

VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE

Fakulta informatiky a statistiky

DIPLOMOVÁ PRÁCE

2011

Jan Bartoš

VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE

Fakulta informatiky a statistiky
Katedra informačních technologií (KIT)



**Master Data Integration hub - řešení pro
konsolidaci referenčních dat v podniku**

Diplomová práce

Diplomant:

Bc. Jan Bartoš

Vedoucí diplomové práce:

Ing. David Slánský

Oponent:

doc. Ing. Jan Pour, CSc.

Rok:

2011

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze kterých jsem čerpal.

V Praze dne 25. 04. 2011

.....

podpis

Poděkování

Na tomto místě chci poděkovat svému vedoucímu diplomové práce panu Davidu Slánskému za to, že mi pootevřel dveře do světa Business Intelligence, za povzbuzení, vstřícnost a ochotu, se kterou zaštitil téma, které jsem si zvolil.

Dále pak kolegovi Jirkovi Trpkošovi za cenné připomínky a postřehy k tématu práce.

A především chci poděkovat svým rodičům za obětavou podporu, kterou mi v průběhu celého studia poskytovali.

Abstrakt

V současných informačních systémech je kladen velký důraz na integraci dílčích aplikací do funkčního celku. Zatímco integrace po stránce komunikační a funkční

je již poměrně dobře pokryta a existují standardní nástroje a technologie pro její realizaci (ESB, message brokery, webové služby), nástroje a postupy pro integraci datových zdrojů dílčích aplikací do společného celku na celopodnikové úrovni se zatím stále bouřlivě vyvíjí.

Master Data Management (MDM) je zastřešujícím přístupem ke správě referenčních (master) dat v podniku. Řeší otázky integrace referenčních dat, datové kvality a konsolidace referenčních dat, evidence a využití metadat při správě referenčních dat, vlastnictví sdílených referenčních dat, zodpovědnosti za referenční data a procesy pro práci s nimi.

Práce se zabývá technologickou stránkou implementace MDM v podniku formou vytvoření centralizovaného úložiště referenčních dat – tzv. Master Data Integration Hubu (MDI hub). MDI hub je systém/aplikace, která umožňuje na základě definovaných workflow a pravidel shromažďovat a konsolidovat referenční data uložená v podnikových systémech a konsolidovaná data poskytovat nazpět. Zároveň poskytuje další služby jako správu číselníků a monitorování datové kvality.

Cílem práce je pokrýt návrh a implementaci MDI hubu, který je konkrétní formou realizace MDM v podniku. V úvodní části práce je řešena motivace pro provádění konsolidace referenčních dat, je upřesněn obsah pojmu MDM a vztah MDI hubu k principům MDM. Dále je uveden souhrn technik používaných při realizaci MDI hubu. Hlavní částí práce je návrh referenční architektury MDI hubu a popis projektových aktivit při realizaci MDI hubu. Zdrojem informací pro práci jsou, kromě odborné literatury (monografie, odborné články, příspěvky z konferencí), také poznatky nabyté při realizaci projektů pro společnosti Adastra a Ataccama a know-how spolupracovníků.

Hlavním přínosem práce je vytvoření uceleného pohledu na budování MDI hubu a návrh referenční architektury MDI hubu, která může posloužit jako východisko při úvahách o konkrétní implementaci. Předpokládané využití práce je jak na straně klientské (firma se pustí do implementace MDI hubu vlastními silami) tak na straně dodavatelské (pro konzultační firmy a integrátory, kteří budou implementovat MDI hub pro zákazníka).

Klíčová slova: datová integrace, Master Data Management, MDM, Customer Data Integration, CDI, Master Data Integration hub, MDI hub, čištění dat, konsolidace dat, referenční data, master data.

Abstract

In current information systems the requirement to integrate disparate applications into cohesive package is greatly accented. While well-established technologies facilitating functional and comunicational integration (ESB, message brokes, web services) already exist, tools and methodologies for continuous integration of disparate data sources on enterprise-wide level are still in development.

Master Data Management (MDM) is a major approach in the area of data integration and referrential data management in particular. It encompasses the referrential data integration, data quality management and referrential data consolidation, metadata management, master data ownership, principle of accountability for master data and processes related to referrential data management.

Thesis is focused on technological aspects of MDM implementation realized via introduction of centrallized repository for master data – Master Data Integration Hub (MDI Hub). MDI Hub is an application which enables the integration and consolidation of referrential data stored in disparate systems and applications based on predefined workflows. It also handles the master data propagation back to source systems and provides services like dictionaries management and data quality monitoring.

Thesis objective is to cover design and implementation aspects of MDI Hub, which forms the application part of MDM. In introduction the motivation for referrential data consolidation is discussed and list of techniques used in MDI Hub solution development is presented. The main part of thesis proposes the design of MDI Hub referrential architecture and suggests the activities performed in process of MDI Hub implementation. Thesis is based on information gained from specialized publications, on knowledge gathererd by delivering projects with companies Adastra and Ataccama and on co-workers know-how and experience.

Most important contribution of thesis is comprehensive view on MDI Hub design and MDI Hub referrential architecture proposal. MDI Hub referrential architecture can serve as basis for particular MDI Hub implementation.

Keywords: data integration, Master Data Management, MDM, Customer Data Integration, CDI, Master Data Integration hub, MDI hub, data cleansing, data consolidation, referrential data, master data.

Obsah

1	Úvod.....	1
1.1	Vymezení tématu práce.....	1
1.2	Cíle práce	2
1.3	Struktura práce	2
1.4	Postup zpracování.....	3
2	Teoretický základ.....	4
2.1	Datová základna podniku – výchozí stav.....	4
2.2	Taxonomie podnikových dat.....	6
2.2.1	<i>Transakční data</i>	6
2.2.2	<i>Referenční data</i>	7
2.2.2.1	Entitní záznamy	8
2.2.2.2	Číselníky.....	8
2.2.3	<i>Metadata</i>	9
2.2.3.1	Business metadata.....	9
2.2.3.2	Technická metadata	10
2.2.4	<i>Relační data</i>	10
2.2.5	<i>Analytická data</i>	11
2.2.6	<i>Administrativní (office) data</i>	11
2.3	Master data management.....	11
2.3.1	<i>Definice MDM</i>	11
2.3.2	<i>Motivace pro zavedení MDM</i>	13
2.3.3	<i>Motivace pro nezavedení MDM</i>	15
3	Použité techniky	16
3.1	Procesní modelování.....	16
3.2	Datové modelování.....	16
3.3	Profiling a explorativní analýza dat	20
3.4	Sběr a analýza uživatelských požadavků	21
3.5	Datová kvalita.....	22
3.6	Metodické řízení vývoje	24
4	Master Data Integration hub.....	26
4.1	Základní požadavky na MDI hub.....	26
4.2	Referenční architektura MDI hubu.....	27
4.2.1	<i>Paradigmata architektury MDI hubu</i>	27
4.2.2	<i>Návrh referenční architektury MDI hubu</i>	28
4.3	Komponenty referenční architektury	31
4.3.1	<i>Staging area</i>	31
4.3.2	<i>Master databáze</i>	32
4.3.2.1	Realizace master databáze	32
4.3.2.2	Modelování master databáze	38
4.3.3	<i>ETL</i>	40
4.3.4	<i>Master data server</i>	41
4.3.5	<i>Klientské aplikace</i>	45
4.3.5.1	Správa číselníků	45
4.3.5.2	Správa výjimek.....	46

4.3.5.3	Administrační konzole	47
4.3.6	Reporting datové kvality	48
5	Projekt implementace MDI hubu	49
5.1	Projektové role	49
5.1.1	Business analytik.....	49
5.1.2	Datový analytik.....	49
5.1.3	Databázový specialista	50
5.1.4	Specialista datové kvality	50
5.1.5	Programátor	50
5.1.6	Solution architect	51
5.2	Fáze projektu.....	51
5.2.1	Analýza	52
5.2.1.1	Katalogizace zdrojových systémů	52
5.2.1.2	Katalogizace entit.....	53
5.2.1.3	Profiling.....	54
5.2.1.4	Analýza a správa uživatelských požadavků.....	55
5.2.1.5	Návrh logického datového modelu master databáze	55
5.2.2	Design	56
5.2.2.1	Návrh cílové architektury MDI hubu	56
5.2.2.2	Návrh pravidel pro čištění dat	56
5.2.2.3	Návrh unifikačních pravidel	57
5.2.2.4	Návrh fyzického datového modelu master databáze	57
5.2.2.5	Návrh rozhraní a služeb.....	58
5.2.2.6	Návrh ETL.....	58
5.2.2.7	Návrh workflow	58
5.2.2.8	Návrh integračních testů.....	59
5.2.3	Implementace.....	59
5.2.3.1	Implementace ETL	59
5.2.3.2	Konfigurování DQ nástroje	59
5.2.3.3	Implementace workflow	60
5.2.3.4	Implementace rozhraní a služeb	60
5.2.3.5	Integrační testování	60
5.2.4	Review	61
5.2.5	Nasazení.....	61
5.2.5.1	Instalace v testovacím/produkčním prostředí	61
5.2.5.2	Úvodní load dat	61
5.2.5.3	Akceptační testy	62
6	Závěr	63
7	Seznam obrázků	66
8	Použitá literatura	67
9	Terminologický slovník.....	69
10	Přílohy	71
10.1	Dodavatelé MDM software, řešení a služeb.....	71
10.2	Standardní workflow MDI hubu	73
10.2.1	WF1 – Import referenčních dat přes dávkové rozhraní.....	73
10.2.2	WF2 – Vložení referenčního záznamu přes online rozhraní	74
10.2.3	WF3 – Řešení nálezu datové kvality	75
10.2.4	WF4 – Aktualizace číselníků.....	76
10.3	Šablony výstupů.....	77

10.3.1	Katalog systémů/datových zdrojů	77
10.3.2	Katalog business entit	78
10.3.3	Přehled zjištění profilingu	79
10.4	Souhrnné reporty datové kvality	80
10.4.1	Souhrnná statistika datové kvality za entitu	80
10.4.2	Labelling	81
10.4.2.1	Labelling na úrovni záznamu – souhrnná statistika	81
10.4.2.2	Labelling na úrovni atributu – souhrnná statistika	81
10.4.2.3	Labelling záznamů na instanční úrovni	82
10.5	Report pro manuální opravu záznamů	83

1 Úvod

1.1 Vymezení tématu práce

Diplomová práce se zaměřuje na uplatnění master data managementu (MDM) při správě podnikových dat, zejména pak na vybudování aplikace pro jednotnou správu referenčních (kmenových, master) dat v podniku – Master Data Integration hub (MDI hub).

MDM je souhrnný pojem pro principy, procesy, technologie a nástroje, které se uplatňují při správě tzv. referenčních dat. Referenční data jsou specifickým typem dat, který reprezentuje reálné objekty (fyzické a právnické osoby, lokality, kontaktní údaje, produkty, hmotný investiční majetek apod.). Tyto reprezentace se mohou napříč různými systémy v rámci podniku i mimo něj lišit (a zpravidla se také liší) z různých důvodů:

- různá úroveň požadovaného detailu informací o daném objektu v různých systémech
- zastarávání údajů
- lidský faktor (chyby při zadávání údajů do systému, různé způsoby zadání téhož údaje)
- fyzická omezení systémů (různé sady popisných atributů, délka datových polí)
- specifické požadavky na zadávaná data (např. zkracování názvů v adresách kvůli omezení délky řádku při tisku na obálky)
- některé údaje nemusí být v okamžiku vytváření záznamu k dispozici

Rozdílné reprezentace reálných objektů vedou k otázkám:

- Reprezentují dva podobné záznamy ze dvou systémů (nebo i ze stejného systému) stejný reálný objekt (adresu, ...), nebo se jedná o dva odlišné objekty?
- Pokud dva záznamy reprezentují stejný reálný objekt a liší se v některých attributech, který ze záznamů obsahuje správné/aktuální údaje?¹
- Znamenají různé hodnoty téhož atributu u reprezentace stejného objektu v různých systémech faktický rozpor, nebo se jedná pouze o interpretační rozdíl? (otázka číselníků a business popisu atributů)
- Kolik unikátních reálných objektů určitého typu je zaznamenáno v systému a podnik s nimi pracuje?

Cílem MDM je vytvořit a efektivně spravovat jednotnou konzistentní bázi referenčních dat v podniku. Existence jednotné báze referenčních dat pak umožňuje zodpovědět výše nastíněné otázky.

Pro podnik je jednotná datová reprezentace reálných entit důležitá z mnoha důvodů:

- jednoznačné určení reálných objektů při komunikaci (uvnitř podniku nebo s třetími stranami)

¹ “A man with a watch knows what time it is. A man with two watches is never sure.” – Segal’s law
(lze volně přeložit: „Pokud máte jedny hodinky, víte, kolik je hodin. Pokud máte dvojce, nikdy si nemůžete být jisti.“)

- reporting
- udržování aktuálních informací o realitě
- možnost použití dalších technik, které pro své správné fungování potřebují co nejsprávnější a nejúplnější referenční data – data mining, specializované marketingové techniky (např. campaign management), risk management.

MDM využívá poznatky a principy dalších specializovaných disciplín:

- Data Quality Management – otázka zajištění souladu mezi skutečnými a požadovanými parametry/vlastnostmi dat
- Data Governance – otázky řízení práce s daty (evidence dat jako specifické formy aktiv, stanovení odpovědnosti za data, řízení a rozvoj procesů v oblasti nákládání s daty, ...)
- Metadata management – otázky sběru, uspořádání, uchování a poskytování metadat

Tyto disciplíny přesahují téma práce a budou zmíněny pouze v rozsahu nutném pro dosažení cíle práce.

1.2 Cíle práce

Cílem této práce je navrhnout a popsat realizaci sdílené báze referenčních dat prostřednictvím implementace MDI hubu v podniku. Dílčími kroky k jeho dosažení jsou:

- vymezení řešení v rámci problémové oblasti (klasifikace dat, identifikace referenčních dat)
- vymezení technik, které budou použity při návrhu a realizaci MDI hubu
- návrh referenční architektury MDI hubu a návrh základních workflow pro činnost MDI hubu
- popis aktivit vykonávaných při budování MDI hubu

Přínos práce spočívá ve spojení teoretických konceptů a informací o MDM a zpětné vazby získané z projektů datové konsolidace u zákazníků v různých odvětvích.

Předpokládané využití práce je při návrhu realizace systému pro správu referenčních dat jak na straně klientské (pokud se firma pustí do vývoje vlastními silami a bude implementovat MDM pro sebe) tak na straně dodavatelské (pro konzultační firmy a integrátory, kteří budou implementovat MDM pro externí zákazníky).

1.3 Struktura práce

První kapitola obsahuje informace o zaměření práce a jejích cílech, a vymezuje vnitřní strukturu práce.

Ve druhé kapitole je vymezena problémová oblast (podniková datová základna), v čem spočívá neuspokojivý stav problémové oblasti (roztržštěnost podnikové datové základny), příčina neuspokojivého stavu, východisko pro jeho možné řešení (referenční data a jejich integrace) a přístup, který lze k řešení vymezeného problému použít (Master Data Management). Zároveň jsou zmíněny další přínosy, které může uvedené řešení přinést. Ve druhé kapitole je též vymezeno, že se práce koncentruje na technologickou stránku řešení problému (implementace MDI hubu).

Ve třetí kapitole jsou uvedeny techniky, které jsou použity při návrhu a realizaci implementace MDI hubu.

Čtvrtá kapitola obsahuje návrh referenční architektury MDI hubu, která může posloužit jako východisko při úvahách o konkrétní implementaci MDI hubu v podmínkách konkrétního zákazníka. Jsou popsány její komponenty a principy, které se v těchto komponentách uplatňují. Pátá kapitola obsahuje přehled rolí, které se implementačního projektu účastní, a přehled aktivit realizovaných v rámci jednotlivých fází projektu spolu s očekávanými výstupy.

V šesté kapitole je uvedeno závěrečné shrnutí výsledků práce.

1.4 Postup zpracování

Úvodní části práce (kapitoly 2 a 3) jsou postaveny převážně na informacích získaných z veřejně dostupných zdrojů (monografie, odborné články, příspěvky konferencí, internetové stránky), které jsou uvedeny v seznamu použité literatury. Hlavní části práce (kapitoly 4 a 5) jsou postaveny především na zkušenostech a zpětné vazbě získané na projektech konsolidace referenčních dat, které byly realizovány společnostmi Adastra CZ a Ataccama Software. Při vytváření práce jsem využil možnost konzultovat aspekty referenčního řešení se zaměstnanci společnosti Ataccama Software, kteří mají v problémové oblasti rozsáhlé zkušenosti.

V práci je v popisu použit koncept dodavatele a zákazníka. Dodavatelem je obecně ten, kdo vytváří hlavní výstupy projektu, takže dodavatelem může být kromě externí firmy i např. interní vývojový tým. Zákazníkem je obecně ten, kdo klade požadavky, platí za projektové dodávky a benefituje z výsledků projektu.

2 Teoretický základ

V této kapitole je vymezena problémová oblast (datová základna podniku), její aktuální stav (roztržštěnost) a příčiny tohoto stavu, proč je aktuální stav nežádoucí a jak je možné ho řešit.

2.1 Datová základna podniku – výchozí stav

V rámci této podkapitoly je popsán postupný vývoj datové základny v typickém podniku. Níže popsany postup vývoje lze brát jako empiricky ověřený.

V době, kdy se začaly informační technologie prosazovat v činnosti a řízení podniků, se uplatňovaly především pro podporu každodenních rutinních operací. Data², se kterými podnik začal nejdříve pracovat a která začal uchovávat, byla data operativní. Jednotlivé aplikace řešily izolované agendy a ke vzájemné výměně dat docházelo velmi zřídka. Běžně se data z jedné aplikace do druhé přepisovala ručně „z papíru“, případně docházelo k ad-hoc výměnám dat prostřednictvím souborových dávek. Protože aplikace nebyly vzájemně propojené, musela si každá aplikace uchovávat kompletní sadu dat, kterou potřebovala pro své fungování – každá aplikace měla svůj vlastní seznam klientů, produktů, své vlastní číselníky atd. V jednotlivých aplikacích se hromadila data, která se mimo danou aplikaci nikde nevyužívala, aplikace fungovaly jako tzv. informační sila. Díky neexistující integraci a komunikaci mezi aplikacemi se v podnikové datové základně začala objevovat nekonzistentní a redundantní data. Rozvoj informačního systému byl řízen diktátem businessu, který požadoval stále nové funkce – rozšiřovaly se stávající aplikace a vyvíjely/installovaly se nové. Rostoucí počet aplikací znamenal rostoucí komplexnost IT infrastruktury podniku a bylo stále obtížnější plnit nové požadavky businessu na IT a realizovat nové IT projekty. Protože vzájemné provázání a integrace jednotlivých aplikací dříve typicky nespádaly mezi hlavní požadavky při rozvoji IS, datová základna podniku byla roztržštěná a nacházela se v atomizovaném stavu. Podle Voříška [Voříšek, 2003, str. 86] je přitom atomizovaná datová základna jedním z významných rizik z pohledu architektury IS/IT:

„Chybou architektury IS/IT je i atomizovaná datová základna podnikového informačního systému. Tato chyba se vyznačuje tím, že jednotlivé aplikace nesdílejí společnou datovou základnu, ale každá aplikace pracuje nad svou datovou základnou. Důsledkem atomizované datové základny je duplicitní sběr dat, obtížné zajištění konzistence, a v konečném důsledku nejednoznačná identifikace objektů.“

Mezi faktory způsobující atomizaci datové základny podniku patří:

- **Neexistující nebo špatná koncepce rozvoje IT v podniku** – v podniku neexistuje jednotná architektura IS/IT, jednotlivé aplikace jsou vyvíjeny/installovány podle potřeby, aniž by se řešila jejich vzájemná integrace. Integrace je realizována pouze na

² Pokud se v textu odkazují na data, mám na mysli data uložená v digitální podobě v informačním systému podniku.

základě dílčí potřeby komunikace mezi jednotlivými systémy pomocí proprietárních prostředků (tzv. spaghetti přístup)

- **Rozvoj databázových systémů, zejména relačních databází** – moderní databázové systémy se snadno instalují a používají. Většina podnikových aplikací přichází s vlastní vestavěnou databází.
- **Rozvoj kancelářských aplikací** – tabulkové kalkulátory a osobní databáze (např. MS Excel, MS Access) jsou stále sofistikovanější a přitom zůstávají snadno použitelné. Tvzení, že nejoblíbenějším manažerským nástrojem je Excel, není daleko od pravdy. Druhou stranou mince je ale to, že dochází k „excelizaci“ nejrozličnějších agend [Imhoff, 2004]. Každý pracovník má svoje vlastní osobní soubory, které ho činí nenahraditelným. Tyto soubory mnohdy obsahují data, která by měla být uložena spíše v podnikových aplikacích. IT nemá tyto osobní datové zdroje pod kontrolou a nemůže zajistit jejich řádné zabezpečení a integraci.

