Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství

Bakalářská práce

Relačně databázová technologie v problematice Master Data Management

Petr HOUF

Česká zemědělská univerzita v Praze | Provozně ekonomická fakulta

Katedra informačního inženýrství | Akademický rok 2009/2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Petr Houf

obor Informatika

Vedoucí katedry Vám ve smyslu Studijního a zkušebního řádu ČZU v Praze čl. 16 určuje tuto bakalářskou práci.

Název práce: **Relačně databázová technologie v problematice Master Data Management**

Osnova bakalářské práce:

1. Úvod
2. Cíl práce a metodika
3. Objasněte teoretické principy relačně db technologie a MDM
4. Navrhněte a realizujte konkrétní uplatnění relačně db technologie v MDM
5. Vymezte a zobecněte přínosy takovéhoto uplatnění
6. Závěr
7. Seznam použitých zdrojů
8. Přílohy

Rozsah hlavní textové části: 30 - 40 stran

Doporučené zdroje:

Adelman,S., Moss,L.T.: Data Strategy. Prentice Hall PTR, 2005.ISBN 0-321-24099-5
Brown, P. C.: Implementing SOA. Addison Wesley Professional, 2008. ISBN 0-321-50472-0
Inmon,W. H., Strauss, D., Neushloss, G.: DW 2.0. Morgan Kaufmann OMG Press, 2008. ISBN 978-0-12-374319-0
Loshin, D.: Master Data Management. Morgan Kaufmann OMG Press, 2009.ISBN 978-0-12-374225-4
Minoli, D.: Enterprise Architecture A to Z. Taylor & Francis Group, 2008. ISBN 978-0-8493-8517-9

Vedoucí bakalářské práce: **Ing. Václav Vostrovský, Ph.D.**

Termín odevzdání bakalářské práce: duben 2011

.....
Vedoucí katedry

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE
Provozně ekonomická fakulta

.....
Děkan

V Praze dne: 15. 1. 2010

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Relačně databázová technologie v problematice Master Data Management" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na jejím konci. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 21. března 2011 _____

Poděkování

Děkuji tímto panu docentovi Vostrovskému za cenné rady v oblasti relačních databázových systémů a dále pak Ing. Vladimíru Kyjonkovi ze společnosti SAS za konzultace a pomoc s výběrem studijních materiálů.

Relačně databázová technologie v problematice Master Data Management

Relational Database Technology facing Master Data Management problems

Souhrn

Relační databázové technologie jsou základním prvkem prakticky všech současných informačních systémů. S růstem jejich komplexnosti a objemů spravovaných dat se však mění požadavky, které jsou na ně kladeny. Platformy jako jsou datové sklady, servisně orientovaná architektura nebo Cloud Computing vyžadují sdílení dat, nyní zpravidla roztříštěných po mnoha vzájemně oddělených datových silech. Dříve přínosné specializované aplikace s dedikovanými datovými úložišti dnes brání zavádění komplexních podnikových procesů a jejich následné automatizaci. Master Data Management je souborem postupů, technologií a organizačních opatření, která umožňuje stávající systémy a aplikace integrovat a vytvořit tak moderní infrastrukturu pro pružnou podporu obchodních cílů organizace.

Summary

Relational database technology is an essential component of almost all existing information systems. With the growth of their complexity and volume of managed data are changing the requirements that are imposed upon them. Platforms such as a Data Warehouses, Service-oriented Architecture or the Cloud Computing requires easy sharing of data, now usually dispersed on a number of mutually separate data silos. Previously useful specialized applications with dedicated data storage currently complicate the implementation of complex business processes and their subsequent automation. Master Data Management is a set of procedures, technologies and organizational arrangements, which allows integrate existing systems and applications to create a modern infrastructure for flexible support of business objectives of the organization.

Klíčová slova:

Relační databázové systémy, SQL, servisně orientovaná architektura, master data management, profilování dat, metadata, podnikový datový slovník, čištění dat, distribuce dat

Keywords:

Relational Database Systems, SQL, Service Oriented Architecture , Master Data Management, Data Profiling, Metadata, Enterprise Data Dictionary, Data Cleansing, Data Provisioning

Obsah

1 Úvod

Oblast informačních technologií patří dlouhodobě mezi jedno z nejdynamičtějších odvětví současnosti. V posledních několika letech však lze pozorovat snahu o efektivnější využití potenciálu, který IT nabízí. Zatímco v minulosti se technologická vyspělost hodnotila především objemem investovaných prostředků, dnes je kladen důraz na zvýšení konkurenceschopnosti společnosti a návratnosti vložených investic. Hlavní požadavky směřované na informatiky lze shrnout do následujících třech oblastí:

Provázanost s obchodními cíli organizace

Inovace v oblasti informačních a komunikačních technologií by dle dnešních CEO a CIO manažerů neměla být samoúčelná, ale konkrétním způsobem přispívat k naplnění plánovaných cílů týkajících se primární činnosti podniku. Například změna ERP systému by neměla být prováděna jen proto, že nový systém je postaven na modernější architektuře, ale například proto, že má nabízí podporu JIT zásobování, jehož zavedením ušetří podnik značnou část nákladů na skladovací prostory.

Flexibilita

Změn, které mohou výrazným způsobem ovlivnit požadavky kladené na informační infrastrukturu organizace je nespočet. Ve většině případů se jedná o neplánované, nebo dlouhodobě obtížně předvídatelné situace (dojít může ke změně vlastníka, změně výrobního programu, změnit se může legislativa nebo požadavky zákazníků či obchodních partnerů). Přizpůsobitelnost a otevřenost je proto dnes jedním z hlavních kritérií při výběru jednotlivých komponent systému, ale i jejich dodavatelů.

Efektivita

Dosažení vytýčených cílů je třeba zajistit s co nejnižšími náklady. Snahou je proto maximálně využít stávající systémy a technologie. Více než dříve je pak kladen důraz na návratnost investic (ROI) a celkové náklady na vlastnictví (TCO), zohledňující veškeré výdaje spojené s provozem a údržbou, technickou podporu jednotlivých systémů a aplikací nebo zaškolení uživatelů.

Výše uvedeným trendům se přizpůsobila nabídka dodavatelů a konzultačních společností a do popředí zájmu se tak dostávají technologie umožňující realizovat výnos respektive úsporu z rozsahu. V dlouhodobém horizontu se jedná především o implementaci Cloud

Computingu a přechod na servisně orientovanou architekturu (SOA), střednědobé projekty jsou pak zaměřeny na virtualizaci prostředků nebo budování datových skladů pro podporu rozhodování (BI).

Kromě modernizace a integrace stávajících systémů lze posun v uvažování podniků pozorovat i v oblasti řízení. Mnohem větší pozornost než v minulosti je dnes věnována procesům, metodikám, využívání ověřených postupů („best practices“) a s nimi spjatých kontrolních mechanismů. I zde je dominantní snaha o maximalizaci užitku z existujících i nově pořizovaných technologií a systémů.

Výše uvedené platí beze zbytku i pro oblast databázových technologií. Současná role relačních databází je ovlivňována především analytickým zpracováním dat (BI), integračními projekty, architekturou SOA a již zmiňovaným Cloud Computingem, jejichž úspěšné nasazení vyžaduje zásadní změny v přístupu k návrhu, implementaci a správě datových úložišť. Disciplínu, která se zabývá touto problematikou, nazýváme Enterprise Data Management (dále EDM) a její velmi důležitou podmnožinu pak tvoří Master Data Management (MDM), jemuž je věnována tato bakalářská práce.

2 Cíl práce a použitá metodika

Cíl práce

Master Data Management je v současnosti hojně využívaným termínem. Lze nalézt velké množství nejrůznějších definic, které jsou však zpravidla formulovány dodavatelskými a konzultačními společnostmi tak, aby co možná nejlépe korespondovaly s nabídkou jejich produktů a služeb. Cílem předkládané bakalářské práce je objektivně objasnit problematiku spojenou s Master Data Managementem (dále jen MDM), popsat jeho jednotlivé součásti a následně demonstrovat a vymezit jeho přínosy na konkrétním řešení.

Členění práce

Vlastní práce bude rozdělena do tří částí. První teoretická část stručně shrnuje historii a principy relačně databázové technologie, aktuálních trendů a požadavků, které jejich nasazení na současné databázové systémy klade. Dále bude rozebrán význam Master Data Managementu a jeho jednotlivé součásti. Druhá část bude věnována především vymezení hlavních činností a postupů používaných během zavádění MDM. Závěrečná část práce popisuje konkrétní uplatnění MDM v rámci informační infrastruktury podniku a přínosy, které jeho implementace přinesla.

Metodika a použité zdroje

Předkládaná práce využívá především metod systémové analýzy a designu IS, konkrétně entitně-relační modelování, modelování datových toků a procesní modelování. Praktická část byla zpracována pomocí databázových nástrojů společnosti Embarcadero nad databázovým strojem Oracle 10g.

Jako zdroj pro vypracování BP posloužily odborné technické publikace doporučené vedoucím práce panem docentem Ing. Vostrovským a panem Ing. Kyjonkou ze společnosti SAS. Dále jsem využil vlastních zkušeností z řady projektů, na kterých jsem se měl možnost podílet během svého působení v roli presales konzultanta a systémového architekta.

3 Teoretické principy relačně db technologie a MDM

Relačně databázové technologie

Relační model

Sběrem a následnou kategorizací dat se lidé zabývali odpradávna. S příchodem výpočetní techniky se otevřely nové možnosti a to především v oblasti zpracování velkých objemů dat. Z počátku byl výzkum v této oblasti iniciován především armádou a státními organizacemi. Za kolébku databázových systémů jsou považovány Spojené státy americké, kde vznikla roku 1960 pod záštitou ministerstva obrany pracovní skupina "Data Systems Languages" a následně pak specializovanější "Database Task Group" v čele s Charlesem Bachmanem (Selinger, 1987).

Jimi publikované výstupy daly vzniknout celé řadě implementací. O tu zásadní se zasloužil zaměstnanec společnosti International Business Machines (dnes známé jako IBM) Edgar Frank Codd. Codd byl velmi kritický k dosud použitým principům a začal proto pracovat na vlastním návrhu. Roku 1970 pak zveřejnil svou práci „A Relational Model of Data for Large Shared Data Banks“, podle které se jím navržený a do současnosti využívaný model nazývá „relačním“ (McGee, 1981). Databázové systémy, které byly následně na bázi relačního modelu navrženy, tvoří dodnes jednu z nejdůležitějších technologií počítačového průmyslu (Groff, 2005, s. 27).

Relační algebra

Coddova relační teorie definuje základních operace pro práci s daty jako je selekce, projekce a spojení, které jsou souhrnně označovány jako relační algebra (Vostrovský, 2008, s. 19 a 20). Mějme dvě jednoduché tabulky "A" a "B" s následující strukturou a vloženými daty:

Tabulka "A"	
dprostredek	**barva**
Motorka	Červená
Motorka	Žlutá
Loď	Červená
Loď	Žlutá

Tabulka "B"		
dprostredek	**barva**	**rychlost**
Auto	Modrá	180
Motorka	Červená	230

Obrázek 1: Obsah v příkladech použitých databázových tabulek (Zdroj: Vlastní)

Selekce - Pomocí selekce lze výběr omezit pouze na řádky, odpovídající kritériím definovaným v klauzuli WHERE.

Příklad SQL dotazu:

```
SELECT * FROM A WHERE barva = 'červená';
```

Vrácený výsledek:

	dprostredek	barva
1	Motorka	Červená
2	Loď	Červená

Projekce - Projekce umožňuje uživateli omezit výsledek db dotazu pouze na výčtem stanovené atributy (sloupce).

Příklad SQL dotazu:

```
SELECT dprostredek, rychlost FROM B;
```

Vrácený výsledek:

	dprostredek	rychlost
1	Auto	180
2	Motorka	230

Spojení - Spojení (Union) slouží ke sjednocení relací. Nutnou podmínkou pro vytvoření spojení více výsledkových sad je jejich shodná struktura, tedy počet a datových typ sloupců, které mají být obsaženy v konečné výsledkové sadě.

Příklad SQL dotazu:

```
SELECT dprostredek, barva
FROM A UNION SELECT dprostredek, barva FROM B;
```

Vrácený výsledek:

	dprostredek	barva
1	Auto	Modrá
2	Loď	Žlutá
3	Loď	Červená
4	Motorka	Žlutá
5	Motorka	Červená

Výše uvedený příklad ukazuje jen jeden z mnoha existujících druhů spojení, které moderní databázové stroje nabízí (Groff,Weinberg, 2005, s. 143-183). Přesto, že konkrétní implementace a použitá syntaxe se mohou částečně lišit podle zvoleného dodavatele,

v nejrozšířenějších RDBMS (Oracle, MS SQL Server, IBM DB2 nebo Sybase ASE) se můžeme setkat s následujícími typy spojení:

- Přirozené spojení (natural join)
- Theta spojení (theta join)
- Polospojení (semijoin)
- Antispojení (antijoin)
- Množinové dělení (division)
- Vnější spojení (Outer Join)
    - Levé vnější spojení (Left Outer Join)
    - Pravé vnější spojení (Right Outer Join)
    - Plné vnější spojení (Full Outer Join)

Kartézský součin - Kartézský součin je specifickým druhem spojení, který vrací úplný výčet kombinací zadaných relací. Jedná se tedy o spojení bez jakékoliv podmínky (join condition). Pokud pro jeho použití neexistují pádné důvody, snažíme se mu v praxi vyhnout pro jeho paměťovou náročnost.

Příklad SQL dotazu:

```
SELECT * FROM B CROSS JOIN A;
```

Vrácený výsledek:

	dprostredek	barva	rychlost	dprostredek	barva
1	Auto	Modrá	180	Motorka	Červená
2	Auto	Modrá	180	Motorka	Žlutá
3	Auto	Modrá	180	Loď	Červená
4	Auto	Modrá	180	Loď	Žlutá
5	Motorka	Červená	230	Motorka	Červená
6	Motorka	Červená	230	Motorka	Žlutá
7	Motorka	Červená	230	Loď	Červená
8	Motorka	Červená	230	Loď	Žlutá

SQL a jeho součásti

Kromě použití relační algebry zároveň Codd definoval základní požadavky na jazyk pro manipulaci s daty. Příkazy měly vycházet z přirozeného jazyka (angličtiny) a být nezávislé na fyzickém uložení dat. V rámci IBM tak postupně spatřil světlo světa jazyk SEQUEL (Structured English Query Language) a v listopadu 1976 pak jeho druhá generace SEQUEL2 (McGee, 1981). Zjednodušením jeho názvu pak vznikl dnešní SQL. Díky

různým proprietárním rozšířením DB dodavatelů se dnes můžeme setkat s mnoha dialekty. Například výše uvedené ukázky využívají SQL dialekt Oracle. Zásadní rysy jazyka jsou však totožné díky standardu ANSI.

ANSI SQL

American National Standards Institute je organizace, která se zabývá schvalováním standardů pro strategická průmyslová odvětví. Ustavení standardů a nezávislost jazyka SQL na konkrétní realizaci či RDMS bylo považováno za důležité především z hlediska přenositelnosti dat. V roce 1987 tak byl standard ANSI SQL přijat jako mezinárodní norma Mezinárodní organizace pro normalizaci (ISO). Nejnovějším standardem je norma ANSI SQL-99. Protože však není závazná, spoléhají vývojáři stále především na ANSI SQL-92, která byla, jak její název napovídá definována roku 1992 (Groff, Weinberg, 2005). Jednotlivé příkazy jazyka SQL je možné rozdělit do několika samostatných skupin:

DDL (Data Definition Language) - Příkazy umožňující definovat datové schéma. Mezi základní příkazy DDL patří:

- CREATE (Vytvoří požadovaný typ objektu)
- ALTER (Umožňuje provést změnu struktury existujícího objektu)
- DROP (Odstraní určený objekt z databáze)

DML (Data Manipulation Language) - Obsahuje set příkazů pro práci s uloženými daty. Mezi základní příkazy DML patří:

- SELECT (Vrací data dle zadaných kritérií)
- INSERT (Vkládá nové záznamy do určené tabulky)
- UPDATE (Provede aktualizaci určených záznamů)
- DELETE (Odstraní příkazem specifikovaná data)

DCL (Data Control Language) - Příkazy pro řízení přístupových práv k jednotlivým objektům databáze. Mezi příkazy DCL patří:

- GRANT (Přiznává uvedená oprávnění určenému uživateli)
- REVOKE (Odebírá uživateli uvedená oprávnění)

TCL (Transaction Control Language) - Příkazy pro řízení transakcí, nebo li sledu operací sdružených do logických bloků. Základní TCL příkazy jsou:

- COMMIT (Provede uložení v rámci transakce provedených změn)
- ROLLBACK (Odvolává transakci a vrací dotčená data do původního stavu)

Implementace databázových transakcí (Loney, Bryla, 2006, s. 260-261) by u většiny současných databázových strojů měla splňovat následující podmínky, které jsou někdy souhrně označovány jako *ACID*:

- *Atomicity* (atomicitu) - Všechny operace vkládání, aktualizace nebo mazání jsou v rámci transakce zpracovány jako jediný příkaz.
- *Consistency* (konzistentnost) - Všechny změny prováděné v transakci jsou úspěšně dokončeny, nebo jsou odvolány a data jsou uvedena do původního stavu
- *Isolation* (izolovanost) - Současně prováděné transakce navzájem neovlivňují.
- *Durability* (trvalost) - Garantuje, že konzistence dat nebude porušena ani v případě havárie v průběhu transakce.

