Mezi další typy databází patří například: post-relační, objektově-relační, objektové, prostorové a časové.

Databázové systémy je možno rozlišovat i podle dalších vlastností a to zejména: rychlost, dostupnost, přenositelnost, cena, transakční zpracování, zabezpečení, atd. Není jednoznačné určit jaká databáze je nejlepší. Vždy je potřeba zjišťovat, jaká databáze je nejvhodnější pro jaký účel. Zde může rozhodovat cena, doba přístupu, nebo velikost db.

Následuje soupis významnějších databázových systémů:

- MySQL,
- PostgreSQL,
- SYSDBA,
- MS SQL,

- Oracle,
- Firebird.

Firebird

Databázový systém Firebird není natolik známý a používaný jako ostatní jmenované systémy, ale bude zde podrobněji rozveden, protože ve společnosti, pro kterou tato diplomová práce vzniká, se používá jako jeden z primárních zdrojů dat.

Firebird (známý též jako FirebirdSQL) je multiplatformní relační databáze, kterou vyvíjí a spravuje Firebird Foundation. Jedná se o odnož databáze InterBase společnosti Borland, která ji uvolnila pod licencí InteBase Public Licence. Licence je odvozena od Mozzila Public Licence. Mezi výhody databázového systému Firebird patří především podpora ze strany různých nástrojů a platforem, např. Delphi, .NET, Java, či PHP. [19]

Systém umožňuje vytvářet pohledy na data, procedury, spouště (angl. trigger) a UDF (User Defined Function), které umožňuji naprogramovat funkci v nějakém nativním jazyce (nejčastěji C) a tu pak přidat do databázového systému v podobě dynamické knihovny. Dále umožňuje transakční zpracování a obsahuje systém zasílání zpráv, resp. z procedury, či spouště je možno volat zprávu, kterou mohou zachytit posluchači. Posluchači musí implementovat rozhraní k tomu určené, připojit se k databázovému serveru a informovat server o jakou zprávu mají zájem. Zpráva je pouze textový identifikátor, který je možno pojmenovat jakýmkoli způsobem. [9]

2.3.2 OLAP

Nejedná se o datové úložiště jako takové, obvykle jsou data v datovém skladu (Data Warenhouse), nicméně někteří autoři tyto dva pojmy přímo spojují v termínech jako "OLAP databáze". [20] Z tohoto důvodu je termín OLAP v této práci vložen do podkapitoly o datových úložištích.

OLAP, z anglického Online Analytical Processing, je technologie uložení dat v databázi, která umožňuje uspořádat velké objemy dat tak, aby byla data přístupná a srozumitelná uživatelům zabývajícím se analýzou obchodních trendů a výsledků (Business Intelligence). [8]

Kapitola 3

Analýza současného stavu softwarového vybavení

Společnost disponuje různým softwarovým vybavením, které je možno rozdělit na následující typy:

- Databáze,
- Informační systémy,
- Specializované servery,
- Specializovaný software.

3.1 Databáze

Ke všem popsaným databázím je možné získat přístup pomocí uživatelského jména a hesla. Ostatní databáze zde nejsou uvedeny. Dále všechny popsané databáze jsou Firebird databázemi, které jsou popsané v 2.3.1.

- Databáze informačního systému,
- Výrobní databáze,
- Nabídková databáze.

3.1.1 Databáze informačního systému

Jedná se o základní datový zdroj informačního systému společnosti. Jsou zde uložená data o uživatelích, jejich přístupová práva, závislosti mezi pracovníky a pracovními místy, pokladní knihy jednotlivých poboček a docházkový systém pro pracovníky mimo sídlo společnosti.

3.1.2 Výrobní databáze

Výrobní databáze obsahuje hlavně informace o zakázkách a zákaznících. Každá zakázka má unikátní označení a skládá se z jednotlivých výrobních pozic. Zakázky se grupují do dávek, podle kterých se vyrábějí.

Bohužel není databáze optimálně navržená, resp. některé informace jsou obsaženy pouze v XML souboru, který se vztahuje ke každé výrobní pozici, a který je uložen jako blob v databázi.

3.1.3 Nabídková databáze

Nabídková databáze má stejnou strukturu jako výrobní databáze (došlo k rozdělení těchto dvou databází kvůli místu, které databáze zabíraly). Navíc oproti výrobní databázi obsahuje tabulku, do které se pomocí databázových spouští (angl. trigger) kopírují data, která jsou potřeba ke sledování nabídek.

3.2 Informační systémy

Informační systém se skládá ze dvou částí, resp. z původního informačního systému, který se aktuálně již téměř nepoužívá a aktuálního informačního systému.

- Původní informační systém,
- Aktuální informační systém.

3.2.1 Původní informační systém

Jedná se o první verzi informačního systému. Informační systém funguje na externím serveru na platformě PHP s databází MySQL. Stále se některé části používají, proto není aktuálně možné ukončit jeho činnost.

Obsahuje základní informace o zakázkách a jejich průběhu. V tomto informačním systému nejsou aktualizovaná data o uživatelích - problémy s přístupovými právy. Data jsou zde obvykle ukládána ručně, jsou nekonzistentní a obsahují chyby.

3.2.2 Aktuální informační systém

Aktuální informační systém funguje na firemním serveru, také na platformě PHP s využitím firemních Firebird databází. Využívá podporu JAVA aplikací u některých operací (např. java applet pro generování čárového kódu).

Informace o zakázkách čerpá z výrobní databáze. Z původního informačního systému si načítá dílčí informace o zakázkách, pomocí webových služeb a snaží se tato data přefiltrovat podle přístupových práv (obtížné kvůli chybám, pomalé).

Informace o nabídkách čerpá z databáze nabídek. Nabídky slouží jen pro přehled vedení společnosti, respektive neslouží k žádnému automatickému zpracování.

Ostatní informace čerpá z databáze informačního systému. Jedná se především o uživatele, jejich přístupová práva a závislost mezi nadřízeným a podřízeným pracovníkem. Činnosti v rámci informačního systému jsou rozvedeny v kapitole 4.

Obsahuje moduly pro získávání údajů z excelových tabulek (soubory ve formátu XLS) a generování excelových tabulek a dokumentů ve formátu pdf.

3.3 Specializované servery

Společnost vlastní několik serverů, specializující se na určitý druh činnosti.

Mezi tyto servery patří:

- Účetnictví POHODA,
- Mzdové učetnictví PAMICA,
- Docházkový systém IKOS.

3.3.1 Účetnictví POHODA

Obsahuje firemní účetnictví, skladové účetnictví a všechny pokladní knihy. Pokladní knihy v účetnictví POHODA a pokladní knihy v rámci informačního systému společnosti nejsou propojeny a společnost tím pádem vede většinu pokladních knih dvakrát.

Účetní a fakturantky mají na svých počítačích klientskou aplikaci, pomocí které pracují s daným účetnictvím na serveru.

Používá MS SQL databázi (přístupové jméno a heslo zná pouze výrobce software) a běží na serveru MS Windows.

Umožňuje, mimo samotného vedení účetnictví, provádět různé exporty a importy dat (ve formátu XML, XLS, CSV), reporty ve formátu PDF, nastavovat práva uživatelům a nastavit pravidelné zálohy.

3.3.2 Mzdové účetnictví PAMICA

Obsahuje firemní mzdové účetnictví. Obecné informace o produktu jsou totožné s účetnictvím POHODA (jedná se o totožného výrobce), navíc obsahuje informace o všech zaměstnancích firmy. Účetnictví POHODA a mzdové účetnictví PAMICA mohou být navzájem propojeny, resp. jednotlivé systémy si mohou navzájem předávat data (pokud by to bylo nutné), právě díky totožnému výrobci.

3.3.3 Docházkový systém IKOS

Zabezpečuje docházkový systém zaměstnanců v sídle firmy. Obsahuje informace o zaměstnancích, jejich přiděleném čipu (zaměstnanci evidují svůj příchod či odchod při vstupu do areálu u terminálu) a do jakého oddělení zaměstnanci patří. Systém umožňuje generovat reporty jednotlivých zaměstnanců.

Každé oddělení musí mít přiděleno vedoucího pracovníka, takže se znovu vytváří hierarchie podřízených a nadřízených pracovníků.

Terminály lze synchronizovat pomocí formátu scv, resp. terminál umožňuje exportovat i importovat data, která má ve vnitřní paměti. Tato paměť je omezená a o danou synchronizaci se stará démon, který beží na serveru.

Data se ukládají do lokální Firebird databáze (přístupové jméno a heslo zná pouze výrobce software).

3.4 Specializovaný software

Následuje výčet specializovaného software, který společnost vlastní:

- Profi plast,
- Evidence,
- OptyWay.

3.4.1 Profi plast

Program vyrobený na zakázku pro potřeby společnosti určený na naceňování zakázek. Je nutným softwarovým základem na každé pobočce společnosti. Umožňuje ze zakázky vytvořit report do PDF určený pro zákazníka. Neustále se vyvíjí a probíhají na něm opravy a úpravy.

Zakázky si program Profi plast ukládá do lokálního souborového systému ve svém vlastním formátu (EDW 2.1.5). Má možnost uložit zakázku do databáze (není automatické - uživatel musí explicitně uložit), což se využívá k nahrávání zakázek do výrobní databáze (v sídle firmy) a pro ukládání nabídek do databáze nabídek (na pobočkách přes VPN).

3.4.2 Evidence

Program taktéž vyrobený na zakázku pro potřeby společnosti stejnou společností jako Profi plast, určený na výrobu zakázek. Umožňuje vytvářet ze zakázek dávky a tyto dávky následně vyrábět. Výroba neprobíhá zcela automatizovaně, resp. program Evidence vygeneruje soubory obsahující instrukce pro stroje k vyrobení dané dávky. Tyto soubory pak pracovníci překopírují ke strojům, které následně vyrábí výrobky.

Program evidence umožňuje kontrolovat stav zakázky. Změnu stavu zakázky provádí zaměstnanci u jednotlivých pracovišť pomocí čtečky čárových kódů.

Mezi další vlastnosti patří především hromadné objednání materiálu na zakázky, které jsou připraveny do výroby. Generuje exporty (v PDF, XLS, DOC), které jsou dodavatelům posílány elektronicky emailem.

3.4.3 Opty-Way

Jedná se o specializovaný program, zabývající se nářezovou optimalizací skel a jejich výrobou. Používá svůj formát pro ukládání optimalizací na lokální souborový systém. Program je ve dvou verzích:

- *Uživatelská* — umožňuje generovat soubory s nářezovou optimalizací, ručně vkládat skla, nebo importovat skla ve formátu XLS. Tento import se provádí ručně a používá se XLS soubor generovaný informačním systémem.
- *Strojová* — umožňuje importovat soubory s nářezovou optimalizací a měnit nastavení stroje. Pracovníci ručně určují, která optimalizace se má použít k nařezání, což vede k možnosti vzniku chyby : mohou některou optimalizaci vynechat, či vyrobit dvakrát, čož je nežádoucí stav.

3.5 Outsourcovaný software

V současné době společnost používá pouze jediný outsourcovaný software a to webovou aplikaci Carcontrol zprostředkovávanou společností O2.

3.5.1 O2 Carcontrol

O2 Carcontrol je webovou aplikací umožňující přidávat, editovat a sledovat jednotlivá auta společnosti. Sledování může probíhat přímo zjištěním polohy aut podle gps, nebo výpisu knihy jízd aut, resp. tras po kterých auto jelo (odkud auto vyjelo, kam dojelo, kolik ujelo kilometrů, atd.). [14]

Systém umožňuje vytvářet exporty knih jízd aut pomocí SOAP protokolu. Tento export lze parametrizovat a získat data za určité období, případně všech aut, či jen jednoho konkrétního auta.

Kapitola 4

Informační požadavky jednotlivých skupin uživatelů

Stávající informační systém byl navržen tak, že práva pro jednotlivé skupiny uživatelů je možno definovat dynamicky přímo v informačním systému, bez nutnosti programového zásahu. Z tohoto důvodu je relevantnější se zabývat otázkou informačních požadavků abstraktních uživatelů, kteří mají nějaké konkrétní právo pro přístup k některému z modulů informačního systému, či jen rozšiřující právo v rámci daného modulu. Nicméně tento přístup není příliš přehledný pro popis, proto se bude tato kapitola dále zabývat hlavně několika pojmenovanými uživateli, kteří obsahují většinu funkčnosti informačního systému.

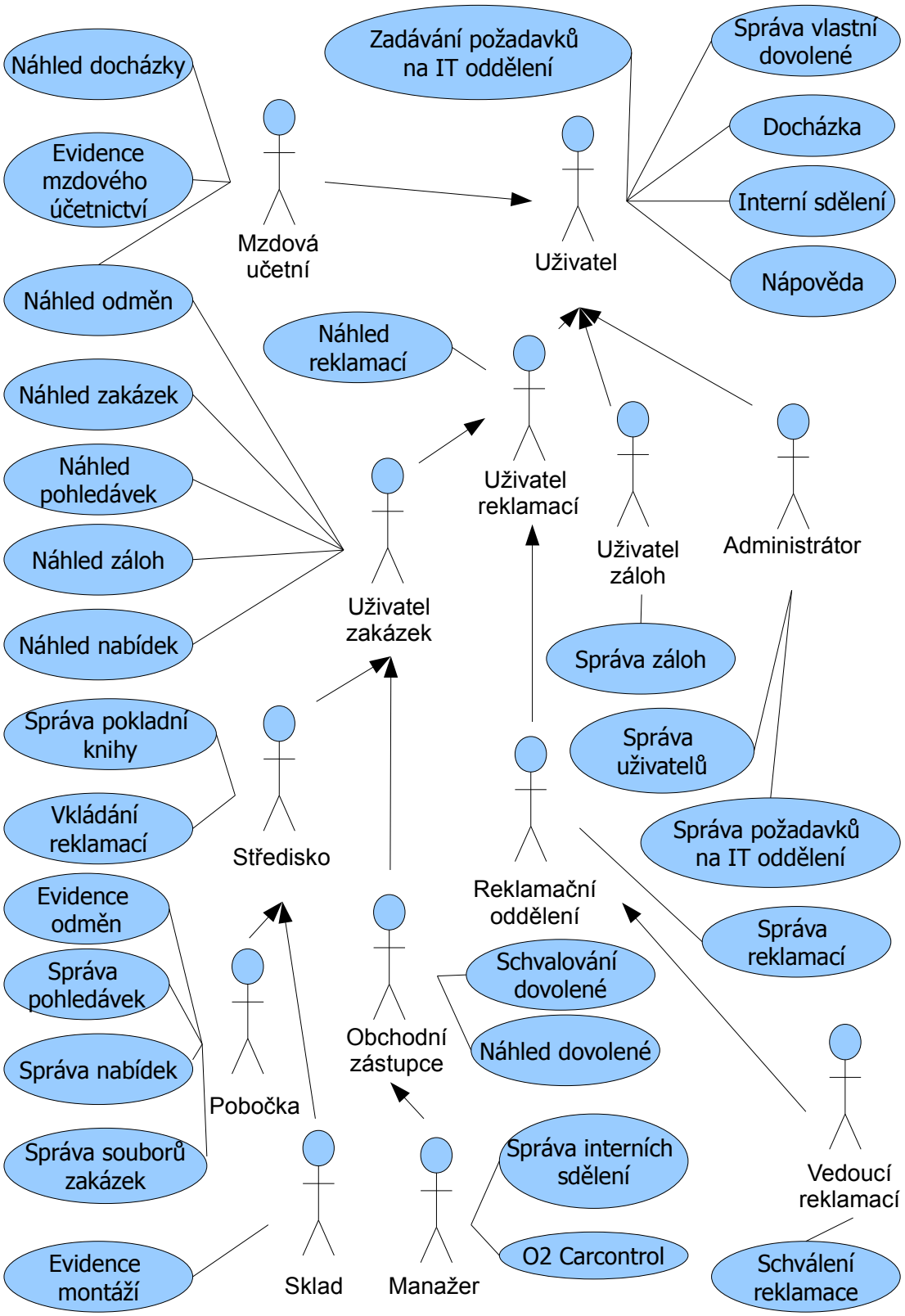
Veškerá data informačního systému musí být filtrována v závislosti na právech daného uživatele, tzn. pod jakou firmu uživatel patří, případně, jesli je uživatel pojmenovaným aktérem informačního systému. Data, na která informační systém umožňuje nahlížet, mohou pocházet z externích zdrojů. Takováto data je pak většinou těžké až téměř nemožné filtrovat (minimálně je toto filtrování značně pomalé), což si odporuje požadavku na rychlý přístup k informacím z informačního systému.