Problém nekonzistentních a redundantních dat v plné šíři odhalil až nástup technologie datových skladů.

Datové sklady se spolu s celou oblastí souhrnně nazývanou business intelligence objevily jako reakce na požadavek managementu více využívat podniková data pro podporu operativního, taktického a strategického rozhodování a řízení. BI pracuje s odlišným typem dat, než jaká se vyskytují v transakčních systémech – s tzv. analytickými daty. Analytická data vznikají transformací z dat operativních. Zatímco operativní data se vyskytují v podobě specifické pro jednotlivé systémy/aplikace (jsou systémově orientovaná), analytická data obsahují sjednocení dat ze všech aplikací členěná podle entit (jsou věcně orientovaná). Právě v procesu transformace operativních dat na analytická se objevily zásadní nedostatky plynoucí z atomizované datové základny podniku:

- Chybí jednoznačná a sdílená business definice entit
- Chybí jednoznačná a sdílená identifikace business objektů napříč systémy
- Jednotlivé systémy používají pro významově stejné atributy různé slovníky (číselníky) a různá označení (názvy sloupců)
- Liší se struktura a granularita evidovaných atributů pro tutéž entitu (např. v jednom systému je jméno klienta uloženo v jednom poli a v jiném systému jsou oddělená pole pro jméno, příjmení, tituly)
- V různých systémech existují pro tentýž reálný objekt různé verze dat a není možné rozhodnout, která verze je aktuální/správná

Dopady uvedených negativ je možné dokumentovat na reálném příkladu:

Triviální otázka – kolik máme zákazníků? – může být v jádru velmi komplikovaná. Pokud položí ředitel podniku tuto otázku vedoucím třech různých oddělení, může dostat až tři různé odpovědi. Pro oddělení marketingu je zákazníkem člověk, se kterým se uskutečňuje marketingová komunikace a na kterého se cílí marketingové aktivity. Do této kategorie mohou spadat i tzv. prospekti (potenciální zákazníci, kteří projeví zájem o produkty společnosti). Pro oddělení prodeje je zákazníkem ten, kdo objednává zboží a kdo za objednávku platí. Pro skladové hospodářství je zákazníkem člověk, který má převzít expedovanou zásilku. Příklad se stává dokonalým ve chvíli, kdy jednotlivá oddělení používají různé aplikace, které nesdílejí zákaznická data. Uvedená situace poukazuje mimo

jiné i na fakt, že napříč podnikem neexistuje jednotná sdílená definice (představa o tom), kdo je pojmem „zákazník“ míněn. Obdobný příklad uvádí Henrik Liliendahl Sørensen v článku na svém blogu³.

Jmenovatelem výše uvedeného problému je neexistující sdílená báze referenčních (master) dat. Master data jsou data, která mají být ze své podstaty sdílená více systémy (v krajním případě všemi systémy). Tato data popisují entity a reálné objekty, se kterými pracuje více systémů současně. Typickou entitou, která má charakter master entity, je „osoba“ (je jedno, zda fyzická nebo právnická). Osoba se může vyskytovat v různých rolích vůči podniku:

- zákazník, potenciální zákazník, dodavatel, odběratel, zaměstnanec
- prospekt, klient, majitel účtu, disponent, ručitel, bývalý klient
- pojištěná osoba, pojistník, obmyšlená osoba
- pacient, lékař

ale stále se jedná o tutéž reálnou osobu. Pokud se jí například změní adresa nebo jméno, měla by se tato změna projevit všude bez ohledu na roli, ve které se příslušná osoba nachází. Také není vyloučeno, že některá osoba bude zastávat i více rolí, a je žádoucí vědět, že i přes různé role se stále jedná o tutéž osobu.

Existují tedy objektivní faktory, které nutí podnik aktivně řešit otázku integrace datové základny.

2.2 Taxonomie podnikových dat

Východiskem pro úvahy o integraci datové základny je fakt, že data v podniku tvoří homogenní masu, ale na základě specifických scénářů používání je možné rozdělit je do kategorií. U každé kategorie dat se dají vysledovat společné rysy – způsob vzniku, formát, místa výskytu (systémy), použití. Různé kategorie podnikových dat mají také rozdílné předpoklady pro integraci – některé má, s ohledem na jejich charakteristické použití, smysl integrovat a některé nikoli. Dyché rozeznává v rámci podnikové datové základny následující typy dat [Dyché, a další, 2006, str. 44]:

- **transakční**
- **referenční**
- **vztahová**
- **metadata**

Tyto kategorie jsou hlavní a reprezentují strukturovaná primární data. Protože ne všechna data v podniku jsou primární a ne všechna jsou strukturovaná, rozšířil jsem seznam o dvě pomocné kategorie dat:

- **analytická** – sekundární data (odvozená z primárních)
- **administrativní** – nestrukturovaná nebo částečně strukturovaná data, která mají charakter pracovních a řídicích dokumentů

2.2.1 Transakční data

Transakční data vznikají při běžné činnosti podniku. Dokumentují jednotlivé operace, které se v podniku odehrávají – ať už se jedná o výrobu automobilu na lince nebo třeba o platební

³ Liliendahl Sørensen, Henrik. Perfect Wrong Answer. Blog: Liliendahl on Data Quality [online]. 9.1.2010, [cit. 2011-04-28]. <http://liliendahl.wordpress.com/2010/01/09/perfect-wrong-answer/>

operace prováděné s účtem v bance. Transakční data dokumentují určitou událost v činnosti podniku, která se jednou v čase odehrála, a od toho okamžiku se stávají součástí historie a jsou archivována.

Transakční data jsou typicky uložena v primárních (transakčních) systémech podniku (ERP, CRM, ostatní produkční systémy). Tyto systémy jsou určeny primárně pro pořizování dat a pracuje se s nimi v reálném čase. U transakčních systémů je kladen důraz na rychlou odezvu, proto je pro ně typické uložení dat ve třetí normální formě, velké množství tabulek a vazeb mezi nimi (převážná většina vazeb je právě mezi tabulkami obsahujícími transakční a referenční data) a snaha o minimální redundanci dat (případná redundance se používá cíleně z výkonnostních důvodů).

Typické rysy transakčních dat:

- transakčních záznamů vzniká velké množství (s ohledem na dynamiku činnosti podniku), je jich řádově více než referenčních záznamů
- transakční záznamy jsou z hlediska vazeb v roli podřízených záznamů (potomků) a odkazují se na referenční záznamy
- má smysl provádět matematické operace nad jejich agregacemi
- slouží jako zdroj pro vytváření faktových tabulek datového skladu
- v popisech byznys aktivit jsou transakční záznamy reprezentovány slovesy (zákazník **koupil** zboží)
- typický životní cyklus transakčního záznamu zahrnuje většinou pouze operace vytvoření a čtení

Transakční data jsou specifická pro jednotlivé systémy, vznikají a jsou používána v rámci jednoho systému a většinou není důvod je sdílet.

2.2.2 Referenční data

Referenční data popisují objekty reálného světa, které do činnosti podniku nějakým způsobem vstupují (např. fyzické osoby, právnické osoby, úřady) a jsou obsluhovány podnikovými procesy. Může se také jednat o hmotný majetek (např. nemovitosti, vozidla) nebo o záznamy konceptů či abstrakcí používaných při činnosti podniku (např. klienti, smlouvy). Referenční data mohou také vymezovat podmínky, za kterých se uskutečňují transakce (např. měna, ve které se transakce provádí, typ účtu, se kterým se transakce provádí, sazebník prováděných operací). Referenční data jsou spolu s transakčními daty uložena v transakčních systémech podniku.

Typickým projevem existence referenčních dat je skutečnost, že datové modely jednotlivých aplikací v podniku obsahují shodně či podobně pojmenované tabulky s podobnými sadami atributů, které mají podobný význam. Tyto tabulky se odkazují na sdílené koncepty, které se objevují v popisu činnosti podniku.

Pro referenční data je typické:

- mají tendenci být spíše statická (nízká frekvence změn), ale mohou se měnit
- referenčních záznamů je řádově méně než transakčních záznamů a vznikají méně často
- referenční záznamy se vyskytují v roli nadřazených (rodičovských) záznamů, na které se transakční záznamy odkazují pomocí databázových vazeb

- nemá smysl provádět matematické operace nad jejich agregacemi
- v popisech byznys aktivit jsou referenční záznamy reprezentovány podstatnými jmény (**zákazník** koupil **zboží**)
- typický životní cyklus referenčního záznamu obsahuje operace vytvoření, čtení, aktualizace a odstranění (CRUD)

Atomizace datové základny se nejvíce dotýká právě tohoto typu dat, protože tato data by měla být kvůli své povaze sdílena napříč všemi aplikacemi a systémy podniku. Práce s referenčními daty je hlavní náplní disciplíny souhrnně označované jako Master Data Management (viz. kapitola 2.3).

Konzistentní pohled na referenční data je důležitý pro:

- komunikaci uvnitř podniku (mezi aplikacemi/systémy/uživateli)
- komunikaci podniku navenek (např. s dodavateli, odběrateli, úřady apod.)
- konzistentní a jednoznačné výstupy reportingu

Referenční data se dále dělí na:

- entitní záznamy – popisují reálné objekty/koncepty
- číselníky – popisují dílčí atributy/charakteristiky/kvality objektů

2.2.2.1 Entitní záznamy

Entitní záznamy popisují instance jednotlivých entit (reálné objekty), se kterými podnik přichází při své činnosti do styku, případně se kterými pracuje. Entitou je zde míněna kategorie/třída business objektů (např. Osoba, Adresa, Smlouva). Entita je nositelem informace o tom, co má určitá skupina business objektů společného, jaké atributy popisují tyto objekty a jak lze jednotlivé objekty od sebe odlišit. Business objekt je pak konkrétním výskytem (instancí) dané entity.

Popis entity zahrnuje:

- **Definici entity** – jednoznačný popis entity, jak je daná entita v podniku chápána, jakých procesů se účastní
- **Vztahy entity** – v jakých vztazích s jinými entitami se může vyskytovat a jaké role může v těchto vztazích mít
- **Seznam atributů** – přehled charakteristik, které popisují konkrétní instanci dané entity
- **Identifikátory entity** – atributy, které mohou sloužit pro jednoznačné určení instance dané entity, případně které umožňují odlišit dvě různé instance téže entity.
- **Informaci o generalizaci/specializaci** – zda má entita nějaký společný nadtyp s jinou entitou, případně zda je nadtypem nějaké jiné entity.

Definice entit jsou součástí sdíleného slovníku pojmů v podniku.

2.2.2.2 Číselníky

Číselník představuje sdílený, jednotně chápaný a interpretovaný popis určité charakteristiky/atributu business objektu. Jedná se o uspořádaný seznam kódů a jim přiřazených významů. Číselníky umožňují popisovat charakteristiky business objektů jednoznačným a srovnatelným způsobem, díky nim je možné business objekty porovnávat a

vytvářet kategorie ekvivalentních objektů z pohledu daného atributu. Číselník obsahuje stabilní a konečný počet položek.

Rozdělení na číselníky a entitními záznamy není vždy jednoznačné, jako pomocná kritéria pro rozlišení je možné použít množství atributů (entitní záznam mívá více atributů), frekvenci změn (číselníky se ve srovnání s entitními záznamy mění méně často), zda má vazba mezi číselníkem a entitním záznamem spíše charakter vztahu nebo zda se jedná spíše o vyjádření vlastnosti.

2.2.3 Metadata

Významnou, avšak často opomíjenou složkou podnikových dat jsou metadata. Lapidárně se označují jako „data o datech“. Metadata popisují obchodní a technické (především z pohledu informačních technologií) aspekty fungování podniku. Marco je rozděluje na [Marco, 2000, str. 49] :

- business metadata – podporují činnost pracovníků v business oblasti
- technická metadata – podporují činnost pracovníků v technické oblasti

2.2.3.1 Business metadata

Business metadata je souhrnný pojem, kterým se označují veškerá strukturovaná data popisující činnost a fungování podniku. Mezi business metadata patří:

- formální definice běžně používaných business pojmů
- vzájemné vazby mezi těmito pojmy
- konceptuální modely a popis procesů
- business popis entit a jejich atributů
- business pravidla a omezení, která se uplatňují na entity
- popis postupů, jak získat informaci o určité entitě (v jakých systémech/aplikacích se nachází data související s danou entitou a jak je interpretovat).

Business metadata slouží potřebám business uživatelů.

Business metadata jsou v podstatě formalizované zaznamenané implicitní znalosti klíčových business pracovníků. Bývají typicky zaznamenána ve formě business dokumentace nebo wiki. Business metadata zajišťují stejné a jednoznačné chápání pojmů a interpretaci dat všemi osobami, které s nimi přijdou do styku. Pokud tato metadata chybí nebo jsou nekompletní, stává se, že:

- stejný pojem se používá pro různé koncepty (pojem zákazník – pro oddělení prodeje ten, kdo provádí objednávku x pro účetní oddělení ten, kdo za objednávku platí x pro oddělení skladového hospodářství ten, komu má být objednávka doručena)
- stejnou hodnotu atributu interpretují různí uživatelé různě (např. atribut cena může někdo chápat jako cenu bez daně a někdo jako cenu s daní)
- každý uživatel získává data k jednotlivým pojmům z jiného zdroje

Chybějící nebo špatná business metadata jsou základní příčinou nedorozumění a představují vážné ohrožení kontinuity činnosti podniku – podnik je závislý na implicitních znalostech svých klíčových zaměstnanců, kteří vědí, „jak se to dělá“. Business metadata umožňují orientovat se v podnikových datech a smysluplně je interpretovat.

2.2.3.2 Technická metadata

Technická metadata popisují informační systém podniku a jeho fungování z technického pohledu. Mezi technická metadata patří:

- Přehled systémů/aplikací v podniku a jejich technických parametrů
- Vzájemné vztahy mezi systémy/aplikacemi (funkční závislosti, datové toky, rozhraní)
- Logické a fyzické datové modely
- Popis hardwarové a síťové infrastruktury

Technická metadata slouží především technickým uživatelům – analytikům, IT architektům, vývojářům. Jsou naprosto nezbytná pro řízení a rozvoj jednotlivých aplikací i informačního systému jako celku. Jenom díky nim je možné analyzovat vzájemné závislosti jednotlivých systémů a to, jaký dopad budou mít změny provedené v jedné aplikaci na ostatní aplikace, případně co je potřeba udělat pro to, aby bylo na určitém místě dosaženo změny. Tato technika se nazývá analýza dopadu (impact analysis). Významným zdrojem technických metadat jsou systémové katalogy relačních databází.

Pro řízení datové základny podniku jsou důležitá především metadata popisující fyzické uložení dat a práci s nimi:

- Forma uložení dat (databáze, textové soubory, XML, proprietární soubory)
- Struktura dat – názvy databázových tabulek, názvy sloupců, datové typy, komentáře sloupců, vazby mezi tabulkami, integritní omezení
- Typy atributů (významové atributy vs. servisní atributy⁴)
- Zdroj, ze kterého data pochází
- Odběratelé dat
- Kde jsou data prezentována business uživatelům (uživatelská rozhraní)

2.2.4 Relační data

Tato data vyjadřují vztahy a vazby mezi business objekty. Mezi relační data patří například:

- **Ad-hoc hierarchie** – umožňují různé pohledy na kategorizaci business objektů nebo hledání objektů s podobnými vlastnostmi. Například v produktových katalozích usnadňují orientaci, hledání substitutů nebo příbuzných produktů (uživatelé, kteří kupují artikl X, kupují často také artikl Y).
- **Householding** – hledání business objektů, které mají mezi sebou určitý vztah relevantní z hlediska činnosti podniku. Například z pohledu banky je relevantní informace o majetkových vazbách mezi právníckými osobami nebo informace o tom, které fyzické osoby jsou členy stejné domácnosti – ekonomicky spjaté skupiny (ESS). S pojmem ESS operují i legislativní normy a například §14 vyhlášky 262/2004 Sb. ESS definuje jako:

„Alespoň dvě osoby, vůči nimž má podnik angažovanost a které jsou navzájem propojeny tím, že jedna z osob ovládá ostatní, nebo jejich vzájemné

⁴ Servisní atributy dokumentují životní cyklus záznamu a jeho atributů. Jsou důležité pro sledování změn záznamů (kdo, kdy a kde a jakým způsobem změnil záznam, odkaz na dřívější verze záznamu)

vztahy jsou takové povahy, že finanční těžkosti jedné z osob mohou způsobit, že alespoň jedna z ostatních nebude řádně plnit svoje platební závazky.“

Pod pojem householding spadá taky sledování majetkové provázanosti společností a z toho plynoucí vliv na rozhodování ve společnostech.

- **Smluvní vztahy** – smlouvy mohou zakládat i poměrně komplexní vztahy/vazby mezi business objekty. Orientace v těchto vztazích je důležitá například při odhalování podvodů nejrůznějšího charakteru.

Informace o vztazích využívají zejména oddělení marketingu a řízení rizik.

2.2.5 Analytická data

Analytická data vznikají odvozením z transakčních dat, takže představují sekundární data. Typické operace, pomocí kterých se transakční data transformují na analytická, jsou výpočty agregací, trendů, průměrných a středních hodnot. Provádí se nad nimi ad-hoc dotazování a používají se pro generování reportů. Do kategorie analytických dat spadají například i výsledky výpočtů z různých prediktivních modelů. Analytická data jsou typicky uložena ve specializovaných systémech – datový sklad, reportingové aplikace.