Statické a dynamické použití jazyka SQL

Přestože SQL není plnohodnotným programovacím jazykem (Groff,Weinberg, 2005), ale specializovaným podjazykem, lze jej používat jako součást jiných programovacích jazyků (C/C++, .NET, Pascal, Java a jiné). Z tohoto pohledu připadají v úvahu dvě možnosti, jak jazyk SQL použít:

Statické použití (přímá invokace) – Jednotlivé SQL příkazy jsou přímo vkládány do kódu, jako by byly součástí hostitelského programovacího jazyka. Za běhu je nelze měnit.
Dynamické použití – V tomto případě je SQL příkaz sestaven programem až za běhu (při kompilaci tedy ještě není znám). Pro tyto účely nabízí SQL speciální sadu příkazů:

- *PREPARE* (Předá SQL příkaz ve formě řetězce databázovému stroji ke kompilaci, příkaz může obsahovat zástupný znak „?“ pro dodatečné vložení parametrů prostřednictvím hostitelských proměnných)
- *EXECUTE* (Předá DB požadavek ke spuštění příkazu dříve zadanému prostřednictvím příkazu PREPARE)
- *EXECUTE IMMEDIATE* (Provede PREPARE a EXECUTE v jediném kroku)

Současný stav

Servisně Orientovaná Architektura (SOA)

Servisně orientovaná architektura je moderní koncept návrhu informačních systémů, který usnadňuje implementaci změn, snižuje závislost na infrastruktuře a výrazně zjednodušuje integraci s dalšími systémy a aplikacemi. V současné době je SOA využívána právě především pro rozsáhlé integrační projekty, kde nahrazuje dřívější přístupy založené na výměně dat pomocí nejrůznějších adaptérů, či synchronizačních nebo replikačních technologiích.

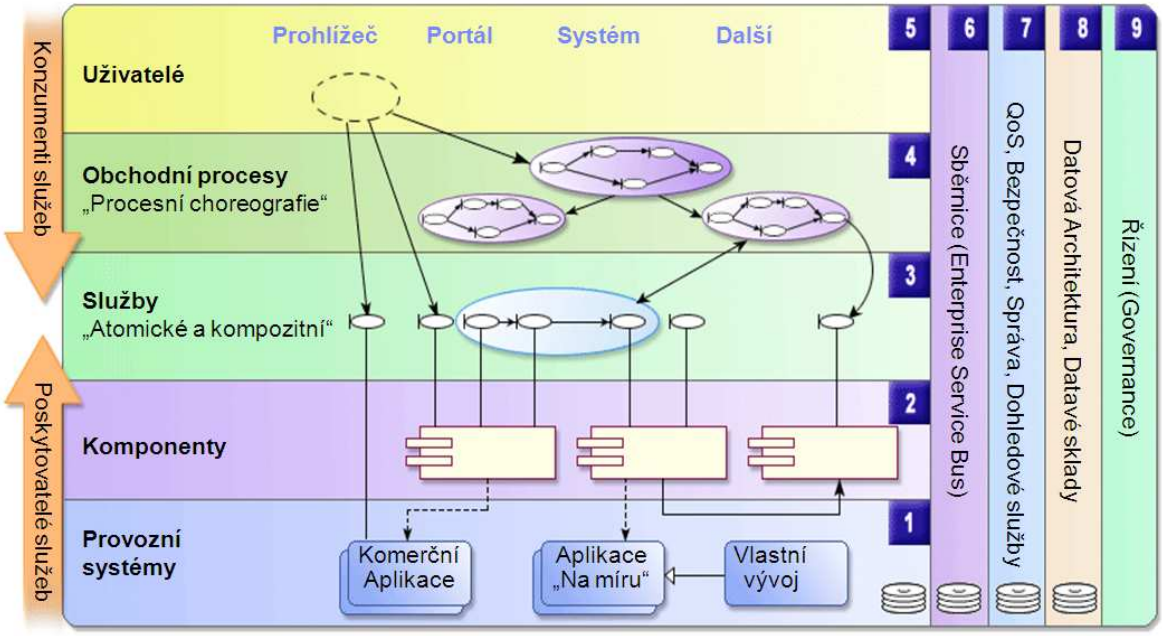
Jak SOA pracuje

Princip servisně orientované architektury (literatura) spočívá v tom, že jednotlivé systémy nezpřístupňují pouze data, ale exaktně popsané funkce (služby), které tak mohou ostatní systémy volat. Tento přístup má následující výhody:

- Ve většině případů není třeba přenášet velké objemy dat. Jejich zpracování proběhne ve zdrojovém systému a konzumentům jsou zasílány již pouze zpracované výsledky.
- Není třeba zásadním způsobem zasahovat do existujících aplikací. Konzumenti služeb jsou naprosto nezávislí na její implementaci.
- Díky možnosti rozmělnit systémy na množství elementární služeb, z kterých lze dle potřeby sestavovat komplexnější aplikace nabízí SOA vysokou míru flexibility.
- Usnadňuje virtualizaci, nasazení aplikací v prostředí "Cloud Computingu" nebo jejich poskytování formou "SaaS".

Na obr. 1 jsou schematicky znázorněny jednotlivé komponenty SOA, jak je definuje IBM. Existující aplikace (1) „vystavují“ své funkce a to buď přímo, nebo ve formě komponent běžících v prostředí aplikačního serveru (2). Služby (3) mohou být prosté (atomické) nebo složené. Ty již mohou být využívány dalšími aplikacemi a systémy. Pokročilejší implementace obsahují takzvanou „procesní choreografii“ (4), která umožňuje měnit chování systému přímo na úrovni procesního modelu a pružně tak reagovat na aktuální potřeby. Skutečným jádrem celého systému bývá společná sběrnice ESB (6), která realizuje komunikaci mezi jednotlivými prvky systému a usnadňuje jejich bezproblémovou integraci. Mezi hlavní úlohy ESB patří směrování a transformace zpráv, zpracování

výjimek, podpora různých protokolů a konektorů pro synchronní i asynchronní komunikaci. V reálném provozu je pak samozřejmě třeba zajistit správu (7) a řízení (9). A nakonec ani SOA se neobejde bez podpory relačních databází a datových skladů (8).



Obrázek 2: Základní prvky Servisně Orientované Architektury (Zdroj: IBM, 2005)

SOA a data

V souvislosti se SOA se nejčastěji hovoří o komunikaci. Hlavním předmětem zájmu bývá společná sběrnice ESB (Enterprise Service Bus), zasílání, transformace a zpracování zpráv. Je však třeba si uvědomit, že tyto zprávy, které řídí chování celého systému, jsou často sestavovány právě z dat uložených v relačních databázových systémech. Jakékoliv problémy s konzistencí nebo kvalitou dat ve zdrojových databázích tak mohou být pro jinak bezchybně navržený systém fatální.

Datové sklady a BI

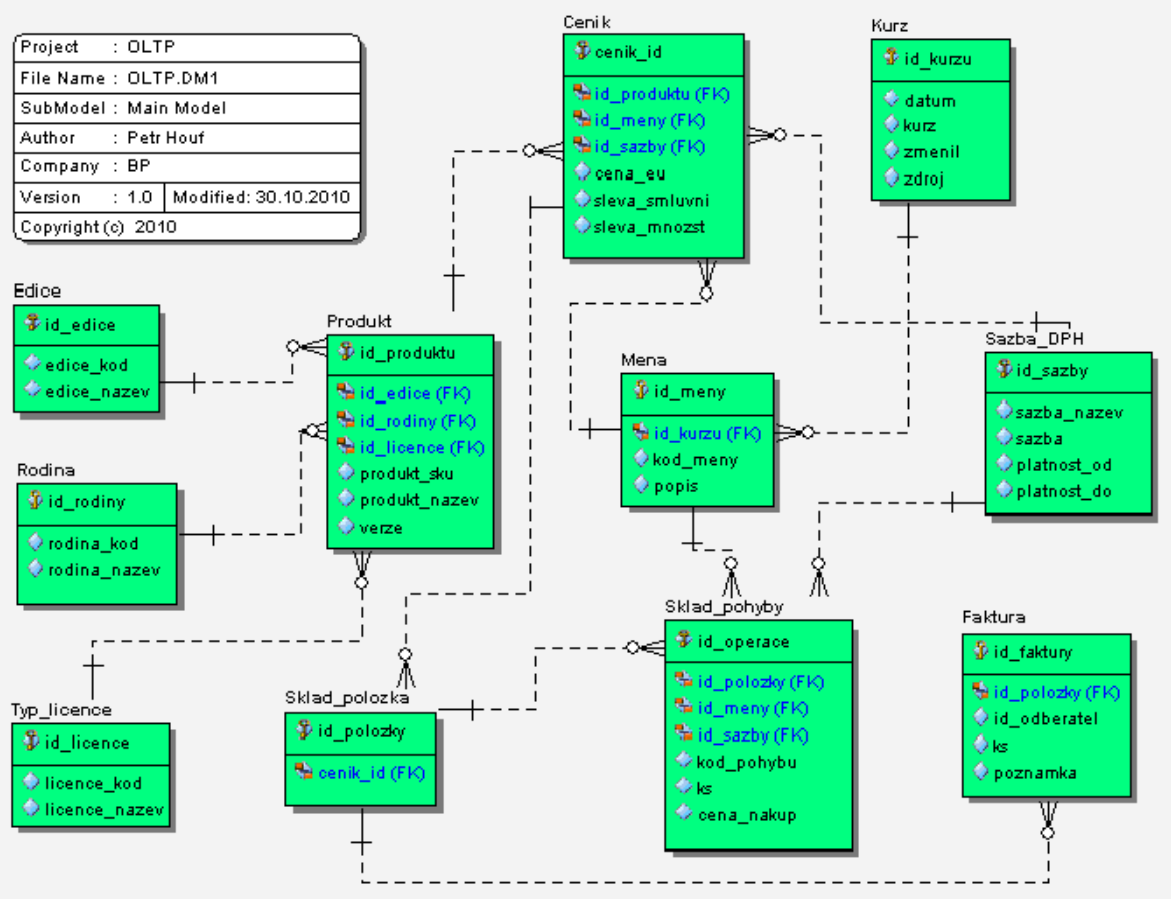
Od doby, kdy se počítače staly běžnou součástí našeho života, se zároveň počala hromadit data. Logicky se proto časem objevil nápad tato nashromážděná data použít pro analýzu a získání kvalitativně nových informací. Možnost opřít se při strategickém či koncepčním rozhodování o historická data se ukázala být významnou konkurenční výhodou. Ačkoli se zdánlivě nejedná o nic jiného než o prosté uložení historických dat do vyhrazené databáze a jejich následné zpřístupnění pro zpracování různými analytickými nástroji, je ve skutečnosti třeba řešit mnoho technických problémů souvisejících s datovou základnou.

Technologie

Datové sklady obvykle uchovávají mnohonásobně větší objemy dat, než je tomu u provozních systémů. Cílem mnohdy velmi komplikovaných dotazů nad mnoha miliony záznamů však zpravidla bývá zkoumání jediné nebo velmi omezeného počtu veličin. Stále více se proto v oblasti datových skladů prosazují specializované databázové stroje umožňující takzvané "sloupcové" zpracování dat (literatura), které přináší až řádové zrychlení. To však znamená, že je třeba počítat s rozdíly (např. datové typy, SQL dialekt a podobně) danými použitím různých technologických platforem u zdrojových systémů a datového skladu.

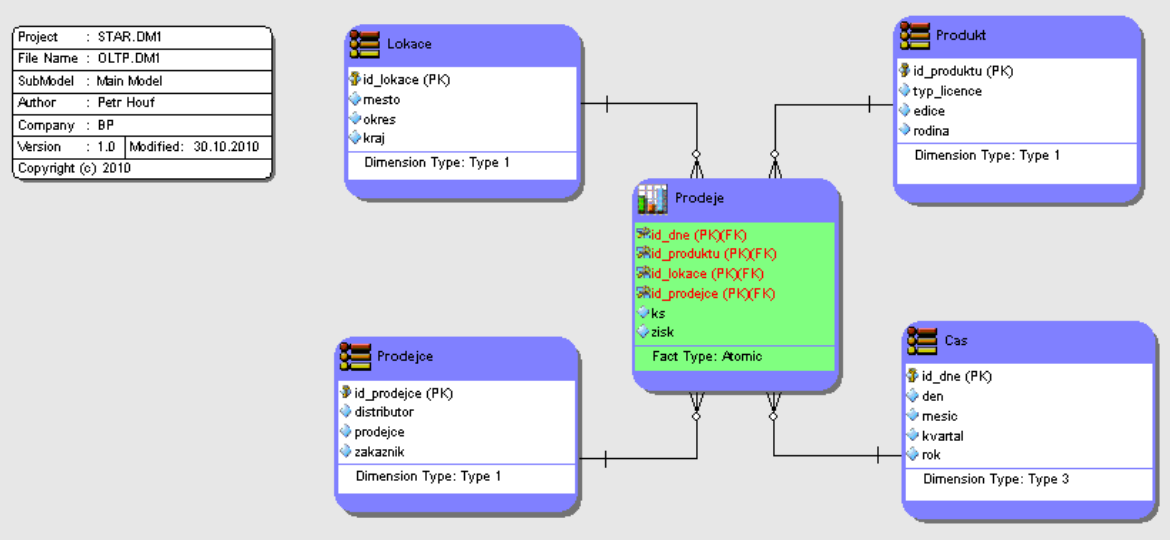
Datové schéma

Struktura dat používaná běžnými provozními systémy je optimalizovaná především pro zabezpečení nekonfliktního zpracování vysokého počtu obchodních operací pro často velké množství současně pracujících uživatelů. Protože toto zajišťujeme za pomoci transakcí, označujeme zpravidla tento druh zpracování jako On-line Transaction Processing (OLTP).



Obrázek 3: Příklad datového modelu části OLTP systému (Zdroj: Vlastní)

Pro OLTP (viz Obr. 2) je typická dobře normalizované datové schéma minimalizující redundanci zpracovávaných dat. Naproti tomu datový sklad bývá optimalizován především pro "dotazování" nad velkými objemy dat. Co se týče redundance, ta je zde naopak nutná, protože jinak by docházelo ke ztrátě historie. Datový sklad tak vyžaduje odlišnou, denormalizovanou datovou strukturu jakou poskytují například hvězdicová nebo vločková schémata (literatura) používaná při více-dimenzionálním modelování (viz Obr. 3).



Obrázek 4: Příklad více-dimenzionálního modelu (Zdroj: Vlastní)

Data

Pro plnění datových skladů nebo datových tržišť je většinou použit více než jeden zdrojový systém. To sebou přináší hned několik úskalí:

- Jednotlivé systémy se mohou lišit technicky a je tak třeba zajistit například konverze kódových stránek nebo datových typů.
- Zdrojové systémy poskytují informace v různých měrných jednotkách, odlišné kvalitě či granularitě
- Určité oblasti dat jsou zachyceny duplicitně ve více systémech (například adresy zákazníků nalezneme jak v CRM, tak v účetnictví).