V následujících podkapitolách budou představeni jednotliví uživatelé informačního systému, jejich činnosti a jaké jsou informační požadavky jedntlivých činností:

- Diagram užití,
- Aktéři informačního systému,
- Specifikace případu užití.

4.1 Diagram užití

Samotný diagram užití je poměrně komplikovaný a nepříliš přehledný. Snahou bylo pokud možno co nejvíce používat abstraktní aktéry pro zvýšení přehlednosti celého diagramu. V případě náhledu docházky se bohužel nepodařilo vytvořit samostatnou vazbu na jednoho aktéra.



Obrázek 4.1: Diagram užití aktérů společnosti.

4.2 Aktéři informačního systému

Ve výše uvedeném diagramu užití (obázek 4.1) je možno nalézt dva druhy aktérů:

- Pojmenovaní aktéři informačního systému,
- Abstraktní aktéři informačního systému.

4.2.1 Pojmenovaní aktéři informačního systému

Pojmenovaní aktéři mají v informačním systému specifické, předem určené postavení. Rozdíl mezi obecným uživatelem a pojmenovaným uživatelem spočívá v tom, že obecný uživatel může vykonávat jakoukoli činnost, která není na první pohled zřejmá, resp. jeho činnost je rozpoznatelná jen podle přidělených práv, kdežto pojmenovaný uživatel se řídí předem určenými a známými právy, podle diagramu užití.

Následuje soupis pojmenovaných uživatelů informačního systému:

- Administrátor,
- Manažer,
- Mzdová účetní,
- Obchodní zástupce,
- Pobočka,
- Reklamační oddělení,
- Sklad,
- Uživatel záloh,
- Vedoucí reklamací.

Administrátor

Uživatelská role administrátora z hlediska informačních požadavků není pro účely této diplomové práce příliš zajímavá, resp. veškeré činnosti provádí v rámci jedné databáze informačního systému.

Přichází do úvahy pouze spojení docházkového systému ve firemním informačním systému a docházkového systému v sídle firmy. Aktuálně musí administrátor vytvářet účty pro docházkový systém v sídle firmy, ke kterém musí přidělovat čipy. Kdežto v informačním systému se uživatel registruje do docházky sám a není s čipy nijak svázán.

Teto princip jasně vede na duplicitu dat, možnost chyby a zatěžování administrátora. Ideálním řešením by bylo, kdyby administrátor mohl k uživatelům v informačním systému přidělit čip a toto nastavení se automaticky přeneslo na docházkový systém, včetně nastavení oddělení a vedoucího.

Manažer

Manažer je nejvyšší dohledovou jednotkou společnosti. Má přístup k informacím o všech pracovnících, či zakázkach, v rámci společnosti na kterou má práva. Společností může takovýto manažer mít na starosti i více, nebo jen jednu. Počet společností, nad kterými manažer provádí dohled, není nijak omezen.

Mzdová účetní

Aktér určený primárně k počítání výplat zaměstnanců prostřednictvím programu Pamica. Na svou činnost využívá: program Pamica, náhled docházky a náhled odměn. Bohužel tato činnost není nijak automatizovaná, takže toto je nutné víceméně provádět ručně.

Obchodní zástupce

Má možnost sledování pouze svých poboček a skladů. Vidí všechny moduly jako pobočka, ale má k těmto modulům vyšší práva, takže s nimi může různě manipulovat nad úroveň pobočky (editovat, mazat).

Relace mezi obchodními zástupci a pobočkami je N:M, resp. pro jednu pobočku může být více obchodních zástupců a naopak, jeden obchodní zástupce obvykle má více poboček.

Pobočka

Aktér s největší četností v systému. Každá pobočka má přidělen unikátní identifikátor (zkratku pobočky), díky kterému lze rozlišovat která zakázka patří které pobočce.

Reklamační oddělení

Reklamační oddělení má k dispozici setříděný seznam reklamací podle aktuálního stavu reklamace v informačním systému. Má možnost reklamaci schválit, zamítnout, nebo nechat přepracovat v případě nesrozvnalostí.

Informace o reklamacích se nacházejí ve výrobní databázi. Vztah reklamací k zakázce je N:1, resp. zakázka může mít více reklamací.

Sklad

Má možnost zapisování a editování dat pouze pro pobočky, spadající pro daný sklad. Nemá právo náhledů aktuálního stavu poboček, tak jako tomu je u obchodních zástupců. Původní relace mezi sklady a pobočkami byla 1:N, resp. jeden sklad obsluhoval několik poboček. Po poslední aktualizaci systému (kvůli změně firemní politiky) se tato relace změnila na N:M, resp. jeden sklad obsluhuje více poboček a jedna pobočka může být obsluhována více sklady.

Uživatel záloh

Primárním úkolem uživatele záloh je přijímat zakázky firemním emailem, tiskout je, zakládat do složek a v případě zaplacené zálohy zapisovat tyto zakázky do starého informačního systému.

Ke své činnosti musí využívat programy: Účetnictví pohoda, Outlook, Profi plast a starý informační systém společnosti.

Vedoucí reklamací

Značně obdobný aktér jako reklační oddělení, hlavní rozdíl spočívá v tom, že aktér vedoucí reklamací má schvalovací právo na reklamace a nese případnou zodpovědnost za schválenou reklamaci. Informační systém eviduje kdo relamaci zpracovával a kdo ji schvaloval, kvůli zpětné kontrole.

4.2.2 Abstraktní aktéři informačního systému

Abstraktní aktéři informačního systému použití v diagramu užití slouží čistě jen pro účely této diplomové práce pro lepší přehlednost diagramu.

Mezi tyto aktéry patří:

- Středisko,
- Uživatel,
- Uživatel reklamací,
- Uživatel zakázek.

Středisko

Středisko je abstraktním aktérem, reprezentující prodejní místo, popř. místo určené pro interakci se zákazníky. Jedná se o reálné místo (pronajaté prostory, apod.) mimo sídlo společnosti, tudíž má každé takovéto středisko vlastní adresu, na kterém se nachází.

Uživatel

Nejzákladnější typ abstraktího aktéra, jehož činnosti používají všichni ostatní uživatelé v rámci diagramu užití. Obecný uživatel (nastavitelný pomocí práv jakýmkoliv způsobem) nemusí nutně obsahovat tyto činnosti.

Uživatel reklamací

Jeden ze základních abstraktních aktérů, jeho činnosti používají jak aktéři reklamačního oddělení, tak aktéři pracující nějakým způsobem se zakázkami.

Uživatel zakázek

Aktér pro různé náhledy a obecně různé činnosti se zakázkami. Jeho činnosti využívají jak výkonné jednotky (pobočky a sklady), tak dohledové jednotky (manažer a obchodní zástupce).

4.3 Specifikace případu užití

Tato podkapitola obsahuje stručný popis jednotlivých případů užití, s jakým softwarem aktér pracuje a k jakým datům přistupuje.

Docházka

Docházku používají bez výjimky všichni zaměstnanci společnosti, potažmo všichni uživatelé informačního systému. Pro evidenci docházky je ve společnosti použit program IKOS a docházkový systém v informačním systému. Zaměstnanci v areálu sídla firmy používají hlavně program IKOS, resp. s přidělenými čipy evidují svůj příchod, či odchod, na terminálu umístěném na vrátnici při vchodu do areálu společnosti.

Zaměstnanci mimo areál sídla společnosti používají docházkový systém v rámci informačního systému.

Evidence montáží

Zaevidování montáže do informačního systému vykonává primárně sklad, který zadává kdy montáž proběhla a který montér prováděl jakou činnost. Na základě této evidence se pak počítají odměny pro montéry.

Samotné zaevidování se ukládá do výrobní databáze, do příslušné tabulky.

Evidence mzdového účetnictví

Není součástí informačního systému. Agenda mzdového účetnictví je vedena v programu PAMICA mzdovými účetními společnosti.

Evidence odměn

Evidenci odměn zadává primárně pobočka. Určuje ke svým zakázkám kdo danou zakázku zaměřil a kdo ji sjednal. Na základě této evidence dostávají pracovníci odměny.

Interní sdělení

Obsahují nová oznámení, od vedení společnosti, určená všem zaměstnancům. Interní sdělení je možno zacílit i na určitou skupinu uživatelů, jako např. pobočky, obchodní zástupce a pod. Díky interním sdělení má každý zaměstnanec okamžitý přístup k novinkám, které se jej týkají, od vedení společnosti z jakéhokoli místa (resp. počítače s připojením na internet).

Náhled docházky

Jak již bylo zmíněno, společnost disponuje dvěma docházkovými systémy, proto náhled docházky probíhá jak v informačním systému, tak v programu IKOS.

Náhled dovolené

Náhled dovolené je určen pro obchodní zástupce a manažera. Tito aktéři mohou sledovat shválenou dovolenou svých podřízených, což umožňuje lépe plánovat záskok za jednotlivé pracovníky, popř. shvalování další dovolené.

Náhled odměn

Umožňuje abstraktnímu aktérovi uživatel zakázek sledovat odměny, buď vlastní, svých podřízených v případě obchodního zástupce a manažera, nebo všech pracovníků v případě mzdové účetní. Slouží i pro zpětnou kontrolu zadaných odměn pro nadřízeného.

Náhled pohledávek

Umožňuje abstraktnímu aktérovi uživatel zakázek sledovat pohledávky jednotlivých zakázek, případně je možno zjistit zda, popř. kdy byla zakázka splacena. Tyto informace se nacházejí ve výrobní databázi ve formě jednotlivých úhrad, které jsou spojené přímo se zakázkou.

Náhled nabídek

Umožňuje abstraktnímu aktérovi uživatel zakázek sledovat nabídky jednotlivých poboček. Tento náhled je spíše určen pro obchodní zástupce a manažera, pro lepší rozhodování zda je pobočka lukrativní, popř. zda je potřeba zvýšit poměr reklamy v dané oblasti, kde se pobočka nachází.

Náhled reklamací

Abstraktní aktér uživatel reklamací má možnost sledovat a filtrovat aktuální reklamace a zjišťovat podrobnosti o těchto reklamacích. Soupisku reklamací ovliňuje typ uživatele, který se na tyto reklamace dívá (např. pobočka vidí jen své reklamace a pod.).

Náhled zakázek

Zobrazuje seznam zakázek a základní informace k nim, jako je např. označení zakázky, datum sjednání a stav v jakém se zakázka nachází. Seznam obsahuje u každé zakázky tlačítko pro přechod k reklamacím. Data o zakázkám se získávají z výrobní databáze.

Náhled záloh

Zobrazuje seznam zaplacených záloh podle datumu zaplacení. Tento seznam obsahuje na posledním řádku sumu prodaných oken s cenou a má možnost filtrace podle datumu, či poboček, takže lze získat přehled prodeje oken a zaplacených záloh za vybrané časové období (zakázka musí mít vždy zaplacenou zálohu).

Nápověda

Každý uživatel informačního systému má k dispozici nápovědu. Jedná se o jednoduché HTML stránky s návody, tipy a radami, spravované většinou administrátorem systému (ne pomocí informačního systému).

O2 Carcontrol

Primárním aktérem kontroly aut je manažer, který se přihlásí do informačního sytému O2 Carcontrol, se svým uživatelským jménem a heslem. Má možnost sledovat aktuální pozici všech firemních aut, jejich knihu jízd, popř. editovat data o firemních autech.

Správa pokladní knihy

Primárním aktérem správy pokladní knihy je abstraktní aktér středisko, který má možnost přidávat záznamy do pokládní knihy, tisk pokladních dokladů a tisk pokladní knihy za vybrané časové období. Pokladní kniha automaticky generuje čísla výdejových a příjmových dokladů, uživateli pak odpadá práce s vypisováním dokladu, případně vzniku chyby.

Schválení reklamace

Nepřidává žádnou informační závislost, či hodnotu. Jedná se pouze o rozšíření práv uživatele schválení reklamace. Při schválení, či zamítnutí reklamace, se uloží informace o tom, který uživatel byl jejím schvalovatelem do tabulky reklamací ve výrobní databázi.

Schvalování dovolené

Schvalování dovolené je obecně určeno pro vedoucí pracovníky, kteří schvalují dovolenou svých přímých podřízených. Jedná se o rozšíření náhledu dovolené, primárním aktérem schvalování dovolené je obchodní zástupce.

Správa interních sdělení

Tuto činnost by mělo provádět pokud možno co nejméně uživatelů, protože slouží k informování všech pracovníků společnosti (mohl by nastat problém se zneužitím a pod.). Primárním aktérem je manažer, který má na starosti informovat své podřízené o aktuálních novinkách, nebo změnách firemní politiky.

Správa nabídek

Primárním aktérem správy nabídek je pobočka, která vede nabídku v programu Profi plast. Nabídku si ukládá v souborovém systému na vlastním počítači a eviduje ji, také přes program Profi plast, do informačního systému společnosti. Vlastní uložení probíhá do databáze nabídek.

Jednotlivé pobočky mohou ke svým nabídkam dopisovat doplňkové informace, které program Profi plast neukládá (poznámku, slevu, apod.).

Správa pohledávek

V rámci informačního systému může aktér pobočka spravovat vlastní pohledávky, resp. může zadat reklamační slevu a slevu z prodlení (obě jsou limitovány do určité ceny zakázky). Správa pohledávek podobně jako tomu je i u schválení reklací pouze rozšiřuje náhled pohledávek o možnost editace.

Správa požadavků na IT oddělení

Aktér administrátor má za úkol spravovat a řešit požadavky na IT oddělení od uživatelů systému. Vzhledem k tomu, že všichni uživatelé informačního systému mají možnost zadání tohoto požadavku, tak může být počet požadavků v určitých situacích značně vysoký. Administrátoři, pokud jich je více, mají možnost si požadavky rozdělit, např. podle společností a nést tak zodpovědnost za řešení požadavků jen pro určitou společnost.

Správa reklamací

Umožňuje aktérovi reklamační oddělení řešit reklamace v informačním systému. Pro zadaný popis reklamace od pobočky doplní reklamační oddělení požadované řešení, popř. co je potřeba objednat pro vyřešení reklamace. Po schválení lze do reklamace doplnit datum dodání zboží, vyskladnění ze skladu, expedice a datum vyřešení.

Reklamační oddělení musí navíc pro svou činnost používat program Evidence, pomocí kterého zjišťují kompletní informace o zakázce, které se v informačním systému nevyskytují. Jedná se především o výrobní výkresy, použité součástky a o smlouvu se zákazníkem.