2.2.6 Administrativní (office) data

Do této kategorie spadají veškerá nestrukturovaná data (volný text, obrázky, schémata), která vytvářejí a používají pracovníci pro provádění svých každodenních úkolů. Jedná se například o:

- e-mailové zprávy
- pracovní dokumenty

Tyto dokumenty zpravidla obsahují odkazy na ostatní typy dat nebo v sobě mohou obsahovat i jejich fragmenty. Administrativní data není možné zařadit do žádné ze čtyř hlavních kategorií, protože jsou buď nestrukturovaná nebo nedostatečně formalizovaná. Administrativní data mohou zachycovat implicitní znalosti klíčových business pracovníků a bývají často důležitým zdrojem business pravidel, která je nutné zohlednit při návrhu MDM řešení.

2.3 Master data management

Předchozí kapitoly popisují neuspokojivý stav panující v podnikových datových základnách (2.1), a zmiňují existenci specifické kategorie dat, kterou lze poměrně přímočaře integrovat – referenční data (2.2). Potenciálním přístupem pro integraci referenčních dat v podniku je Master Data Management.

2.3.1 Definice MDM

Termín MDM je poměrně frekventovaným pojmem a téměř každý si pod ním představuje něco jiného. Na začátek uvedu několik definic, ze kterých se pokusím dobrat společných rysů. Definice uvedené v anglickém jazyce jsou uvedeny v původní nepřeložené podobě.

„Master data management is a collection of best data management practices that orchestrate key stakeholders, participants, and business clients in incorporating the business

applications, information management methods, and data management tools to implement the policies, procedures, services, and infrastructure to support the capture, integration, and subsequent shared use of accurate, timely, consistent, and complete master data.”

[Loshin, 2009, str. 8]

„Master data management (MDM) comprises a set of processes and tools that consistently defines and manages the non-transactional data entities of an organization (which may include reference data). MDM has the objective of providing processes for collecting, aggregating, matching, consolidating, quality-assuring, persisting and distributing such data throughout an organization to ensure consistency and control in the ongoing maintenance and application use of this information.“

[Wikipedia]

„Master data management (MDM) comprises a synoptic set of business proceedings (processes, policies and tools) which centrally and persistently define the key data of an organization (also called Reference data). The objective of MDM is to collect from, and supply to various processes, unique instances of each entity (single version of the truth). Uniqueness in this context is an aspect of data normalization which eliminates outright redundancy due to collection of instances from dissimilar sources. MDM ensures the consistency, accuracy, stewardship and accountability for the core information of the enterprise, enabling organizations to eliminate endless debates about “who’s data is right.”

[Adastra Corporation]

Syntézou lze z těchto definic odvodit, že:

1. MDM je sada postupů a nástrojů
2. Jeho cílem je poskytovat přesná, kompletní a aktuální referenční data
3. Zahrnuje procesy, aplikace i uživatele

V řízení podnikových dat se uplatňuje MDM, pokud:

- Činnost podniku je strukturovaně popsána (procesní model). Existuje katalog definic business entit, které se používají v popisu činnosti podniku. Každá business entita je strukturovaně popsána a součástí popisu je i informace, kde jsou data business entit uložena – **Metadata management (business metadata)**
- Je zdokumentováno, jaká data podniková datová základna obsahuje, a jejich vzájemné závislosti. Jsou zdokumentovány jednotlivé aplikace, jejich rozhraní a vzájemná komunikace – **Metadata management (technická metadata), Enterprise Architecture**
- Je implementován systém, který umožňuje sdílet a efektivně pracovat s referenčními daty, a je vyřešena komunikace tohoto systému s ostatními podnikovými aplikacemi – **MDI hub**
- Jsou nastavena pravidla pro řízení a kontrolu kvality dat, je definováno, co jsou kvalitní data z pohledu business aktivit – **Data Quality Management**
- Jsou nastavena pravidla, jak se s referenčními daty nakládá a jsou stanoveny role, které zodpovídají za kvalitu referenčních dat a dodržování pravidel nakládání s nimi – **Data Governance**

Práce se zabývá technologickou stránkou řešení MDM z pohledu implementace MDI hubu. Procesní a organizační stránku MDM řeší Data Governance, o které je možné se dozvědět více v [Dyché, a další, 2006], [Berson, a další, 2007] nebo [Loshin, 2009]. Problematikou metadat se zabývá [Marco, 2000].

2.3.2 *Motivace pro zavedení MDM*

Mít data v pořádku není samo o sobě přínosem, a málokterý manažer bude ochotný vynaložit prostředky jen kvůli tomu, že dodržovat standardy a best-practices je dobré. Pokud je ale možné kvantifikovat přínos skrze snížení nebo trvalou udržitelnost nákladů na zpracování/správu dat nebo zvýšení spokojenosti zákazníků, je šance prosadit zavedení MDM u business sponzorů. Zavedení MDM v podniku je pouze prostředkem, který má přispět k realizaci obchodních cílů podniku.

Motivační faktory pro zavedení MDM je možné rozdělit na:

- **Interní**
 - o individuální přístup k zákazníkovi (kompletní přehled o zákazníkovi, lepší cílení marketingových kampaní)
 - o přesnější reporting více odpovídající realitě (počty zákazníků, přesnější obrázky o každém individuálním zákazníkovi)
 - o jednodušší vývoj nových aplikací
 - o konsolidace zákaznického kmene při fúzi dvou společností (konsolidace salesforce)
 - o sjednocení pohledu na různé entity napříč společnostmi (klienti, produkty, smlouvy,...)
 - o sjednocení popisných charakteristik reprezentovaných číselníky
- **Externí**
 - o regulační faktory (povinný reporting, udržování historie záznamů)
 - o compliance (soulad s právními předpisy – např. pravidla opt-in, opt-out, souhlas s použitím dat ke specifickému účelu)

Přínosy zavedení Master data managementu v podniku se projevují zprostředkovaně při dosahování strategických cílů, proto bývá obtížné najít business case a kvantifikovat přínos. Prvotní úvaha vždy je: Jaká oblast činnosti podniku je nejvíce závislá na datech a může být potenciálně ohrožena nízkou kvalitou dat buď z pohledu nerealizovaných výnosů nebo z pohledu reálných ztrát (neefektivně vynaložené náklady, ztráty způsobené podvody, pokutami, snížením tržeb v důsledku odchodu zákazníků). Pro představu uvedu několik příkladů, ze kterých lze vycházet:

Lepší znalost zákazníka – díky integraci dat z různých zdrojů je k dispozici kompletní obrázek o interakci zákazníka s podnikem, a na základě těchto dat lze odvodit například hodnotu zákazníka (Customer Lifetime Value), pravděpodobnost nákupu (Propensity To Buy), cross-selling a up-selling příležitosti, je možné lépe zacílit marketingové kampaně (nabízet určité zboží a služby pouze zákazníkům, kteří o ně budou mít s vysokou pravděpodobností zájem), případně lze lépe vyhodnotit rizika spojená se zákazníkem (nejrůznější skóring, detekce podvodného chování).

Lepší obsluha zákazníků – zákazník automaticky předpokládá, že oddělení péče o zákazníky má k dispozici kompletní informace o veškeré interakci mezi ním a podnikem, a očekává, že pracovníci budou schopni přesně a pohotově reagovat na veškeré jeho požadavky (mají po ruce informace o všech jeho objednávkách, fakturách, atd.). Toto má velký vliv na loajalitu zákazníka a bývá to klíčové zvláště u společností, které nabízí větší portfolio služeb (banky, pojišťovny, telekomunikační operátoři). Tento přínos je možné kvantifikovat z pohledu call-centra například jako zkrácení průměrné délky telefonního hovoru s call-centrem nebo snížení průměrného počtu hovorů.

Konsolidace zákaznického kmene při fúzi dvou společností – po fúzi je potřeba konsolidovat data o klientech a jejich produktech ze dvou různých zdrojů do jednotné báze zákaznických dat. Bez této operace není možné zodpovědět, kolik unikátních klientů má nový subjekt vzniklý fúzí, protože samozřejmě existují osoby a firmy, které byly klienty obou původních subjektů. Zároveň je tato operace nezbytná proto, aby bylo možné obsloužit klienty obou původních subjektů na společných přepážkách pomocí jednotného systému. Tento problém typicky řeší společnosti v oblasti bankovníctví, pojišťovnictví, telekomunikací a utilitách (distribuce el. energie, plynu, vody). Příkladem z českého prostředí jsou například bankovní fúze – fúze Živnobanky a HVB Bank (UniCredit Bank), fúze ČSOB a Poštovní spořitelny nebo fúze Raiffeisen Bank a eBanky. Ze slovenského prostředí lze uvést příklad fúze zdravotních pojišťoven Dohvera a Apollo.

Konzistentní reporting – duplicita a vzájemná nekonzistence mezi stavem reportovaným a skutečným dokáží znehodnotit i ten nejpropracovanější report. Manažeři činí na základě reportů rozhodnutí, a pokud nejsou reporty přesné, nemusí mít přijatá opatření očekávaný efekt. Pokud manažeři ztratí důvěru v přesnost a pravdivost reportingu, je to závažný problém, který komplikuje rozhodovací proces a vede k hledání postranních řešení (viz. v úvodu zmiňovaná „excelizace“). Protože kdo není schopen kontrolovat, nemůže efektivně řídit. Dalším jevem, který narušuje důvěru v reporting, je vzájemný nesoulad mezi různými reporty popisujícími stejnou oblast/jev.

Dodržení regulatorních požadavků – existují zákonné normy, které musí podniky dodržovat. Tyto normy se týkají například zavedení jednoznačné identifikace klientů finančních institucí (zákony proti legalizaci výnosů z trestné činnosti – v české legislativě zákon č. 284/2004 Sb.), korporátního finančního reportingu a oceňování rizik (Sarbanes-Oxley Act, Basel II) či opt-in/opt-out marketingových sdělení a pravidla nakládání s osobními údaji (zákony na ochranu osobních údajů – v české legislativě zákon č. 101/2000 Sb.)

Pružnější a rychlejší zavádění nových produktů a služeb – pokud je k dispozici centralizovaný katalog business entit a jejich atributů, je snadné říci, zda jsou k dispozici data potřebná pro realizaci určitého záměru (produktu či služby), případně co je potřeba udělat pro získání těchto dat. Pokud se používá konzistentní způsob práce s daty, předchází se redundancím a „nežádoucí lidové tvořivosti“ – je poměrně běžné, že se kvůli nové službě nasadí nová aplikace, která funguje jako další informační silo. Příkladem takové nové služby může být například „Express půjčka“ – pokud má banka dostatek údajů pro předběžný skóring klienta a pro navržení úvěrového rámce, může ji zrealizovat.

Snížení nákladů na uchování a správu dat – redundantní uložení dat s sebou nese vícenáklady, ať už v podobě diskového prostoru, prostoru pro servery včetně klimatizace a zabezpečení, nadbytečných licencí na software (typicky databáze nebo ETL nástroje) nebo v podobě úsilí zaměstnanců, které musí vynaložit na udržování dat (např. pracovníce na přepážce banky, která musí znovu vyplnit veškeré identifikační a kontaktní údaje zákazníka, když mu zakládá nový produkt).

Monitorování a udržování kvality dat – umožňuje předcházet potenciálním ztrátám či nákladům v budoucnosti. Pokud například nejsou k dispozici spolehlivé kontaktní údaje, není možné s klientem efektivně komunikovat. U větších podniků tak může například nedoručená pošta představovat nemalý náklad. Pokud se kvalita dat sleduje a ví se, že některé údaje u konkrétního klienta nejsou k dispozici, je možné např. prostřednictvím front-desk aplikace informovat pracovníce na přepážce, aby tyto údaje od klienta vyžádala při jeho další návštěvě.

Jednodušší integrace nových aplikací – není potřeba opakovaně vynalézat kolo. Pokud je práce s daty řešena standardní cestou, šetří se tím čas a tedy i náklady na vývoj. Pokud může aplikace využít pro svou činnost již existující aplikační infrastrukturu nebo datové zdroje, snižují se náklady na její integraci s ostatními systémy a náklady na následný provoz a údržbu.

Výstavbou business-case pro implementaci MDI hubu se zabývá například [Dyché, a další, 2006], [Putchala, 2007], [Berson, a další, 2007].

2.3.3 *Motivace pro nezavedení MDM*

V minulé podkapitole je uvedena řada důvodů pro zavedení MDM. Pro zachování objektivity jsou v této podkapitole uvedeny alespoň dva základní případy, kdy nemá smysl MDM zavádět.

Náklad investice do zavedení MDM převyšuje její potenciální přínos – implementace MDM je investice jako každá jiná a proto se při jejím schvalování vyhodnocuje její návratnost. Pokud náklady na zavedení (návrh řešení, změna IT infrastruktury a procesů, vytvoření potřebné aplikační podpory) a provoz MDM převyšují finanční vyčíslení pozitivního dopadu potenciálních benefitů uvedených v předchozí podkapitole anebo doba návratnosti je příliš dlouhá, nemá smysl se pouštět do zavádění MDM.

Firma nenaráží na problémy s daty, které má MDM řešit – v takovém případě není nutné o MDM uvažovat. Tato situace je typická spíše pro malé firmy, které se nepotýkají s problémem jednoznačné identifikace zákazníků napříč systémy a otázka synchronizace datových zdrojů pro ně není důležitá (třeba proto, že si vystačí s 1-2 core systémy, nevyžadují speciální reporting, apod.), případně nepodléhají žádným regulatorním požadavkům, které by vyžadovaly zavedení MDM.

3 Použité techniky

V této kapitole jsou popsány techniky, které jsou použity jednak při návrhu referenční architektury MDI hubu a jednak v rámci aktivit prováděných při samotné implementaci.

Mezi tyto techniky patří:

- Procesní modelování – popisuje dynamický pohled na řešení
- Datové modelování – popisuje statický pohled na řešení
- Profiling a explorativní analýza dat
- Řízení uživatelských požadavků – popisuje uživatelský pohled na řešení
- Řízení datové kvality
- Metodické řízení vývoje

3.1 Procesní modelování

Procesní modelování je technika, která se používá pro formální a konzistentní popis procesů. Norma ISO 9000:2005 definuje proces jako sadu souvisejících nebo vzájemně na sebe působících aktivit, které transformují vstupy na výstupy. Jedním z cílů procesního modelování je strukturovaně popsat aktivity nutné pro vytvoření výstupu takovým způsobem, aby bylo možné je provádět opakovaně standardním způsobem s dosažením očekávaného výstupu nezávisle na tom, kdo je provádí. Způsobem popisu procesů a použitím grafické notace se zabývají standardy:

- Business Process Modelling Notation (BPMN)⁵
- Unified Modelling Language (UML)⁶ a jeho rozšíření pro modelování business procesů [Eriksson, a další, 2000], [Engels, a další]

Procesní modelování je použito pro popis workflow v MDI hubu. Pro popis je použita BPMN procesní notace. V návrhu metodiky implementace MDI hubu není použita grafická notace, protože aktivity při vývoji a implementaci neprobíhají sekvenčně, ale spíše paralelně, existují mezi nimi různé závislosti a mohou se opakovat. Místo grafické notace je použit strukturovaný popis aktivit, které vytváří projektové výstupy. U dílčích aktivit se předpokládá, že člověk v roli, která vykonává danou aktivitu, má oborovou znalost potřebnou pro úspěšné vytvoření očekávaného výstupu. Procesním modelováním se podrobněji zabývá například [Řepa, 2007].

Procesy hrají v MDM důležitou roli. Jejich popisy jsou jedním z východisek pro identifikaci business objektů, se kterými podnik pracuje. Nastavení procesů má přímý dopad na kvalitu a způsob práce s referenčními daty v podniku. Problematikou přenastavení procesů při zavádění MDM v podniku se zabývá Data Governance (viz. odkazy na konci kapitoly 2.3.1).

3.2 Datové modelování

Datové modelování je technika, která se používá pro formální popis strukturovaných dat skrze datové modely. Model je obecně obraz reality, který zachovává pouze

⁵ <http://www.bpmn.org/>

⁶ <http://www.uml.org/>

charakteristiky/atributy podstatné pro daný účel. Datový model je grafickým popisem strukturovaných dat o business entitách a jejich vzájemných vztazích. Je pro datovou základnu tím, čím je pro budovu plán či technický výkres.



Obrázek 1 – Modely a kompromisy při jejich realizaci [Simsion, a další, 2005]

Datové modelování se používá především kvůli:

- podpoře porozumění datové reprezentaci business objektů a konceptů, jejich vnitřní struktury a vzájemným vztahům
- omezení redundance v definici dat (různá místa a způsoby reprezentace stejných business konceptů) a podpoře znovupoužitelnosti dat
- komunikaci návrhu datové základny v rámci týmu a s business uživateli (je tak možné ověřit, zda model správně reprezentuje realitu)
- ověření logické správnosti návrhu (případně změn) datové základny aplikace, splnitelnosti business požadavků s ohledem na dostupná data, kontrole dodržení business pravidel, kontrole vzájemné konzistence různých pohledů na aplikaci/informační systém
- dokumentaci aktuálního stavu datové základny aplikací a návrhu možných změn
- generování odvozených modelů a jiných artefaktů (databázové tabulky, procedury pro jejich plnění, definice rozhraní a mapování, obrazovky uživatelského rozhraní atd.)⁷ – toto generování lze realizovat automatizovaně s použitím CASE nástrojů (Sybase PowerDesigner, CA Erwin DataModeller, SparxSystems Enterprise Architect apod.)