Význam relačních DB v moderních systémech

SOA a BI nejsou jen typickými zástupci aktuálních trendů, ale zároveň ukazují dva charakteristické pohledy na dnešní využití relačních databázových systémů. SOA (a další vícevrstvé architektury) přesunují datovou aplikační logiku z databáze na úroveň aplikačních serverů. Aplikace je tak méně závislá na konkrétním databázovém stroji, který

následně plní jen roli datového úložiště. Více než rozsah poskytovaných funkcí (Java v databázi, podpora zpracování XML či integrace webových služeb) očekávají uživatelé co nejvyšší výkon při správě velkých objemů dat a minimální nároky na údržbu. Stále častěji zvažují podniky také použití specializovaných databází pro warehousing a využití technologie sloupcového ukládání dat.

S klesajícím zájmem o databáze roste na straně druhé význam samotných dat, protože převážná část současných systémů, včetně těch založených na SOA je řízena právě daty. Rostoucí popularitu datových skladů můžeme naopak vnímat jako důkaz toho, že si stále více uvědomujeme cenu, kterou mají informace ukryté v hromadících se provozních datech.

Master Data Management

Na počátku byla výpočetní technika (reprezentovaná sálovými počítači) využívána téměř výhradně k hromadnému zpracování dat. Sama data byla jen jakousi vstupní surovinou, která po provedení definovaných operací pozbyla jakýkoliv další smysl, a nebylo třeba ji dále uchovávat. Dnes se již s takovým přístupem setkáme jen zřídka. Jak je zřejmé z aktuálních trendů popisovaných v předchozí kapitole, mimo využití dat jako jednorázového vstupu mají dnes data význam jako zdroj pro nejrůznější analýzy a v řadě systémů vystupují zároveň jako řídící prvek.

Data jako zdroj informací

Významný posun přineslo rozšíření výpočetní techniky do podnikové sféry. Podniky záhy zjistily, že provozní data (včetně těch historických) mohou být zdrojem důležitých informací a tím i konkurenční výhody. Začaly vznikat specializované aplikace a systémy zaměřené na určitý segment dat a jejich maximální zhodnocení (EIS, CRM, CMS a řada dalších).

Data jako řídící prvek

V posledních letech získávají data ještě další význam a to řídící. S cílem minimalizovat své náklady či poskytnout služby co nejširšímu okruhu zákazníků, snaží se podniky automatizovat co nejvíce firemních procesů a agend. Systém není řízen uživatelem, ale rozhoduje se na základě vstupních dat a stanovených pravidel.

Data a informační systémy

Současné informační systémy využívají zpravidla data obou výše uvedených typů a jsou tak vysoce závislé na jejich kvalitě. Většina z nás již někdy zažila situaci, kdy jsme na opakovaný dotaz obdrželi rozdílnou odpověď, určitá informace nám byla předána vícekrát, nebo naopak vůbec. Ať se již jednalo o cenu poptávaného výrobku, pozvánku na společenskou akci, dotazník nebo výzvu k úhradě nedoplatku za elektřinu, byla pravděpodobně na vině redundantní nebo naopak neúplná či chybějící data.

Pokud jsou data spravovaná jediným systémem, jedná se zpravidla o chyby způsobené během jejich pořizování. V takovém případě většinou postačí data jednorázově „vyčistit“ a vhodným způsobem nastavit validační pravidla pro vstup nových dat.

I přes existenci komplexních podnikových balíků bychom však jen obtížně hledali větší společnost nebo organizaci, která je schopna své potřeby pokrýt jediným systémem. Příkladem může být budování zmiňovaných datových skladů, integrace se systémy dodavatelů, odběratelů nebo společností získaných v rámci akvizice. S prosazováním principů a technologií jako jsou datové sklady, SOA nebo nově Cloud Computing se ukázalo, že bez kvalitní datové základny nebude jakkoliv dobře navržený systém nebo aplikace poskytovat uživatelům hodnověrné údaje. V reakci na tyto problémy byly hledány cesty a postupy jak o datovou základnu pečovat, aby k těmto neduhům nedocházelo. Brzy se začalo hovořit o Enterprise Data Managementu (EDM) a Master Data Managementu (MDM). Zatímco EDM se zabývá řízením životního cyklu všech podnikových dat, MDM se soustředí právě na problémy související se zpracováním dat uložených ve více zdrojových systémech.

Spravují li systémy informace o stejných klíčových subjektech, a tvoří li jejich průnik neprázdnou množinu, je zde velké riziko chyb. Je li množství těchto chyb významné co do počtu nebo závažnosti, a působí li podniku ať již přímé nebo nepřímé finanční škody, může být právě MDM vhodným řešením. V případě, že má kvalita dat dopad na analytické výstupy hovoříme o analytickém MDM, pokud se jedná chyby v datech ovlivňujících provozní agendy, hovoříme o provozním MDM.

Analytický MDM

Analytický MDM je zaměřeno na oblast datových skladů a BI (Business Intelligence). MDM zde řeší typický problém takzvané „jediné verze pravdy“. Datové sklady shromažďují data z nejrůznějších systémů pro účely jejich analytického zpracování. Takto získané informace pomáhají podnikům například při optimalizaci cenové politiky, přizpůsobení výrobního programu požadavkům zákazníků nebo podobných strategických rozhodnutích. Pokud by se však výstupy z datového skladu významně lišily od informací poskytovaných jednotlivými zdrojovými systémy, budou pro výše uvedené účely nepoužitelné a investice do zavedení BI přišly nazmar.

Operativní MDM

Podobně jako Analytické MDM řeší unikátnost a kvalitu dat. U datového skladu však nemusí být zpravidla data vždy stoprocentně aktuální a je akceptovatelná dávková aktualizace (např. jednou za hodinu, nebo jednou denně). Naproti tomu u provozních

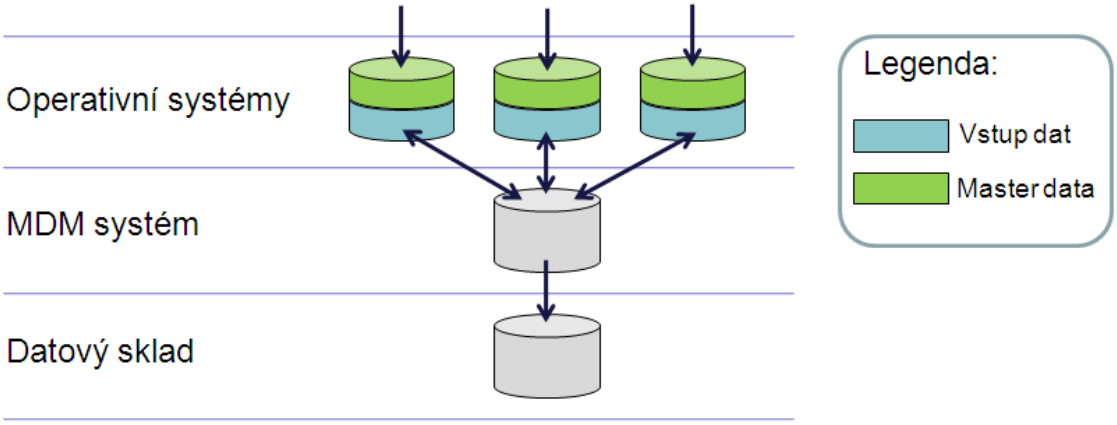
systémů může mít zpoždění propagace ověřených a vyčištěných dat zpět do jednotlivých datových zdrojů vážné důsledky na právě probíhající transakce. Proto u provozního MDM vystupuje do popředí právě problém dostupnosti a latence „Master Dat“. Nejčastějším řešením je vytvoření takzvaného MDM Hubu, který má poskytování (někdy též označované jako federaci) Master Dat na starosti. Z pohledu technické realizace se federace MD opírá většinou o použití webových služeb.

Způsoby implementace MDM

Možností a způsobů, jakým způsobem MDM ve společnosti zavést je samozřejmě více (White, 2007, s. 7). Nejčastěji se hovoří MDM registry, MDM Repository nebo takzvaném hybridním řešení.

MDM jako číselník (Registry Style)

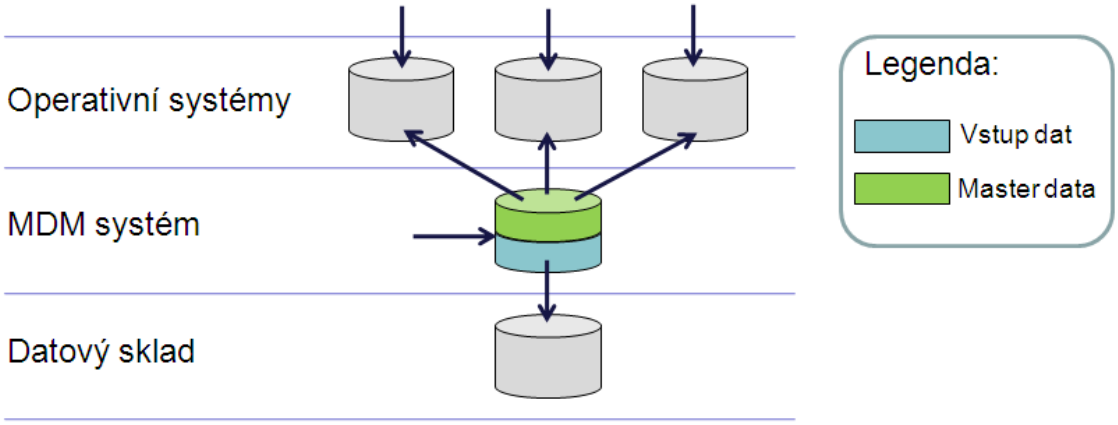
V situaci, kdy jsme pro každou oblast Master Dat schopni využít jako důvěryhodný zdroj některý ze stávajících systémů, není třeba zbytečně udržovat jejich další kopii. V tomto případě MD databáze neobsahuje vlastní data, ale pouze reference na příslušná Master Data, pomocí kterých se na ně následně odkazují ostatní systémy.



Obrázek 5: Realizace MDM jako registru odkazů (Zdroj: White, 2007)

MDM jako centrální DB (Repository Style)

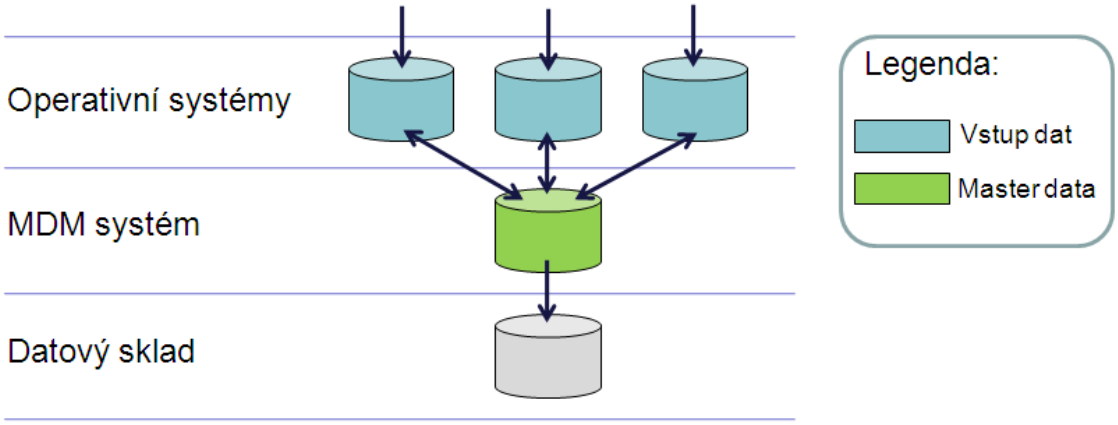
V tomto případě jsou MasterData udržována v centrálním repository a ostatní systémy na ně pouze odkazují. Master Data Repository tak pro jednotlivé systémy plní roli jakéhosi centrálního číselníku. Zdrojové systémy se na centrálně uložená referenční data pouze odkazují.



Obrázek 6: Realizace MDM jako centrálního repository (Zdroj: White, 2007)

Kombinované řešení (Hybrid Style)

V reálné praxi není často z technických nebo organizačních důvodů použít ani jedno z výše uvedených řešení. Hybridní MDM je kompromisem mezi oběma přístupy. Data jsou udržována jak ve zdrojových systémech, tak v Master Data Repository.



Obrázek 7: MDM jako hybridní řešení (Zdroj: White, 2007)

Identifikace Master Dat

Z výše uvedeného vyplývá, že MDM tedy nezavádíme samoúčelně, ale v reakci na konkrétní typ identifikovaných obchodních problémů spojených s kvalitou a dostupností dat, která jsou klíčová pro fungování a řízení podniku. Za určující symptomy vedoucí k úvahám o implementaci MDM lze tedy považovat:

- Použitelnost výstupů je degradována vinou nekvalitních dat
- Data jsou neúplná a znemožňují tak jednoznačně identifikovat sledovaný subjekt

- Jedná se o data, která jsou klíčová pro hlavní aktivity společnosti
- Zdrojová data jsou uložena ve dvou a více systémech
- Informace a jejich struktura ve zdrojových systémech nejsou disjunktní

Master Data

Slovo "Master" má v angličtině celou řadu významů (mistr, pán, řídící, ovládající, originál a mnoho dalších). Ve většině případů se proto bude jednat o „páteřní“ data, která na základní úrovni popisují určitý objekt našeho zájmu jako je třeba firma, osoba nebo výrobek. Na tato data jsou pak napojeny další rozšiřující údaje. Vzhledem k jejich významu není neobvyklé, když taková data figurují ve více systémech. Hojně citovaným příkladem dat, která jsou pořizována, udržována a zpracovávána souběžně ve více systémech jsou osoby. Pomineme-li náklady spojené s pořizováním dat, potřebným úložným prostorem a podobně, zvyšuje se tím rovněž riziko zavlečení nejrůznějších chyb. Dochází k nekorektnímu párování, vzniku duplicit a dalším problémům. Se zvyšující se složitostí systémů rostou rovněž škody, které může tento typ chyb způsobit. Vrátíme-li se ke zmiňovaným osobám, zjistíme, že se jedná o velmi důležitý vazební prvek, prostřednictvím kterého lze skloubit informace z řady zdrojů a cíleněji tak například oslovovat klienty. Pokud mají informace o osobách hrát roli master dat, je třeba aby:

- existovaly v jediné platné verzi
- byly uloženy tak, aby je mohli sdílet všichni uživatelé, aplikace a systémy
- musí vykazovat maximální kvalitu

Proces implementace MDM

Vlastní zavedení Master Data Managementu ve společnosti není jednorázovým úkolem, ale jedná se spíše o trvalý proces, který lze rozdělit do několika základních fází:

Analýza

Analýza je iterativním procesem, který probíhá na několika úrovních. Cílem analýzy je zachytit a pečlivě prozkoumat všechna fakta potřebná pro realizaci definovaných požadavků (Conolly, 2009, 125). Jak již bylo uvedeno, zahájení implementace MDM by mělo být iniciováno obchodními potřebami. Shromážděné požadavky uživatelů je třeba podrobit důkladnému rozboru, jehož cílem je jednak porozumět zadání a následně navrhnout vhodný způsob jeho realizace. V průběhu návrhu a implementace navrženého

řešení je třeba zajistit zpětnou vazbu. Takto získané informace se opět analyzují a jsou použity ke korekci stávajícího zadání. Práce analytiků nekončí ani po zprovoznění MDM. Systém musí obsahovat dostatečné mechanismy sledování, které budou poskytovat dostatek podkladů pro vyhodnocení dosažených výsledků. Ty pak spolu s novými požadavky, které přináší zkušenosti uživatelů se zavedeným systémem, poslouží k dalšímu rozvoji a zkvalitňování MDM.

Hlavním zdrojem pro analýzu aktuálního stavu jsou metadata. Z pohledu MDM je třeba realizovat následující kroky (Loshin, 2009, 137):

- Posbírat maximální množství metadat popisujících ty části jednotlivých systémů, které obsahují plánované zdroje pro získání Master Dat
- Vyřešit případné konflikty ve struktuře dat a identifikovat potřebné úpravy
- Navrhnout vhodný společný model, který sjednotí master data jak po stránce jejich reprezentace tak sémantiky tak, aby při jejich konsolidaci nedocházelo k žádným kolizím

Metadata

Metadata jsou informace o struktuře a vlastnostech dat v systému, aplikaci nebo třeba konkrétním souboru. V literatuře (Minoli, 2008, 104) bývá často používána jednoduchá definice „Data o datech“. Metadata nám umožňují jimi popsaná data správně zpracovávat a interpretovat.