Správa souborů zakázek

Nejedná se přímo o činnost v rámci informačního systému, ale má ji na starost aktér pobočka. Soubory zakázek (i nabídek) si musí vést pobočka v souborovém systému vlastního

počítače. Jakmile je zakázka uzavřena a je zaplacena záloha zákazníkem, pošle pobočka emailem soubor zakázky do sídla společnosti, kde dojde k zaevidování do záloh ve starém informačním systému. Pobočka si musí sama určovat vlastní unikátní název zakázky. Označení zakázky v rámci firmy je generováno automaticky, výše zmíněný unikátní název zakázky slouží pouze pro vlastní potřebu pobočky. Generování označení zakázky probíhá až v sídle společnosti a unikátní název pak pobočce slouží pouze pro kontrolu, že se jedná o jednu a tu samou zakázku.

Správa uživatelů

Umožňuje aktérovi administrátor vytvářet uživatele, měnit skupinu ke které patří, resetovat heslo a zneaktivnit uživatelský účet. Uživatelé jsou rozděleni na uživatele a uživatele, kteří jsou zároveň střediskem (pobočka a sklad).

Správa vlastní dovolené

Slouží pro všechny aktéry informačního systému pro zadávání dovolené a kontroly, zda byla dovolená v daný termín schválena. Neschválená dovolená může obsahovat poznámku se zdůvodněním od nadřízeného pracovníka.

Správa záloh

Uživatel záloh eviduje zálohy ve starém informačním systému. Podklady k evidenci záloh získává z emailu, na který mu chodí uzavřené zakázky, které si otevže v programu Profi plast a přepíše relevatní informace do starého informačního systému. Následně uloží soubor zakázky do souborového systému serveru, ze kterého si pak zakázku převezme výrobní proces (objednání materiálu, výroba, atd.).

Vkládání reklamací

Aktér uživatel zakázek má možnost ke každé zakázce vytvořit reklamaci. Reklamace vyžaduje textový popis a zadaný typ reklamace. Mezi nepovinné atributy, které uživatel nemusí zadat při vytváření reklamace, patří: fotografie závady, viník, nebo sleva, kterou zákazník požaduje. Fotografie se ukládají do souborového systému informačního systému.

Zadání požadavků na IT oddělení

Všichni uživatelé informačního systému mají možnost zadat požadavek na IT oddělení, v případě nahlášení chyby v informačním systému společnosti, nainstalování aktualizace, chyba tiskárny a pod. Primárně tyto požadavky řeší administrátoři systému, pomocí práv lze nastavit, aby měl daný administrátor v nabídce požadavky všech firem, nebo jen některých vybraných.

Kapitola 5

Způsoby zpracování dat

Společnost nezpracováva svá data žádným automatizovaným prostředkem, nevlastní datový sklad, ani nástroje pro podporu business intelligence. Data k analýze obvykle zpracovávají vedoucí pracovníci daného útvaru ručně do excelovských tabulek (soubory ve formátu XLS), popř. se využívají reporty z informačního systému.

Z těchto důvodů bude kapitola věnována pouze popisu zpracování informací a předávání dat ve společnosti a reportům z informačního systému.

- Zpracování informací a předávání dat ve společnosti,
- Reporty z informačního systému.

5.1 Zpracování informací a předávání dat ve společnosti

- Zpracování zakázek,
- Zpracování mezd.

5.1.1 Zpracování zakázek

V této podkapitole budou popsány způsoby manipulace a zpracování dat týkajících se zakázek od počáteční vytoření nové zakázky, přes výrobní procesy, až po celkové dokončení zakázky.

Cenová nabídka

Takováto cenová nabídka vzniká získáním poptávky od zákazníka, ať už telefonicky, emailem, či při osobním setkání.

Nacenění probíhá v programu Profi plast. Uživatel na pobočce má možnost si cenovou nabídku uložit do lokálního souborového systému ve speciálním formátu edw (formát je popsán v 2.1.5) a později pozměnit, pokud si to zákazník žádá. Zároveň má uživatel povinnost zaevidovat nově vzniklou nabídku do databáze nabídek. Přímo v programu Profi plast je tlačítko, které přes VPN uloží danou nabídku do databáze.

Vznik zakázky

Než je založena zakázka na některé z poboček, tak obvykle vzniká cenová nabídka, podle které se zákazník rozhoduje, zda za daných podmínek takovouto zakázku podepíše, či nikoli.

Pokud se zákazník rozhodne podepsat zakázku na základě cenové nabídky, uživatel změní v programu Profi plast nastavení z nabídky na zakázku. Pak již nelze takovýto soubor v programu Profi plast změnit.

Automaticky se vygeneruje smlouva o dílo (šablona smlouvy je uložena ve formátu rtf a probíhá textovým nahrazením podle údajů o zakázce), která bude součástí souboru se zakázkou.

Takto vzniklá zakázka je poslána do sídla společnosti emailem k dalšímu zpracování.

Předzpracování zakázky v sídle společnosti

Zakázka se nachází, jak již bylo výše napsáno, na firemním emailu, dokud nedojde k zaplacení zálohy (je propacena zálohová faktura). Jakmile je záloha zaplacena, zakázku je nutno ručně zaevidovat do starého informačního systému a soubor se zakázkou v emailu uložit do souborového systému na firemním serveru na místo, které je určené k zápisu nových zakázek do výrobní databáze.

Uložení zakázky

Uložení zakázky do výrobní databáze má na starosti další pracoviště, jehož úkolem je vzít soubor ze souborového systému na serveru, otevřít jej v programu Profi plast a uložit jej do výrobní databáze. Soubor je pak pracovníky přesunut do jiné složky, určené na zpětné dohledávání těchto souborů (s těmito soubory se již obvykle dále nepracuje).

Objednání materiálu

Po uložení zakázky do výrobní databáze je možno vytvořit hromadnou objednávku materiálu v programu Evidence. Hromadná objednávka je ve formátu XLS a je poslána pracovníky firmy do správné firmy (objednávek se generuje více, podle druhu materiálu). Pracovníci zaevidují do starého informačního systému datum objednání materiálu u jednotlivých zakázek.

Výrobní proces

Každá směna má svého mistra, který plánuje dávku zakázek, které se mají vyrábět. Plán se tvoří dopředu, aby objednané zboží stihlo dorazit od dodavatelů. Vytvořením plánu vznikají soubory pro výrobní stroje, které musí mistr nakopírovat na sdílený disk, ze kterého tyto soubory stroje použijí.

Jednotlivá pracoviště skenují čárové kódy na produktech pomocí čteček čárových kódů do programu Evidence a tím oznamují, jaká fáze zakázky je již hotova (projevuje se především ve výrobní databázi). Pokud jsou všechny položky zakázky vyrobeny, je zakázka připravena k montáži.

Montáž zakázky

Po montáži zakázky pracovníci na skladu zaevidují dílčí práce na montáži do výrobní databáze a doplní doplatek do pokladní knihy. Obě činnosti provedou v rámci informačního

systému.

5.1.2 Zpracování mezd

Mzdové účetnictví se eviduje v programu PAMICA, ve kterém jsou obsaženy všechny potřebné informace o zaměstnancích.

Mzdové účetní potřebují znát pro výpočet mezd informace o docházce zaměstnanců a jejich případných odměnách. Informace o docházce se nacházejí ve 2 docházkových systémech (informační systém a IKOS) a informace o odměnách se nacházejí v informačním systému.

Sběr a použití těchto informací provádí mzdové účetní ručně, zde vzniká prostor pro zautomatizování, popř. alespoň částečné zautomatizování, celého procesu.

5.2 Reporty z informačního systému

Důležitým faktem je, že reporty nejsou implementovány jako webové služby, nicméně je tu možnost úpravy stávajícího řešení po dohodě s IT oddělením.

Reporty z informačního systému lze rozdělit na dva logické celky:

- Obchodní reporty,
- Výrobní reporty.

5.2.1 Obchodní reporty

Obchodní reporty umožňují zobrazovat obchodní informace, které jsou filtrovány podle jednotlivých poboček, obchodních zástupců a firem. Reporty zobrazují informace za určité zvolené období (určitý den, měsíc, kvartál, celý rok).

Tyto reporty nejsou určeny pro samotné pobočky, ale pro obchodní zástupce a management společnosti. Obchodní zástupce může získat pouze reporty stavů svých poboček, resp. jednotliví uživatelé si mohou filtrovat pouze ta data, na která mají nárok.

Obvykle hlavní informace, které tyto reporty přináší, je počet zakázek a jejich cena. Jednotlivé informační celky, které tyto reporty zobrazují:

- Zakázky,
- Pohledávky,
- Zálohy,
- Nabídky,
- Montáže,
- Odměny.

5.2.2 Výrobní reporty

Výrobní reporty jsou určeny především pro divizi zabývající se výrobou, tudíž zde není předmětem zájmu filtrování podle poboček, obchodních zástupců a firem, ale spíše rozdělení na jednotlivé směny, které co vyráběly.

Obykle je pro tyto reporty nejpodstatnější druh výrobku a počet vyrobených kusů.

Kapitola 6

Strategie rozvoje informačního systému

Z předchozích kapitol je patrné, jak celkový systém softwarových řešení, tak i samotný informační systém jsou poměrně složité a nákladné záležitosti.

Z důvodů finanční krize se společnost snaží šetřit finanční prostředky ve všech odděleních, včetně IT oddělení. Snahou IT oddělení je nezávádět žádná nová softwarová řešení, ale spíše aktualizovat či upravovat stávající řešení popř. hledat open sourcové alternativy.

Hlavní požadavky společnosti na informační systém jsou:

- Automatizace procesů,
- Automatické plánování montáží,
- Zavedení vlastních informačních systémů pro dceřinné společnosti.

6.1 Automatizace procesů

Velké množství procesů ve společnosti je řešeno různými těžkopádnými a neefektivními řešeními. Optimálním cílem je zredukovat ruční činnosti uživatelů a tyto činnosti zautomatizovat.

Aktuálně se pracuje na zápisu záloh z programu Profi plast do nové databáze záloh. Jedná se o první krok k zautomatizování celého procesu zapisování záloh. Tento krok bude vykonáván opět ručně, ale odpadá nutnost přepisovat informace o zakázce ze souboru, který obsahuje informace o zakázce, ve formátu pdf do starého informačního systému. Tím se zásadně zredukuje chybovost a zvýší efektivita tohoto procesu.

Nutná automatizace by měla nastat ve výrobních procesech. Jak již bylo dříve zmíněno, některé dílčí činnosti vykonávají ručně pracovníci, čímž hrozí nebezpečí chybovosti.

6.2 Automatické plánování montáží

Plánování montáží aktuálně plánují ručně zaměstnanci ve skladech. Společnost by chtěla, aby informační systém dokázal naplánovat zakázky, které jsou již připraveny k montáži tak, aby byly naplánovány pokud možno co nejefektivněji a zároveň musí splňovat podmínky pro montáž.

- Podmínky pro montáž,
- Efektivita plánování.

6.2.1 Podmínky pro montáž

Podmínky, za kterých je možno montáž uskutečnit jsou poměrně složité. Je nutno brát v potaz, aby se co nejvíce zakázek vešlo do auta s co nejmenšími přejezdy mezi jednotlivými montážemi a zároveň aby zakázka, která se vejde do jednoho auta, nebyla rozdělena do více aut.

Pokud se zakázka do jednoho auta nevejde, je nutné, aby to stejné montážní auto provedlo domontování následující pracovní den. V případě větší zakázky (přesahující určitý předem dohodnutý limit), je nutné použít co nejvíce montážních aut a provést montáž co v nejkratším čase.

Automatické předplánování by měl systém provést týden předem. Uživatelé budou mít možnost měnit montáže do dvou pracovních dnů, pak již nelze naplánované montáže změnit, pouze zrušit celou montáž.

Změna uživateli informačního systému by měla být optimalizována tak, aby systém nenabízel neefektivní varianty termínu montáže.

6.2.2 Efektivita plánování

Z předchozích podmínek je patrné, že výpočetní náročnost tohoto procesu je značná a na první pohled ji lze přirovnat ke známému problému obchodního cestujícího. Více informací o problému obchodního cestujícího lze nalézt v [13].

Problém lze optimalizovat vhodným seřazením zakázek podle velikosti, nicméně výpočetní náročnost by byla pořád značná, což je problém hlavně při výpočtech v reálném čase (u přeplánovávání uživateli je nutné spočítat všechny podmínky, kterými se má montáž dané zakázky řídit).

6.3 Zavedení vlastních informačních systémů pro dceřinné společnosti

Zavedení vlastních informačních systémů pro dceřinné společnosti je nákladný a časově náročný projekt. Jedná se spíše o plán do budoucna.

Celkový systém by pak měl vypadat tak, že každá společnost bude mít vlastní informační systém a bude existovat jeden centrální informační systém, se kterým se budou všechny ostatní synchronizovat. Centrální infomační systém by měl sloužit pro jednodušší vedení celé obchodní skupiny.

Výše popsaná strategie pro zavedení informačních systémů pro dceřinné společnosti byla platná v období tvorby semestrálního projektu, v průběhu tvorby diplomové práce došlo ke změně této strategie ze strany vedení společnosti.

Implementace informačních systémů pro jednotlivé společnosti v době vytváření této diplomové práce již započala a prioritou společnosti je zavedení integrace všech informačních systému do jednoho původního. Není zamýšlena plná integrace všech komponent jednotlivých informačních systémů (např.: Zakázky, které dceřinná společnost objednává u hlavní

společnosti, musí být evidovány jak v informačním systému dceřinné společnosti, tak v informačním systému hlavní společnosti, kdežto zakázky, které objednává u jiné společnosti, nesmí být zaevidovány v informačním systému hlavní společnosti.).

Kapitola 7

Návrh integrace softwarových prostředků do jednoho funkčního celku

Integrace softwarových prostředků se soustředí na dvě základní myšlenky a to: integrace aplikací uvnitř firmy (EAI, enterprise application integration) a integrace mezi různými subjekty (B2B, business to business). [15]

V kapitole budou postupně vysvětleny principy integrace, jejich druhy a modely a samotný návrh integrace.

- Druhy integrace,
- Modely integrace,
- Návrh integrace.

7.1 Druhy integrace

- Datová integrace,
- ETL integrace,
- Aplikační integrace.

7.1.1 Datová integrace

Datová integrace se vyznačuje především spolehlivostí a vysokým výkonem. Výhodou takovéto integrace je fakt, že není zapotřebí zasahovat do programů, což se v praxi může hodit, zejména pokud není možné do takovýchto programů zasahovat (např. u produktů třetích strany).

Takovéto řešení je schopno rychle rozšířit změnu v datech do míst, které se o změně mají dozvědět. S výhodou lze tohoto efektu využít např. v situacích, kdy je jedno softwarové řešení používáno na více místech a má existovat jeden centrální systém, který získává informace od všech ostatních a naopak změna v hlavním systému ovlivňuje podřízené systémy. [5]

7.1.2 ETL integrace

ETL integrace (popř. ETL transformace) obvykle řeší předávání dat mezi různými datovými typy (specializované databáze, různé datové formáty, či rohraní). Typickou úlohou takovéto transformace je přenášení velkého množství informací z relačních databází do datového skladu. [5]

ETL transformace je zkratkou z anglických slov extraction, transformation a load, z čehož plyne, že se nejedná o pouhé replikování dat jako je tomu u datové integrace, ale je nutné data jak získat, tak je obvykle i přetransformovat před následným uložením.

Existuje celá řada ETL nástrojů, které se liší jak cenou, tak způsobem použití. Základní rozdělení nástrojů je: ETL nástroje první generace, které na základě navrženého transformačního předpisu vygenerují kód, který se kompiluje a spouští většinou na zdrojové platformě, a ETL nástroje druhé generace, jejichž jádrem je transformační engine realizující ETL procesy podle objektově definovaného předpisu uloženého v katalogu metadat navrženého vývojářem. [21]

7.1.3 Aplikační integrace

Aktuálním trendem v integraci softwarových řešení je nejen integrace na úrovni dat jako takových, ale i na úrovni samotných aplikací. Problémem je, že aplikace musí být vhodně navržena, aby bylo možno takovouto integraci realizovat.