Při modelování se používají různé úrovně detailu, aby se zachovala čitelnost, srozumitelnost a celková konzistentnost modelu. Norma ANSI75 definuje tři úrovně podrobnosti modelů [Summary of current work ANSI/X3/SPARC/study group: database systems, 1974]:

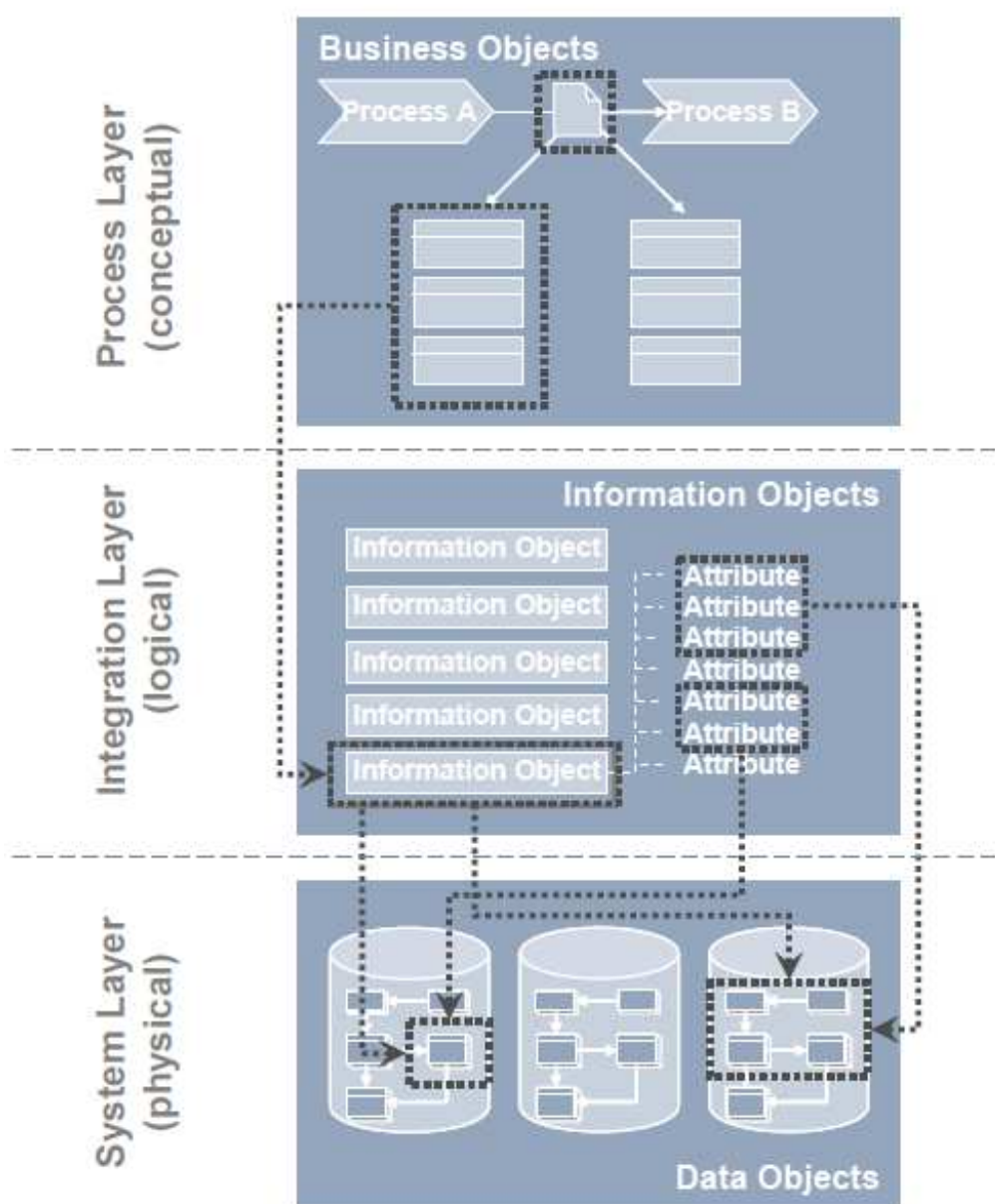
⁷ Na této myšlence je založena vývojová disciplína Model Driven Engineering (MDE). Úvod do problematiky MDE lze nalézt v [Schmidt, 2006]

- **Konceptuální** – představuje základní zobrazení business entit a konceptů, vzájemných vztahů mezi nimi a jejich nejdůležitějších atributů. Typickým představitelem konceptuálního datového modelu je například entity-relationship diagram (ERD). Konceptuální model slouží především business analytikům a uživatelům. Při jeho vytváření se používá top-down přístup. Používá se pro popis korporátního datového modelu [Kowalski, 1999].
- **Logická** – je určena pro návrh logické reprezentace dat, která je nezávislá na konkrétní platformě/databázi. Obsahuje větší úroveň detailu než konceptuální model – řeší se způsob realizace vazeb (vazební tabulky, cizí klíče), konceptuální atributy jsou dekomponovány na atomické datové atributy⁸. Na této úrovni se přidávají i servisní atributy (metadata) pro potřeby auditu, historizace apod., které jsou odvozeny z nefunkčních požadavků (viz. kapitola 3.4) Pro pojmenování entit a atributů ještě používá přirozený jazyk. Pro dosažení maximální konzistence se používá technika normalizace datového modelu. Logický datový model slouží především datovým analytikům. Při použití top-down přístupu vzniká konkretizací konceptuálního modelu, při použití bottom-up přístupu vzniká zpětnou normalizací fyzického modelu a převedením jmen atributů a entit/tabulek do přirozeného jazyka.
- **Fyzická** – je určena pro popis realizace uložení dat popsaných logickým modelem v konkrétním databázovém systému (DBMS). Jedná se o rozpra na úroveň databázových tabulek (entity) a sloupců (atributy). Pro pojmenování tabulek a sloupců se již používají zkratky a kódy odpovídající konvencím zvoleného DBMS. Fyzický datový model dále obsahuje informace o datových typech, integritních omezeních (primární klíče, cizí klíče, unikátnost, neprázdnost), dalších databázových artefaktech (triggery, procedury, pohledy apod.) a informace o zvolených proprietárních optimalizacích uložení dat (použití indexů, partitioningu, databázových pohledů, LOB⁹). Z důvodů optimalizace uložení dat může také dojít k denormalizaci oproti logickému modelu. Fyzický datový model slouží především databázovým specialistům.

Vztah mezi modely na jednotlivých úrovních podrobnosti (vrstvách) vyjadřuje Obrázek 2.

⁸ Atomický datový atribut již není možné dekomponovat na podrobnější úroveň, bývá v databázi realizován jako sloupec.

⁹ Large objects – forma uložení většího objemu dat ve formě databázového sloupce. Oproti běžným databázovým sloupcům má určitá omezení ve fyzické implementaci (nedá se např. indexovat ani používat přímo ve WHERE podmínkách SQL dotazů)



Obrázek 2 - Vztah mezi vrstvami modelu [Harmonizing company-wide Information Objects, 2008]

Jednotlivé úrovně modelu by měly být mezi sebou konzistentní. Základní pravidla kontroly konzistentnosti úrovní modelů lze lapidárně shrnout větou: „V modelu se nám nic neztrácí ani se nám tam nic samo od sebe neobjevuje“. Existují dobře zdokumentované postupy [Simsion, a další, 2005], jak lze kontrolovat vzájemnou konzistentnost různých vrstev modelu, případně jak provádět transformace při přechodu mezi vrstvami.

Datové modely bývají cenným zdrojem informací o podnikové datové základně, protože dokumentují způsob uložení dat v dílčích aplikacích/systémech. Tyto lokální datové modely vznikaly buď při návrhu aplikací nebo jako součást dokumentace. Lokální datové modely je možné i odvodit metodou tzv. reverse-engineeringu [Reverse Engineering and Design

Recovery: A Taxonomy, 1990], [Allen, a další, 2005] přímo z relačních databází (s použitím CASE nástrojů – např. Sybase PowerDesigner¹⁰).

Technika datového modelování je použita pro návrh/odvození master data modelu, návrh logického a fyzického uložení referenčních dat v MDI hubu.

3.3 Profiling a explorativní analýza dat

Pojmem profiling se souhrnně označuje sada technik a postupů, jejichž cílem je prověření a zdokumentování současného stavu dat ve zdrojových systémech. Ve své podstatě se jedná o součást reverse-engineeringu. V rámci profilingu se zkoumají zdrojová data a veškerá dostupná metadata s nimi související (katalogy metadat v DBMS, dokumentace formátu souborů, dokumentace/manuály zdrojových systémů). Také je možné analyzovat kód, který obsluhuje zkoumaná data, a uživatelská rozhraní, přes která uživatelé nahlíží na data v systémech. Při profilingu se pracuje buď s kopiemi zdrojových úložišť dat nebo s jejich extrakty. Pro analýzu se používají především různé statistické metody a ověřují se také vzájemné závislosti mezi různými sadami atributů.

Při profilingu se shromažďují zjištění ve dvou oblastech:

- **Technologická oblast** – zabývá se technologickými aspekty uložení dat (specifikace technologického řešení datových zdrojů (databáze, XML, textové a binární soubory), pojmenování sloupců a tabulek, datové typy, šířky sloupců, dodržení referenční integrity a dalších integritních omezení, objemy dat apod.)
- **Business oblast** – řeší význam uložených dat a jejich interpretaci (vyplněnost/nevyplněnost, popisy a vysvětlení datových elementů, formát dat (například data (sg. datum), čísla apod.), použité číselníky apod.), může také identifikovat business pravidla ovlivňující data

Výsledky profilingu je možné prohlížet ze dvou různých dimenzí:

- **Po zdrojových systémech** – tento pohled se používá při provádění profilingu
- **Po entitách** – tento pohled se používá při sumarizaci výsledků profilingu, při hledání problémů datové kvality a při odvozování mappingu mezi datovými strukturami zdrojových systémů a datovými strukturami MDI hubu.

Výstupem profilingu jsou informace:

- které systémy obsahují data o business entitách
- jaká data o business entitách jsou k dispozici
- jaké atributy lze použít pro identifikaci business entity
- jakým způsobem se transformuje struktura zdrojových dat do tzv. kanonické formy (Master data model)
- souhrnné statistiky datových tabulek a atributů (počty záznamů, vyplněnost)
- informace o vazbách mezi tabulkami a integritních omezeních
- informace o použitých číselnících
- informace o souvisejících business pravidlech, která ovlivňují hodnotu atributů nebo jsou jimi ovlivňována

¹⁰ <http://www.sybase.com/products/modelingdevelopment/powerdesigner>

Profiling úzce souvisí také s analýzou datové kvality. Otázkou profilingu a explorativní analýzy dat se zabývá například [Allen, a další, 2005], [Kotek, 2008].

3.4 Sběr a analýza uživatelských požadavků

Aby bylo možné naplnit očekávání zákazníka spojená se zavedením master data řešení, je potřeba znát jeho potřeby a požadavky. Základní informace o zákaznických očekáváních jsou součástí zadávací dokumentace projektu. Tato očekávání je potřeba transformovat do strukturované podoby, ověřit jejich úplnost a doplnit je o tzv. implicitní (nevyslovená) očekávání.

Požadavky je možné rozdělit na funkční a nefunkční:

- **Funkční požadavky** – funkční požadavky popisují funkce vyvíjeného systému, které musí tvůrce systému implementovat, aby uživatelé systému mohli plnit svoje úkoly a byly tak uspokojeny jejich (business) požadavky [Wiegers, 2003]. Funkční požadavky jsou zohledněny v návrhu systému.
- **Nefunkční požadavky** – popisují chování a parametry systému, nikoli funkcionalitu (co by měl systém dělat, jak by se měl chovat v určitých situacích). Implicitní očekávání spadají do této kategorie. Nefunkční požadavky jsou zohledněny v architektuře systému. Typické nefunkční požadavky jsou logování a audit, konkrétní cílová platforma, škálovatelnost, SLA (doba odezvy, dostupnost, spolehlivost, aktualizace), rozšiřitelnost, dokumentace, způsob komunikace s ostatními systémy.

Uživatelské požadavky říkají, co se má udělat, nikoli jak se to má udělat (to je předmětem návrhu a implementace).

Ke sběru uživatelských požadavků slouží různé metody:

- **Analýza dokumentace** – slouží mimo jiné i k odhalení implicitních (nevyřčených) požadavků, funkčních i nefunkčních
 - o **Zadávací dokumentace** – obsahuje základní informace o tom, co zákazník požaduje
 - o **Business procesy** – z této dokumentace je možné získat informace o tom, jak podnik realizuje svoji hlavní činnost (business), s jakými business entitami operuje, jaké informace při své činnosti potřebuje a jaká data má k dispozici, jaká business pravidla se v podniku uplatňují a jak se tato pravidla projevují v datech.
 - o **Zdrojové systémy** – z této dokumentace je možné získat informace o tom, co je v jednotlivých aplikacích/systémech uloženo, při jakých činnostech se používají, jakým způsobem s nimi uživatelé pracují, jaké technické prostředky aplikace využívají.
 - o **Architektura IS/IT** – z této dokumentace je možné zjistit, jaké technologie a platformy se v podniku používají, informace o hardwarové a síťové infrastruktuře, jak spolu jednotlivé zdrojové systémy komunikují.
- **Interview** – při nich se zjišťují požadavky osob reprezentující relevantní zájmové oblasti (tzv. stakeholders – business sponzoři, management, uživatelé, architekti IS, analytici, správci aplikací) a hledají se odpovědi na otázky vzešlé z analýzy zadávací

dokumentace. Umožňují doplnit „bílá místa“ nalezená v dokumentaci. Interview se typicky účastní malé množství osob (maximálně 2 za zákazníka a 2 za dodavatele – jeden vede interview a druhý zapisuje), větší množství osob komplikuje efektivní komunikaci a získávání informací. Pro křížové ověření a konfrontaci získaných informací se používá workshop (viz dále).

- **Pozorování** – při sledování toho, jak uživatelé plní své rutinní úkoly, je možné odhalit informace, které se nedají vyčíst z dokumentace (zvyklosti v používání aplikací, dodržování nepsaných pravidel apod.)
- **Workshop** – slouží spíše pro ověření požadavků než pro jejich sběr. Při workshopech se provádí review závěrů vyvozených v průběhu sběru a analýzy požadavků zákazníkem a ověření funkční specifikace. Workshopu se typicky za stranu dodavatele i zákazníka účastní více lidí (

Uživatelské požadavky je vhodné konfrontovat s best-practice i se zkušenostmi z předcházejících projektů a tím předcházet známým rizikům. Technikami práce s uživatelskými požadavky se zabývá [User requirements analysis_A review of supporting methods, 2002].

Uživatelské požadavky lze formálně modelovat pomocí use-cases diagramů [Cockburn, 2000]. Tyto diagramy zachycují činnosti/akce, které aktéři (uživatelé, ostatní aplikace) provádí na implementovaném systému (MDI hubu). Use-case popisuje chování systému v reakci na podnět, který pochází z jeho okolí. Use-cases tvoří součást funkční specifikace. Díky formálnosti jejich zápisu je možné provést mapování mezi use-cases, navrženými funkcemi systému a jejich fyzickou realizací (modul systému, metoda/funkce, výkonný kód). Řadu užitečných informací k praktickému používání use-case UML lze nalézt v [Ambler]

Uživatelské požadavky je nutné zaznamenávat a formálně dokumentovat. Jenom tak je možné zákazníkovi dokladovat, že dodaný systém splňuje to, na čem se s dodavatelem dohodl. Provedením analýzy uživatelských požadavků nepřebírá dodavatel zodpovědnost za dosažení efektů očekávaných zákazníkem, pouze minimalizuje riziko, že zákazníkova očekávání nebudou naplněna.

Uživatelské požadavky jsou evidovány v tzv. katalogu uživatelských požadavků. Každý požadavek je strukturovaně popsán a sleduje se stav jeho řešení.

3.5 Datová kvalita

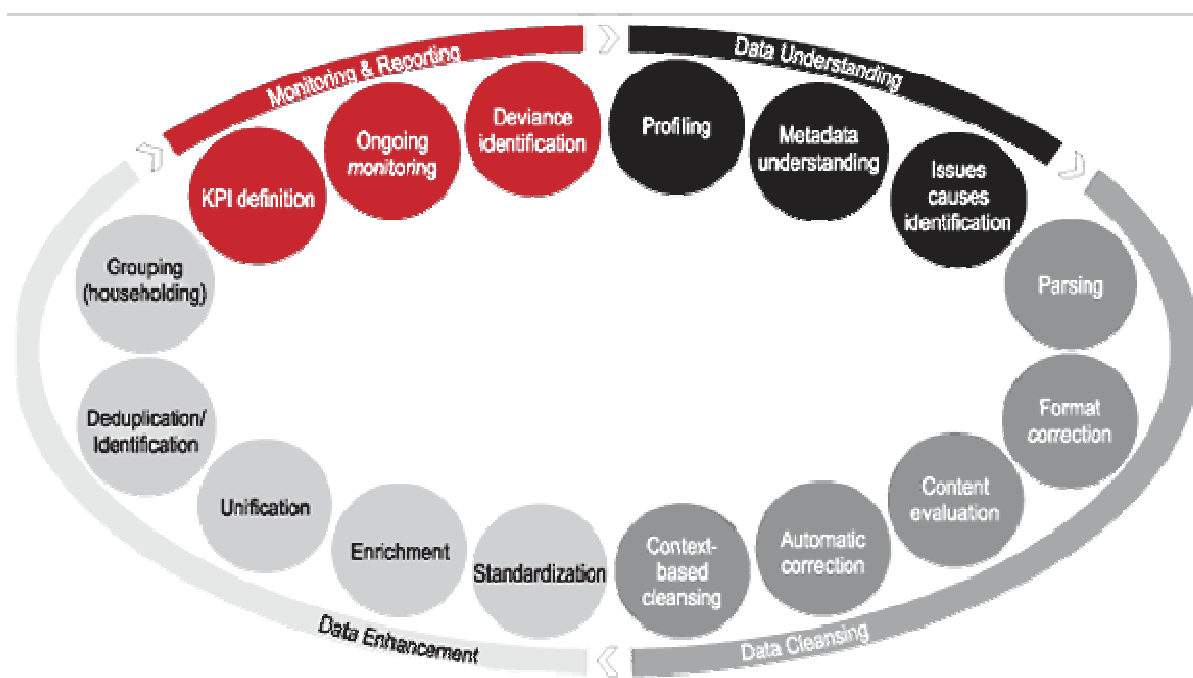
Datová kvalita je svojí povahou abstraktní pojem, jehož význam lze vyjádřit tvrzením: „Data jsou kvalitní natolik, nakolik vyhovují svému účelu a použití.“ Účel a použití jsou veličiny, které se v čase vyvíjejí. Je proto nezbytné vydefinovat určitou základní sadu charakteristik, které se mají naplnit. Tato sada se skládá z požadavků MDI hubu a z vybraných uživatelských požadavků. Pokud není datová kvalita dostatečně definována, stává se zakládlem a zástupným problémem, který s oblibou ostatní oddělení označují za příčinu svých případných neúspěchů.

Z pohledu MDI hubu je klíčové, aby referenční data byla ve stavu, který umožňuje bezpečné a spolehlivé párování záznamů (unifikaci), tj. mají správně vyplněny a standardizovány

klíčové identifikační atributy. Z pohledu zdrojových systémů je klíčové, aby data co nejlépe vyhovovala účelu, pro který jsou v nich uložena a tomu, jak se používají.

Obrázek 3 zachycuje vyčerpávajícím způsobem cyklus řízení datové kvality z pohledu společnosti Adastra. V řízení datové kvality lze vymezit dvě hlavní oblasti, které mají přímý dopad na fungování MDI hubu:

- **Čištění** (Data Cleansing, Standardization) – transformace vedoucí ke zlepšení datové kvality a rozlišení kvalitních a méně kvalitních dat. U méně kvalitních dat bývá též identifikován důvod, proč byla data zařazena do této kategorie. Operace probíhající při čištění jsou podrobněji vyjmenovány v kapitole 4.3.4 v části „Čištění, unifikace a konsolidace dat“. O různých formách datové nekvality a jejich nápravě pojednává [Data Cleaning: Problems and Current Approaches, 2000]
- **Unifikace** (Unification, Deduplication) – proces identifikace záznamů, které reprezentují stejný reálný objekt. Platí, že čím kvalitnější jsou data, tím přesnější a úplnější jsou výsledky unifikace. Unifikace je založena tzv. business klíči – **business klíč** tvoří sada významových atributů, která umožňuje s poměrně vysokou spolehlivostí spojit dva instanční záznamy reprezentující stejný reálný objekt, případně rozlišit od sebe instanční záznamy reprezentující různé reálné objekty. Unifikaci definuje sada business klíčů a pravidel, jak tyto klíče porovnávat.