Metadata je samozřejmě možné vytvářet a spravovat jakýmkoliv způsobem. S rostoucí složitostí systémů se však rychle ukázalo jako praktické rozšířit metadata o vizuální složku (diagram) a metadata udržovat ve formě modelů. Z hlediska databází je nejdůležitějším typem modelu datový ER model. Při návrhu, implementaci a řízení informačních systémů je však využíváno mnoho dalších modelů (procesní, datových toků, UML). Metodikou použití jednotlivých typů modelů se zabývají MDA (Modelem řízené architektury) z nichž asi nejznámější jsou Zachman Framework (Minoli, 2008, 58) nebo IBM RUP.

Pro podporu MDM lze využít několik typů modelů. V převážné většině se jedná o modely datové, ale některé důležité informace lze rovněž získat například z procesního modelu.

Nejčastěji používané typy modelů

Procesní model – Procesní model se typicky využívá k lepšímu pochopení chodu společnosti a jeho zákonitostí. Lze jej zpracovávat na různých úrovních podrobnosti, takže

se lze setkat s procesními modely, které mimo sledu jednotlivých činností popisují i toky dat. Většina dnešních CASE nástrojů umožňuje na základě zachycených informací vygenerování takzvané CRUD matice. Ta přehlednou formou ukazuje, kdo jaká data vytváří (C - Create), čte (R - Read), modifikuje (U - Update) nebo maže (D - Delete).

S využitím takto získaných informací lze například vhodně nastavit přístupová práva a bezpečnostní politiku obecně.

	Zakázka	Firma	Kategorie	Forma	Osoba	Ceník	Produkt	Typ	Verze	Skupina	Okec
Prodej											
Vytvoření zakázky	C, R										
Vložení položky	C,R,U					C,R,U	C,R,U	C,R,U	C,R,U	C,R,U	
Odebrání položky	R,U,D					R,U,D	R,U,D	R,U,D	R,U,D	R,U,D	
Založení nového zákazníka	R,U	C,R,U	C,R,U	C,R,U	C,R,U						C,R,U
Vložení zákazníka	R,U	R,U	R,U	R,U	R,U						R,U
Odebrání zákazníka	R,U,D	R,U,D	R,U,D	R,U,D	R,U,D						R,U,D
Zrušení zakázky	R,U,D										
Nákup											
Vytvoření objednávky		C,R,U	C,R,U	C,R,U	C,R,U						C,R,U
Vložení položky						C,R,U	C,R,U	C,R,U	C,R,U	C,R,U	

Obrázek 8: Příklad jednoduché CRUD matice (Zdroj: Vlastní)

Konceptuální datový model – popisuje strukturu dat nezávisle na následně zvolené technologii (databázový stroj, strukturovaný soubor, …). Protože se jedná o obecný model, je s výhodou používán v situacích, kdy způsob konkrétní implementace není ještě znám, nebo tam, kde má navrhovaný systém nabízet podporu více platforem.

Logický datový model – tvoří určitou mezivrstvu mezi konceptuálním a fyzickým datovým modelem. Přesto, že si zachovává nezávislost na konkrétní implementaci (fyzickém uložení dat), není již nezávislý technologicky. Jinak tedy bude vypadat logický datový model pro XML a jinak pro uložení dat v databázi.

Fyzický datový model – fyzický datový model rozšiřuje model logický o fyzickou implementaci. Do fyzického modelu se tak například promítají konkrétní vlastnosti zvoleného RDBMS a cílem bude co nejlepší optimalizace modelu právě a jen pro něj.

Model datových toků – hlavním cílem modelu je zachycení transformací dat mezi systémy. Tyto informace jsou neocenitelné například pro dopadovou analýzu při nutnosti změn ve zdrojových nebo cílových systémech a aplikacích.

Profilování dat (Data profiling)

Kromě metadat získaných z existujících modelů nebo zpětným inženýringem je dalším velmi důležitým zdrojem informací profilování dat. Charakteristiky, které lze s využitím profilování získat, mimo jiné umožňují:

- Exaktně posoudit existující kvalitu dat a předem odhadnout s ní spojené potenciální problémy, které by mohly zavedení MDM komplikovat
- Jednodušeji a rychleji identifikovat problematická data
- Sledovat, kontrolovat a kontinuálně zlepšovat procesy a definovaná pravidla týkající se pořizování a zpracování dat
- Rozkrývat a zdokumentovat podrobné informace o datových elementech a jejich vzájemných vztazích

Výstupem profilování dat je celá řada statistických charakteristik, z nichž ty nejčastější uvádí tabulka 1. Data získaná pomocí profilování lze dále využít pro specifické typy analýz. Patří sem například doménové analýzy nebo analýzy primárních a cizích klíčů.

- *Doménová analýza* - Pro domény, které definují výčet platných hodnot, je možné využít takzvanou doménovou analýzu. Ta typicky prověřuje, zda se ve sloupci nevyskytují jiné než definované povolené hodnoty. Zároveň poskytuje základní informace o jejich distribuci, tedy počtu výskytů jednotlivých hodnot v rámci analyzovaného sloupce
- *Analýza primárních klíčů (PK)* - Primární klíč (Primary Key) poskytuje unikátní odkaz na jeden konkrétní řádek v DB tabulce. Cílem PK analýzy je nalezení všech sloupců, které by mohly být s vysokou mírou pravděpodobnosti použity jako primární klíč (takzvaných PK kandidátů).
- *Analýza cizích klíčů (FK)* - Cizí klíč (Foreign Key) je integritním omezením, které garantuje, že do daného sloupce tabulky může být vložena výhradně hodnota primárního klíče jiné určené tabulky. FK analýza vyhledává potenciální kandidáty na FK a prověřuje je z hlediska referenční integrity.
- *Křížová doménová analýza* - Tento typ analýzy identifikuje překryvy či redundance dat v rámci jednoho nebo více sloupců v jedné nebo více tabulkách. Vlastní analýza probíhá ve dvou krocích. V prvním jsou vyhledány všechny potenciální páry sloupců, v druhém je pak zevrubně testována jejich vzájemná kompatibilita.

Tabulka 1: Přehled nejběžnějších charakteristik profilování dat

Parametr	Význam
Data type	Název uživatelského datového typu či domény
Native type	Základní datový typ, který byl použit k odvození
Length	Rozsah datového typu (počet znaků/bytů, počet číslic před a za oddělovačem apod.)
NotNull	Určuje, zda pro daný sloupec byla nastavena volba "NotNull" zamezující vložení prázdné hodnoty.
Formát	Způsob zobrazení, například použité oddělovače, způsob seskupování znaků a podobně
Max	Maximální hodnota ve sloupci (závisí na datovém typu a hodnotové stupnici konkrétního DB stroje)
Min	Minimální hodnota ve sloupci (závisí na datovém typu a hodnotové stupnici konkrétního DB stroje)
Mod	Nejčastěji se vyskytující hodnota pro daný sloupec tabulky
Row Count	Počet všech řádků v tabulce
Null Count	Počet řádků, které mají v daném sloupci hodnotu "Null"
Null %	Percentuální podíl "Null" hodnot na celkovém počtu řádků.
Cardinality	Stupeň unikátnosti. Určuje se jako podíl počtu různých hodnot a počtu všech řádků v tabulce.
Selectivity	Podíl počtu všech řádků v tabulce a počtu unikátních hodnot.

Extrakce a konsolidace dat

Po získání všech dostupných informací o datových strukturách, které mají plnit úlohu vzorových (master) dat můžeme přistoupit k jejich konsolidaci. Pro konsolidovaná data můžeme vytvořit nový dedikovaný systém, ponechat je v původních zdrojových systémech, nebo pro jejich uložení můžeme zvolit některé ze stávajících datových úložišť, typicky to, jehož data vykazují nejvyšší kvalitu.

Struktura MDM úložiště

Struktura sloučeného záznamu je typicky sjednocením atributů udržovaných ve zdrojových systémech. Nejedná se pouze o výčet atributů, ale například i jejich datové typy, granularitu a význam. Jde v zásadě o to, aby při přenosu dat ze zdrojových systémů nedocházelo k jejich poškození či dezinterpretaci (Loshin, 2009, 141). Pozornost je třeba věnovat především:

- *Datovým typům* – Datový typ konsolidovaného úložiště musí poskytovat dostatečný rozsah pro uložení dat ze všech plánovaných zdrojů. Například u znakových položek by jinak mohlo dojít k nechtěnému „ořezu“ vkládaných dat.
- *Omezením datového modelu* – Pokud je struktura dat (datový model) vycházející z jednotlivých zdrojových systémů nedostatečná a neumožňuje tak ukládat vhodným způsobem požadovaná data, je třeba model rozšířit. V rámci analýzy můžeme například identifikovat problém s ukládáním informací o cizincích, kteří mimo u nás běžně užívaného jména a příjmení mají ještě jméno střední. Tento atribut je tedy třeba do stávajícího modelu přidat.
- *Uloženým jednotkám* – Některé informace mohou být v jednotlivých zdrojích uloženy v agregované podobě nebo v jiných měrných jednotkách. Například jedna z charakteristik výrobku může být objem. Jeden systém jej bude uvádět v litrech, druhý pak v kubických centimetrech. V případě, že by konsolidovaný systém převzal taková data bez vhodné transformace, byla by výsledná data výrazně zkreslená.
- *Kódům a návěštím* – V centrálním úložišti je třeba zajistit správnou interpretaci kódů a návěští zdrojových systémů. Ty reprezentují určitou hodnotu, která však nemusí být pro všechny systémy shodná. Například číslo 17 označuje ve dvou zdrojových systémech barvu. V jednom je má význam „Zelená“ a v druhém „Modrá“. Naproti tomu různé hodnoty návěští (flagu) „1“, „Y“, „T“ nebo „A“ označují stejnou hodnotu typu „boolean“.
- *Formátům dat* – Pro konsolidovaná data je také třeba zvolit vhodný formát. Typickým příkladem mohou být položky pro ukládání datumových hodnot. Datum 12-01-97 může v jednom systému reprezentovat 12. ledna 1997 (dd-mm-rr), v druhém ale 1. prosince (mm-dd-rr).
- *Granularitě* – Pro atributy, které jsou v různých zdrojích uloženy v různé úrovni detailu, je třeba provést refaktoring a najít jednotný způsob jejich zachycení, splňující požadavky všech dotčených systémů. Typickým příkladem může být ulice a číslo domu. V řadě aplikací se jedná o jedinou alfanumerickou položku. Pro společnost, která se zabývá doručováním zásilek, může být vhodnější rozčlenění na samostatné položky pro „název ulice“, „orientační číslo“ a „číslo popisné“.

Výsledná struktura sloučeného záznamu může být buď jednoduchá (plochá), nebo hierarchická. Hierarchickou strukturu bychom použili například pro produkty, které se prodávají jednak samostatně, ale zároveň také v sadách složených z několika produktů. Pro vlastní konsolidaci dat je třeba zvolit vhodnou technologii. Rozhodující jsou v tomto směru především požadavky na latenci dat v konsolidovaném zdroji. Pro účely analytického MDM může být dostačující dávková konsolidace, v některých případech pak dokonce jednorázová.

sloučený záznam	titul_před	jméno	střední	příjmení	rodné	titul_za	datum_narození	pohlaví	rodné_č
systém A	titul	jméno		příjmení			datum_narození	pohlaví	
systém B	tituly	jméno	střední	příjmení	rodné				rodné_č
systém C		jméno		příjmení					rodné_č

Obrázek 9: Princip sloučení záznamů z více systémů do ploché struktury (Zdroj: Vlastní)

U operativního CRM se nejčastěji setkáme s konsolidací průběžnou. V současnosti se v praxi můžeme setkat s několika technologiemi pro extrakci a následnou konsolidaci dat. Typicky se jedná o různé varianty replikací, nasazení specializovaných ETL nástrojů nebo různá vlastní aplikační řešení.

Replikace – Replikace (od replika = zmnožení nebo zdvojení) je technologie umožňující řízenou duplikaci dat. Změny v primární databázi (vložení, aktualizace nebo smazání záznamu) jsou současně propagovány do určených cílových databází. Vlastní replikace může být realizována (Replication Strategies, 2003) na úrovni záznamů databáze (např. dvoufázový commit), transakčních logů (Sybase Replication Server) nebo fyzického úložiště – média (Hewlett-Packard, IBM).

ETL – Zkratkou ETL označujeme specializované serverové nástroje pro datovou integraci, jejichž název je odvozen od tří hlavních činností, které zabezpečují (Lacko, 2003, 60). Jedná se o:

- *Extrakci* – Získání dat z relačních databázových systémů a dalších strukturovaných, ale i nestrukturovaných datových zdrojů za pomoci nejrůznějších ovladačů, adaptérů nebo konektorů.
- *Transformaci* – Proces zajišťující převod původní struktury zdrojových dat do tvaru zvoleného pro konsolidované úložiště. Některé ETL nástroje jsou během procesu transformování dat provádět i další operace jako je například standardizace, čištění či obohacování.

- *Load* – Nahrání (uložení) extrahovaných a transformovaných dat do určeného konsolidovaného úložiště.

Vlastní vývoj – Další možností je vytvoření vlastního řešení (Lacko, 2003, 61) na bázi procedurálních nadstaveb jazyka SQL (Transact-SQL, PL/SQL, …) nebo některého z programovacích jazyků (C/C++, Java, Pascal, …).
Dalším problémem, který je třeba vyřešit, je zachování referencovatelnosti zdrojových dat. Dostupnost informace o tom, z kterého systému data pochází a které systémy je využívají, mají zásadní význam pro korektní řízení životního cyklu master dat. Typicky tak struktura v master data repository obsahuje hned několik unikátních identifikátorů:

- Identifikátor zdrojového systému
- Identifikátor záznamu v rámci zdrojového systému
- Identifikátor záznamu v rámci master data repository

Čištění, deduplikace a obohacování MD

Data získaná konsolidací z jednotlivých zdrojových systémů budou s vysokou pravděpodobností obsahovat nemalé procento duplicitních záznamů a mnoho dalších chyb. Aby mohla být použita jako „vzorová“, je třeba je vyčistit a standardizovat tak, aby odpovídala všem zjištěným požadavkům a pravidlům.

Čištění dat

Čištění dat je proces, během kterého se snažíme data, která jsme během analýzy identifikovali jako chybná, neúplná nebo nesrozumitelná, upravit tak, abychom co nejvíce zvýšili jejich kvalitu a odstranili maximum ze zjištěných nedostatků. Chyby a metody jejich odstranění lze rozdělit do dvou základních kategorií dle původu dat:

- Jedná se o samostatný zdroj dat, chyby jsou zpravidla zaviněny lidským faktorem (uživateli, kteří údaje vkládali).
- Data pochází z více zdrojů. Zdroje, byť kvalitní, popisují stejnou entitu, následkem čehož konsolidovaná data obsahují větší či menší procento duplicit. Jejich odstranění pak může komplikovat fakt, že se jednotlivé zdroje liší například metodikou pořizování dat (vstupní pravidla, jednotky, identifikátory) nebo časem, kdy byla pořízena.

Při zavádění MDM se bohužel typicky setkáváme s kombinací obou uvedených případů, tedy konsolidací dat z dvou a více nekvalitních zdrojů. I v takovém případě je však vhodné rozdělit proces čištění dat do uvedených dvou kroků, tedy provést standardizaci a čištění dat jednotlivých zdrojů a teprve následně jejich sjednocení a případnou deduplikaci. Mezi nejčastější zdroje chyb, na které je třeba se zaměřit, patří například:

- *Data v nesprávných sloupcích* – Může se jednat o položky, pro které není v datové struktuře k dispozici žádný atribut (viz omezený datový model), a tak jej uživatel zapíše do atributu, který je pro daný záznam nevyužitý. Zjednodušeně řečeno, jedná se o správná data, nicméně uvedená ve špatném sloupci.
- *Překlepy* – Tento typ chyb se nejčastěji vyskytuje ve spojitosti s typováním. Často se vyskytují překlepy spojené s použitím diakritiky (např. 8 místo á), přehození znaků při rychlém psaní (sk místo ks) nebo záměna znaků, které jsou na klávesnici vedle sebe.
- *Chyby vzniklé přepisem* - V případě vkládání informací získaných například během telefonátu připadají v úvahu také chyby u položek, jejichž fonetika není shodná s psanou formou.