Aby takováto integrace fungovala, je nutné stanovit jakým způsobem budou spolu aplikace komunikovat — např. v případě, že jedna čeká na oznámení o dokončení druhé a pod. Návrh vzájemné komunikace aplikací je pro tuto integraci kritický. Při chybném návrhu přestane systém fungovat jako celek.

Vhodným řešením tvorby či výběru aplikací se zdá použití architektury založené na službách (SOA) a její implementace pomocí webových služeb (WS). Takováto architektura má jasně definován způsob komunikace a je tudíž vhodná pro integraci, což ovšem neznamená, že odpadná problém s chybným návrhem.

7.2 Modely integrace

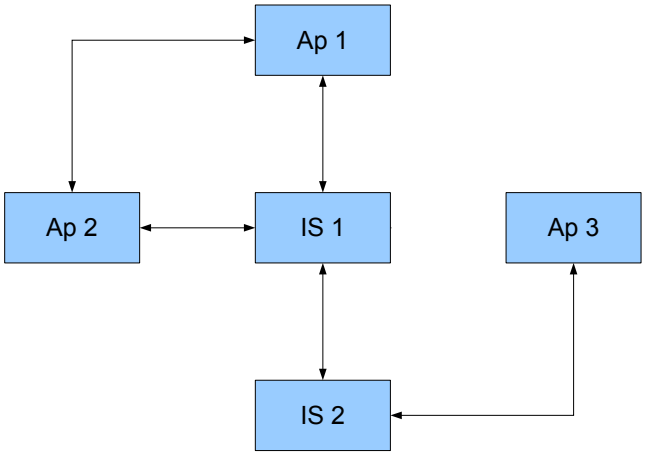
- Klasické pojetí systémové integrace,
- Nové pojetí systémové integrace — Softwarová integrace.

7.2.1 Klasické pojetí systémové integrace

Po analýze softwarového vybavení společnosti obvykle vzniká spojení aplikací N:M podle jejich vzájemných závislostí. Integrační prostředky řeší dílčí problémy s integrací mezi jednotlivými aplikacemi a ne celkem jako takovým.

Tento princip je výhodný jednodušším návrhem při realizaci, resp. je možé postupně řešit jednotlivé úlohy aplikačních závislostí.

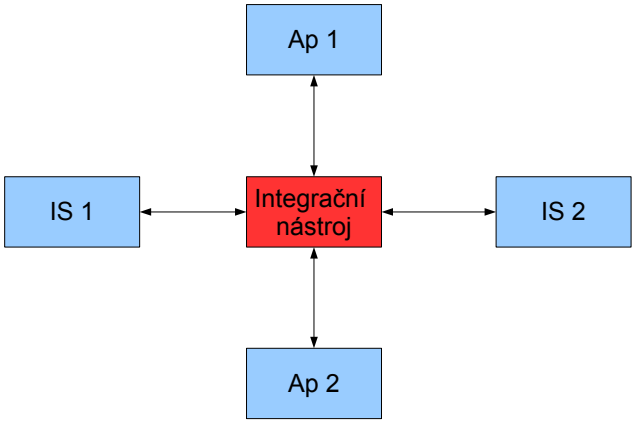
Naopak nevýhodou tohoto principu je skutečnost, že při jakékoli změně v systému je nutné opětovně realizovat integraci aplikací, kterých se změna týká primárně a zároveň zkontrolovat integraci aplikací, kterých se týká sekundárně (aplikace mající závislost se změněnou). Tedy jinými slovy je nutné zkontrolovat celou řadu integračních prostředků, což může vést v praxi k chybám.



Obrázek 7.1: Klasický model systémové integrace N:M.

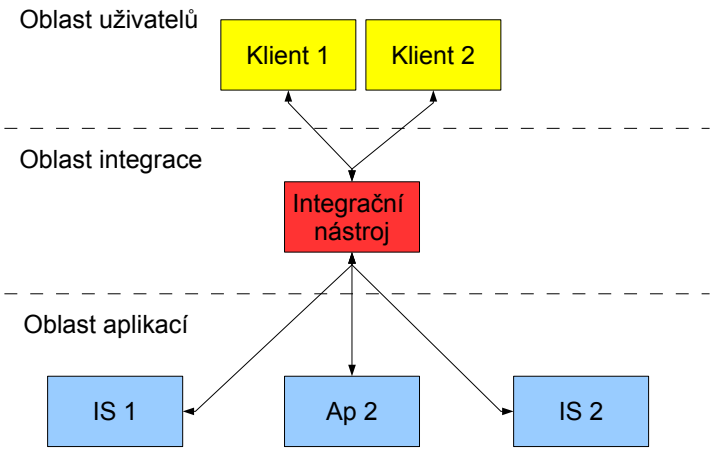
7.2.2 Nové pojetí systémové integrace — Softwarová integrace

Po analýze softwarového vybavení společnosti vzniká spojení aplikací s integračním nástrojem 1:N a integrační nástroj se stává hlavní částí celého systému.



Obrázek 7.2: Softwarová integrace — model 1:N.

Z pohledu uživatelů je tento typ integrace chápán tak, že se jednotlivé klientské aplikace, které uživatelé používají, připojují k integračnímu nástroji, který zprostředkovává komunikaci mezi ostatními částmi systému.



Obrázek 7.3: Nové pojetí integrace z pohledu uživatelů.

7.3 Návrh integrace

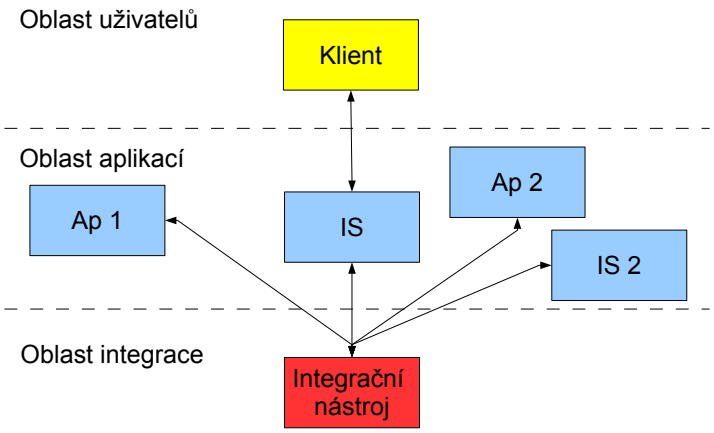
Tento návrh integrace se bude zabývat pouze EAI integrací z důvodu, že společnost B2B integraci aktuálně nepotřebuje, resp. EAI integrace má pro společnost v současné době vyšší význam.

- Konceptuální návrh,
- Návrh jednotlivých integrací,
- Celkový pohled na návrh jednotlivých integrací.

7.3.1 Konceptuální návrh

Návrh integrace vychází z nového pojetí integrace, popsaném v 7.2.2, s mírnou modifikací tohoto konceptu, vzhledem k tomu, že stávající informační systém je poměrně rozsáhlý a bylo by komplikované vytvářet znovu klientské prostředí.

Jinými slovy bude integrace provedena do databáze informačního systému a v informačním systému se jen poupraví datové zdroje, případně přidají moduly na obsluhu těchto nových dat.



Obrázek 7.4: Konceptuální návrh integrace z pohledu uživatelů.

7.3.2 Návrh jednotlivých integrací

Následuje popis návrhu jednotlivých integrací, které budou součástí celkového integračního nástroje.

- Integrace výrobních dat,
- Integrace systémů s personálními daty zaměstnanců,
- Integrace docházkových systémů,
- Integrace starého informačního systému,
- Integrace procesu zápisu záloh,
- Integrace knihy jízd z O2 Carcontrol,
- Integrace hlavního informačního systému s informačními systémy dceřinných společností.

Integrace výrobních dat

Na integraci výrobních dat z výrobní databáze do databáze informačního systému je vhodné použít ETL transformaci. Data je potřeba před samotnou integrací transformovat z důvodu nekonzistence dat, proto nelze použít pouze datovou transformaci.

Není potřeba používat všechna data z výrobní databáze, resp. databáze obsahuje mnoho dat, které se reálně nepoužívají.

Samotná integrace by neměl být problém, obě databáze jsou přístupné v rámci firemní sítě. Navíc jsou obě databáze Firebird (které jsou popsány v 2.3.1), takže je možno použít stejnou strukturu jako má výrobní databáze (kromě chybných návrhů, které bude nutné pozměnit).

Integrace systémů s personálními daty zaměstnanců

Data o zaměstnancích jsou v různé podobě uložena v informačním systému, ale nejsou zde všichni zaměstnanci a stav databáze nemusí být úplně aktuální. Aktuální data o zaměstnancích jsou uloženy v programu PAMICA, ze kterého jsou vypočítávány mzdy. Z toho plyne, že tato data musí být vždy aktuální a jsou vhodná pro použití jako bázová data.

PAMICA umožňuje provádět exporty dat, takže s vhodným použitím ETL transformace je možno exportovat data do databáze informačního systému. Tady může nastat problém s poznáváním zaměstnanců, kteří již jsou nějakým způsobem uložení v informačním systému.

Variantou je nechat rozpoznat zaměstnance automaticky podle jména a příjmení a zbytek co se nerozpozná, tak rozdělit ručně.

Integrace docházkových systémů

Informační systém obsahuje docházkový systém pro externí pracovníky. Na integraci tohoto systému s docházkovým systémem v sídle firmy je nutné použít dvě metody integrace:

- *Aplikační integrace*, která bude komunikovat se službami, které běží na docházkovém serveru a umožňují tento docházkový systém ovládat.
- *ETL transformaci*, která bude načítat exporty z jednotlivých terminálů. Tyto exporty jsou ve formátu CSV.

Na ucelení docházkových systémů je nutné použít předchozí integraci systémů s personálními daty, aby bylo možno jasně identifikovat pracovníka firmy.

Integrace starého informačního systému

Předpokladem této integrace je dokončená integrace výrobních dat. S využitím webových služeb starého informačního systému je možné použít aplikační integraci s aktuálním informačním systémem.

Hlavním smyslem je synchronizovat ta data z databáze informačního systému, která se do starého informačního systému zadávají ručně a zároveň již jsou obsažena v databázi informačního systému. Zmenší se tím chybovost dat, obsažených ve starém informačním systému a zároveň pracovníkům ušetří práci navíc.

Integrace procesu zápisu záloh

Předpokladem této integrace je dokončení zápisu záloh, popsaném v 6.1, do nové databáze. Samotná integrace bude založena na dvou metodách integrace:

- *Aplikační integrace*, která bude komunikovat s webovými službami, které umožňují nahrávat změny dat do starého informačního systému.
- *ETL transformaci*, která bude data uložená v nové vzniklé databázi transformovat a ukládat do databáze informačního systému.

Po integraci do informačního systému je nutné doplnit informaci o úhradě k zakázce, které se tato záloha týká, a zároveň přepočet zaplacené částky zakázky. Tato integrace ovlivňuje také moduly pohledávek a odměn, kvůli možnosti zaplacení zakázky v plné výši.

Integrace knihy jízd z O2 Carcontrol

Společnost požaduje, aby informace o trasách firemních aut bylo možno sledovat přímo v informačním systému společnosti. Informace o těchto trasách lze získat z exportu knihy jízd v aplikaci O2 Carcontrol. Data je nutné pravidelně synchronizovat z O2 serveru do databáze, protože přímé použití exportu v rámci firemního informačního systému není možné kvůli časovému omezení mezi jednotlivými dotazy ze strany O2 serveru.

Integrace hlavního informačního systému s informačními systémy dceřinných společností

Hlavním důvodem této integrace jsou data zakázek, které mají nejvyšší prioritu. Společnost zajišťuje výrobu zakázek, ale dceřinné společnosti si objednávají zakázky od více dodavatelů, což byl důvod vzniku informačních systémů u dceřinných společností.

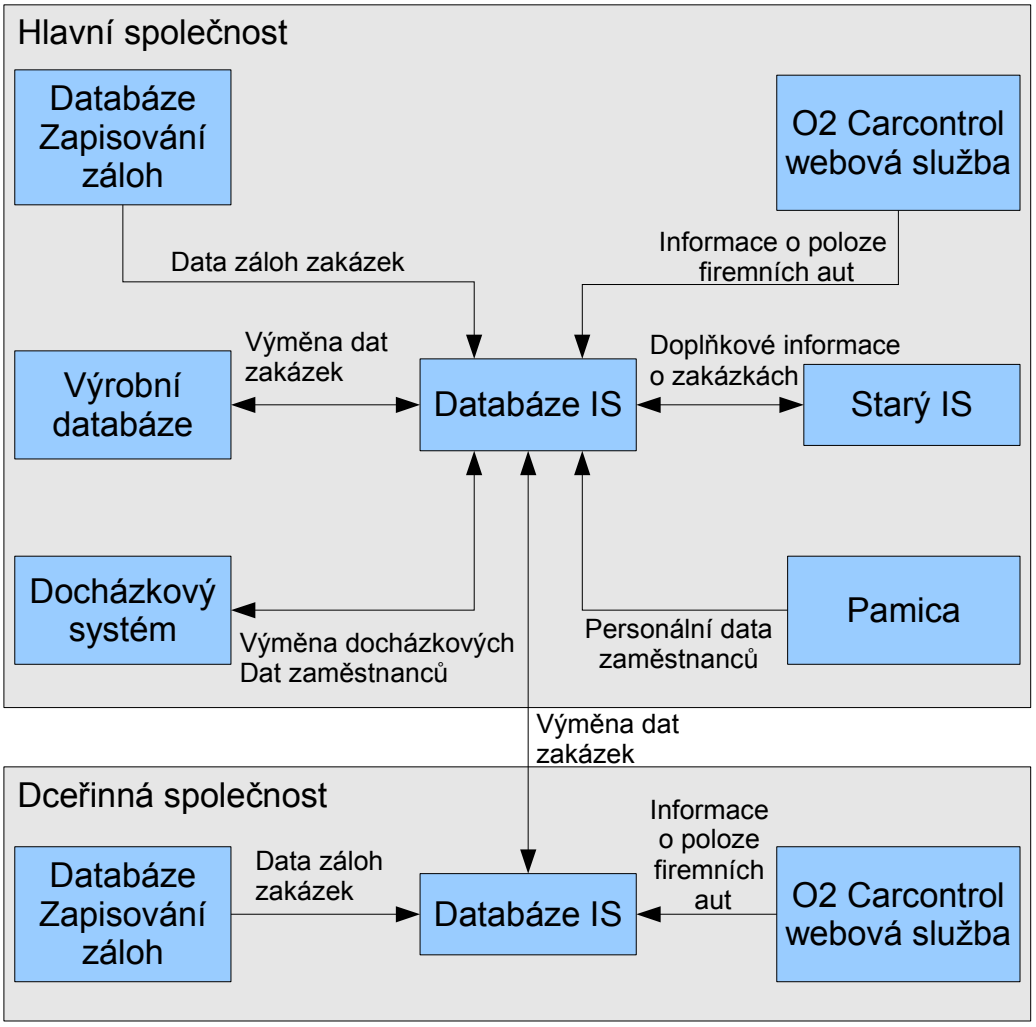
Integrace dat bude probíhat obousměrně. Z dceřinné společnosti do hlavní se odešle informace o nové zakázce s potřebnými daty a naopak z hlavní společnosti do dceřiné společnosti se budou posílat informace o výrobě zakázky, resp. jestli již je objednáno zboží, je naplánovaná do výroby, je vyrobená, apod.