Obrázek 3 - Cyklus řízení datové kvality [Adastra Corporation]

Při čištění je hodnocena kvalita jednotlivých atributů z pohledu jeho použitelnosti a automatických oprav. Tento proces se nazývá **labeling datové kvality**. Při labelingu datové kvality se jednotlivé atributy a instance označují štítky, které představují business pohled na stav kvality daného atributu/instance. Základní rozlišované kategorie jsou:

- OK – atribut splňuje veškeré implicitní kontroly (vyplněnost, správný formát, platnost podle číselníku, vzájemný soulad s ostatními atributy instance), a business pravidla vztahující se na daný atribut.
- CORRECTED – atribut nesplňuje implicitní kontroly nebo business pravidla, ale je možné ho v procesu čištění bezpečně automaticky opravit (např. na základě korekčních číselníků nebo odvození z ostatních atributů). Co se považuje za bezpečnou opravu, bývá definováno business pravidly pro čištění dat.
- NOT_OK – atribut nesplňuje kontroly nebo business pravidla, a není možné ho bezpečně automaticky opravit. Atribut musí být opraven manuálně.
- EMPTY – atribut je nevyplněn.

Kategorie CORRECTED a NOT_OK mají ještě detailní úroveň, na které je přesně vymezena příčina přiřazení štítku (např. atribut IČO má přidělen štítek NOT_OK, protože je vzájemně nekonzistentní s atributem NÁZEV_FIRMY). Labeling datové kvality je důležitý pro reporting datové kvality a pro identifikaci podstaty nekvality atributu. Labeling se používá pro označení tzv. nálezů datové kvality. **Nález datové kvality** je v podstatě jakékoli zjištění týkající se kvality zdrojových dat, které má určitý business význam/dopad a které může vyžadovat nějakou akci/řešení ze strany uživatele (např. doplnění chybějícího údaje nebo oprava chybného údaje).

O datové kvalitě a její systematické analýze je možné se více dozvědět v [Kotek, 2008].

3.6 Metodické řízení vývoje

Metodické řízení vývoje je nezbytné, zvláště pokud na projektu participuje více pracovníků. Usnadňuje jim společné fungování a kooperaci při vytváření výstupů. Jedná se v podstatě o souhrn pravidel, která se na projektu dodržují. Tato pravidla by měla být na jednu stranu dostatečně flexibilní, aby nepřekážela efektivnímu plnění úkolů, na druhou stranu dostatečně konkrétní, aby si je různí členové týmu nevykládali různě. Metodika má mimo jiné za úkol omezit nežádoucí projevy „lidové tvořivosti“ a napřít tvořivost členů týmu správným směrem.

Mezi přínosy uplatnění metodického řízení patří, že:

- Jednotliví pracovníci mluví „stejnou řečí“, používají stejné označení pro koncepty užívané při vývoji
- Při obměně v projektovém týmu se noví členové rychleji zorientují
- Výstupy jednotlivých činností mají konzistentní podobu bez ohledu na to, kdo je vytváří
- Metodika v sobě kumuluje poznatky z projektů a užitečné znalosti pracovníků, kteří na nich participovali
- Opakovaně se vyskytující problémy se řeší stejným (konzistentním) způsobem
- Při vytváření výstupů je zohledňována jejich znovupoužitelnost

V rámci metodického řízení se stanovují:

- **Sdílený slovník pojmů**
- **Jmenné konvence**
 - o způsob pojmenování entit, atributů, databázových tabulek, sloupců
 - o adresářové struktury, způsob pojmenování souborů

o pojmenování ostatních vývojových artefaktů

- **Způsob práce se sdílenými projektovými prostředky** (databáze, servery, modely, dokumenty)
- **Pravidla vytváření dokumentace** (co se dokumentuje, kde se dokumentuje, v jakém jazyce se dokumentuje, v jakém formátu se dokumentuje)
- **Šablony výstupních dokumentů**
- **Postupy „krok za krokem“ popisující opakované standardní činnosti** (např. profiling datových zdrojů)
- **Práce s verzemi výstupů**

Mezi podpůrné nástroje, které pomáhají s uplatňováním metodiky při realizaci projektu, patří:

- **Aplikace pro správu obsahu** (aplikace typu wiki¹¹, např. Confluence¹²) – slouží pro aktivní správu a sdílení dokumentace a znalostí
- **Aplikace pro správu verzí** (Subversion¹³, CVS¹⁴) – slouží pro správu verzí výstupů (pracovní dokumenty, vyplněné formuláře s výstupy analýz, zdrojové kódy apod.)
- **Aplikace pro sledování požadavků** (Bugzilla¹⁵, Jira¹⁶) – umožňují sledovat průběh řešení požadavků (uvnitř týmu i vůči zákazníkovi), způsob jejich vyřešení a osoby zodpovědné/podílející se na řešení. Požadavkem obecně může být cokoli – instrukce pro autora nějakého výstupu s žádostí o opravu/rozšíření, požadavek na doplnění určité funkčnosti do vyvíjené aplikace apod.

Metodika není dogma, ale v průběhu projektu se obohacuje – pokud má smysl některou oblast projektových činností omezit určitými pravidly nebo ji řešit standardní cestou, metodika je rozšířena nebo přizpůsobena prostředí, ve kterém projekt probíhá.

¹¹ Wiki - označení webů (nebo obecněji hypertextových dokumentů), které umožňují uživatelům přidávat obsah podobně jako v internetových diskusích, ale navíc jim také umožňují měnit stávající obsah [zdroj: Wikipedia – The Free Encyclopedia]

¹² <http://www.atlassian.com/software/confluence>

¹³ <http://subversion.tigris.org/>

¹⁴ <http://savannah.nongnu.org/projects/cvs>

¹⁵ <http://www.bugzilla.org/>

¹⁶ <http://www.atlassian.com/software/jira/>

4 Master Data Integration hub

Master Data Integration hub (MDI hub) je konkrétní realizací systému, který má sloužit jako jednotné uložení referenčních dat a poskytovat služby pro práci s referenčními daty. Nejprve jsou shrnuty **základní požadavky** na MDI hub, které se opakují napříč realizovanými projekty. Tento seznam je možné použít jako východisko pro upřesňování zákaznických požadavků. Na základě základních požadavků je navržena referenční architektura MDI hubu, která je popsána v další kapitole. Jsou zmíněny alternativní přístupy ke zvolené formě realizace MDI hubu (registry, hybridní paradigma). Dále jsou popsány komponenty referenční architektury.

4.1 Základní požadavky na MDI hub

Požadavky jsou jedním ze vstupů, který je zohledněn při návrhu architektury výsledného řešení MDI hubu. Kromě specifických uživatelských požadavků existuje určitá sada požadavků, která se vyskytuje napříč projekty implementace MDM. Tato sada je označena jako základní požadavky.

Základní požadavky, které jsou na MDI hub kladeny, jsou:

- **Dávková synchronizace referenčních dat se zdrojovými systémy**
 - Import referenčních dat ze zdrojových systémů
 - Rozdílové snímky (inkrementy)
 - Plné snímky
 - Export referenčních dat ve struktuře zdrojových systémů
 - Export referenčních dat v kanonické struktuře
- **Online synchronizace referenčních dat se zdrojovými systémy**
 - Validace referenčních záznamů
 - Identifikace referenčních záznamů (vůči bázi referenčních dat)
 - Vkládání nových referenčních záznamů
- **Čištění, ověření a konsolidace referenčních dat**
- **Uživatelské rozhraní pro prohlížení a správu referenčních dat**
- **Sledování datové kvality a řešení nálezů datové kvality**
 - Souhrnný reporting
 - Reporting pro opravu záznamů
- **Správa číselníků**
 - Import externích číselníků
 - Údržba číselníků (aktualizace)
 - Synchronizace číselníků s ostatními systémy
 - Historizace číselníků
- **Správa nálezů datové kvality**
 - definice workflow pro identifikaci, evidenci a řešení nálezů datové kvality
 - uživatelské rozhraní pro práci s nálezy datové kvality (není podmínkou)
- **Historie změn v referenčních záznamech**
- **Podpora pro definování a spouštění workflow**

4.2 Referenční architektura MDI hubu

Návrh architektury vychází ze základních požadavků a z uživatelských požadavků. Při návrhu architektury je nutné průběžně ověřovat, že skutečně realizuje vymezené požadavky. Nejprve je ale třeba zvolit, jakou formou bude MDI hub realizován. Kapitola obsahuje přehled navrhované architektury, její dílčí komponenty jsou popsány v následující kapitole (4.3).

4.2.1 Paradigmata architektury MDI hubu

Existují různé přístupy k návrhu architektury MDI hubu, jejich základní rysy shrnují tři různá paradigmata (architekturní styly) realizace MDI hubu [Dyché, a další, 2006, str. 47-50], [Loshin, 2009, str. 165-171]:

- **Registry (virtuální)** – hub slouží jako index, který odkazuje přímo na instance záznamů ve zdrojových systémech. Hub obsahuje pouze minimální sadu atributů nutnou pro identifikaci instančních záznamů (tzv. párovací klíče) a informaci o seskupení instančních záznamů. Každý nově vytvořený instanční záznam v systému projde zařazením do indexu. Při dotazu se konsolidovaný záznam business objektu vytváří za běhu z jednotlivých instančních záznamů uložených v různých systémech pomocí tzv. distribuovaného dotazu (federated query). Toto řešení je typické minimálním objemem dat uložených v hubu. S referenčními záznamy se pracuje pouze ve zdrojových systémech. Výhodou je minimální zásah do zdrojových systémů, nevýhodou je vyšší výpočetní náročnost (neustálé přepočítávání konsolidovaných záznamů za běhu) a vyšší nároky na komunikaci mezi zdrojovými systémy a hubem. Autoritativním zdrojem dat o entitách je některý ze zdrojových systémů.
- **Repository (transakční hub)** – hub slouží jako jediné místo, kde jsou uložena referenční data. Každý business objekt má pouze jediný instanční záznam, který perzistuje v hubu. Aplikace přistupují k referenčním datům a provádějí změny pouze přes služby hubu. Transakční hub se nejprve naplní referenčními daty z jednotlivých systémů, ta se zkonsolidují, vytvoří se konsolidované záznamy, a aplikace pak dále pracují již jen s těmito konsolidovanými záznamy. S referenčními záznamy se pracuje pouze v hubu. Ve srovnání s typem registry je náročnější na analýzu a implementaci. Model transakčního hubu je také více zatížen množstvím atributů, které jsou specifické třeba pouze pro jeden zdrojový systém. Výhodou je, že transakční hub je jediné místo uložení referenčních záznamů (snížení objemu zálohovaných dat, snadnější zabezpečení a řízení přístupu k datům), nevýhodou je nutnost velkého zásahu do zdrojových systémů a následně jejich úplná závislost na transakčním hubu. Transakční hub může fungovat i v režimu s replikovanými referenčními daty, kdy si jednotlivé aplikace ponechávají synchronizované kopie referenčních dat pouze pro čtení a veškeré úpravy referenčních dat se odehrávají v transakčním hubu. Autoritativním zdrojem dat o entitách je hub.
- **Hybridní řešení** – hub v sobě drží párovací klíče instančních záznamů (registry), navíc ale obsahuje i některé (nejfrekventovanější/nejčastěji používané) atributy instančních záznamů (repository). Specifické atributy se ponechávají pouze ve

zdrojových systémech. Dochází k obousměrné synchronizaci mezi hubem a zdrojovými systémy. Výhodou je relativní nezávislost zdrojových systémů a hubu, hub není zatížen komplexitou záznamů ve zdrojových systémech, nevýhodou jsou vyšší nároky na synchronizaci dat. Autoritativním zdrojem dat o entitách je hub. Někdy je rozeznáván ještě podtyp hybridního řešení, u kterého nedochází ke zpětné propagaci vyčištěných a konsolidovaných dat zpět do zdrojových systémů.

Protože jsem se v praxi setkal pouze s hybridními řešeními MDI hubu, je v návrhu referenční architektury MDI hubu použito hybridní paradigma.

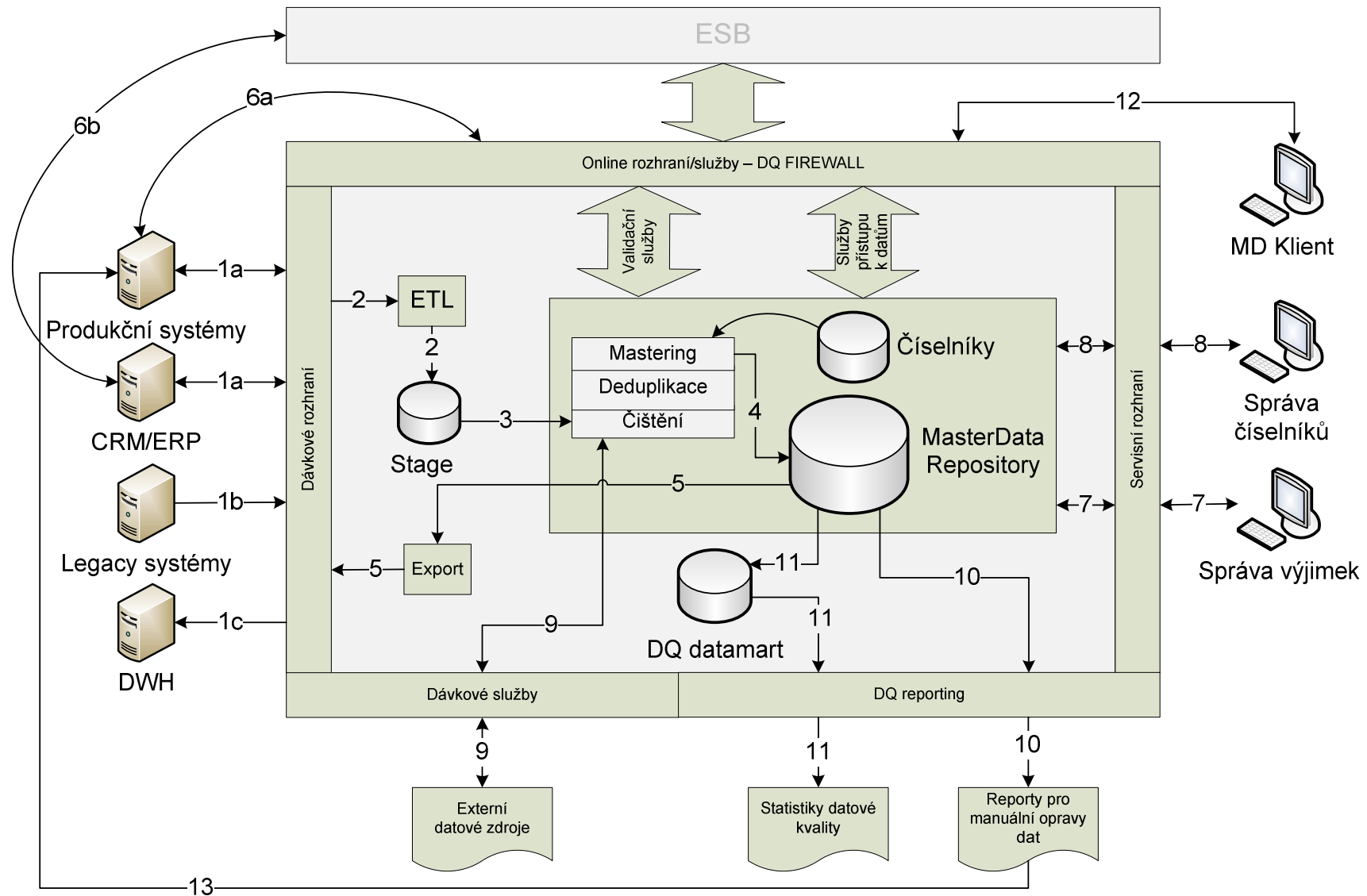
4.2.2 Návrh referenční architektury MDI hubu

Návrh referenční architektury znázorňuje Obrázek 4. Schéma zachycuje podstatné komponenty řešení MDI hubu a datové/komunikační toky, které probíhají mezi nimi. MDI hub tvoří tyto komponenty:

- **Staging area** – dočasné úložiště dat importovaných do master databáze
- **ETL (Extract Transform Load)** – proces pro import do master databáze
- **Master databáze** – hlavní úložiště referenčních dat
- **Master data server** – souhrnné označení pro aplikaci, která obsluhuje master databázi, spravuje číselníky, vystavuje dávková i online rozhraní pro práci s daty a provádí čištění a konsolidaci dat
- **Data Quality Firewall** – souhrnné označení pro online validační služby se standardním rozhraním (web service), které mají zabránit (nebo přinejmenším ztížit) vkládání nekvalitních dat do zdrojových systémů. Tyto služby jsou realizovány Master Data serverem a je možné je integrovat do uživatelských rozhraní zdrojových systémů.
- **Klientské aplikace** – aplikace pro prohlížení obsahu master databáze, pro řešení nálezů datové kvality a pro správu číselníků
- **DQ datamart** – datamart obsahující informace o datové kvalitě, který je plněn master data serverem

Jednotlivé komponenty jsou podrobněji popsány v kapitole 4.3. Pro doplnění celkového pohledu na architekturu MDI hubu je nutné ještě popsat jednotlivé toky mezi komponentami:

- Podnikové aplikace mohou komunikovat s MDI hubem buď přes dávkové nebo online rozhraní. U dávkového rozhraní rozeznáváme režim obousměrné dávkové synchronizace (1a), dávkový zápis (1b) a dávkové čtení (1c). Dávkový zápis je typický pro historické aplikace, které nemají programové rozhraní pro aktualizaci vnitřních dat. Dávkové čtení je typické pro systémy, ve kterých žádné referenční záznamy nevznikají (například datový sklad).
- Proces načtení dávky pomocí ETL (2) je typicky spuštěn buď časovaně (scheduler) nebo událostí (někdo zavolá rozhraní ETL). Vstupní dávka je nejprve načtena do stage, odkud je následně přenesena do rozdílových tabulek master databáze. Pokud není vstupní dávka v režimu inkrement a nejedná se o úvodní load, je ve stage nejprve vypočten inkrement.



Obrázek 4 - Referenční architektura MDI hubu

- Inkrement je poté nahrán do master databáze (3), kde je nejprve vyčištěn, deduplikován a zkonsolidován. Následně je inkrement sloučen s obsahem master databáze (4).
- MDI hub je schopen na požádání vytvořit dávkový export, např. pro synchronizaci se zdrojovým systémem (5).
- Zdrojové systémy mohou také využívat online služby poskytované MDI hubem, ať už přímo prostřednictvím rozhraní webové služby (6a) nebo zprostředkovaně přes Enterprise Service Bus (6b) za předpokladu, že ESB je v podniku realizována. Validační služby jsou označovány souhrnným pojmem Data Quality Firewall (DQ firewall).
- MDI hub vystavuje také rozhraní pro řízení datové kvality (7), které umožňuje řešení nálezů datové kvality a manuální opravy v attributech referenčních záznamů nebo v jejich seskupení. Nad tímto rozhraním může fungovat buď webový klient nebo jiná aplikace. Podobným způsobem je řešeno také rozhraní pro správu číselníků (8). Číselníky je možné buď přímo editovat přes klientskou aplikaci nebo jsou dávkově importovány.
- V určitých případech může dávat smysl požadavek zpracovat určitý vstupní soubor podle pravidel master databáze a poskytnout výsledek, aniž by zpracovávaný soubor ovlivnil vlastní obsah master databáze (9). Zároveň může být získána informace o tom, jak se záznamy ze vstupního souboru spárují s obsahem master databáze.
- Protože ne vždy se dá použít rozhraní pro řízení datové kvality, vytváří se i reporty pro manuální opravy referenčních dat ve zdrojových systémech (10). Důvody pro použití reportů pro manuální opravy jsou následující:
 - o Některé opravy v datech není možné provést automatizovaně (chybí dodatečné informace, jedná se o zásadní zásah do dat záznamu, který může změnit jeho význam)
 - o Uživatelská rozhraní některých systémů neumí spolupracovat s DQ Firewalllem
 - o Některé systémy neumožňují dávkovou synchronizaci referenčních dat (např. legacy systémy nebo některé kritické produkční systémy)

Reporty pro manuální opravy pak používají uživatelé, kteří buď opravují referenční data v cílovém systému přímo skrze jeho uživatelské rozhraní (13) nebo tyto opravy zaznamenávají do reportů, odkud se pak berou pro automatické zpracování.