Deduplikace (Matching)

Po normalizaci a „vyčištění“ jednotlivých zdrojů dat lze přikročit k párování jednotlivých záznamů. Pro párování je vhodné rozdělit atributy do kategorií, které lze nazvat např. „zásadní“, „důležité“ a „podružné“. Zásadní jsou atributy, které jsou pro danou entitu signifikantní. Jedná se zpravidla o v čase neměnné atributy (u osob např. rodné číslo, datum narození, jméno a podobně). Důležitá data se mohou v čase měnit (číslo OP, bydliště, zaměstnavatel), jsou však z hlediska zpracování podstatná. Mezi podružné řadíme atributy, které jsou důležité pouze v rámci konkrétního systému (např. platové ohodnocení v účetním mzdovém systému), ale mají nízký nebo nulový význam pro identifikaci subjektu, v tomto případě osoby. Protože se informace o entitě nachází ve více systémech, může v řadě případů dojít k situaci, kdy atribut není zadán v obou porovnávaných zdrojích, nebo se zadané hodnoty navzájem liší. Je proto vhodné nastavení váhy jednotlivých zdrojů dat. Hlavními faktory by měli být:

- *Relevance dat* – např. systém personalistiky vs. kniha jízd
- *Aktuálnost* – např. průměrná doba od poslední aktualizace

- *Naplnění, úplnost* – percentuální poměr mezi počtem prázdných a vyplněných hodnot

Obohacování

Během procesu čištění a deduplikace dat je zároveň možné rozšířit stávající datovou základnu o data získaná z externích zdrojů. Například můžeme použít seznam poštovních směrovacích čísel tak, jak je zveřejňován Českou poštou. Ten pro každé PSČ uvádí kromě města a městské části rovněž okres a kraj. Rozšířením původního modelu o tyto údaje získáme nový pohled (dimenzi) pro vyhodnocování prodejů.

Poskytování (provisioning, federace) dat

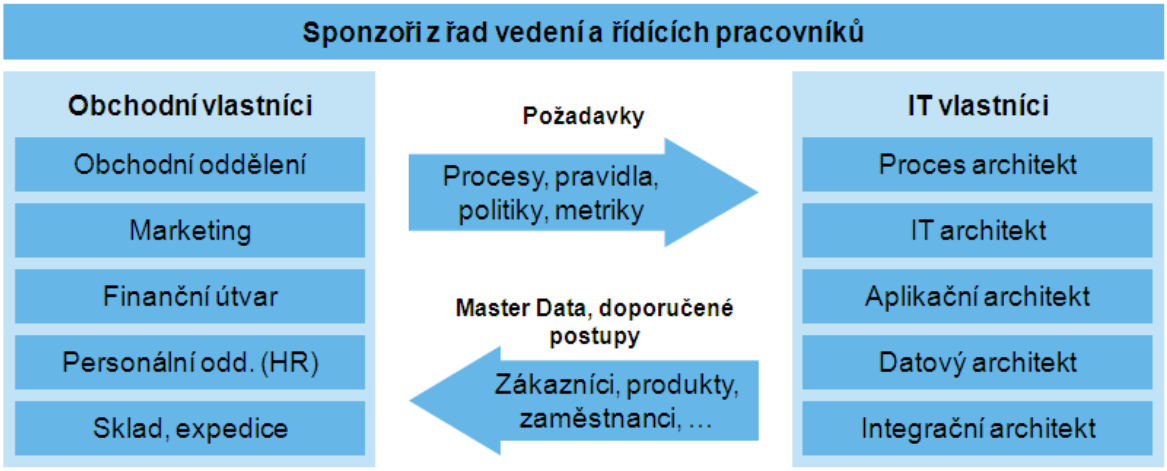
V závislosti na zvoleném typu architektury pro zavedení MDM je třeba kromě konsolidace zajistit rovněž propagaci standardizovaných a vyčištěných dat zpět do provozních systémů. Tento přístup se týká řešení s centrálním repository kmenových dat a hybridních řešení. Technicky je možné poskytování dat realizovat podobnými prostředky jako jejich konsolidaci. V praxi převládají řešení založená na službách. Aplikace nebo specializovaný server (MDM Hub) nabízí sadu funkcí pro práci s kmenovými daty:

- *Add* – Ošetřuje požadavek provozních systémů na vložení nového kmenového záznamu
- *Update* - Ošetřuje požadavek provozních systémů na aktualizaci existujícího kmenového záznamu
- *Delete* - Ošetřuje požadavek provozních systémů na odstranění existujícího kmenového záznamu

Hub může samozřejmě nabízet mnoho dalších funkcí pro správu, sledování a auditing.

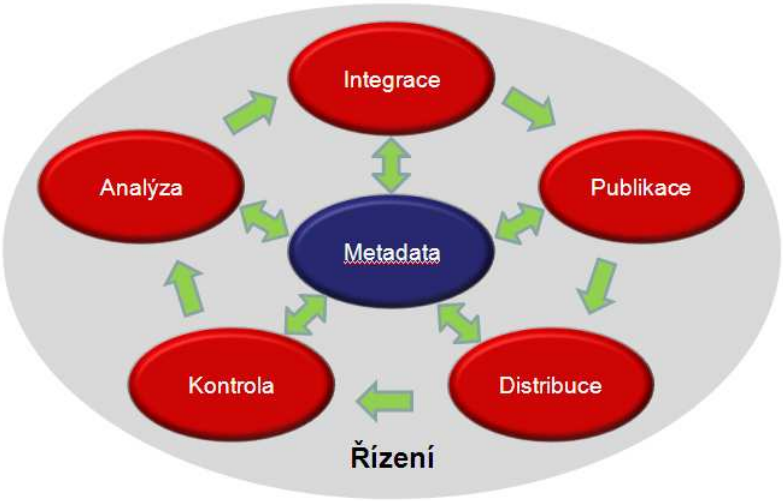
Řízení (Governance)

Aby bylo zavedení MDM v organizaci úspěšné, je nezbytně nutná podpora řídících pracovníků. Motivací pro budování MDM musí být vždy obchodní potřeby, ale vlastní implementace je plně v kompetenci IT pracovníků. Jedná se tedy o rozdílné skupiny pracovníků s často velmi odlišným pohledem na chod organizace, priority ale i používanou terminologií. Bez jasně deklarované podpory managementu a organizačního řízení by tyto rozpory mohly být zásadním problémem. Zcela zásadní je pak zajištění hladké komunikace mezi jednotlivými pracovníky, respektive rolemi, které zastávají.



Obrázek 10: Řízená komunikace mezi věcnými a technickými vlastníky (Zdroj: Loshin)

K MDM by mělo být přistupováno jako k projektu. Měly by tedy být jasně definovány cíle, časový harmonogram, zodpovědnosti a metriky pro průběžné vyhodnocování. Zavádění MDM je trvalým procesem s řadou stupňů zralosti (Loshin, 2009, 55-60), který lze dekomponovat na jednotlivé činnosti.



Obrázek 11: Základní činnosti procesu zavádění MDM (Zdroj: Loshin)

4 Návrh a realizace MDM pro stávající RDBMS

Vzhledem k omezenému rozsahu práce jsou použity pouze dílčí části dotčených systémů. Výběr byl zúžen tak, aby vyhovoval cíli práce, tedy demonstraci principů a technik používaných při implementaci MDM.

Výchozí stav

Ve společnosti jsou používány tři samostatné systémy. Jedná se interní informační systém pro podporu prodeje, jehož hlavním modulem je webový obchod (e-shop) a informační portál pro zákazníky. Druhý systém slouží především technikům pro evidenci a řízení případů technické podpory. Posledním systémem je software určený k vedení účetnictví. Ten je zároveň využíván také k řízení skladové evidence, fakturaci a k udržování informací o dodavatelích a odběratelích. Každá z uvedených aplikací využívá vlastní dedikovaný databázový stroj. Přestože jsou všechny systémy funkční, trápila společnost vysoká míra chyb při zpracování běžných agend, která jednak snižovala produktivitu a navíc poškozovala společnost v očích jejích zákazníků. Primárním cílem bylo sjednotit záznamy o zákaznících, aby bylo možné provádět cílené kampaně, vyhodnocovat reálné zisky očištěné o náklady na čerpanou technickou podporu a podobně. Zvažována byla řada variant integrace. Finálně bylo rozhodnuto implementovat MDM, prozatím v omezeném rozsahu formou pilotního projektu.

Základní analýza

Všechny systémy využívají databázový server Oracle Enterprise 10g. S ohledem na shodu v nastavení nls parametrů (nastavený jazyk, znaková sada, formát ukládání datumových položek) je možné sdílet data bez provádění konverzí, což integraci Master Dat značně zjednodušuje.

Jako zdroj pro získání potřebných informací byly kromě požadavků uživatelů použity datové modely jednotlivých systémů (příloha 1-3). Vybírány byly tabulky, zachycující identifikační informace o společnostech a osobách, které v systémech vystupují v roli zákazníků. Z datových modelů tak byly identifikovány následující objekty/tabulky:

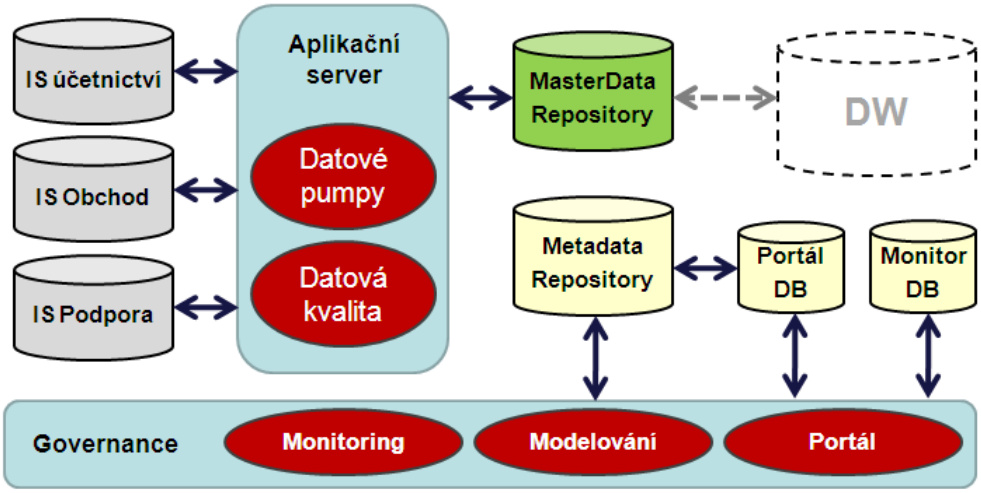
- *Firma* (systém "Support")
- *Kontakt* (systém "Support")
- *DNA* (systém "Sales")
- *Odběratel* (systém "Accounting")

- *Osoba* (systém "Accounting")
- *Adresa* (systém "Accounting")

Navrhované řešení

Pro první fázi projektu bylo rozhodnuto, že se implementace zaměří na oblast informací týkajících se zákazníků. Dva ze systémů jsou udržovány (včetně vývoje) vlastními silami. Třetí systém (účetnictví) byl vyvinut "na míru" externím dodavatelem, v rámci platné technické podpory je však možné nárokovat případné úpravy. Pro realizaci byl zvolen hybridní přístup, který je náročnější na implementaci, ale jeho nasazení předpokládalo nejmenší dopad na stávající aplikace.

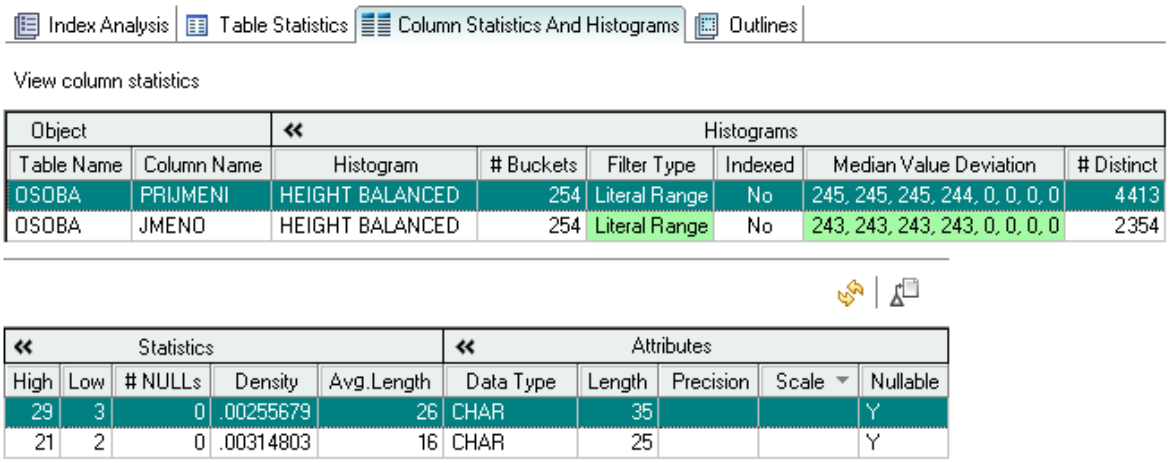
Operativní podnikové systémy „účetnictví“, „obchod“ a „podpora“ byly propojeny aplikačním serverem, který realizuje transport dat (datové pumpy) mezi nimi a úložištěm kmenových dat. Zároveň provádí potřebné transformace a poskytuje funkce pro čištění, deduplikaci a obohacování dat (Datová kvalita). Master Data Repository využívá databázový stroj Oracle a je připraveno na napojení plánovaného datového skladu (DW). Menší dedikované DB Servery slouží pro uložení metadat (Metadata Repository), agregovaných dat pro publikaci metadat (Portál DB) a také nasbíraných charakteristik (Monitor DB). Nad těmito databázemi pracují nástroje PerformanceCenter (Monitoring), ER/Studio (Modelování) a Embarcadero Enterprise Portál, který poskytuje webový přístup k metadatům prostřednictvím předpřipravených i add hoc sestav (portál).



Obrázek 12: Schéma navrženého řešení (Zdroj: Vlastní)

Profilování dat

Sloupce, které byly vybrány jako možní kandidáti pro použití v rámci kmenových dat (Master Data Repository) byly dále analyzovány za pomoci profilování dat. Pro profilování byl použit nástroj DB Optimizer. Získané charakteristiky poskytly základní informace týkající se kvality dat v jednotlivých zdrojových databázích.



Obrázek 13: Výstup profileru pro sloupce „Jméno“ a „Příjmení“ (Zdroj: Vlastní)

Na základě získaných údajů bylo rovněž možné určit relevanci jednotlivých zdrojových systémů. Tyto informace byly zásadní pro návrh skriptů pro čištění a deduplikaci záznamů založených na metodě BoB (nejlepší z nejlepších z anglického „Best of Best“). Cílem této techniky je sestavit každý jednotlivý záznam v Master Data repository z kombinace nejspolehlivějších, nejkvalitnějších, nejaktuálnějších nebo nejčastěji se vyskytujících atributů.

Metadata Repository

Bohužel ani sebelepší návrh nemůže vyloučit potřebu budoucích změn ve struktuře dat. Zvláště v rozsáhlejších systémech může i drobná změna vyvolat dominový efekt. Pro MDM je proto schopnost řídit změny komplexně pro všechny dotčené systémy zásadní. Aby to bylo možné, bylo vytvořeno společné metadata repository. Pro jeho realizaci byl použit nástroj Embarcadero ER/Studio XE. Zároveň bylo rozhodnuto, že metadata repository bude zároveň plnit roli podnikového datového slovníku.