Implementace této integrace bude pravděpodobně provedena datovou integrací přímo mezi jednotlivými databázemi společností pomocí virtuálni privátní sítě (VPN), přes kterou jsou všechny servery navzájem propojeny.

7.3.3 Celkový pohled na návrh jednotlivých integrací

Jednotlivé integrace budou prováděny přes integrační nástroj, který není na následujícím diagramu uveden kvůli zjednodušení diagramu a lepší přehlednosti.

Diagram naznačuje propojení hlavní a dceřinné společnosti a jednotlivých softwarových částí v rámci daných společností. Integrace softwarových prostředků v rámci dceřinných společností (u všech dceřinných společností probíhá integrace stejným způsobem, proto je v diagramu uvedena pouze jedinná dceřinná společnost) je přímou podmnožinou integrací v hlavní společnosti, navíc je jen integrace dat zakázek s nadřízenou společností.



Obrázek 7.5: Celkový návrh jednotlivých integrací.

Kapitola 8

Implementace integrace softwarových prostředků

Kapitola obsahuje popis existujících integračních nástrojů, implementaci vlastního nástroje a popis jakým způsobem byly jednotlivé integrace, které jsou popsány v kapitole návrhu integrací 7, implementovány.

Výše zmíněný obsah této kapitoly obsahují následující podkapitoly:

- Stávající integrační nástroje,
- Implementace vlastního integračního nástroje,
- Integrace jednotlivých softwarových prostředků.

8.1 Stávající integrační nástroje

Mezi stávajícími integračními nástroji jsem nenašel žádný vhodný pro potřeby společnosti. Problematická byla především cena, či podpora Firebird databáze.

Mezi hlavními produkty ETL na ICT trhu můžeme zařadit Ab Initio (Ab Inition Software), DataStage (Ascental Software), ETL v rámci DB2 (IBM), Informatica (Informatica), SSIS jako součást MS SQL Server (Microsoft) a Oracle Warenhause Bilder (Oracle). [17]

Následuje stručný přehled některých známých integračních nástrojů a jejich následný popis:

- SQL Sever Integration Services,
- Oracle Warehouse Bilder,
- DataStage.

8.1.1 SQL Sever Integration Services

SQL Sever Integration Services (SSIS) je ETL nástrojem, který je součástí MS SQL serveru 2005 a vyšších, od společnosti Microsoft. Nástroj podporuje tzv. drag-and-drop ovládání, které je v současné době velmi rozšířené a oblíbené uživateli. Mezi další výhody SSIS patří např. podpora ve vývojovém nástroji Visual Studio, na které jsou vývojáři na Microsoft platformách zvyklý. [23]

Nástroj podporuje:

- **ERP systémy**: SAP, Oracle, Epicor, Siebel, JD Edwards, Lawson, Dynamics, Sage, Hyperion a další.
- **Databázové systémy**: MS SQL, Oracle, MS Access, Teradata, IBM DB2, PostgreSQL, MySQL a další.
- **Soubory na počítači**: Textové soubory, XML, Excel soubory, CSV a další.
- **Počítačové protokoly**: HTTP, FTP, SMTP a další.

8.1.2 Oracle Warehouse Bilder

Oracle Warehouse Bilder (OWB) od společnosti Oracle je nástrojem, který je součástí databázového systému Oracle a umožňujícím generovat PL/SQL skripty a schémata umožňující vytvářet ETL transformace.

Díky úzkému spojení s databázových serverem, grafickému uživatelskému rozhraní, generování kódu a integrátorů (např. SAP) umožňuje nástroj provádět různé ETL integrace Oracle databázového systému s ostatními systémy, popř. datovými formáty. [7]

Nástroj, který dokáže integrovat data z oblíbených datových zdrojů, obchodních aplikací a databází, mimo jiné podporuje: IBM DB2, Teradata, MS SQL Server, SAP Business Information Warehouse, Microsoft Analysis Services, XML data a nestrukturovaná data. OWB kombinuje úložiště metadat, datovou integraci a nástroj Enterprise data quality, k vytvoření integrace optimalizované pro databázový systém Oracle. [12]

8.1.3 DataStage

ETL nástroj DataStage od společnosti Ascental Software je rozdělen na klientskou a serverovou část, kde serverová část zajiťuje samotné ETL integrace a klientská umožňuje konfiguraci těchto integrací pomocí grafického uživatelského rozhraní. Klientská část není multiplatformní, resp. funguje pouze na operačních systémech MS Windows XP a vyšších, což je zjevnou nevýhodou tohoto nástroje. [18]

8.2 Implementace vlastního integračního nástroje

Integrační nástroj je implementován v jazyce Java na platformě Java SE. Použitím jazyku Java jsem se rozhodl kvůli předchozím zkušenostem, platformové nezávislosti, podpoře databázových ovladačů, snadné práci s více vlákny a snadnou manipulací se soubory ve formátu xml. Předností této platformy je i velmi dobře zpracovaná dokumentace, rozsáhlá komunita, mnoho open source knihoven a velké množství příkladů různých řešení.

Druhou možností bylo použití .NET technologie od společnosti Microsoft, ale to se ukázalo jako nevhodné, z důvodů, že se nejedná platformově nezávislé řešení (servery společnosti fungují na operačním systému Linux, konkrétně CentOS) a navíc neexistuje .NET ovladač pro databázi Firebird, které jsou ve společnosti využívány jako jeden z hlavních datových zdrojů. Stránky výrobce Firebird databáze sice obsahují odkaz na .NET ovladač, nicméně tento odkaz není funkční.

Implementace vlastního integračního nástroje je rozdělena na knihovnu, která je nadstavbou JDBC, obsahující JDBC ovladače od významnějších databázových systémů, umožňující rozpoznání typu databáze podle connection stringu a následnému zavedení příslušného

ovladače, a na knihovnu implementující JDBC ovladač k souborům ve formátu XLS. Zdrojové kódy vlastně vytvořených knihoven a samotného nástroje je možno nalézt na přiloženém CD v adresáři src.

Druhou částí je samotná implementace integračního nástroje, který využívá výše zmíněné knihovny a další open source knihovny, o kterých bude zmínka v následujícím textu.

Popis implementace je rozdělen na následující logické části:

- Použité knihovny,
- Integrační relace,
- Extrakce dat,
- Transformace dat,
- Uložení dat,
- Konfigurovatelnost nástroje,
- Průběh činnosti nástroje,
- Aktivní synchronizace,
- Omezení nástroje.

8.2.1 Použité knihovny

Program využívá externí knihovny především k práci s xml soubory. V rámci implementace vlastní knihovny pro práci s datovými zdroji používá knihovny výrobců jednotlivých databázových úložišť, které zde nejsou uvedeny, protože se ve většině případů jedná o standardní JDBC ovladače k jednotlivým databázím. Popis jednotlivých databází je možno nalézt v 2.3.1.

Následuje popis použitých knihoven pro práci s XML:

- XStream,
- Xerces DOM parser,
- Apache Commons IO,
- Apache POI,
- DBConnector,
- JXlsDriver,
- WebServices.

XStream

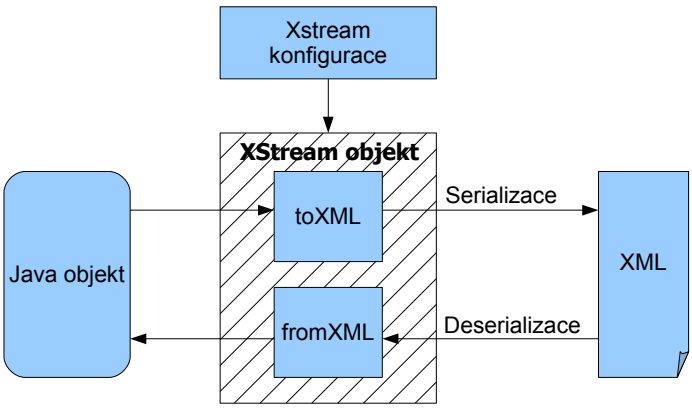
XStream je open source knihovna umožňující serializaci obecných Java objeků do XML a zpět. Její použití je poměrně jednoduché. Knihovna umožňuje nastavit různé parametry určující tvar XML do kterého se mají objekty uložit (případně ze kterého se mají načíst), aby bylo možno použít jakékoli obecné XML s využitím vhodných parametrů.

Výsledné XML je generováno tak, že z jednotlivých objektů jsou přímo ukládána primitiva (čísla a řetězce) jako hodnoty XML uzlu a atributy objektu, které odkazují na jiný objekt, mohou být uloženy přímo v XML uzlu jako struktura objektu, nebo se může na již definovaný objekt jen odkazovat.

Odkaz na objekt může být realizován přidělováním vnitřních identifikátorů každému definovanému objektu, nebo odkazování pomocí XPath. Pro člověka je čitelnější odkazování pomocí vnitřních identifikátorů, XPath cesta nemusí být na první pohled zřejmá kam vlastně vede.

Knihovna počítá i s dědičností a rozhraními. Lze ukládat objekty, které implementují nějakého rozhraní, do názvu uzlu podle daného rozhraní a v atributu uzlu je pak definována konkrétní třída objektu.

Podstatnou výhodou knihovny je fakt, že pokud jsou všechna data uložena v jednom objektu, pak je serializace, popř. deserializace, provedena jediným voláním knihovní metody, což značně ulehčuje např. vytváření konfiguračních souborů a pod. záležitostí s předem danou strukturou. [1]



Obrázek 8.1: Serializace a deserializace java objektu pomocí knihovny XStream.

Xerces DOM parser

Knihovna slouží pro parsování XML dokumentů. Umožňuje procházet strukturu dokumentu přes jednotlivé uzly, načítat hodnoty a atributy těchto uzlů. Není tzv. thread safe (není bezpečná při paralelním užívání), ale tento problém lze v Javě, v rámci implementace vlastní aplikace, poměrně efektivně řešit pomocí zámků (synchronized metody). Lze ji použít i pro serializaci objektů, ale na serializaci je již v projektu použita knihovna XStream, která je poměrně snadná na použití. [2]

Apache Commons IO

Knihovna obsahuje spoustu užitečných metod, které mají co dočinění s nějakou formou vstupu a výstupu. Jedná se především o převody mezi různými vstupními a výstupními proudy, práci se soubory, nebo v některých případech ulehčení práce se síťovými spojeními. [3]

WebServices

WebServices je vlastní poměrně jednoduchou knihovnou zprostředkovávající komunikaci přes HTTP protokol. Zdrojové kódy jsou přiloženy na CD v adresáři src/WebServices.

Apache POI

Knihovna umožňuje přístup k dokumentům od společnosti Microsoft a podporuje datové formáty Office 2007 (XLS, DOC, PPT, XLSX, DOCX a PPTX). Výhodou je, že se neustále vyvíjí a reaguje na změny ve formátech dokumentů. [4]

JXlsDriver

Implementace vlastního JDBC ovladače pro soubory ve formátu xls. Tyto soubory se pak chovají jako samostatné databáze. Přístup k těmto souborům je realizován pomocí knihovny Apache POI. Ovladač není plně implentován, resp. nebylo snahou vytvořit plně hodnotnou implementaci JDBC ovladače. Snahou bylo především realizovat ovladač, který je dostatečně funkční k potřebám nově vzniklého ETL nástroje. Existují nějaké implementace tohoto ovladače, ale ty jsou buď placené, nebo mají nedostatečnou funkčnost. Zdrojové kódy knihovny jsou na CD v adresáři src/JXlsDriver.

DBConnector

Vlastní knihovna, která především sjednocuje databázové ovladače do jedné knihovny a vytváří nádstavbu nad JDBC. Knihovna dokáže rozpoznat z connection stringu o jaký typ databáze jde, zavede příslušný databázový ovladač a vrací abstraktní připojení. Zdrojové kódy knihovny jsou na CD v adresáři src/DBConnector.

8.2.2 Integrační relace

Pojem integrační relace, v kontextu nově implementovaného násroje, znamená jeden integrační úkon, resp. definuje jednu ETL transformaci. Hlavními prvky této relace jsou definice zdrojového datového zdroje pro extrakci, transformační pravidla a cílový datový zdroj.

Všechna výše popsaná nastavení integrační relace lze nastavit v rámci konfigurace programu v konfiguračním souboru. Další informace o konfigurovatelnosti integračního nástroje se nachází v 8.2.8.

8.2.3 Extrakce dat

Pro extrakci dat lze použít libovolný datový zdroj, který aplikace podporuje. Aktuálně mezi takovéto zdroje patří především datové zdroje, k nimž je možno připojit se pomocí JDBC, tedy různá databázová řešení.

Aplikace ovšem není omezena pouze na JDBC datové zdroje. Navržené rozhraní není nijak svázáno přímo s JDBC, takže je možné vytvořit jakýkoliv obecný datový zdroj, pouze třída řešící extrankci dat musí implementovat toto rozhraní.

Aktuálně je konfigurace pro definici zdroje pro extrakci dat navržená spíše pro databázový datový zdroj, než pro obecný datový zdroj. Neznamená to, že není možné vytvořit vlastní konfigurační třídu, která bude implementovat rozhraní nastavení zdroje dat.

8.2.4 Transformace dat

Rozhraní transformace, v rámci implementace, je navrženo pokud možno co nejobecněji. Nejedná se o pouhý převod hodnoty ze zdroje do cíle s nějakou transformací, ale je zde možnost vytvořit transformaci, kde z jednoho řádku dat na vstupu (uvažujeme-li na vstupu a výstupu databázový zdroj) zniká více řádků na výstupu, nebo i žádný. Takto naprogramovaná transformace lze pak v praxi využít na různé druhy úloh.

Příkladem transformace, ve které z jednoho řádku vstupních dat vzniká obecně N řádků na výstupu může být např.: K transformaci se využívá databázová procedura, které na vstupní data reaguje tak, že může mít obecně 0 až N řádků na výstupu. Podobných příkladů lze jistě nalézt více.

Ke každé výstupní hodnotě lze navíc přidat tzv. konvertor, který umožňuje jednoduchou konverzi jedné hodnoty, pokud je transformace reálná, na jinou. Příkladem takovéto konverze může být např. ukládání datumu v různých formátech a následné převody mezi nimy.

Aktuálně je transformace navržena jako obecné rozhraní, předpokládající, že chybějící transformace bude doprogramována. Do budoucna jako další rozšíření by bylo dobré vytvořit co nejobecněji konfigurovatelnou transformaci, nastavitelnou z konfiguračního XML, umožňující deklarativně vytvořit jakoukoli obecnou tranformaci.

Toto neznamená, že by jedinou aktuálně implementovanou transformaci (SOAP) nebylo možno konfigurovat. Konfigurace není obecná, protože se vztahuje přímo k dané transformaci. Fakt, že konfigurace transformace není obecná, není chybou. Pouze se nejedná o obecné řešení, nicméně nalezení obecného řešení tohoto problému přesahuje rámec této diplomové práce a navíc není potřebné pro splnění jejího cíle.

8.2.5 Uložení dat

Uložení dat, podobně jako extrakce, přepokládá databázový zdroj, ke kterému je možné se připojit pomocí JDBC, což stejně jako u extrakce dat není podmínkou viz. 8.2.3. Datový zdroj (nebo též v tomto případě cíl) musí podporovat transakční zpracování, což je jedinou významnější změnou proti datovému zdroji popsanému v podkapitole Extrankce dat.

Nic nebrání ve vytvoření jiného typu datového zdroje pro uložení dat, který transakční zpracování nepodporuje, pomocí implementace vhodného rozhraní. Nicméně ignorování transakcí může vést na nedefinované chování.