- Kromě reportů pro manuální opravy dat je důležitý také souhrnný reporting datové kvality, který je realizován skrze Data Quality Datamart (11). Tento reporting ukazuje vývoj datové kvality v čase a umožňuje ji sledovat z různých dimenzí (např. po organizačních jednotkách nebo po zdrojových systémech). Na výstupy reportingu datové kvality pak mohou být navázána KPI pracovníků zodpovědných za datovou kvalitu.

Přímo do master databáze je možné nahlížet pomocí MD klienta, který umí vyhledávat a zobrazit konsolidované referenční záznamy, dokáže zobrazit, z jakých instančních záznamů je konsolidovaný záznam složen a další důležitá metadata záznamu (zdrojový systém,

datum poslední změny, kvalitu jednotlivých atributů nebo celých záznamů) (12). Klient také může umožňovat zásahy do referenčních dat jako zakládání nových instačních záznamů, úpravu konsolidovaných záznamů nebo ad-hoc zásahy do párování záznamů v rámci matchingu (spojování a rozpojování záznamů). Jednotlivé komponenty MDI hubu jsou popsány v následující kapitole.

4.3 Komponenty referenční architektury

4.3.1 Staging area

Staging area (stage) má v MDI hubu stejnou roli, jakou má v datových skladech, to je přechodné uložení extrahovaných dat ze zdrojových systémů. Má zajistit rychlou a efektivní extrakci dat [Novotný, a další, 2005]. Staging area slouží pro:

- Transformaci struktury zdrojových dat na základní strukturu (viz. kapitola 4.3.2)
- Import nedatabázových zdrojů dat do databázové struktury (textové soubory, XML extrakty)
- Výpočet inkrementu ze zdrojů, které nedokáží poskytovat data v inkrementální podobě

Staging area tvoří logickou nultou vrstvu master databáze (viz popis vrstev master databáze v kapitole 4.3.2).

Datové importy mohou přicházet do MDI hubu ve dvou formách:

- **Snímek** (full load, snapshot) – jedná se o kompletní snímek dat ve zdrojovém systému k určitému okamžiku v čase
- **Inkrement** – obsahuje pouze záznamy, které vznikly nebo se změnily během definovaného časového intervalu (typicky od posledního exportu). Zároveň obsahuje informaci o tom, které záznamy byly v tomto intervalu smazány.

Z pohledu zatížení MDI hubu se jako optimální strategie v práci s importy jeví zpracování úvodního snímku (initial load, initial migration) a následné zpracování inkrementů. Po určité době nebo v případě potřeby je možné provést tzv. synchronizační full load. Ten musí fungovat pro případ nastání tzv. worst-case scénáře (když se všechno pokazí a jediné řešení je nahrát data znovu).

Pro zdroje, které nedokáží poskytovat extrakty v inkrementálním režimu, simuluje stage vytváření inkrementů. Z tohoto důvodu je stage rozdělena na 2 logické části:

- **Dočasná** – do této části se nahraje aktuálně zpracovávaná dávka
- **Permanentní** – tato část obsahuje poslední zpracovanou dávku (poslední známý stav zdrojového systému). Permanentní část se s každým loadem aktualizuje a při synchronizačním loadu se kompletně přepíše.

Výpočet inkrementu se provádí porovnáním dvou snímků (předchozího a aktuálního). Pro výpočet inkrementu je zvolena jedna ze strategií v závislosti na instančních metadatech, která jsou ve zdroji k dispozici:

1. **Časová značka poslední změny záznamu a unikátní klíč záznamu, který je stabilní v čase** (jakmile je klíč jednou přidělen, už ho nemůže dostat žádný jiný záznam)
 - o Záznamy, jejichž klíč se nenajde v permanentní části, jsou označeny jako nové

- o Záznamy, jejichž klíč se najde v permanentní části, a v dočasné mají novější časovou značku, jsou označeny jako aktualizace
 - o Záznamy, jejichž klíč existuje v permanentní části, ale nenajde se v dočasné, jsou označeny jako smazané
2. **Zdroj obsahuje pouze unikátní klíč stabilní v čase**
 - o Strategie pro vložení a smazání je stejná jako v předchozím případě
 - o Pro záznam, jehož klíč se najde v permanentní i v dočasné části se porovnají všechny atributy, které se importují do master databáze (spojené do řetězce) – pokud se některý atribut liší, je záznam označen jako aktualizovaný
 3. **Zdroj neobsahuje unikátní klíč stabilní v čase** – v takovém případě se jako alternativní klíč použije kombinace atributů, které nejlépe identifikují záznam (lze zjistit např. profilíngem). Použije se strategie 1 s náhradním klíčem.
 4. **Zdroj neobsahuje unikátní klíč stabilní v čase ani časovou značku poslední změny** – použije se strategie 2 s náhradním klíčem.

Přístupy 3 a 4 jsou uvedeny pouze pro úplnost, jejich použití je přinejmenším riskantní.

Výsledný inkrement se pak zpracuje v master databázi.

4.3.2 Master databáze

Master databáze je relační databáze, ve které jsou uložena referenční data v kanonické struktuře nezávislé na jednotlivých zdrojích. K datům uloženým v master databázi se přistupuje skrze rozhraní poskytovaná master data serverem (viz kapitola 4.3.4).

4.3.2.1 Realizace master databáze

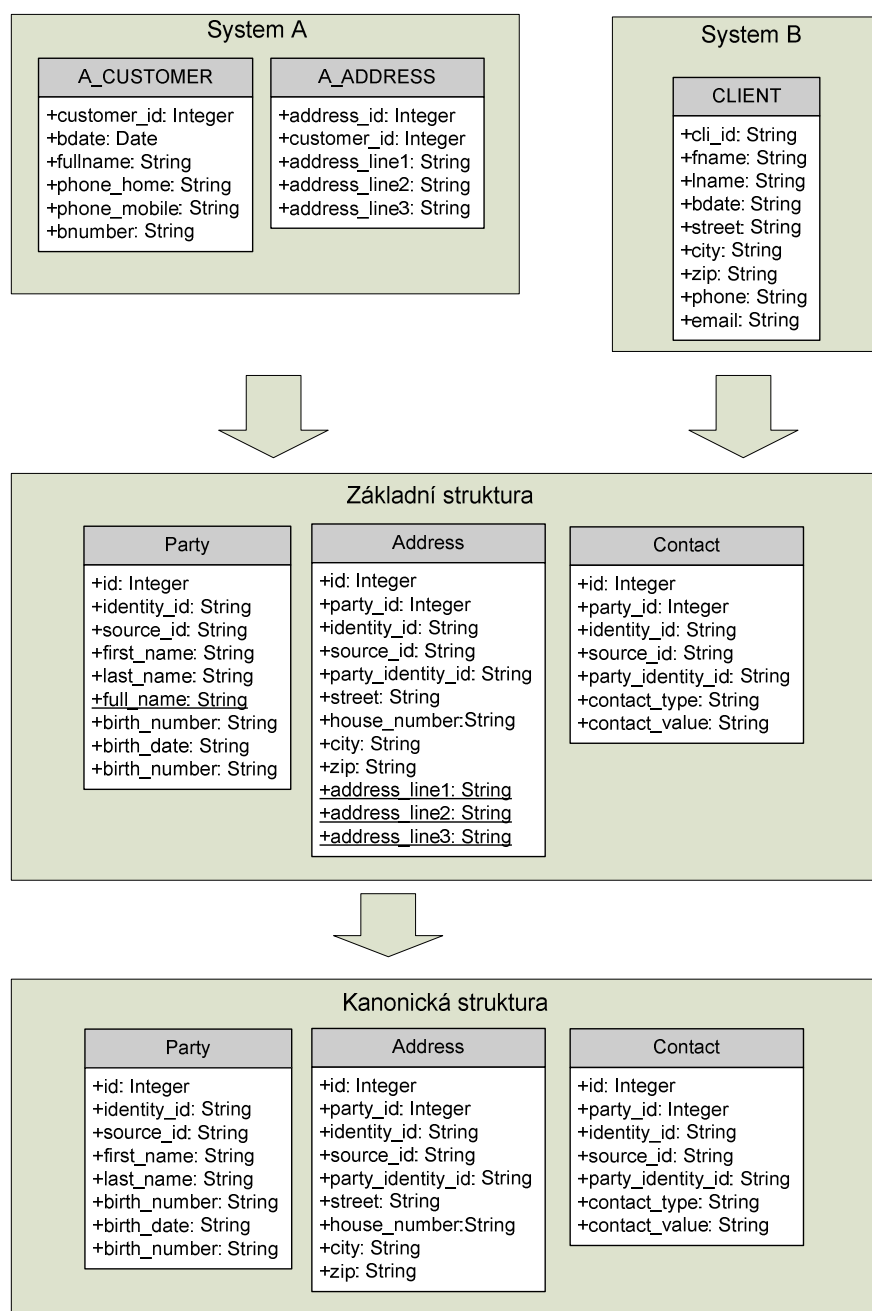
V následujícím popisu jsou rozlišeny tři typy struktury referenčních dat:

- **Struktura zdrojového systému** – každý zdrojový systém má vlastní sadu tabulek, ve které jsou uložena referenční data, a používá vlastní pojmenování atributů. Struktura uložení je optimální z pohledu zdrojového systému.
- **Základní struktura** – mezikrok mezi strukturou zdrojového systému a kanonickou strukturou. Obsahuje tabulky v kanonické struktuře a názvy atributů jsou kanonické, obsahuje navíc ale také složené atributy nebo sémanticky nejasné atributy ze zdrojových systémů, které bude možné transformovat na kanonické atributy až v průběhu čištění (např. full_name, address_line1, address_line2, address_line3). Pokud jeden záznam ze zdrojového systému popisuje více entit, jsou při převodu na základní strukturu rozděleny (jeden záznam z tabulky CLIENT se rozpadne na jeden záznam v tabulce Party, jeden záznam v tabulce Address a dva záznamy v tabulce Contact).
- **Kanonická struktura** – standardní struktura nezávislá na konkrétním systému, ve které jsou uloženy referenční záznamy. Každá entita je uložena v samostatné tabulce a každý atribut má jasně definovanou sémantiku. Pojmenování tabulek i atributů je sémantické a nezávislé na konkrétním systému.

Vztah mezi jednotlivými typy struktur znázorňuje Obrázek 5.

V master databázi se vyskytují dva typy referenčních záznamů:

- **Instanční záznam** – reprezentuje referenční záznam na úrovni zdrojového systému. Tyto záznamy mohou obsahovat duplicity (jeden business objekt může být reprezentován více referenčními záznamy – napříč všemi systémy i v rámci jednoho systému se mohou vyskytovat duplicity).
- **Konsolidovaný záznam** – reprezentuje business objekt. Při konsolidaci vznikají z instančních záznamů skupiny, které reprezentují tentýž business objekt. Jednotlivé záznamy mají různě kvalitní atributy, takže je možné provést rekombinaci záznamů, vybrat nejkvalitnější atributy jednotlivých instancí a vytvořit konsolidovaný (zlatý, reprezentativní) záznam.



Obrázek 5 – Zdrojová, základní a kanonická struktura

Tabulky v master databázi je možné rozdělit na:

- **Referenční tabulky** – obsahují vlastní referenční záznamy, tvoří jádro
- **Servisní tabulky** – obsahují řídicí a konfigurační data nezbytná pro fungování a řízení MDI hubu

Další odstavce se vztahují pouze k referenčním tabulkám.

Ke zpracování referenčních záznamů dochází v několika fázích, proto je možné rozdělit referenční tabulky do čtyř logických vrstev. V každé vrstvě se nachází tabulky, které obsahují specifická data. Vrstvy L1, L2 a L3 obsahují instanční záznamy, vrstva L4 obsahuje konsolidované záznamy (může ale obsahovat i instanční záznamy, viz. dále). Při fyzické realizaci se data mezi vrstvami nereplikují (tj. logicky mohou být na druhé vrstvě vidět vyčištěné atributy a k nim zdrojové hodnoty, fyzicky jsou ale uloženy odděleně).

- **První vrstva (L1)** – data transformovaná do základní struktury. Při nahrání dat do první vrstvy nedochází k žádným transformacím, pouze k mapování zdrojových atributů na atributy základní struktury 1:1. První vrstva je v základní struktuře. Kromě toho, že je zdrojem pro plnění vrstvy L2, se používá k dohledání původních hodnot atributů zdrojových záznamů.
- **Druhá vrstva (L2)** – vyčištěná a standardizovaná zdrojová data. V této fázi dochází k aplikaci transformací na data. Mezi hlavní transformace patří:
 - o oprava překlepů
 - o formátování a standardizace hodnot atributů
 - o přesun hodnot mezi atributy (např. tituly se oddělují od jména a přesunují se do samostatného pole)
 - o sémantizace atributů – hodnoty nalezené v generických polích (address_line1, address_line2, address_line3) se přesouvají do polí s jasnou sémantikou (street, house_number, city, zip_code).
 - o generování štítků pro označení nálezu datové kvality

První a druhá vrstva bývají někdy z výkonnostních důvodů spojovány do společné, tzv. **instanční vrstvy (L1+L2)**.

- **Třetí vrstva (L3)** – standardizovaná data doplněná o informaci, které instanční záznamy reprezentují stejný business objekt. Ve třetí vrstvě jsou uložena unifikační metadata (id skupin, označení pravidel uplatněných při seskupování).
- **Čtvrtá vrstva (L4)** – doménové pohledy, které obsahují konsolidované i instanční záznamy. Master databáze prezentuje referenční data odběratelům formou **doménových pohledů**. Doménový pohled prezentuje konsolidovaná data ve struktuře požadované uživatelem doménového pohledu (osoba, aplikace/systém) při zachování vazeb na nekonsolidované entity/tabulky zdrojového systému (např. pro konsolidovaného bankovního klienta je možné vidět seznam smluv/produktů, které se vážou k jeho instancím). V doménovém pohledu se vyskytují tři typy logických tabulek:
 - o **Zlaté** – obsahují konsolidované záznamy. Jsou realizovány jako plnohodnotné tabulky. Tímto způsobem se realizují i M:N vazby mezi zlatými záznamy.
 - o **Stříbrné** – obsahují deduplikované vazby mezi nekonsolidovanými instančními záznamy a konsolidovaným záznamem (např. pro konsolidovaného klienta

všechny jeho unikátní adresy nebo telefonní čísla – pokud měly například dvě instance, ze kterých vznikl konsolidovaný klient, stejnou adresu, adresa se objeví v této tabulce pouze jednou). Jedná se o vazební tabulky, které se odkazují na instanční vrstvu.

- o **Bronzové** – obsahují vazby mezi nekonsolidovanými instančními záznamy a konsolidovaným záznamem. Používají se typicky tam, kde nedává smysl deduplikovat instance určitého typu (např. Produkt, Smlouva). Jedná se o vazební tabulky, které se odkazují na instanční vrstvu.

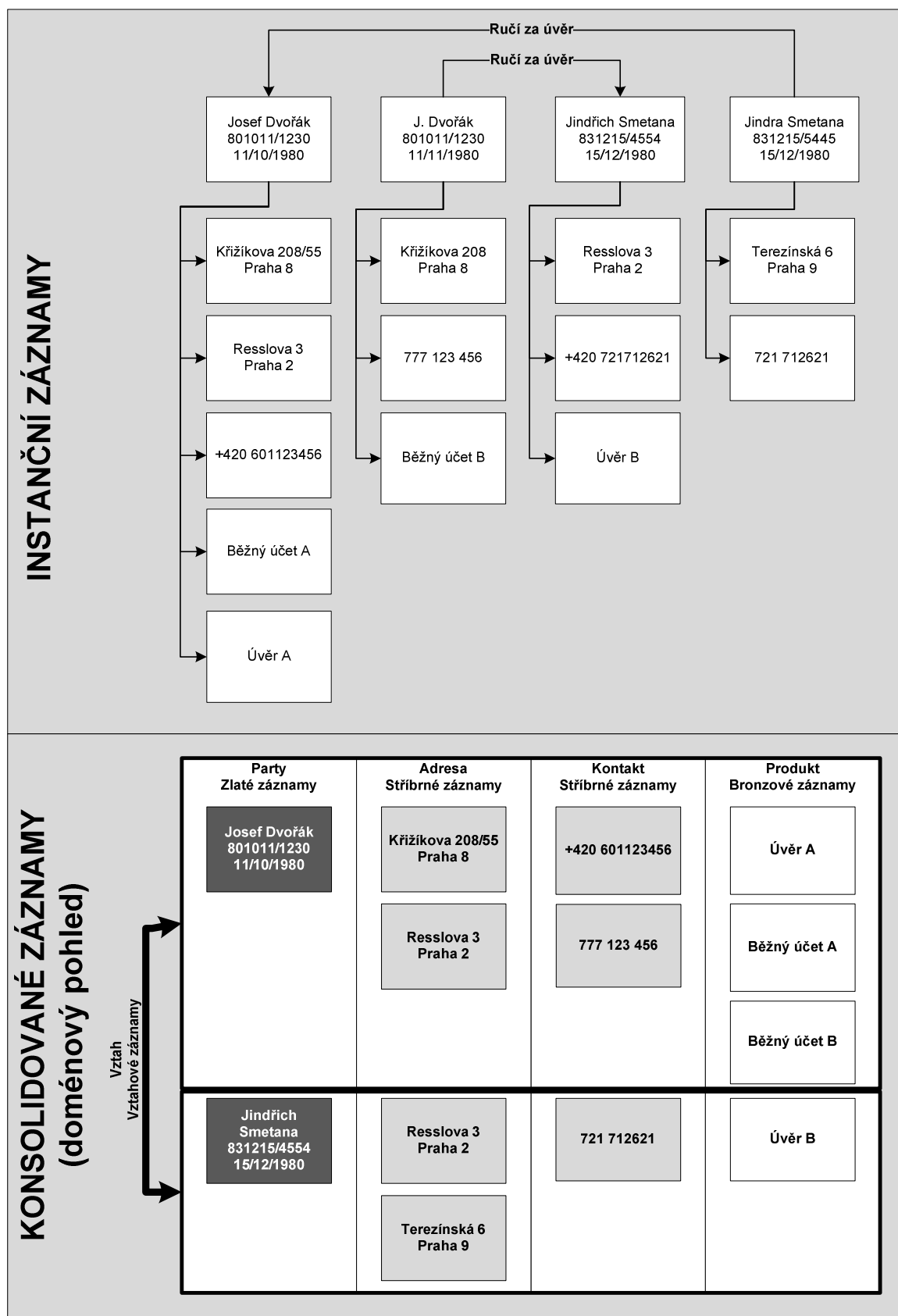
Význam jednotlivých typů tabulek v doménovém pohledu znázorňuje Obrázek 6 na následující straně.