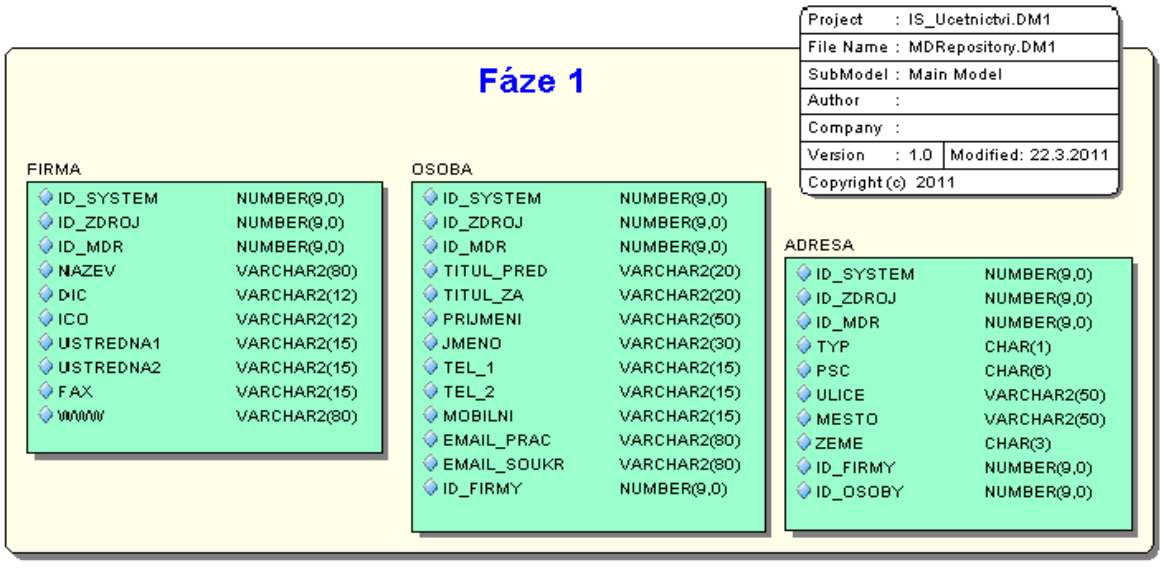
Podnikový datový slovník (Enterprise Data Dictionary)

Jak je patrné z datových modelů (přílohy 1-3), byly shodné informace ukládány pod různými názvy a někdy i v odlišném formátu (datový typ, pravidla, kontroly, …). Naopak

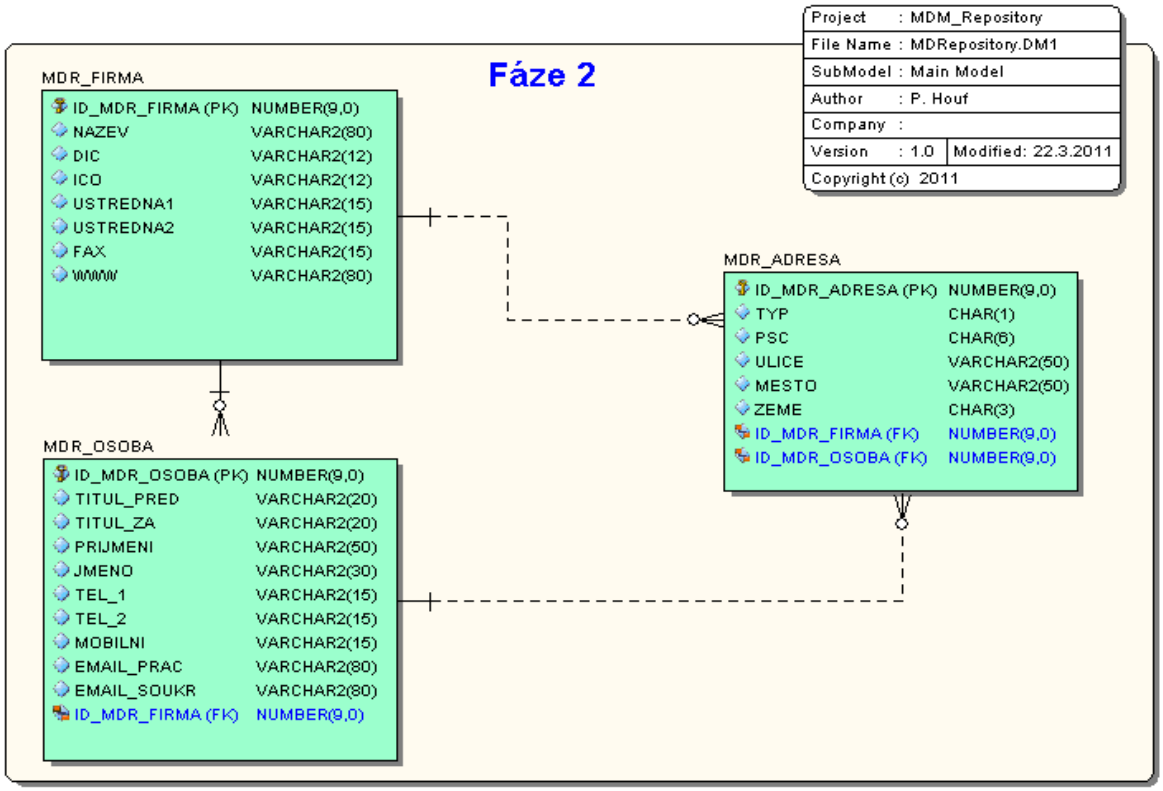
atributy se shodnými jmény se svým obsahem lišily. Tento stav je důsledek odděleného návrhu a vývoje jednotlivých systémů, kdy je každý datový prvek definován znovu a znovu jako unikátní. Nevýhodou takového přístupu je, že sebou kromě vyšších časových nároků přináší i riziko nekonzistence a následně také obtížné udržovatelnosti. Datový slovník definuje katalog předpřipravených atributů, které jsou následně k dispozici pro opakované použití. Základním prvkem datového slovníku je doména (domain). Doména je komplexním popisem daného datového elementu. Doména může být určena jedním nebo více parametry jako jsou například datový typ, rozsah hodnot, pravidlo nebo použití jmenných konvencí. Definované domény lze následně opakovaně používat při tvorbě datového modelu. Datový slovník také udržuje informace o tom, v kterých modelech se jím spravované domény používají.

MasterData Repository (Systém of Record)

Za jednotlivé systémy byli zvoleni datoví stewardi, zodpovědní za návrh zdrojů a pravidel pro řízení master dat a řízení jejich datové kvality. Na základě jejich požadavků a připomínek byla v rámci datového slovníku provedena standardizace datových modelu a byl navržen datový model pro Master Data Repository. Master Data Repository je koncipováno jako dvouúrovňové.

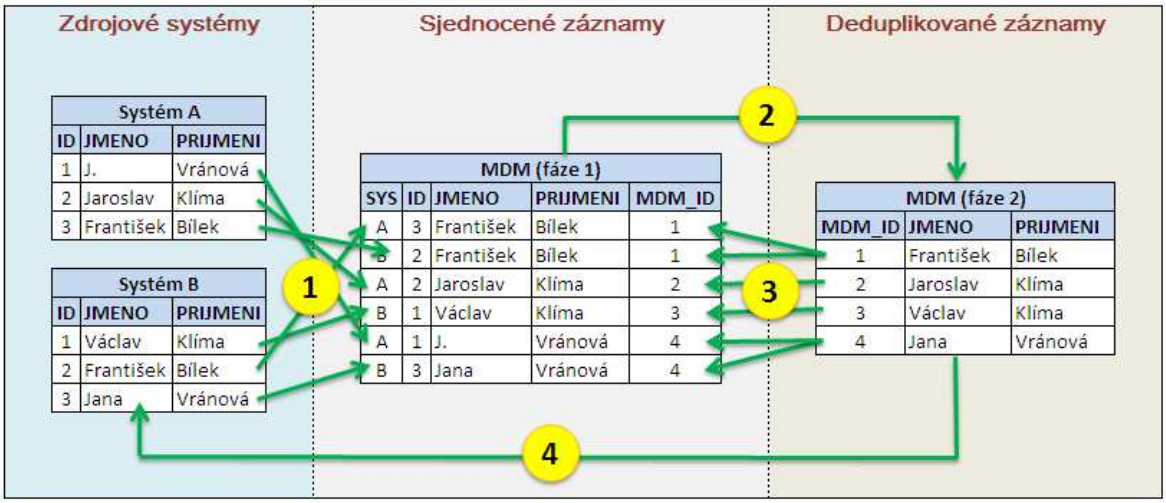


Obrázek 14: Model MDR pro konsolidaci kmenových záznamů (Zdroj: Vlastní)



Obrázek 15: Model MDR pro uložení deduplikovaných kmen. záznamů (Zdroj: Vlastní)

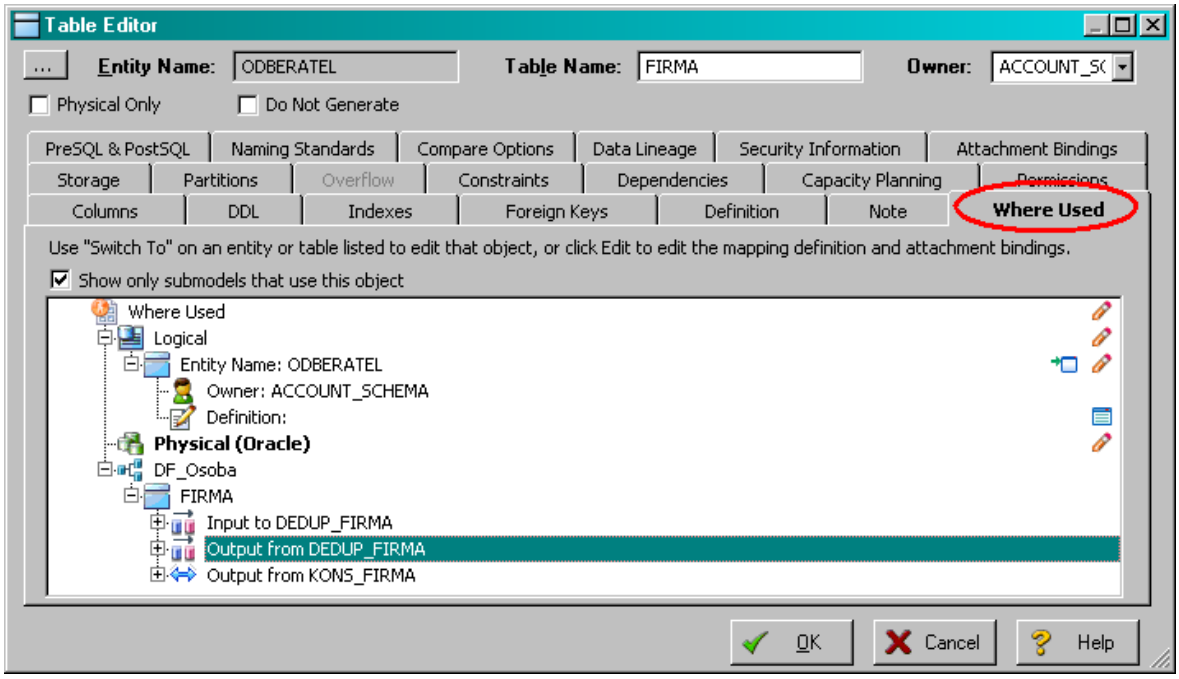
V první fázi (obr. 14) se jedná o plochou strukturu bez referencí, které jsou do modelu doplněny až po deduplikaci. Hlavní úlohou první fáze (Stage) je shromáždění (konsolidace) záznamů ze zdrojových systémů a jejich standardizace. Druhá fáze (obr. 15) již slouží k finálnímu uložení „vyčištěných“ a deduplikovaných Master Dat. Toto řešení zajišťuje zpětnou identifikaci zdrojových dat, tak jak byly získány z původních systémů.



Obrázek 16: Zajištění zpětné identifikace zdrojových záznamů (Zdroj: Vlastní)

Naplnění MD Repository

Jak je patrné ze schématu (obrázek 17), naplnění daty nelze provést přímo. Bylo proto třeba navrhnout vodnou aplikační logiku (transformační skripty). Základem byly opět datové modely zdrojových systémů a centrálního repository, které sloužily jako zdroj respektive cíl pro model datových toků a transformací (příloha 4). Tento model byl rovněž začleněn do metadata repository. Při požadavku na změnu v některém z dotčených systémů je tak možné provést snadno a rychle dopadovou analýzu.



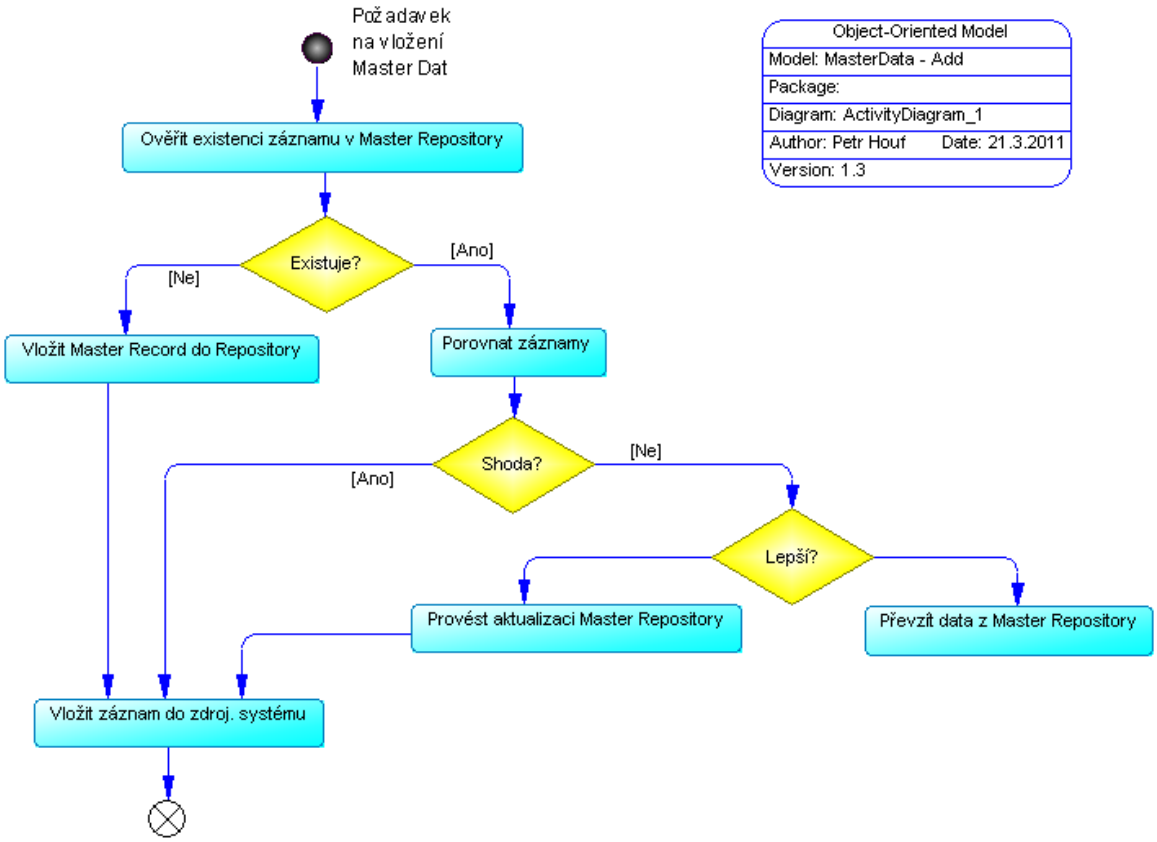
Obrázek 17: Ukázka dopadové analýzy (Zdroj: Vlastní)

Master Data Management Hub

Pro účely pilotního projektu byl provisioning realizován jako aplikace poskytující funkce pro základní řízení kmenových dat v centrálním repository. Aplikace realizuje:

Vložení nového záznamu - Požadavek na vložení nového záznamu obsahujícího kmenová data není realizován provozním systémem, ale je předán MDM repository. Zde je ověřeno, zda se jedná o nový nebo již existující subjekt. Pokud subjekt neexistuje, je doplněn do repository a volajícímu systému je vrácena stejná sada parametrů, které obsahoval požadavek. V opačném případě jsou parametry volání nahrazeny existujícími hodnotami uloženými v MDM repository a ty jsou použity volajícím systémem.

Aktualizace záznamu - Aktualizace záznamu obsahujícího kmenová data probíhá obdobným způsobem jako jeho vložení. Předané parametry jsou porovnány s existujícími kmenovými daty, a jsou-li na kvalitativně vyšší úrovni, použijí se pro jejich aktualizaci. Pokud tomu tak není, jsou volanému systému vracena kmenová data, a ta se použijí pro aktualizaci.