Takovéto nedefinované chování může nastat např. v situaci, kdy ze zdroje dat získáváme nově vzniklé záznamy a do cílového zdroje dat má být zapsána tato informace. Při použití transakčního zpracování by se při vzniku jakékoli chyby použilo odstranění změn (rollback) a veškeré změny by nebyly aplikovány, ale při nevhodné implementaci transakčního zpracování (resp. žádné) se může stát, že nově vzniklý záznam se bude vyskytovat v cílovém datovém zdroji několikrát, místo právě jedenkrát, jak je požadováno.

Datový zdroj pro integraci je zároveň zdrojem dat (časových razítek poslední úspěšné synchronizace) a zároveň slouží k uložení nového časového razítka. V tomto datovém zdroji

je transakční zpracování podmínkou zcela nutnou, protože v případě chyby by se mohla uložit informace o úspěchu a část dat by se pak nemusela vůbec sesynchronizovat viz podkapitola omezení.

8.2.6 Průběh činnosti nástroje

Program po spuštění načítá konfiguraci z konfiguračního souboru do konfiguračních tříd. Struktura konfiguračních tříd je taková, že se do hlavní kořenové třídy ukládají seznamy tříd s konfigurací datových zdrojů a jednotlivých integračních relací. Program postupně projde seznam konfigurací integračních relací, podle tohoto nastavení vytvoří objekty jednotlivých integračních relací.

Celý program funguje paralelně, resp. každý objekt integrační relace provádí svou činnost ve vlastním vlákně. Tato vlákna není potřeba navzájem zamykat, vlákna jen čtou ze sdílené paměti a nikdy nezapisují, navíc si každé vlákno vytváří vlastní spojení s databázemi, kvůli zjednodušení alogoritmu a aby odpadly problémy s paralelismem.

Vlákno provede definovanou akci a uspí se na určitý časový interval, popř. pokud má relace nastaven čas synchronizace, tak nastaví uspání na zbytek do tohoto času (čas synchronizace může být např. 9:30). Spojení s databázemi vytváří vlákna jen když jej potřebují během vykonávání akce, ve stavu čekání není spojení aktivní.

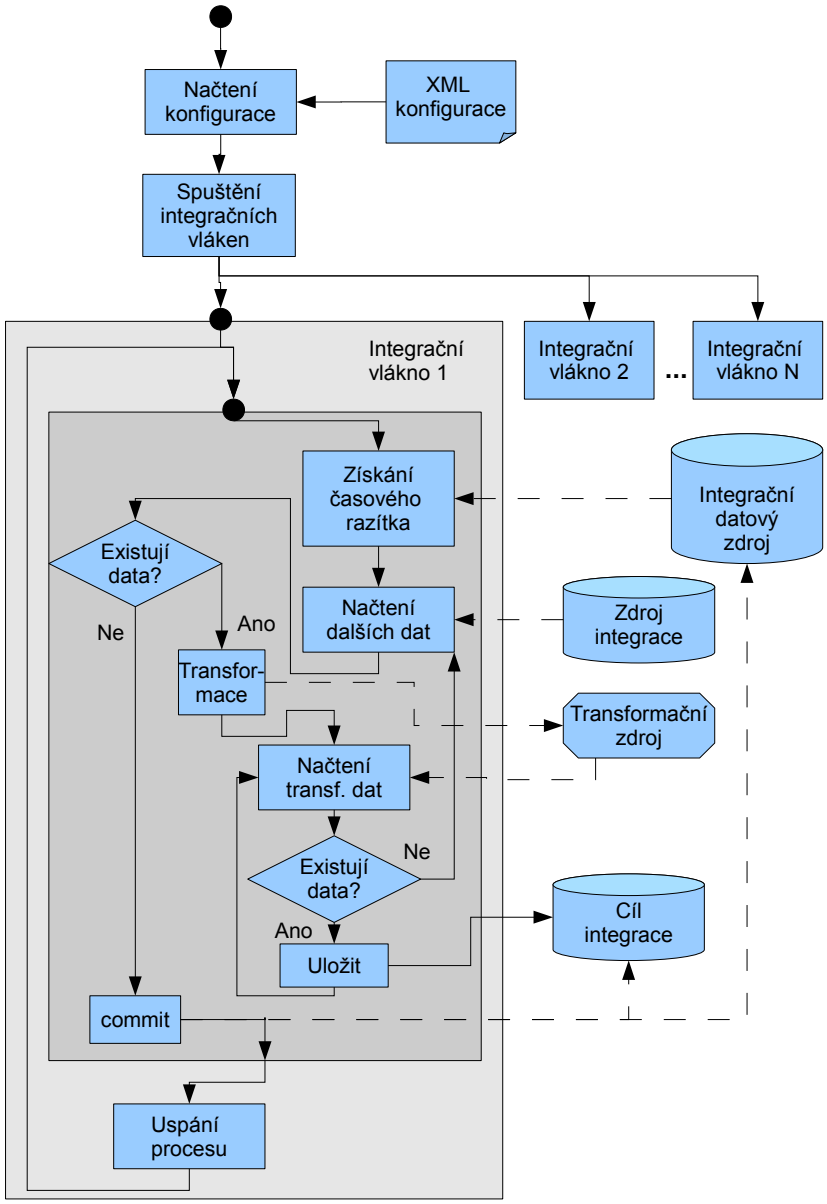
Při provádění akce, pokud je nastaven název časového razítka, se načte časové razítko poslední synchronizace dané relace a toto časové razítko se používá pro filtraci záznamů, které se načítají z datového zdroje. Pokud název časového razítka nastaven není, pak se časové razítko poslední synchronizace nenačítá a záznamy nejsou nijak filtrovány (není podle čeho) z datového zdroje.

Veškerá aktivita nad datovými zdroji (ať už synchronizačními, zdrojovými, či cílovými) probíhá, pokud to daný datový zdroj podporuje, v transakcích. Při jakékoli chybě při provádění akce se na všechny otevřené transakce zavolá odstranění změn (rollback), potvrzení změn (commit) je použito jen pokud se celá akce vykoná bez jakékoli chyby (toto platí i v případě, že se nepřenáší žádná změna ze zdrojového datového zdroje do cílového).

Záznamy načtené ze zdrojového datového zdroje se postupně načítají, použije se na ně transformace (pokud je definovaná) a ukládají se do cílového zdroje. Výsledkem transformace záznamu nemusí být jen jeden záznam, ale i více záznamů či žádný záznam, což vede k lepší použitelnosti a variabilitě při konfiguraci daného integračního nástroje.

Po úspěšném provedení akce se aktualizuje hodnota časového razítka poslení provedené akce dané relace (v rámci transakce).

Po provedení odstranění změn (rollback), který nastává po chybě (jak již bylo zmíněno výše), se nástroj pokusí informace o neúspěchu uložit do logovací tabulky v integračním datovém zdroji, ve kterém se nachází tabulka s časovými razítky. Uložení chyby do logu se nezdaří např. v případě, že chyba nastane kvůli ztrátě konektivity s databází, ve které se logovací tabulka nachází. Mezi informace, které nástroj do logovací tabulky ukládá, patří: název relace, důvod chyby a jakým způsobem byla chyba vyvolána (např. sql dotaz pro databázi). Tyto informace pomáhají odladit celý systém, popř. najít neočekávaná data, která způsobila chybu integrace. Dokud nedojde k odstranění takovéto chyby, nebude daná integrační relace, ve které chyba nastala, fungovat a bude tzv. zaseknutá. Zaseknutá integrační relace znamená, že se bude opakovaně, s určitým nakonfigurovaným zpožděním, opětovně pokoušet o integraci stejných dat, která opět skončí neúspěchem. Problém zaseknuté relace je blíže specifikován v podkapitole zabývající se omezeními programu v 8.2.9.



Obrázek 8.2: Diagram průběhu činnosti ETL nástroje.

8.2.7 Aktivní synchronizace

Nástroj podporuje aktivní synchronizaci, resp. vykonání synchronizační relace aktivovanou událostí. Rozhraní je definováno obecně, ale samotná implementace této činnosti je vytvořena pouze pro databázový systém Firebird, který posílání událostí na serveru a poslech u klienta (pro Javu v rámci knihovny JayBird) podporuje.

Mírný problém těchto událostí v databázovém systému Firebird je ten, že událost je vyvolána z dosud neuzavřené transakce, takže není možné okamžitě reagovat na tuto událost, protože dokud není transakce dokončena, tak se změny v databázi neprojeví, proto je reakce na tuto událost zpožděna o několik sekund. Není to moc dokonalé řešení, protože u transakce, která trvá delší dobu, nemusí stačit ani zmíněných několik sekund, ale mělo by to být dostatečné pro většinu jednoduchých změn v rámci databáze.

Problém paralelismu, resp. aby nenastala situace, že synchronizační relace běží a měla by být znovu spuštěna ve stejném čase na základě příchozí události, je vyřešen tak, že reakce na událost nevolá přímo vykonání synchronizační relace, ale pouze se pokusí probudit vlákno, které se o tuto relaci stará. Pokud vlákno je ve stavu spánku, tak se vlánko vzbudí a relace proběhne, a pokud není ve stavu spánku, tak se nestane nic, protože již relaci vykonává. Eventuelně by mohlo dojít k situaci, že relace, která probíhá řeší synchronizaci dat, která je ještě před příchodem události, nicméně v tomto případě hrozí jen v nejhorším případě pozdržení synchronizace o 1 synchronizační interval, kdežto případné problémy s paralelismem by mohli být poměrně fatální, způsobit dualitu záznamů a pod.

8.2.8 Konfigurovatelnost nástroje

Konfigurační souboru musí mít aktuálně název "config.xml"a být uložen v místě odkud se program spouští. Z názvu konfiguračního souboru vyplývá, že je konfigurace uložená ve formátu xml. Program obsahuje tzv. konfigurační třídy, obsahující informace o nastavení jednotlivých částí (datové zdroje, integrační relace, transformace, atd.). Pro načtení konfigurace z konfiguračního souboru do konfiguračních tříd se s výhodou používá knihovna XStream 8.2.1.

Konfigurace je rozdělena na datové zdroje a nastavení jednotlivých integračních relací. Datovými zdroji je myšlen datový zdroj, který je podporován integračním nástrojem k získávání dat. V konfiguraci musí být uvedeno jaký datový zdroj je tzv. integrační, resp. obsahuje tabulku s časovými razítky posledních integrací jednotlivých relací a tabulku s logy chyb.

V rámci zápisu v xml konfiguračního souboru podporuje knihovna XStream vytváření nových objektů, či odkazů na již definované objekty v xml pomocí XPath odkazů na ně. Tento zápis není úplně ideální, resp. není moc čitelný pro člověka, lepší možností by bylo využít vlastnosti knihovny XStream: odkazu na objekty pomocí určení identifikace objektu a reference na něj. Bohužel již nezbyl čas na úpravu stávajícího řešení, tato úprava by byla vhodná jako budoucí rozšíření.

Odevzdané konfigurační soubory k dceřinným společnostem a hlavní společnosti obsahují xml komentáře, které popisují význam jednotlivých záležitostí v konfiguračním souboru a objasňují význam činnosti programu pokud některá vlastnost není zadána.

8.2.9 Omezení nástroje

Nástroj obsahuje určitá omezení, nebylo účelem diplomové práce vytvořit obecný nástroj, který řeší jakoukoli integraci, ale nástroj s jehož pomocí bude integrace proveditelná.

Omezení nástroje jsou následující:

- Integrační datový zdroj,
- Zaseknutí integrační relace,
- Transformace.

Integrační datový zdroj

Integrační datový zdroj je ve stávající fázi omezen na databázový datový zdroj. Datový zdroj jiného typu buď nebude fungovat vůbec, nebo minimálně není zaručena bezchybná funkcionalita. Nejedná se o moc výrazné omezení vzhledem k tomu, že společnost disponuje několika databázemi.

Nástroj netvoří automaticky záznam v tabulce s časovými razítky integrací, takže při definování nové relace nebude relace funkční a je nutno tento záznam ručně vložit. Taktéž toto omezení není moc výrazné, protože přidání nějaké relace není standardní operace.

Odstraněním zmíněných dvou omezení nad integračním datovým zdrojem by bylo vhodným rozšířením, pokud se v budoucnu bude poračovat ve vývoji programu.

Zaseknutí integrační relace

Tento problém se týká takové integrační relace, která není schopna vykonávat svou činnost kvůli vyjimečné události. O zaseknuté relaci má smysl mluvit pouze u relace, která bez zásahu administrátora systému se po čase sama nezprovozní. Bez zásahu se může relace zprovoznit např. při restartu databázového serveru když se z nedosupného datového zdroje stane dostupný a pod. Naopak pokud nastane neočekávaná hodnota ve vstupních datech, kterou cílový datový zdroj nepříjmá, pak je nutno tento problém ručně vyřešit (např. úpravou hodnoty konfliktní hodnoty ve zdrojovém datovém zdroji).

Problém zaseknutí integrační relace je poměrně závažný a nenašel jsem vhodný způsob jak obecně řešit jakoukoli neočekávanou chybu. Vhodným rozšířením programu by byla detekce relací, které jsou zaseklé, s aktivním upozorněním pověřené osoby (např. administrátora systému).

Transformace

Program má navržené rozhraní na obecnou transformaci, nicméně jedinou implementovanou transformací je transformace pomocí SOAP, kde SOAP službě se předají parametry ze zdrojového datového zdroje a hodnoty z výsledku služby se nahrají do cílového datového zdroje. Není zde možnost nakonfigurovat obecnou transformaci, ale s využitím implementace vhodné služby je možné dosáhnout libovolné reálné transformace, čímž se mnohem zvyšují možnosti integrace pomocí nově implementovaného nástroje.

Vhodným rozšířením programu by bylo přidání možnosti definice více transformací, v předem určeném pořadí jejich aplikací, a rozšíření palety možných transformací (převody mezi různými formáty datumu, využití regulárních výrazů, apod.).

8.3 Integrace jednotlivých softwarových prostředků

Vzhledem k faktu, že nově implementovaný integrační nástroj, který je použit na realizaci integrace softwarových prostředků ve společnosti, aktuálně nemá moc velkou podporu v

obhlasti transformace, tak pro většinu transformací jsou využity prostředky databázových systémů - spouště, procedury a pohledy.

Konfigurace a sql skripty k jednotlivým integracím se nachází na přiloženém cd v adresářích: konfigurace a sql. Konfigurace jednotlivých integrací je uložena ve 3 konfiguračních souborech, kde jeden je určen pro hlavní společnost, druhý pro dceřinnou a třetí pro jednorázové převody.

Následuje popis implementace jednotlivých integrací, navrhovaných v kapitole 7:

- Integrace výrobních dat,
- Integrace systémů s personálními daty zaměstnanců,
- Integrace docházkových systémů,
- Integrace starého informačního systému,
- Integrace procesu zápisu záloh,
- Integrace knihy jízd z O2 Carcontrol,
- Integrace hlavního informačního systému s informačními systémy dceřinných společností,
- Automatický přepočet odměn.

8.3.1 Integrace výrobních dat

Integrace výrobních dat proběhla ve 2 fázích:

- Synchronizace dat,
- Jednorázový převod dat.

Synchronizace dat zakázek

Synchronizace dat zakázek z výrobní databáze do databáze informačního systému je nastavena tak, že z pohledu, který v sobě sdružuje data zakázky a zákazníka, ve výrobní databázi se podle časového razítka poslední změny přesouvají změny do procedury v databázi informačního systému. Tímto způsobem je překonáno omezení transformací nově vzniklého ETL nástroje, resp. transformace s daty se provádí přímo na úrovni jednotlivých databází a ETL nástroj pouze předává data.