Při plnění doménového pohledu se zohledňuje:

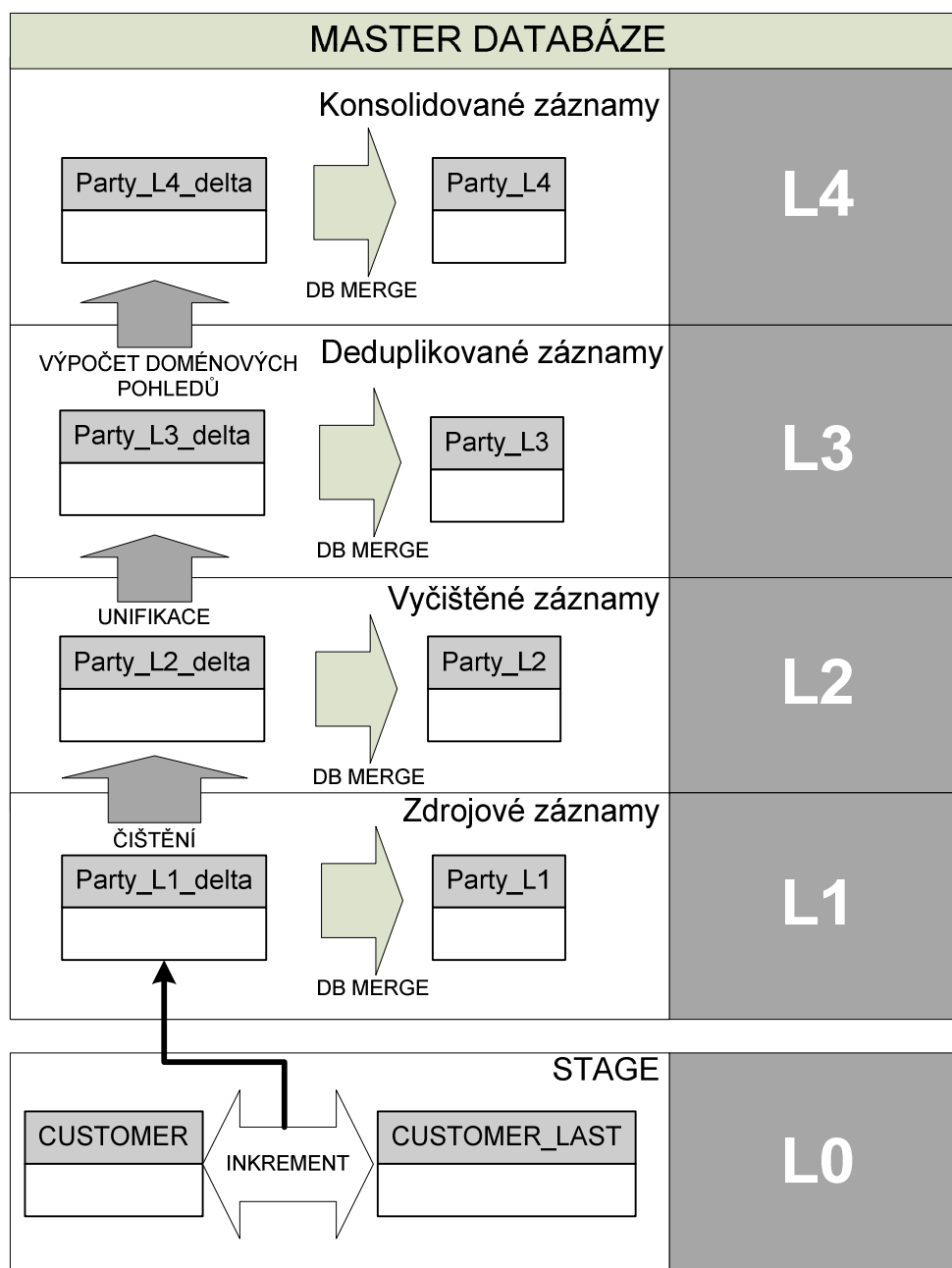
- seskupení instančních záznamů (výsledek unifikace)
- které záznamy smí uživatel doménového pohledu vidět (např. pouze konsolidované záznamy skupin, které obsahují alespoň jeden záznam z daného systému nebo pouze záznamy, ke kterým má souhlas s použitím údajů)
- v jaké struktuře chce uživatel vidět data (tabulky, sloupce)
- v jaké podobě chce uživatel vidět atributy (zdrojové, vyčištěné, konsolidované)

Doménové pohledy mohou být realizované jako virtuální (např. pomocí databázových view) nebo jako perzistentní formou datamartu. Při rozhodování o způsobu realizace hraje roli výpočetní náročnost doménového pohledu, očekávaná frekvence využívání a požadavky na synchronizaci doménového pohledu s obsahem nižších vrstev (dávkově vs. v reálném čase).

Při fyzické realizaci master databáze se zpravidla oddělují také **bázové tabulky** a **tabulky obsahující inkrement (delta)**. Výhodou tohoto oddělení je, že se při čištění a unifikaci zpracovávají pouze nová a změněná data a je případně možné pozdržet sloučení inkrementu s bází do chvíle, než se ověří celková kvalita dat v inkrementu. Tím je možné zabránit poškození báze referenčních dat nahráním vadného inkrementu. Schéma fyzické realizace tabulek zachycuje Obrázek 7.



Obrázek 6 - Reprezentace konsolidovaných zdrojových dat v doménovém pohledu



Obrázek 7 - Schéma fyzického uložení dat v master databázi

4.3.2.2 Modelování master databáze

Při modelování master databáze je cílem na základě dostupných informací o referenčních datech vytvořit jejich kanonický model (tzv. master data model). Při návrhu struktury master databáze se používá technika datového modelování (viz. kapitola 3.2). Informace o referenčních datech lze získat z různých míst:

- **Popisy procesů** – obsahují odkazy na všechny entity, se kterými podnik přichází do styku
- **ERD modely dílčích systémů** – obsahují informace o evidovaných entitách a jejich vazbách na úrovni jednotlivých systémů
- **Korporátní datový model** – model, který pokrývá celou datovou základnu podniku nebo přinejmenším její podstatnou část [Simsion, a další, 2005]. Existuje na konceptuální a logické úrovni. Korporátní datový model je důležitým nástrojem pro řízení rozvoje podnikové datové základny a architektury podnikových systémů.

Při vytváření master data modelu je možné využít koncept harmonizace informačních objektů, kterou navrhuje Schmidt a Otto v [Harmonizing company-wide Information Objects, 2008]. V článku jsou definovány tři typy objektů podle úrovně modelu, na které existují:

- business objekty na konceptuální úrovni
- informační objekty na logické úrovni
- datové objekty na fyzické úrovni.

Vztah mezi jednotlivými typy objektů vyjadřuje Obrázek 2. Středem pozornosti jsou informační objekty, které jsou z pohledu modelování master databáze v podstatě prvky master data modelu. Schmidt a Otto navrhuje metodiku, která řeší integraci a harmonizaci informačních objektů na podnikové úrovni (METIO). Metodika zahrnuje přehled metadatových atributů nezbytných pro popis informačního objektu i návrh procesu pro harmonizaci popisu a použití informačního objektu napříč podnikem.

Další možností je odvození master data modelu z korporátního datového modelu. O technice vytváření korporátního datového modelu pojednává [Kowalski, 1999], techniky užitečné při vytváření korporátního datového modelu jsou popsány též v [Simsion, a další, 2005].

Master databáze se modeluje na třech úrovních:

- **Konceptuální** – zachycuje pouze entity, jejich atributy/skupiny atributů a vazby mezi entitami. Vazby obsahují kardinality a role. V modelu se identifikují entity, které ve skutečnosti představují různé role téže entity – společného nadtypu (například Pojištěný i Pojistník jsou speciální případy/role entity Osoba). Používá se hlavně pro zachycení podstatných entit a jejich atributů a pro komunikaci s business uživateli.
- **Logický** – jednotlivé atributy jsou reprezentovány více sloupci, které prezentují různé stavy zpracování atributu – zdrojová podoba, předčištěná podoba, ověřená a standardizovaná podoba, jsou doplněny pomocné sloupce pro označení:
 - o Zdroje záznamu (zdrojový systém/rozhraní) – pokud se jedna entita může vyskytovat ve více rolích, které jsou uloženy ve zdrojovém systému v různých tabulkách, zdroj je třeba identifikovat až na úroveň zdrojové tabulky (např.

pokud se Osoba může vyskytovat v rolích Pojištěný¹⁷ a Pojistník¹⁸, které jsou uloženy ve zdrojovém systému na odlišných místech, je potřeba indikovat, v jaké roli se daný záznam ve zdroji vyskytl)

- o Datum poslední změny záznamu
- o Kdy byl záznam zpracován, verze použitých čisticích a konsolidačních pravidel
- o Stav, zda je záznam aktuální nebo smazaný
- o Transformací provedených na atributu v rámci čištění (explanations)
- o Kvality atributu a nálezu datové kvality (labelling)
- o Kvality celého záznamu
- o Zařazení záznamu do určité skupiny v rámci matchingu a pravidlo, které se při zařazení uplatnilo

Používá se pro zachycení veškerých uživatelských požadavků na data v největším detailu (na základě něj je možné rozhodnout, zda jsou data pro realizaci uživatelského požadavku v master databázi k dispozici)

- **Fyzický** - dokumentuje fyzickou realizaci logického pohledu na data v databázi

Při vytváření master data modelu se používají tato pravidla:

1. **Ne všechny entity musí být uloženy v master databázi** - protože není účelné ukládat do master databáze všechny entity identifikované ve zdrojových systémech, je nutné rozhodnout, které entity a atributy budou uloženy v master databázi. Vstupem pro rozhodování je logický master data model a výsledky analýzy datových modelů zdrojových systémů a profilingu. Z analýzy datových modelů zdrojových systémů máme k dispozici katalog entit (entita, koncept reprezentovaný entitou a způsob uložení ve zdrojovém systému - viz kapitola 5.2.1.2). Pro vyhodnocení, které entity má smysl udržovat v master databázi, lze použít kritéria:

- Entita se vyskytuje ve více systémech
- Na entitu se odkazují další entity
- Entitu má smysl konsolidovat (může existovat více instancí, které ve skutečnosti reprezentují tentýž reálný objekt)
- Entita sdílí podstatnou část svých atributů s jinou entitou

Pro výběr atributů, které má smysl ukládat v master databázi, lze použít kritéria:

- Atribut může být použit při identifikaci instance
- Atribut má více odběratelů
- Atribut se vyskytuje ve více zdrojových systémech a je možné vybrat jeho správnou/nejllepší variantu

2. **Používají se umělé primární klíče** - Pro jednoznačnou identifikaci instantních záznamů v tabulkách master databáze a pro vazby mezi záznamy se používají umělé klíče (surrogate key), které si master databáze generuje sama. Důvodem je ochrana proti nespolehlivé referenční integritě zdrojových systémů a případné absenci nebo

¹⁷ Pojištěný - osoba, na kterou se pojistná smlouva vztahuje (objekt smlouvy)

¹⁸ Pojistník - osoba, která uzavřela pojistnou smlouvu (subjekt smlouvy)

nestabilitě primárních klíčů ve zdrojovém systému. V některých zdrojích nemusí být klíče přítomny, protože vztahy mezi entitami jsou definovány implicitně (např. podrízená entita je v XML vnořeným podstromem nadřízené entity)

3. **Každý instanční záznam musí mít identitu** – Pro každou entitu je definován tzv. identitní klíč (jednoduchý nebo složený klíčový atribut), který umožňuje v rámci daného zdrojového systému jednoznačně identifikovat záznam kvůli výpočtu inkrementu, zpětné propagaci vyčištěných dat a dohledání záznamu ve zdrojovém systému. Instanční záznam je pak v rámci MDI hubu jednoznačně identifikován pomocí kombinace „zdrojový systém + identitní klíč“. Toto pravidlo nefunguje v případech, kdy je ve zdroji porušena unikátnost klíče.
4. **Při práci s instančními záznamy je třeba zohlednit historii změn** – historizace se řeší podobným způsobem jako v případě slowly-changing dimensions (SCD) datových skladů. V případě MDI hubu je upřednostňována SCD typu 4 (instanční tabulka obsahuje pouze aktuální záznamy a historické záznamy jsou uloženy v oddělené tabulce), protože umožňuje sledovat kompletní historii záznamů a instanční tabulka není zatížena historickými záznamy, které jsou uloženy odděleně.

U referenčního záznamu může hrát roli povaha změny hodnoty. Proto sledujeme dva typy změn:

- **Formální (správnost)** – oprava, která nemění význam hodnoty (např. oprava překlepu ve jméně).
- **Faktické (platnost)** – změna, která mění význam hodnoty (např. změna adresy při přestěhování)

Každá změna je buď prvního nebo druhého typu. Tento údaj může hrát roli ve chvíli, kdy potřebujeme tyto dva typy změn odlišit, protože na ně jsou navázány nějaké procesy (např. změna místní příslušnosti k úřadu nebo pobočce). Rozdíl formální a faktické změny dokumentuje následující příklad:

ADDRESS_HISTORY					
ID_adresy	Ulice	Město	PSČ	Platné do	Správné do
123	Křížíkova 55	Praha 8	180 00	<i>Null</i>	6.7.2008
123	Křížíkova 208/55	Praha 8	186 00	1.1.2010	<i>null</i>

ADDRESS					
ID_adresy	Ulice	Město	PSČ	Platné do	Správné do
123	Resslova 3	Praha 2	120 00	<i>Null</i>	<i>null</i>

Díky rozlišení správnosti a platnosti můžeme říct, že klient se přestěhoval pouze 1.1.2010 a nikoli 6.7.2008.

4.3.3 ETL

ETL (Extract-Transform-Load) je komponenta, která zajišťuje dávkové přesuny dat. Používá se pro dávkovou extrakci a transformaci dat ze zdrojových systémů a jejich uložení do stage.

Zároveň provádí výpočet inkrementu pro zdroje, které nedokáží poskytovat data v inkrementálním režimu.

ETL je možné realizovat různými prostředky:

- Prostředky DBMS - MS SQL Server Integration Services, Oracle Data Integrator
- Skriptovací/programovací jazyk - VB.Script, C#, Java, PL/SQL, TSQL
- Specializovaný nástroj (Talend Open Studio, Clover ETL, Pentaho Data Integration, IBM InfoSphere Data Stage, Informatica Power Center, Ab Initio)

ETL může fungovat v režimech:

- Pull - po přijetí signálu si aktivně vytáhne data přímo ze struktur zdrojového systému
- Push - ETL zpracovává extrakt, který mu zdrojový systém připraví

Požadavky na ETL:

- Možnost definovat workflow - sekvence operací, možnost reagovat na různé výsledky vykonání jednotlivých operací (úspěšně ukončeno/neúspěšně ukončeno)
- Čtení/zápis databázových tabulek
- Provádění dalších databázových operací (vytváření a rušení indexů, volání DB procedur a funkcí, ...)
- Čtení/zápis textových souborů s oddělovači
- Čtení/zápis XML souborů
- Čtení/zápis dalších formátů souborů (xls, dbf)
- Provádění souborových operací - přesouvání, mazání
- Provádění notifikací - mailem, voláním webové služby
- Mapování atributů a transformace

Úloha ETL:

- Zajistit exekuci workflow spojených s importem dat do master databáze
- Provést kontrolu správnosti formátu dat a úplnosti importu (byly přeneseny všechny záznamy, které jsou součástí importu)
- Provést úvodní transformaci dat (pouze mapování 1:1) - používají se informace o mappingu získané při profilingu zdrojů
- Zajistit případný výpočet inkrementu pro neinkrementální zdroje
- Naimportovanou dávku zaregistrovat do fronty pro další zpracování MD serverem
- Logovat průběh importu

4.3.4 Master data server

Termínem Master data server se souhrnně označuje aplikace, která zajišťuje obsluhu master databáze. Může se jednat o aplikaci vyvinutou vlastními silami nebo o zakoupené typové řešení, případně se může jednat o kombinaci obojího (nakoupené komponenty řešící dílčí úkoly propojené do funkčního celku).

Funkční požadavky na master data server:

- **Správa uložení dat v master databázi** - master data server odstiňuje uživatele a ostatní systémy od specifik způsobu uložení instančních a konsolidovaných záznamů, procesu jejich vytváření a udržování historie záznamů

- **Čištění, unifikace a konsolidace dat** – master data server realizuje proces čištění a konsolidace dat na základě nastavených business pravidel. Tato komponenta master data serveru je vhodným kandidátem na realizaci formou zakoupení specializované aplikace. Součástí čištění a konsolidace dat je:
 - o Odstranění zástupných (dummy) hodnot a tzv. blacklistových¹⁹ hodnot
 - o Překlad číselníkových hodnot na jednotný tvar (např. atribut pohlaví – M x F, 1 x 0, M x Ž nebo kódy států)
 - o Kontrola formátu a validity atributu (regulární výrazy, kontrolní součty – např. kontrolní součet u rodného čísla, kontrola formátu čísla občanského průkazu)
 - o Parsing atributu (dekompozice atributu na jeho dílčí složky a jejich ověření vůči číselníkům)
 - o Oprava překlepů (korekční náhrady – Božny Němcové → Boženy Němcové)
 - o Standardizační náhrady (TGM, T.G.Masaryka, T.G.Mas., Tomáše G. Masaryka, Masaryka, Masarykova → Masarykova)
 - o Ověření v číselníku
 - o Ověření vzájemné konzistence atributů (např. IČO a název firmy, rodné číslo a datum narození)
 - o Příprava párovacích hodnot pro unifikaci (odstranění znaků neměnicích význam atributu, převedení na velká písmena, použití slovníku synonym)
 - o Kontrola splnění business pravidel

Součástí unifikace je:

- o Přiřazení záznamu do konkrétní skupiny na základě unifikačních klíčů (atributy, které musí mít záznamy stejné, aby tvořily jednu skupinu) a pravidel pro porovnávání klíčů s tolerancí (odlišnosti v pořadí slov ve jméně, překlepy, slovo navíc, zkratky apod.)

Proces čištění, unifikace i konsolidace musí být auditovatelný (lze doložit průběh transformací provedených na instančním a master záznamu i pravidlo, které se uplatnilo při zařazení instančního záznamu do určité skupiny při matchingu).