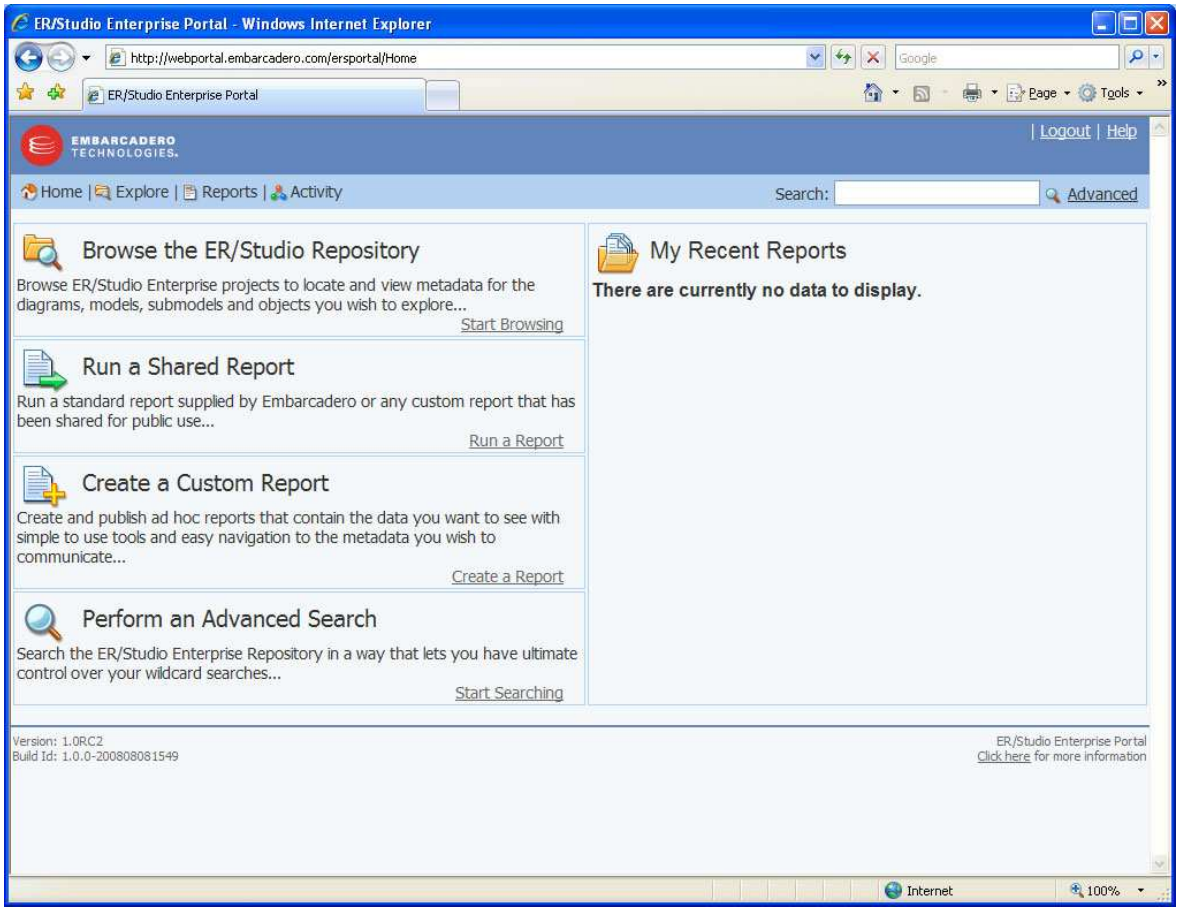
Obrázek 18: Proces vložení nového záznamu (Zdroj: Vlastní)

Smazání záznamu - Vzhledem k objemu dat bude požadavek na odstranění záznamu realizován pouze ve zdrojovém systému. Záznam v Master Data Repository ponecháme. V situaci, kdy by bylo nutné záznamy odstraňovat i repository (např. z důvodu úspory úložného prostoru), muselo by samozřejmě předcházet ověření, že rušený záznam není používán žádným z ostatních informačních systémů využívajících MDM repository.

Data Governance

Jak bylo zmíněno v teoretické části práce, pro úspěšnou implementaci a následné zkvalitňování MDM je důležité, aby zainteresovaní pracovníci mohli snadno sdílet potřebné informace. Pro potřeby řízení a pravidelného vyhodnocování bylo vybudované metadata repository rozšířeno o informace získávané profilováním a informace

poskytované nástroji pro monitorování. Za vhodnou komunikační platformu byl zvolen webový portál, prostřednictvím kterého lze zobrazovat, prohledávat a komentovat modely a další metadata uložená v metadata repository. Portál dále umožňuje vytvářet ad-hoc výstupní sestavy a zajistit jejich distribuci.



Obrázek 19: Rozhraní portálu pro práci s metadaty (Zdroj: Vlastní)

5 Přínosy implementace MDM

Navržené řešení se soustředilo pouze na oblast dat, týkajících se zákazníků a obchodních partnerů. Podobným způsobem by bylo možné konsolidovat a sdílet data týkající se komunikace se zákazníky, data o dodávaných produktech a podobně. Přes omezený rozsah měla implementace MDM pro stávající infrastrukturu společnosti tyto přínosy:

- Implementace MDM celkově pozvedla kvalitu operativních dat. Odpadly tak například chyby při automatizovaném zpracování objednávek přijatých prostřednictvím webového obchodu. Nedochází již ani k problémům s duplikací objednávek podaných současně přes e-shop a písemně.
- Díky sjednocení kmenových dat je možné implementovat procesy realizované přes více systémů/datových úložišť a jednodušeji tak zajistit jejich softwarovou podporu a automatizaci.
- V souvislosti se zavedením MDM byl vytvořen podnikový datový slovník, který standardizuje datový model a umožňuje jeho lepší řízení. Možnost dopadové analýzy a znovupoužití datových prvků zkvalitňuje a usnadňuje realizaci požadovaných změn a další rozvoj jednotlivých systémů.
- Analýzy prováděné během MDM projektu vedly k důkladnému zdokumentování podnikových procesů a používaných systémů. Další rozvoj IT architektury již bude prováděn výhradně v souladu s MDA.
- Zavedení MDM zároveň vytvořilo podmínky pro plánované vybudování datového skladu. Ten v budoucnosti umožní analyzovat data, která jsou aktuálně roztroušena po samostatných datových silech.
- Zásadní snížení chybovosti a možnost automatizovat řadu činnosti, které bylo dříve nutné provádět ručně, vedlo ke snížení provozních nákladů.
- Zkvalitnění a homogenizace datové základny umožnilo automatizovat řadu činností. Zrychlení, zlevnění a zkvalitnění služeb poskytovaných zákazníkům výrazně zvýšilo konkurenceschopnost společnosti.

Jak je patrné, význam MDM se projevuje téměř ve všech oblastech fungování podniku. Přestože hlavní motivací pro implementaci MDM je především podpora obchodních cílů organizace, dopady jeho zavedení lze kladně hodnotit i v rovině finanční a technické.

6 Závěr

Stále více se prosazující technologie, jako jsou datové sklady, servisně orientovaná architektura nebo Cloud Computing kladou zásadní důraz na sdílení dat, která jsou v současnosti zpravidla roztříštěna po mnoha oddělených datových silech. Dříve přínosné specializované aplikace s dedikovanými datovými úložišti tak dnes paradoxně brání zavádění komplexních podnikových procesů a jejich následné automatizaci. Společnosti proto hledají nejrůznější způsoby, jak získat z takto fragmentovaných dat ucelené a nezkreslené informace. V této souvislosti se stále častěji hovoří právě o Master Data Managementu.

Master Data Management je iterativním procesem zavádění a soustavného zlepšování technických a organizačních opatření, cílem kterých je především:

- Zajištění existence jediné platné verze kmenových dat
- Zajištění použití shodných kmenových dat napříč zvolenými informačními systémy a aplikacemi
- Zajištění maximální možné kvality kmenových dat (jejich přesnosti, věrnosti, úplnosti a aktuálnosti)
- Zajištění vhodného způsobu řízení celého životního cyklu kmenových dat

Master Data Management je tedy souborem postupů, technologií a organizačních opatření, která vedou k sjednocení klíčových dat a umožňují tak snáze integrovat stávající systémy a aplikace a vytvořit tak moderní infrastrukturu pro pružnou podporu obchodních cílů organizace. Výsledky dosahované tradičními integračními přístupy jako je datová nebo aplikační integrace včetně aktuálně upřednostňované servisně orientované architektury jsou často devalvovány právě nízkou kvalitou dat v konsolidovaných systémech. Je samozřejmě možné snažit se eliminovat tyto nedostatky v každém ze systémů samostatně, MDM však nabízí komplexní řešení. Klade důraz nejen na klíčová data, která si jednotlivé systémy vyměňují, ale také na řízení celého jejich životního cyklu. Dochází tak k přirozené konsolidaci dat, která jsou pro integrované systémy společná, a naopak specifická data zůstávají plně v kompetenci specializovaných aplikací.

Nedílnou součástí zavádění MDM je analýza, která se opírá především o modelování (ať již procesní, datové nebo objektové). Vedlejším, ale velmi důležitým produktem MDM je tak poznání firemních procesů a datových struktur, které využívají. Předností MDM je také

fakt, že v maximální možné míře využívá stávající zdroje a díky široké škále možných řešení umožňuje dosáhnout požadovaných cílů bez zásadního dopadu na existující aplikace a systémy.

Praktická část předložené bakalářské práce rovněž prokázala, že MDM není řešením pouze pro velké společnosti, ale že jej lze implementovat cíleně jen pro zvolenou oblast dat, nebo postupně po jednotlivých etapách. Obecně lze MDM doporučit společnostem, jejichž informační infrastruktura pro podporu svých hlavních aktivit využívá tří a více systémů.

7 Seznam použitých zdrojů

Seznam obrázků

Seznam tabulek

Přehled použitých termínů

JIT	Just In Time - Způsob zásobování, kdy dodavatel přebírá zodpovědnost za včasné dodávky a odběratel tak nemusí udržovat skladové zásoby.
ROI	Return Of Investements – Návratnost vložených investic. Zpravidla se uvádí časový úsek, během kterého zisk nebo úspora vzniklé nasazením dané technologie či zařízení umoří náklady související s jejím pořízením.
TCO	Total Cost Ownership – Celkové náklady na vlastnictví, které kromě pořizovací ceny zohledňují všechny další výdaje související s pořízením výrobku či technologie.
Cloud Computing	Moderní koncept kombinující výhody virtualizace hardwarových prostředků a SaaS.
BI	Business Intelligence – Využití analytických a reportovacích nástrojů pro vytěžení informací z existujících dat.
SaaS	Software as a Service – Způsob licencování software, kdy je použití SW aplikace účtováno jako služba.
ESB	Enterprise Service Bus – Komunikační sběrnice zajišťující řízenou výměnu zpráv mezi jím propojenými systémy.
OLTP	OnLine Transactional Processing – Způsob ukládání a zpracování dat v relačních databázových systémech.
CRM	Customer Relationship Management – Systém pro řízení vztahů se zákazníky.
EIS	Ekonomický Informační Systém – Jedná se zpravidla o rošířený účetní SW nebo jeho nadstavbu, která umožňuje analýzu finančních dat.
MDA	Model Driven Architecture – Souhrné označení metodik pro řízení životního cyklu IS za pomocí modelování (např. Zachman Framework, RUP a další).

Seznam literatury

ADELMAN, Sid, MOSS, Larissa, ABAI, Majid *Data Strategy*, Prentice Hall PTR, 2005, ISBN 0-321-24099-5

BROWN , Paul C. *Implementing SOA*, Addison Wesley Professional, 2008, ISBN 0-321-50472-0

CONOLLY, Thomas, BEGG, Carolyn, HOLOWCZAK, Richard *Profesionální průvodce tvorbou efektivních databází* Computer Press, 2009, ISBN 978-80-251-2328-7

GROFF, J.R., WEINBERG, P.N. *SQL Kompletní průvodce* Computer Press, 2005, ISBN 80-251-0369-2

HUMPRIES, Mark a kol. *Data warehousing – návrh a implementace* Computer Press, 2002, 80-7226-560-1

IBM, Global Business Services *SOA Scenarios – An Introduction*, IBM, 2005

INMON, William, STRAUSS, D., *DW 2.0* Morgan Kaufmann OMG Press, 2008, ISBN 978-0-12-374319-0

KEEN, Martin *Implementing an SOA Using an Enterprise Service Bus* IBM, 2004, ISBN 0738490008

LACKO, Luboslav *Datové sklady, analýza OLAP a dolování dat* Computer Press, 2003, ISBN 80-7226-969-0

LONEY, Kevin, BRYLA Bob *Mistrovství v Oracle Database 10g* Computer Press, 2006, ISBN 80-251-1277-2

LOSHIN David *Master Data Management* Morgan Kaufmann OMG Press, 2009, ISBN 978-0-12-374225-4

McGEE, William C. *Data Base Technology* IBM Journal of Research & Development 25, 1981, ISSN 0018-8646

MINOLI ,Daniel *Enterprise Architecture A to Z* Taylor & Francis Group, 2008, ISBN 978-0-8493-8517-9

Replication Strategies Technical White Paper, Sybase Inc., 2003, L02038 MIL6143

SELINGER, P.G. *Database technology* IBM Systems Journal č.26, 1987, ISSN 0018-8670