Interval synchronizací je nastaven na 6 minut. Změny nemusí být propagovány okamžitě, ve většině případů by stačila i synchronizace jednou za hodinu, nebo i pomalejší. Při realizaci této synchronizace bylo snahou především pokud možno co nejvíce omezit práci s výrobní databází (během pracovní doby bývá poměrně vytížená), ale jedna pracovní pozice (evidence dodaného materiálu) vyžadovala kratší interval mezi jednotlivými synchronizacemi.

Problémem dlouhého intervalu mezi jednotlivými synchronizacemi bylo, že by uživatel po naskladnění do skladu musel čekat např. hodinu na sychronizaci databází, než si mohl vytisknou dodací listy. Po několika testování s hodnotou synchronizačního intervalu v praxi se tento interval ustálil na rozumném kompromisu 6 minut, protože uživatel má na starosti více činností, takže 6-ti minutové omezení na tisk ničemu nevadí a jeden dotaz nad výrobní databází za 6 minut nepředstavuje vůbec žádnou zátěž. Interval by mohl být nastaven i na kratší dobu (z hlediska zátěže, by to nemělo hrát příliš velkou roli), ale nebylo to potřeba.

Jednorázový převod doplňkových dat

Výrobní databáze obsahovala některá data, která patřila logicky k zakázkám. Po dokončení synchronizace dat zakázek bylo možné převést tyto zbývající záležitosti. Mezi doplňková data lze zařadit především tabulky s daty o evidenci motáží (práce jednotlivých montérů), odměnách zaměstnanců za jednotlivé zakázky a data reklamací.

Převod byl s výhodou realizován obdobně jako synchronizace dat zakázek, ale s tím, že po převedení bylo možno tuto relaci vyřadit z konfigurace nástroje, resp. po jednorázovém převedení byly přenastaveny datové zdroje pro manipulaci s těmito daty na databázi informačního systému, takže ve výrobní databázi již nadále needocházelo ke změnám v těchto datech.

Při realizaci tohoto převodu se ukázalo, že převádění všech dat v rámci jediné transakce je poměrně zdlouhavé a vytváří enormní zátěž na databázový systém, proto byl do nástroje naimplementován tzv. limit relace. Tento limit umožňuje v konfiguraci nastavit maximální počet záznamů, které se mají synchronizovat v rámci jedné relace. Převod pak proběhl v několika menších transakcích. Tento princip byl aplikován i u synchronizace zakázek, protože při prvotním převodu bylo zapotřebí uložit statisíce záznamů.

8.3.2 Integrace systémů s personálními daty zaměstnanců

Tato integrace není úplně dotažena do konce, resp. do nově vytvořené tabulky v databázi informačního systému se kopírují, popř. modifikují, záznamy zaměstnanců z reportu ve formátu xls ze mzdového účetnictví PAMICA. Hlavním klíčem, pomocí kterého se data párují, je rodné číslo.

Problém dokončení této integrace tkví v tom, že v informačním systému se již zaměstnanci vyskytují v různých formách (docházka, odměny, apod.) a není možné automatizovaně přiřadit uživatele z informačního systému k zaměstnanci ze mzdového účetnictví PAMICA. Řešením by bylo jedině jednorázově ručně pospojovat tyto uživatele a další pak již generovat v databázi na základě tabulky se zaměstnanci. Takovýmto způsobem by šlo automatizovaně zakládat nové uživatelské účty, popř. je rušit, pokud se již zaměstnanec nevyskytuje ve mzdovém účetnictví (může se stát, že se informaci o ukončení pracovního poměru s nějakým zaměstnancem administrátor vůbec nedozví).

8.3.3 Integrace docházkových systémů

Tuto integraci se jako jedinou z návrhu nepodařilo realizovat. Program IKOS, který se používá na evidenci docházky v sídle společnosti nemá žádné vhodné prostředky, kterými by bylo možné nějaké integrace dosáhnout. Program sice využívá démony, pracující na pozadí, které umožňují jeho ovládání, ale společnost nemá k dispozici žádnou softwarovou dokumentaci k programu IKOS. Není jasné jak vlastně tito démoni fungují, jaký využívají komunikační protokol a pod.

Jediná teoretická možnost realizace byla pomocí reportu z jednotlivých terminálů v CSV formátu, ale tento report lze nechat vygenerovat pouze ručně a navíc obsahuje identifikaci pracovníků ve formě vnitřního identifikátoru, který není v informačním systému znám. Administrátor by pak musel ručně provazovat identifikátor zaměstnance v rámci informačního systému a vnitřní identifikátor zaměstnance v rámci programu IKOS, což by pravděpodobně vedlo k chybovosti a vyšší zátěži na administrátora systému.

8.3.4 Integrace starého informačního systému

Tato integrace v průběhu vytváření implementace jednotlivých integrací přestala být aktuální. Společnost přestala starý informační systém zcela používat, takže jakákoli snaha o integraci je již nadále bezpředmětná. Původní myšlenkou realizace této integrace bylo předávat data do starého informačního systému pomocí HTTP požadavku POST, ve kterém by byla předána potřebná data, stejně tak pro získání dat jakožto ze zdroje obdobnou cestou. Získaná data by mohla být např. ve formátu XML, ale do této fáze implementace se tento zdroj dat nedostal, resp. zíkávání dat z tohoto zdroje není aktuálně funkční (nebylo potřeba jej implementovat).

V nástroji je aktuálně implementována funkce vytvoření HTTP požadavku GET jakožto cílový datový zdroj, na které jsem testoval funkčnost tohoto řešení. Příklad konfigurace synchronizační relace s takovýmto datovým zdrojem je možno nalézt v příloze C.

8.3.5 Integrace procesu zápisu záloh

Tato integrace navazuje na dokončení zápisu záloh pomocí programu Profi plast, popsaném v 6.1. Pracovník zápisu záloh nyní vůbec nepracuje se starým informačním systémem a neeviduje zálohy ručně, že by musel přepisovat údaje zakázek do informačního systému.

Zápis nyní probíhá poloautomatizovaně, resp. uživatel si otevře z emailu soubor se zakázkou v programu Profi plast a díky připravenému tlačítku provede zápis všech relevantních informací do nové databáze záloh. Ukládání do nové databáze je z důvodu, že se využívá zápis do výrobní databáze (s některými doplňujícími daty), pouze nasměrován na jinou databázi.

V této fázi nastává činnost nástroje, resp. synchronizace dat z databáze záloh do databáze informačního systému. Při převodu nově zapsané zakázky do databáze informačního systému, dojde ihned automaticky, pomocí databázové spouště, k založení nové zakázky, která bude následně upravována podle výrobní databáze, jak bude postupně procházet jednotlivými výrobními procesy.

Požadavkem převodu mezi databázemi byla pokud možno co nejrychlejší propagace změn. Řešením by bylo buď nastavit velmi krátký interval mezi jednotlivými relacemi, ale nepřislo mi vhodné provádět velmi často dotazy nad třemi databázemi, proto je využito vlastnoti aktivní synchronizace nástroje popsané v 8.2.7.

Navazující zlepšení v rámci informačního systému spočívá v tom, že zakázka je založena hned ve fázi zápisu zálohy, tudíž se nečeká na zapsání do výrobní databáze (obvykle docházelo k prodlení několik pracovních dnů), pobočka si může vyplnit odměny k této zakázce a ihned vidí přidělenou odměnu v náhledu odměn. Dříve s několikadenní prodlením nebyly odměny pro zaměstnance příliš transparentní a často se stávalo, že odměnu, kterou měli dostat k určitému měsíci dostali až v následujícím.

8.3.6 Integrace knihy jízd z O2 Carcontrol

Integrace knihy jízd z O2 Carcontrol, která je popsána v 3.5.1, vychází z návrhu této integrace v 7.3.2, tedy pomocí příslušné webové služby budou stahovány knihy jízd do databáze informačního systému.

Webová služba, která zprostředkovává report s daty knihy jízd, má několik vstupních parametrů jako uživatelské jméno, heslo a za jaké období (počáteční a koncový den) má report obsahovat data. Parametry jméno a heslo jsou statické, resp. při případné změně by nemělo být problém zeditovat konfigurační soubor, kdežto období je dynamické.

Integrace využívá procedury v integrační databázi, která vrací počáteční a koncový den, který je potřeba synchronizovat. Procedura funguje tak, že počáteční den a koncový den nastaví o jeden den více, než datum poslední synchronizace. Pokud je koncový den shodný s aktuálním dnem, tak se koncový den nastaví podle datumu poslední synchronizace, díky čemuž počáteční datum bude větší, než koncové datum a služba nevrátí žádné výsledky a nedojde k synchronizaci. Je patrné, že synchronizace tudíž probíhá den zpětně, aby se již záznamy v knize jízd neměnily.

V příloze D je vidět nastavení příslušné SOAP transformace. Parametry pro webovou službu jsou odeslány jednoduše ve formě XML, ve kterém jsou jak statické hodnoty parametrů, tak dynamické hodnoty. Dynamické hodnotu mají formu #{NAZEV}, kde NAZEV je název sloupce ze zdrojové tabulky. Před odesláním dotazu dojde k jednoduchému textovému nahrazení za hodnotu ze zmíněného sloupce.

Konfigurace zdrojových a cílových sloupců je vidět v příloze E. Zdrojové sloupce nejsou v cílových vůbec požity. Cílové sloupce používají jako zdroj dat jen výsledky transformace. U jednoho ze sloupců je použita konverze, protože získaný časový údaj z O2 serveru má formát ISO 8601, ale cílový sloupec je databázové časové razítko (timestamp), což není přímo kompatibilní, resp. Firebird databáze si s tímto údajem sama neporadí.

Na této integraci lze dobře demonstrovat situaci, že pro jeden řádek ze vstupu, který je generován vždy jen jeden procedurou, je možné vytvořit několik řádků na výstupu pomocí transformace. Počet řádků na výstupu je závislý na počtu záznamů v XML souboru, který zasílá webová služba pomocí SOAP. Pokud XML neobsahuje uzel označený v konfiguraci SOAP transformace jako dataSeparator, tak se relace považuje za nezdařenou a nemění se časové razítko poslední synchronizace. Tato situace nastává např. když O2 server zasílá zprávu o chybě, nebo když chce nástroj synchronizovat dnešní den.

8.3.7 Integrace hlavního informačního systému s informačními systémy dceřinných společností

Jak již bylo zmíněno v návrhu, tak by tato integrace měla probíhat oboustraně. Nástroj přímo nepodporuje konfiguraci pro obousměrnou integraci, takže je toto nutné realizovat dvěmi synchronizačními relacemi, kde jedna reprezentuje směr od dceřinné společnosti do hlavní společnosti a druhá směr opačný.

Směr od dceřinné společnosti lze realizovat s výhodou pomocí již vytvořené integrace záloh, resp. s použitím obdobné konfigurace s mírnými modifikacemi datových zdrojů. Místo toho aby byl zdroj databáze záloh a cíl databáze informačního systému v rámci jedné společnosti, tak zdrojem bude tabulka s daty záloh v databázi informačního systému v dceřinné společnosti a cílem bude ta samá tabulka v databázi informačního systému hlavní společnosti (tabulka záloh má stejnou strukturu jako v databázi záloh, tak v databázi informačního systému). Výsledkem bude simulace normálního zápisu zálohy v hlavní společnosti, ale zdrojem bude dceřinná společnost.

Opačný směr je značně limitován, resp. největším předmětem zájmu je stav výroby, ve kterém se zakázka nachází. Tato integrace je poměrně snadná, změny v databázi hlavní společnosti se propagují do databáze dceřinné společnosti voláním procedury s parametry: označení zakázky a stav. Pokud zakázka existuje v dceřinné společnosti a její stav je různý od příchozího, pak se nalezené zakázce modifikuje stav.

Realizace by mohla probíhat i přímou synchronizací mezi tabulkami s nastavením klíče na číslo zakázky, ale bylo by nutné filtrovat zakázky pro danou dceřinnou společnost. Pomocí procedury odpadá nutnost tyto zakázky filtrovat. Přímá oboustranná synchronizace

mezi tabulkami, aby nedocházelo k cyklení synchronizace, je ošetřena tak, že se sql příkaz update nad daným záznamem zavolá jen v případě, že je alespoň 1 hodnota, která má být modifikována, různá od uložené hodnoty v záznamu. Jinými slovy nástroj neprovede změnu, pokud záznam je již ve stavu v jakém by měl být po aplikování změn, díky čemuž se nezmění časové razítko poslední změny záznamu.

8.3.8 Automatický přepočet odměn

Nejedná se o integraci jako takovou a tudíž není součástí návrhu. Jedná se spíše o demonstrativní využití implementovného nástroje.

Jádro problému u odměn byl fakt, že odměny nejsou fixní, ale jsou přímo závislé na celkové částce, za kterou obchodník prodal zboží za měsíc, resp. podle této částky se určuje kolik procent ze zisku má daný obchodník získat. Takovýmto způsobem počítat odměny vždy, když na ně chce někdo nahlížet je značně neefektivní a pomalé, proto byly odměny předpočítávány při změně.

Pomocí předpočítávání fungoval přepočet odměn po dokončení integrace výrobních dat a integrace procesu zápisu záloh. Problémem tohoto přístupu bylo, že pokud nastala změna u zakázky v částce (v případě chyby a pod.), muselo dojít k přepočítání odměn. Přepočítání bylo rovnou voláno v rámci databáze příslušnou spouští a bylo poměrně pomalé.

Nastavení synchronizační relace, aby volala jednou za určitý interval proceduru v databázi informačního systému, která přepocet provede, řeší neustálé problémy s rychlostí výpočtu odměn v reálném čase. Takovýmto způsobem lze vykonávat opakovaně s určitou nastavenou prodlevou nad databází nějakou činnost. Obvykle takovouto činnost databázové stroje nepodporují. Konfigurace relace pro přepočet odměn je možno nalézt v příloze B.

Kapitola 9

Případová studie

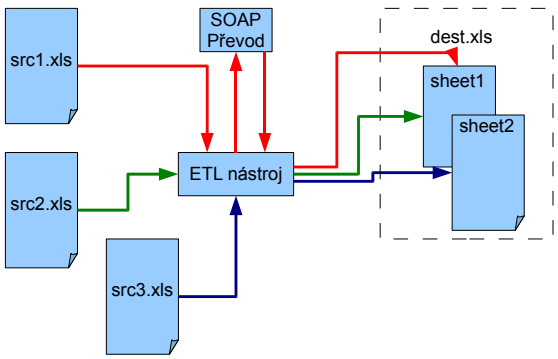
Tato kapitola obsahuje jednoduchou případovou studii, na které je demonstrována činnost nově implementovaného integračního nástroje. Složitější integrační konfigurační skripty jednotlivých integračních relací jsou použitelné jen v rámci firemní sítě, kvůli přístupu k databázovému serveru a jiným softwarovým prostředkům.

Všechny soubory případové studie, včetně aplikace a knihoven, je možno nalézt na přiloženém CD v adresáři: pripadova_studie. Demonstrativní příklad je přímo připraven k použití. Při zapnutém ETL nástroji se budou změny ve zdrojových souborech propagovat v určitých intervalech do cílového souboru.

Základem integrace jsou 4 soubory ve formátu XLS. Tento formát je zvolen, protože jako jediný z podporovaých datových zdrojů nově vzniklého integračního nástroje nevyžaduje spojení s databázovým serverem.

Nástroj chápe jednotlivé soubory v tomto formátu jako samostatné databáze a listy (sheets) jako tabulky. Dále vždy používá první řádek jako definiční pro názvy sloupců jednotlivých tabulek. Následující řádky již obsahují data.