WHITE, Colin *Using Master Data in Business* BI Research, 2007

8 Přílohy

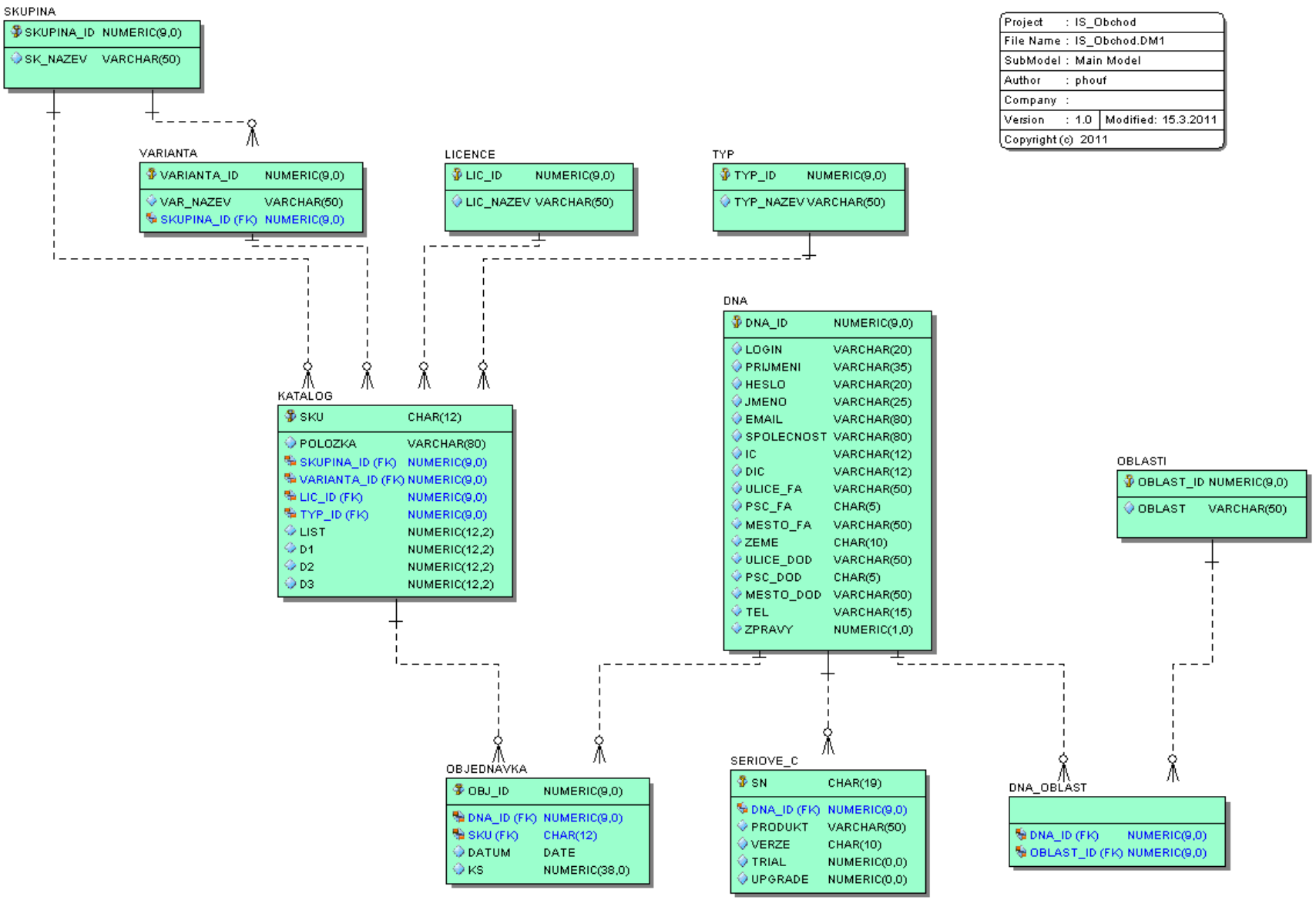
Příloha 1: Datový model IS Obchod

Příloha 2: Datový model IS Support

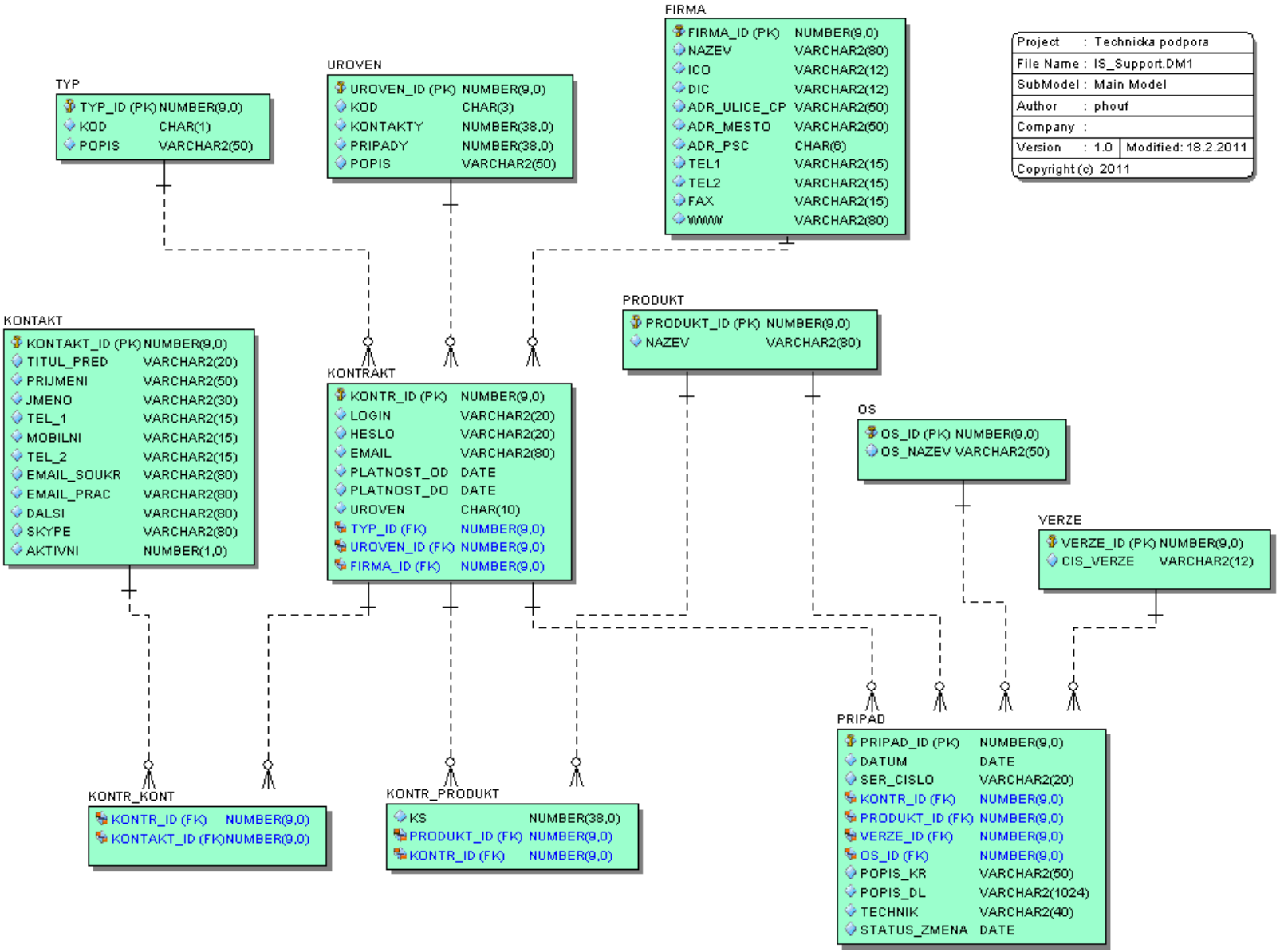
Příloha 3: Datový model IS Účetnictví

Příloha 4: Model datových toků

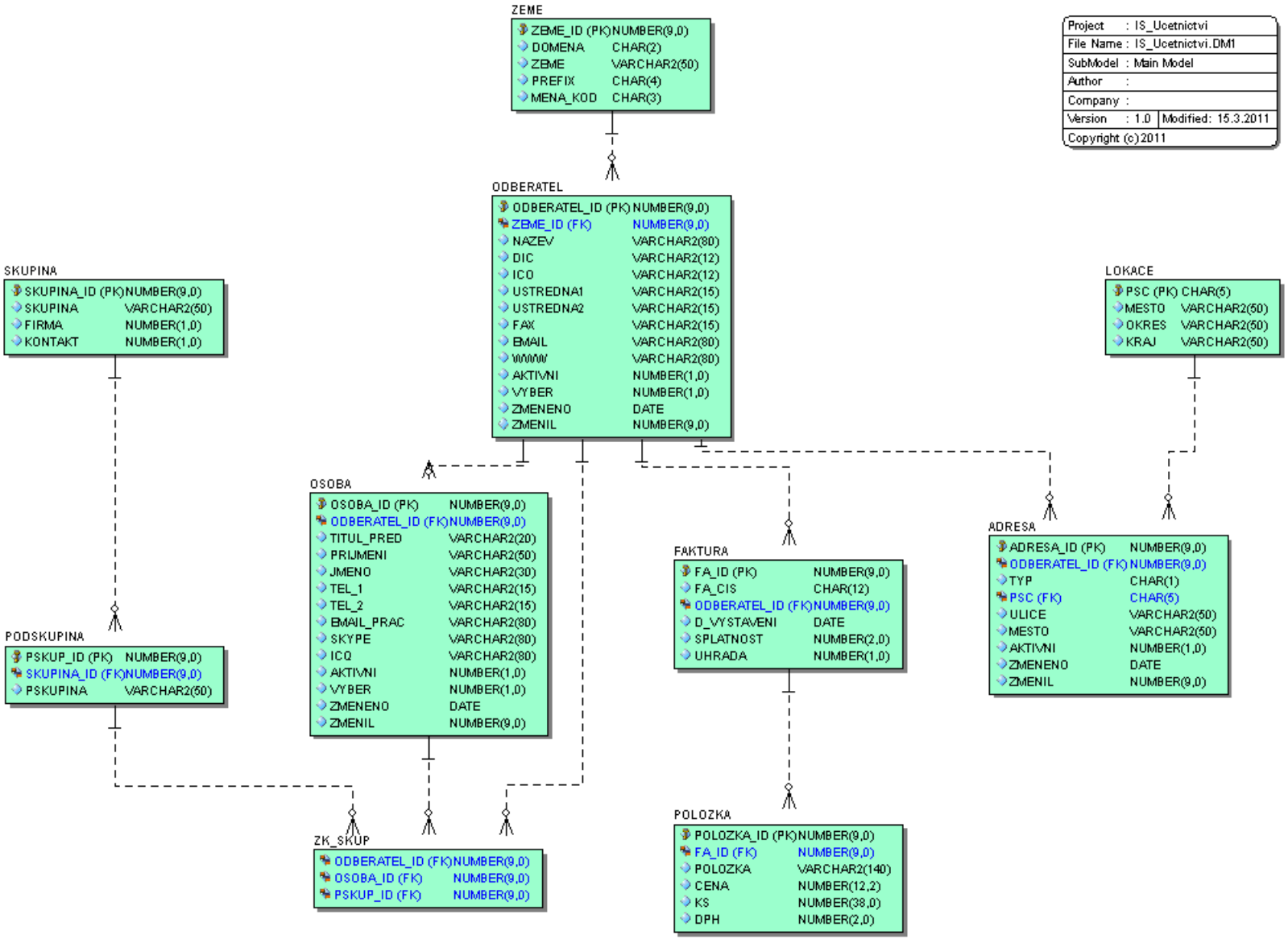
Příloha 5: Enterprise Data Dictionary



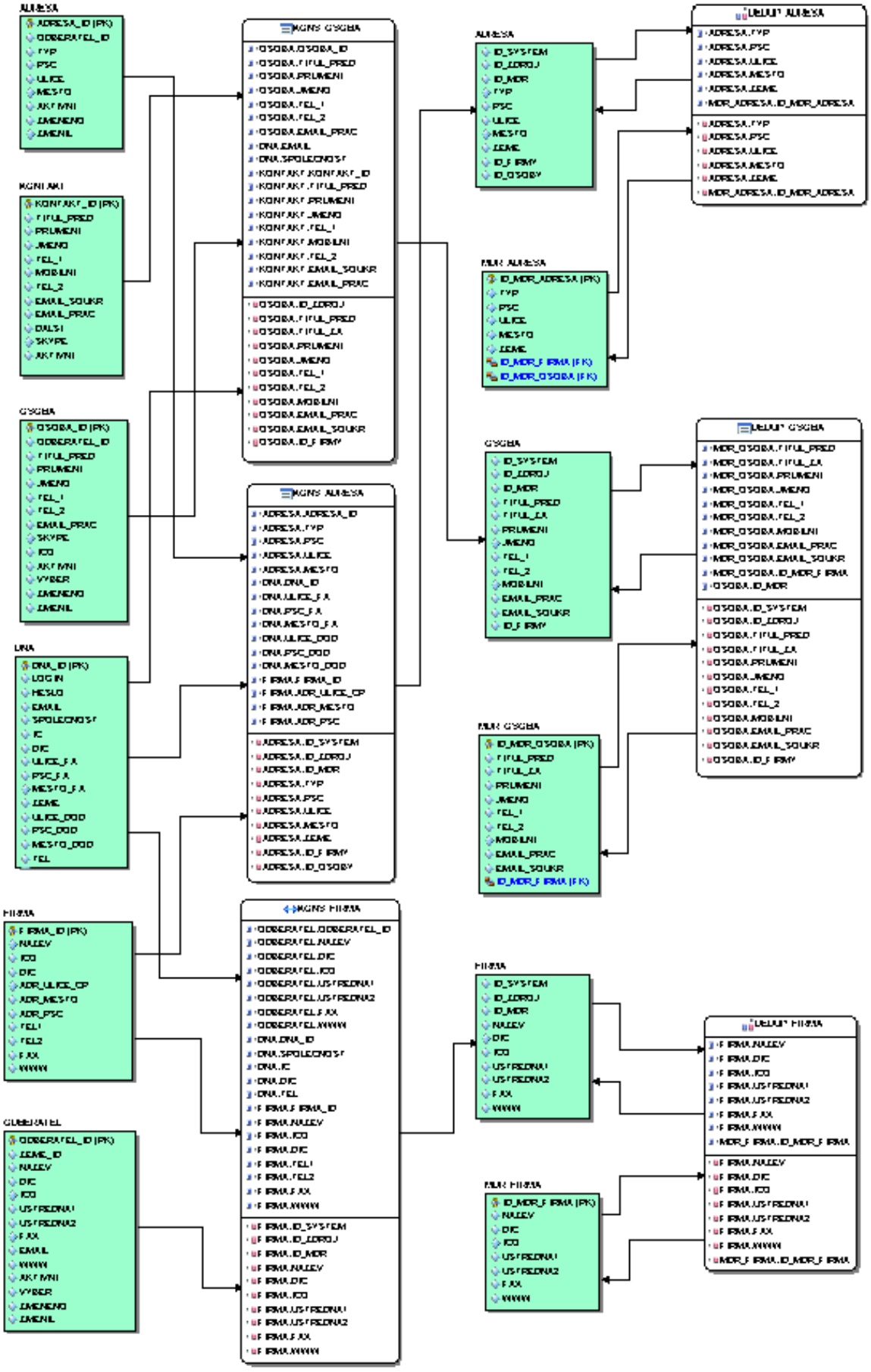
Příloha 1: Datový model IS Obchod



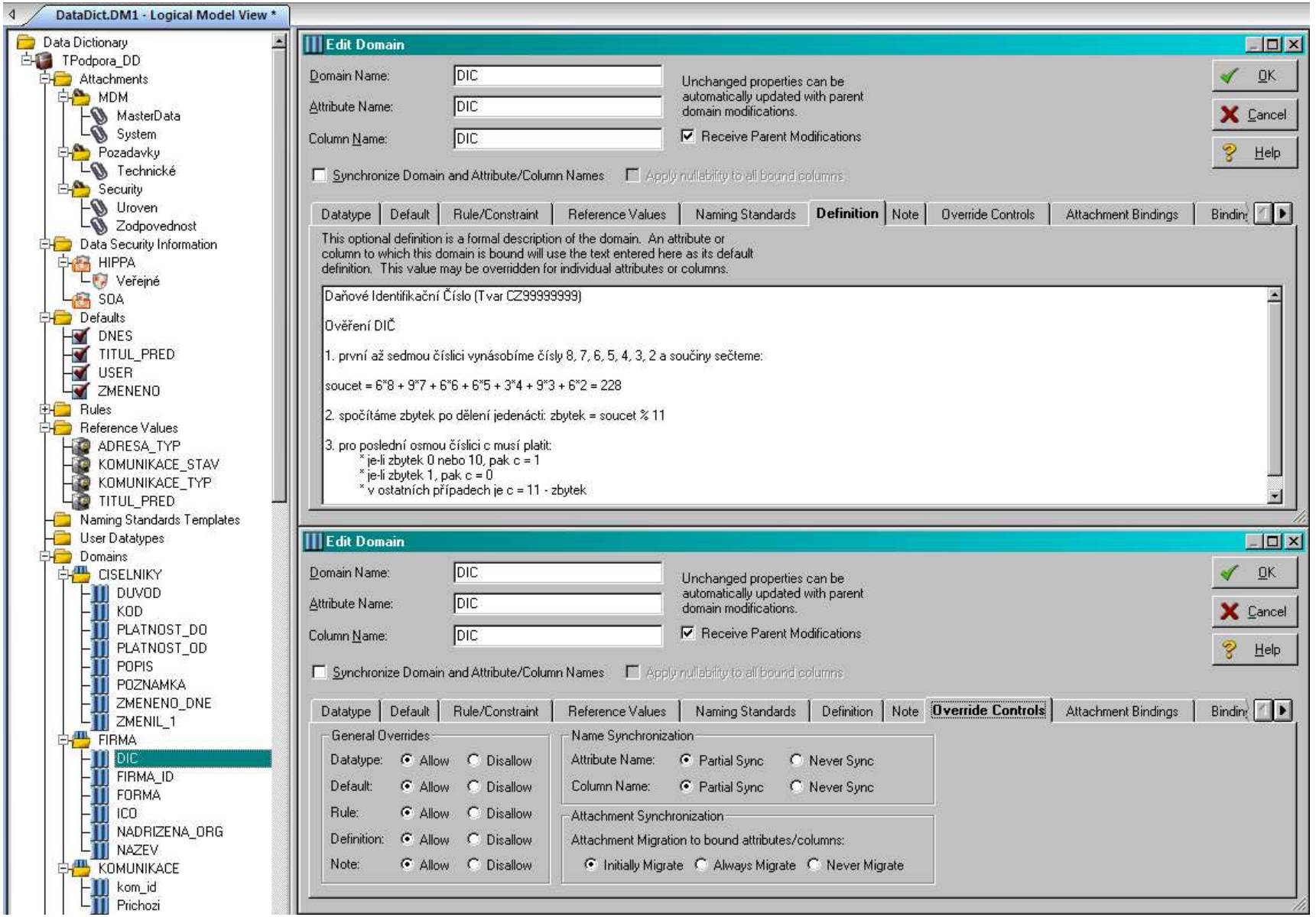
Příloha 2: Datový model IS Support



Příloha 3: Datový model IS Účetnictví



Příloha 4: Model datových toků



Příloha 5: Enterprise Data Dictionary