Zdrojové databáze se nazývají src1.xls, src2.xls a src3.xls a cílová databáze se nazývá dest.xls. Případová studie demonstruje integraci dvou různých tabulek z dvou různých databází do jedné tabulky v cílové databázi podle klíče, který je nastaven pro jeden ze slouců. Navíc na integraci dat z databáze src1 je použita transformace dat pomocí veřejné SOAP služby (tato integrace vyžaduje, aby počítač, na kterém je ETL nástroj spuštěn měl přístup na internet). Integrace tabulky ze třetí databáze demonstruje jednoduchou synchronizaci s tabulkou v cílové databázi.



Obrázek 9.1: Diagram průběhu činnosti ETL nástroje v případové studii.

Kapitola 10

Závěr

Hlavním předmětem zájmu této diplomové práce bylo navrhnout integraci klíčových částí systému, které aktuálně společnost potřebuje. Bylo by jistě možné vymyslet další možnosti integrace, ale ty by nebyly natolik aktuální a použitelné.

Tato práce postupně popisuje datové typy, rozhraní, datová uložiště a informační požadavky jednotlivých uživatelských skupin, jaký je aktuální stav softwarového vybavení společnosti, nastiňuje kam společnost míří (co se týče IT technologií), snaží se objasnit základní otázky integrace a možnosti integrace pro tuto společnost. Práce obsahuje návrh jakým způsobem se budou obecně integravat jednotlivé softwarové prostředky. Každá navrhovaná integrace popisuje zdrojový a cílový prostředek, zdůvodnění integrace a obecný způsob provedení dané integrace.

Problémy při implementaci navrhlé integrace byly především z důvodů, že softwarové vybavení společnosti, jak je vidět z přecházejícího textu, je poměrně disharmonické. Data často obsahují chyby a jsou v nekonzistentním stavu. Mnoho softwarových prostředků, které společnost využívá, s integrací do většího celku nepočítá, takže pak vzniká problém jakým způsobem nějakou integraci s takovýmto prostředkem vlastně vytvořit. Hlavními problémy jsou především speciální datové formáty, nepodléhající žádnému standardu, nebo nemožnost provedení automatizovaného importu, či exportu dat. Přes tyto problémy se integrace alespoň některých systémů zdařila. Některé integrace vypadaly slibně pouze ve fázi plánování, ale později ve fázi implementace se ukázaly být jako nereálné a v praxi neproveditelné.

Přínosem této diplomové práce je vytvoření několika integrací softwarových prostředků a následné zefektivnění práce zaměstnanců společnosti, odstranění některých chyb ze strany uživatelů a zlepšení kontroly nad zaměstnanci, či aktuálním stavem společnosti, ze strany vedení společnosti. Dále jako hlavní přínos této diplomové práce bych označil nově vzniklý ETL nástroj, který pracuje i s databázovým systémem Firebird. Do budoucna by bylo vhodné, jako rozšíření tohoto nově zvniklého ETL nástroje, dále zapracovat na jeho možnostech konfigurace, zvýšit jeho možnosti transformace dat a rozšířit paletu možných zdrojů dat.

Literatura

[1] About XStream [online]. `http://xstream.codehaus.org/`, 2008 [cit. 2011-04-12].

[2] The Document Object Model [online]. `http://xerces.apache.org/xerces2-j/dom.html`, 2010 [cit. 2011-04-03].

[3] Commons IO [online]. `http://commons.apache.org/io/`, 2010 [cit. 2011-04-07].

[4] Apache POI - the Java API for Microsoft Documents [online]. `http://poi.apache.org/`, 2011 [cit. 2011-05-05].

[5] Integrace softwarových systémů [online]. `http://www.profinit.eu/cz/it-reseni/integrace-sw-systemu`, [cit. 2010-11-25].

[6] Bříza, V.: *Acrobat 7: podrobně a prakticky*. Snadno a rychle, Grada, 2006 [cit. 2010-11-01], ISBN 9788024719382.

[7] Connolly, T.; Begg, C.: *Database systems: a practical approach to design, implementation, and management*. International computer science series, Addison-Wesley, 2005 [cit. 2010-11-14], ISBN 9780321210258.

[8] Constantinos, M.: *OLAP Technology [Online]*. University of Athens, [cit. 2010-11-22]. URL `http://cgi.di.uoa.gr/~kmorfo/DownloadFiles/Cube.pdf`

[9] Císař, P.: *Podrobná příručka Interbase/Firebird*. Computer Press, 2003 [cit. 2011-01-02], ISBN 80-7226-946-1.

[10] Elliote R. Harold, S. W. M.: *XML V KOSTCE, Pohová referenční příručka*. GRADA, 2002 [cit. 2010-10-28], ISBN 80-7226-712-4.

[11] Hall, M.: *Java servlety a stránky JSP*. [cit. 2010-12-05], ISBN 80-86-330-06-0.

[12] Hammergren, T.; Simon, A.: *Data Warehousing for Dummies*. For Dummies, John Wiley & Sons, 2009 [cit. 2010-11-14], ISBN 9780470407479.

[13] Hanzálek Zdeněk, c. P.: Problém obchodního cestujícího [online]. `http://edux2.felk.cvut.cz/modules/edux/get_file_from_dms.php?function=view&FileID=1737&source=p` , [cit. 2011-01-05].

[14] Jergl, V.: Vozový park je třeba nejenom sledovat, ale především řídit. *CIO BUSINESS WORLD*, ročník 2011, č. 5, 2011 [cit. 2011-05-01], ISSN 1803-7321.

[15] Juřek, M.: Moderni integrace aplikací [online].
http://download.microsoft.com/download/8/6/c/86c09926-affc-4e14-bec0-3c45cd989436/Moderni_integrace.pdf, [cit. 2010-12-02].

[16] Kosek, J.: *XML a PHP*. Grada, 2009 [cit. 2010-12-19], ISBN 9788024711164.

[17] Pour, J.; Toman, P.: *Podniková informatika: počítačové aplikace v podnikové a medzipodnikové praxi, technologie informačních systémů, řízení a rozvoj podnikové informatiky*. Expert (Grada), Grada, 2006 [cit. 2010-11-29], ISBN 9788024712789.

[18] Prasad, K.: *Data Warehouse Development Tools*. Dreamtech Press, 2005 [cit. 2010-11-14], ISBN 9788177226430.

[19] Procházka, D.: *Oracle*. Grada, [cit. 2011-04-07], ISBN 9788024727622.

[20] Roman, D.: *OLAP*. IDOC, [cit. 2010-11-22].
URL http://homel.vsb.cz/~dan11/isys/Danel%20-%20IS%20-%20OLAP.pdf

[21] Schiller, M.: Co se skrývá pod zkratkou ETL? [online].
http://www.systemonline.cz/clanky/co-se-skryva-pod-zkratkou-etl.htm, 2003 [cit. 2010-11-13].

[22] Toman, I.: *XML technologie: principy a aplikace v praxi*. Průvodce (Grada), Grada, 2008 [cit. 2010-11-01], ISBN 9788024727257.

[23] Withee, K.: *Microsoft Business Intelligence For Dummies*. For Dummies, John Wiley & Sons, 2009 [cit. 2010-12-10], ISBN 9780470526934.

Příloha A

Obsah CD

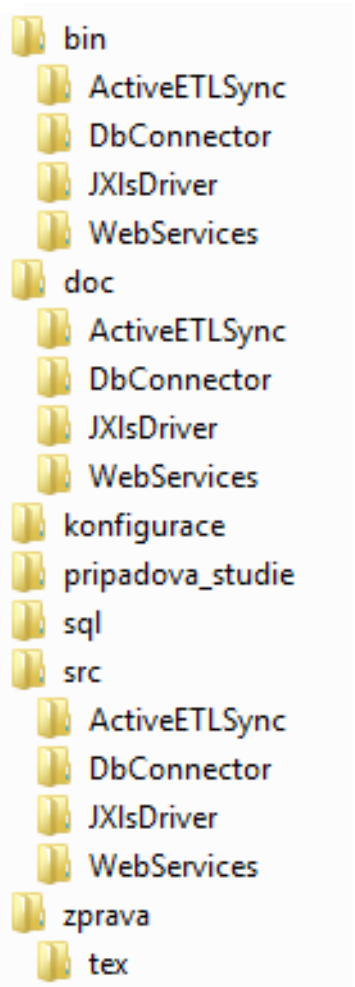
Diplomovou práci tvoří 4 projekty, kde ActiveETLSync je hlavním projektem a ostatní jsou pomocné knihovny. V adresáři doc se nachází dokumentace k jednotlivým projektům, v adresáři bin jsou binární spustitelné soubory projektů a v adresáři src jsou kompletní projekty. Každý projekt má v těchto adresářích vlastní adresář podle jeho názvu.

Adresář pripadova_studie obahuje demonstrační příklad, na kterém lze vyzkoušet funkcionalitu implementovaného ETL nástroje.

Adresář konfigurace obsahuje konfigurační skripty pro jednotlivé integrace. Konfigurační soubor pro hlavní společnost se nazývá: config-company.xml, pro dceřinnou společnost: config-subcompany.xml a jednorázový konfigurační soubor se nazývá: config-onetime.xml.

V adresáři sql je možno nalézt jednotlivé procedury, které jsou použity v rámci databázových systémů ve společnosti.

Adresář zprava obsahuje text diplomové práce a adresář tex, ve kterém je obsah v latexu, ze kterého je možno tuto zprávu vygenerovat.



Obrázek A.1: Adresářová struktura přiloženého CD.

Příloha B

Ukázka konfigurace přepočtu odměn

```
<!-- @Author: Bc. Jarosav Strouhal -->
<!-- Relace: prepocet odmen -->
<syncRelation>
        <!--
                Nazev synchronizacni relace - podle tohoto nazvu se uklada timestamp
                posledni synchronizace v synchronizacni databazi (tabulka sync)!
        -->
        <name>prepocet_koeficientu</name>
        <!-- datovy zdroj - reference je XPath -->
        <dataSrc class="database" reference="../../../dataSources/database"/>
        <!-- datovy cil - reference je XPath -->
        <dataDest class="database" reference="../../../dataSources/database"/>
        <!-- Zdrojova tabulka -->
        <srcTable>cas_vcera</srcTable>
        <!-- Cilova tabulka -->
        <destProcedure>prepocet_koeficientu</destProcedure>
        <!-- Synchronizacni timestamp ve zdrojove databazi -->
        <syncTimeStamp>LSTPD</syncTimeStamp>
        <!-- Synchronizacni interval -->
        <interval>3000000</interval>
</syncRelation>
```

Příloha C

Ukázka konfigurace HTTP GET požadavku

```
<!-- @Author: Bc. Jarosav Strouhal -->
<!-- Relace: datumu vyrobeni skla na web -->
<syncRelation>
	<!--
		Nazev synchronizacni relace - podle tohoto nazvu se uklada timestamp
		posledni synchronizace v synchronizacni databazi (tabulka sync)!
	-->
	<name>skla_web_sync</name>
	<!-- datovy zdroj - reference je XPath -->
	<dataSrc class="database" reference="../../../dataSources/database[3]"/>
	<!-- Datovy zdroj, ktery odesila data HTTP GET protokolem -->
	<dataDest class="httpGet">
		<!-- url adresa dane webove sluzby -->
		<url>http://www.eurojordan.com/cz/pobocky/vyrobeno_sklo.php</url>
	</dataDest>
	<!-- Zdrojova tabulka -->
	<srcTable>SKLA_WEB_SYNC</srcTable>
	<!-- Synchronizacni timestamp ve zdrojove databazi -->
	<syncTimeStamp>LSTPD</syncTimeStamp>
	<!-- Synchronizacni interval -->
	<interval>1200000</interval>
	<!-- Vyjmenovani sloupcu, ktere je treba synchronizovat -->
	<cols>
		<col>
			<srcCol>ZAKAZKA</srcCol>
			<destCol>CS</destCol>
		</col>
		<col>
			<srcCol>STOJAN</srcCol>
		</col>
		<col>
			<srcCol>DATUM</srcCol>
		</col>
	</cols>
</syncRelation>
```

Příloha D

Ukázka konfigurace soap transformace

```
<!-- @Author: Bc. Jarosav Strouhal -->
<transformation class="soapExport">
  <url>https://carcontrol.cz.o2.com/ssl/Export.asmx</url>
  <encoding>utf-8</encoding>
  <msg><![CDATA[
    <GenerateExportXML xmlns="http://tempuri.org/">
      <jmeno>****</jmeno>
      <heslo>****</heslo>
      <jazyk>cs</jazyk>
      <exportID>export_kniha_jizd_zakladni_firma</exportID>
      <paramKeys>
        <string>datum_od</string>
        <string>datum_do</string>
        <string>vozidla_id</string>
        <string>gps</string>
      </paramKeys>
      <paramValues>
        <string>#{DATUM_OD}</string>
        <string>#{DATUM_DO}</string>
        <string></string>
        <string>0</string>
      </paramValues>
    </GenerateExportXML>
    ]]>
  </msg>
  <dataSeparator>
        <name>Radky</name>
        <path>
              GenerateExportXMLResponse/GenerateExportXMLResult/NewDataSet
        </path>
  </dataSeparator>
```

```
<transformSources>
   <transformSource class="soapSource">
      <node><name>vozidlo_spz</name></node>
   </transformSource>
   <transformSource class="soapSource">
      <node><name>vyjezd_datum</name></node>
   </transformSource>
   <transformSource class="soapSource">
      <node><name>prijezd_datum</name></node>
   </transformSource>
   <transformSource class="soapSource">
      <node><name>mista</name></node>
   </transformSource>
   <transformSource class="soapSource">
      <node><name>ucel</name></node>
   </transformSource>
   <transformSource class="soapSource">
      <node><name>ridic</name></node>
   </transformSource>
   <transformSource class="soapSource">
      <node><name>delka_osobni</name></node>
   </transformSource>
   <transformSource class="soapSource">
      <node><name>delka_firemni</name></node>
   </transformSource>
   <transformSource class="soapSource">
      <node><name>delka</name></node>
   </transformSource>
   <transformSource class="soapSource">
      <node><name>tachometr</name></node>
   </transformSource>
   <transformSource class="soapSource">
      <node><name>pocet_projetych_sekund</name></node>
   </transformSource>
  </transformSources>
</transformation>
```

Příloha E

Ukázka konfigurace transformace sloupců

```
<!-- @Author: Bc. Jarosav Strouhal -->
<cols>
  <!-- zdrojový sloupec, nekopíruje se do cíle -->
  <col><srcCol>DATUM_OD</srcCol><destColCopy>False</destColCopy></col>
  <col><srcCol>DATUM_DO</srcCol><destColCopy>False</destColCopy></col>
  <!-- nemá nastavený zdroj, data se berou z transformace -->
  <col>
    <transformSource class="soapSource"
        reference="../../../transformation/transformSources/transformSource[2]" />
    <destCol>VYJEZD</destCol>
    <conversions>
      <conversion class="iso8601ToTimeStamp"></conversion>
    </conversions>
  </col>
  <col>
    <transformSource class="soapSource"
        reference="../../../transformation/transformSources/transformSource[3]" />
    <destCol>PRIJEZD</destCol>
    <conversions>
      <!-- Konverze hodnoty z transformace ISO 8601 na databázový timestamp -->
      <conversion class="iso8601ToTimeStamp"></conversion>
    </conversions>
  </col>
  <col>
    <transformSource class="soapSource"
        reference="../../../transformation/transformSources/transformSource[4]" />
    <destCol>MISTA</destCol>
  </col>
  <col>
    <transformSource class="soapSource"
        reference="../../../transformation/transformSources/transformSource[9]" />
    <destCol>DELKA</destCol>
  </col>
  <col>
    <transformSource class="soapSource"
        reference="../../../transformation/transformSources/transformSource" />
    <destCol>SPZ</destCol>
  </col>
</cols>
```