- **Dávková vstupní a výstupní rozhraní** – dávková rozhraní slouží především pro synchronizaci dat mezi MDI hubem a zdrojovými systémy. Dávkové rozhraní je typicky realizováno jako sada databázových tabulek nebo soubor/y s exportovanými daty. Jako dávkové rozhraní může být chápán i doménový pohled.
- **Online rozhraní** – jsou typicky realizována pomocí webových služeb a zahrnují
 - o Služby pro čtení záznamů uložených v master databázi
 - o Služby pro identifikaci a zápis referenčního záznamu
 - o Služby pro čištění a validaci záznamu – služby slouží pro kontrolu dat zadávaných do zdrojové aplikace. Používají stejná pravidla, jako se používají při dávkovém čištění a konsolidaci dat. Tyto služby se souhrnně označují jako Data Quality Firewall (DQ Firewallu). Koncept DQ Firewallu je založen na myšlence, že kontrolu datové kvality je nejefektivnější provádět již při

¹⁹ Seznam zakázaných hodnot, které se nesmí pro atribut použít.

ukládání referenčních dat do zdrojového systému. DQ Firewall má zabránit průniku nekvalitních dat do zdrojových systémů. Díky interaktivnímu režimu je možné uživateli předkládat rovnou ke schválení i návrhy oprav (varianty) jeho vstupu – například adresa „Křižíkova 55, Praha 8 – Karlín, 186 00“ se může uživateli na první pohled jevit jako dostatečně přesná, protože neví, že tato adresa ve skutečnosti reprezentuje dva různé domy – „Křižíkova 208/55“ a „Křižíkova 55/65“).

Pro koexistenci online rozhraní a dávkových importů je naprosto klíčové řešení vzájemného souběhu batch a online služeb pro ukládání dat. MDI hub musí mít vyřešen konkurenční přístup k datům a tzv. race conditions, aby vracel konzistentní výstup při deduplikaci záznamů.

- **Definování a exekuce workflow** – pod pojmem workflow je v tomto případě míněna posloupnost operací automatizovaně vykonávaných komponentami MDI hubu, která řeší specifickou úlohu. Jedná se o technické pojetí workflow, které, narozdíl od byznys pojetí workflow, abstrahuje od interakce s uživatelem. Interakce s uživatelem je vytěsněna do prostoru „mezi workflow“, tj. před začátek nebo za konec workflow. Akce uživatele může hrát roli spouštěcí události. Posloupnost prováděných operací nemusí být lineární, může se větvit na základě výsledku vykonání dílčích operací (úspěšné provedení kroku vs. chyba). Workflow je spuštěno buď událostí (příchod požadavku) nebo časovaně (pomocí scheduleru). Výsledek vykonávání jednotlivých operací i celého workflow je sledován a logován.

Exekuce workflow může být koordinována

- o skriptem operačního systému (shell)
- o databázovým plánovačem (DB scheduler)
- o specializovanou aplikací/modulem MDI hubu

Koordinátor má za úkol spouštět jednotlivé kroky v pořadí definovaném ve workflow plánu. Mezi typické operace ve workflow patří:

- o Manipulace se soubory – kopírování, přesouvání, komprimace, přejmenování, mazání
- o Volání databázových příkazů, které nevrací množinu výsledků (např. db procedury)
- o Vykonání databázového dotazu, který vrací množinu výsledků (např. SELECT)
- o Čekání na událost (příchod souboru, objevení záznamu v databázové tabulce)
- o Spuštění příkazu operačního systému
- o Spuštění vnořeného workflow

Workflow se typicky vykonává v synchronním režimu (každý krok se spustí teprve poté, co skončí předcházející krok), spuštěná vnořená workflow se vykonávají asynchronně.

Workflow se skládá z izolovaných operací, proto není možné použít koncept transakcí známý z prostředí relačních databází. V případě selhání workflow se

provádí tzv. kompenzace – vykonání operací, které mají uvést prostředí do stavu, v jakém bylo před výskytem selhání.

Základní workflow realizovaná MDI hubem jsou navržena na konceptuální úrovni v Příloze (10.2):

- o Import referenčních dat ze zdrojových systémů (WF1)
 - o Online aktualizace referenčních dat (WF2)
 - o Řešení nálezů datové kvality (WF3)
 - o Aktualizace číselníků (WF4)
- **Plánování úloh a notifikace** – plánování úloh je interní služba, která má za úkol v daném čase spustit vykonání určité sady operací (spustit workflow). Spuštění může být zahájeno také nějakou událostí (příchod požadavku na určité rozhraní). Notifikace má za úkol upozornit odběratele oznámení, když nastane určitá situace. Notifikace může zahrnovat např. odeslání mailu, zavolání webové služby, odeslání zprávy pomocí messaging infrastruktury (MQ, JMS). Notifikace může být alternativně řešena sledováním auditního logu MDI hubu.

Nefunkční požadavky na master data server:

- **Robustnost**
 - o aplikace dokáže reflektovat nové uživatelské požadavky
 - o design aplikace počítá s možností implementace nových funkcí, aniž by tím byla ohrožena stabilita aplikace
 - o aplikace reaguje na nestandardní situace (např. na pád komponenty/aplikace) standardním a předvídatelným způsobem
- **Škálovatelnost**
 - o z pohledu výkonu (doba odezvy, doba zpracování) a objemu uložených dat
 - o aplikace v maximální míře využívá zdroje, které potřebuje pro svou činnost (paměť, procesorový čas, disková rozhraní, síťová rozhraní)
 - o je možné efektivně zvyšovat výkon aplikace přidáním nového hardware
 - o při jaké zátěži a objemech dat již není možné zvýšit výkon prostým přidáním hardware (limity škálovatelnosti)
- **Auditovatelnost**
 - o je možné sledovat provedené změny a identifikovat původce změn (kdo změnu provedl, kdy změnu provedl, jakou změnu provedl, jak je možné změnu anulovat)
 - o lze realizovat pomocí auditních tabulek nebo auditních log souborů
- **Řízení přístupu**
 - o k datům
 - o ke službám/rozhraním
 - o napojení na adresářové služby – např. ActiveDirectory, LDAP
- **SLA**
 - o garantované parametry poskytované služby (průměrná a mezní doba odezvy, dostupnost, kapacita, aktuálnost poskytovaných dat apod.) – př. služba má garantovanou dostupnost denně od 6:00-22:00 (22:00-6:00 je servisní okno,

kdy dostupnost není garantovaná), průměrná doba odezvy max. 500 msec, mezní doba odezvy max. 2 sec, kapacitní vytížení 20 tisíc dotazů denně

- **Správa verzí, nasazování nových verzí**
 - o změny v modelu master databáze – musí existovat konzistentní postup, jak provádět aktualizace struktury master databáze, aniž by byla ohrožena referenční data uvnitř
 - o změny čistících a konsolidačních pravidel
 - o aplikace změn v čistících a konsolidačních pravidlech a v číselnících na záznamy již uložené v master databázi (tzv. retrofitting)

4.3.5 *Klientské aplikace*

Aby mohli běžní (business) uživatelé spravovat MDI hub a pracovat s daty v něm, potřebují k tomu nástroje, které by jim umožnily provádět běžné operace uživatelsky přívětivou formou. Mezi základní (generické) aplikace patří:

- Správa číselníků
- Správa výjimek
- Administrační konzole

Tyto aplikace je vhodné realizovat jako webové, protože se snadněji udržují, je možné je centralizovaně spravovat, nemusí se instalovat ani se kvůli nim nemusí instalovat na klientské stanice žádný speciální software, a jsou přístupné odkudkoli.

Kromě výše uvedených generických aplikací je samozřejmě nezbytné integrovat služby MDI hubu do aplikací, které uživatelé běžně používají pro správu svých agend.

4.3.5.1 *Správa číselníků*

MDI hub se neobejde bez číselníků, používá je při čištění vstupních dat, při interpretaci vstupních hodnot atributů importovaných ze zdrojových systémů a při překladu výstupu MDI hubu do „řeči“ zdrojového systému. Číselníky mohou pocházet z interních nebo externích zdrojů.

Interní číselníky vznikají v podniku a slouží ke standardizaci zápisu hodnot vybraných atributů. Často bývají definovány pouze na úrovni systému (standardizace atributu v rámci jednoho systému), proto plní svou roli pouze částečně. Při implementaci MDI hubu je nutné číselníky harmonizovat na úrovni celého podniku – to zahrnuje sjednocování business významu číselníků, kódování hodnot a sjednocování granularity (míry detailu) číselníků. Harmonizace spočívá ve vytvoření překladových slovníků pro jednotlivé systémy, aby MDI hub mohl překládat před zpracováním vstupní hodnoty na standardní podobu a při plnění doménových pohledů ji pak zase dokázal konvertovat zpět do dialektu konkrétního systému. Pokud v podniku existuje aplikace pro centrální správu číselníků, musí s ní MDI hub bezpodmínečně komunikovat. Pokud v podniku taková aplikace neexistuje, může MDI hub převzít její roli.

Externí číselníky jsou vytvářeny a aktualizovány vnějšími subjekty či úřady. Slouží pro sjednocení popisu/klasifikace vybraných atributů, které jsou důležité například pro legislativní reporting. Slouží také jako podklad pro ověřování existence/správnosti

některých údajů (např. Registr Ekonomických Subjektů²⁰ – RES slouží pro ověření IČ a názvu firmy).

V MDI hubu se používá pro práci s daty řada číselníků, především v procesu čištění a konsolidace referenčních dat:

- **Korekční** – náhrady řetězců (opravy překlepů)
- **Parsovací** – slouží pro rozdělení vstupního řetězce na komponenty s určitým sémantickým významem
- **Ověřovací (validační)** – slouží pro ověření, zda je vstupní hodnota známá či nikoli (např. ověřování IČO v RES), případně zda je určitá kombinace vstupních hodnot validní či nikoliv
- **Překladové (standardizační)** – používají se pro překlad atributů na standardizovanou hodnotu (překlad různých povolených forem zápisu atributu do jednotné podoby, převod číselníkových hodnot specifických pro dílčí systém do jazyka „sdílených číselníků“ a nazpět)
- **Obohacovací** – na základě klíče doplňují k záznamu další informace/atributy
- **Blacklisty a whitelisty** – číselníky obsahující nepovolené (blacklist) a povolené (whitelist) výjimky
- **Komplexní číselníky** – slouží pro práci se složenými atributy (např. adresy), umožňují různé kontroly konzistence mezi komponentami složeného atributu, případně rekonstrukci chybějících/vadných komponent složeného atributu. Příkladem komplexního číselníku je např. Územně identifikační registr adres spravovaný Ministerstvem práce a sociálních věcí ČR (UIR-ADR)²¹

Typické operace při správě číselníků jsou:

- **Přidávání nových položek**
- **Aktualizace stávajících položek**
- **Import číselníků z externích zdrojů**
- **Historizace číselníků** – není možné zahazovat neaktuální hodnoty, protože v systému mohou existovat hodnoty, které vznikly v době, kdy platila dnes již neaktivní hodnota.

Uživatelské rozhraní by mělo tyto operace podporovat a zároveň by mělo umožnit prohlížení, filtrování a vyhledávání v číselnících.

4.3.5.2 Správa výjimek

Aplikace pro správu výjimek tvoří uživatelské rozhraní k workflow pro řešení nálezů datové kvality (WF3). Jedná se o issue tracking aplikaci s vestavěnou podporou pro validace vstupů pomocí online služeb. V aplikaci lze definovat workflow pro manuální opravy různých typů chyb a je možné definovat role odpovídající za požadavky v jednotlivých stavech. Typické stavy jsou: nový/přiřazen/vyřešen/uzavřen/ke schválení/schválen/znovu otevřen. Zároveň je možné sledovat stav řešení jednotlivých zpracovávaných nálezů datové kvality.

²⁰ <http://registry.czso.cz/irsw/>

²¹ <http://forms.mpsv.cz/uir/>

ataccama
 Home | My Issues | All Issues | Reports | Logout
 Navigation: Home > Issues > For supervisor > ID #21 (Shipping: 21)

Address

Original data

Address line 1: 595 Piedmont Ave
 Address line 2:
 City: Atlanta
 State: GA
 ZIP code: 30308-2454

Corrected data

Street: 595 Piedmont Ave NE
 Attn.:
 City: Atlanta
 State: GA
 ZIP code: 30308-2478

Severity **Message**

Medium ZIP+4 code does not match the address.
 Low ZIP code was fixed.

Comment:

Proposal selection

Select one of the proposed values:

	Street	Attn.	City	State	ZIP code
select	215 North Ave NE		Atlanta	GA	30308-2454
select	595 Piedmont Ave NE		Atlanta	GA	30308-2478

History

Date: 2010-04-21 23:56:29 User: webuser Action: Send to supervisor
 Not sure which address is the right one.
 Date: 2010-04-06 23:09:34 User: DQC Action: create
 (empty message)

Obrázek 8 - Ataccama DQIT - aplikace pro řešení nálezů datové kvality [Ataccama, 2010]

4.3.5.3 Administrační konzole

Pod pojmem Administrační konzole (Klient) se skrývá klientská aplikace, která umožňuje nahlížet na obsah master databáze uživatelsky přívětivým způsobem. Mezi základní podporované operace patří:

- Vyhledávání konsolidovaného záznamu a souvisejících instančních záznamů
- Vyhledání instančního záznamu a nadřazeného master záznamu
- Listování seznamem instančních a konsolidovaných záznamů
- Prohlížení zdrojových a vyčištěných hodnot atributů

Pokud má Klient sloužit i pro úpravu referenčních dat, je nutné navrhnout konzistentní způsob, jak toto realizovat. Zásahy do instančních a konsolidovaných záznamů mimo zdrojový systém a MDI hub (např. editace hodnot atributů zpracovaných MDI hubem) jsou nepřijatelné, protože komplikují audit a synchronizaci záznamů a jsou zdrojem nekonzistencí. Klient musí fungovat jako další zdrojový systém, který může zakládat vlastní instanční záznamy na základě instančních záznamů z jiných systémů a konsolidovaných záznamů. Instanční záznamy pocházející z Klienta pak dostanou vyšší prioritu při vytváření konsolidovaných záznamů. Klient pak bude sloužit jako system-of-record, tj. autoritativní zdroj referenčních dat pro vybrané entity.

MDI hub provádí čištění a konsolidaci dat podle dané sady pravidel a jeho výstup je konzistentní. Klient neslouží k tomu, aby uživatelé mohli upravovat konsolidované referenční záznamy do podoby, kterou chtějí vidět. Pokud uživatel chce dosáhnout změny na úrovni konsolidovaných referenčních záznamů, musí provést potřebné změny na úrovni instančních referenčních záznamů ve zdrojových aplikacích. Tyto změny se pak při zpracování v MDI hubu promítnou i do konsolidovaného referenčního záznamu. Pokud se

objeví požadavek na editaci konsolidovaných referenčních záznamů, poukazuje to na hlubší problém v návrhu zdrojového systému (Proč chce uživatel vidět data jinak, než jak jsou uložena ve zdrojovém systému?)

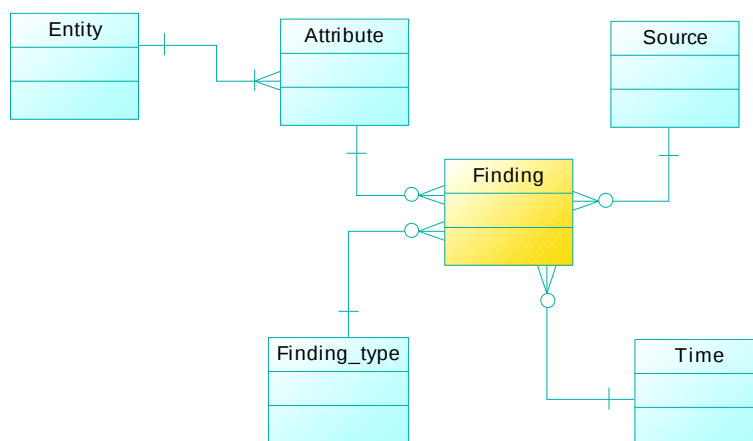
Klient může také sloužit jako nástroj pro administraci MDI hubu (nasazování, správa a sledování služeb, nasazování, spouštění a sledování workflow, nasazování a správa čistících a unifikačních pravidel).

4.3.6 Reporting datové kvality

Reporting datové kvality slouží k:

- sledování úrovně datové kvality v jednotlivých zdrojích
- sledování vývoje datové kvality v čase
- identifikaci problematických entit a atributů z pohledu kvality evidovaných dat (je možné zjistit, jaký atribut se evidoval/eviduje špatným způsobem a navrhnout např. metodický pokyn, jak atribut správně zapisovat)
- identifikaci nejfrekventovanějších defektů v datech

Souhrnný reporting datové kvality je realizován pomocí DQ datamartu. Tabulku faktů v datamartu představují nálezy datové kvality na úrovni atributů jednotlivých instančních záznamů.



Obrázek 9 – Zjednodušený logický model DQ datamartu

Díky této konstrukci je možné prohlížet reporty nálezů datové kvality z různých dimenzí, případně provádět drill-down až na konkrétní problematický záznam. Ukázka souhrnného reportu datové kvality se nachází v příloze práce (viz Příloha 10.4).

Součástí reportingu datové kvality jsou i reporty pro manuální čištění dat. Ty se používají:

- jako podklady pro manuální opravy prováděné přes uživatelské rozhraní zdrojového systému
- jako místo, kde jsou nejprve opravy zaznamenány, a teprve po jejich ověření jsou propagovány

Ukázka reportu pro manuální opravy dat je v příloze práce (viz Příloha 10.5).

5 Projekt implementace MDI hubu

V předchozí kapitole je popsáno, jak vypadá referenční architektura MDI hubu. V této kapitole je popsáno, z jakých aktivit se skládá projekt implementace MDI hubu. Popis jednotlivých aktivit je konceptuální (high-level) a dílčí aktivity jsou popsány volně.

5.1 Projektové role

Pro dodávku projektu jsou potřeba lidé s expertizou v různých oblastech. Role nejsou pevně ohraničené, je výhodou, pokud má pracovník v dané roli přesah některé z ostatních oblastí (např. business analytik do oblasti solution architekta a naopak, datový analytik do oblasti datové kvality a naopak, atd.)

5.1.1 Business analytik

Business analytik je člověk, který má přehled a zkušenosti v předmětné oblasti podnikových aktivit a dokáže komunikovat s business pracovníky „jejich řečí“. Zároveň by měl mít zkušenosti s procesním modelováním a s vytvářením zadání pro pracovníky vývoje. Měl by ovládat vytváření konceptuálních schémat v některém z běžně používaných diagram editorů (např. Microsoft Visio) a používání CASE nástrojů pro modelování (PowerDesigner, EnterpriseArchitect). Zodpovídá zejména za:

- Sběr a ověřování uživatelských požadavků
- Transformaci uživatelských požadavků do podoby funkční specifikace
- Získávání relevantní business dokumentace pro ostatní členy týmu
- Sběr business pravidel a získávání zpětné vazby na jevy nalezené v datech
- Metodické řízení návrhu reportů
- Návrh změn v business procesech, které budou ovlivněny nasazením MDI hubu

5.1.2 Datový analytik

Datový analytik je člověk, který má zkušenosti s explorativní analýzou dat, dokáže systematizovat poznatky o struktuře dat získané analýzou datových extraktů, umí číst datové modely na všech úrovních a dokáže vytvářet odvozené datové modely. Má smysl pro kontrolu konzistence mezi modely. Má zkušenosti s top-down i bottom-up metodami konstrukce datových modelů.

Zodpovídá za:

- Analýzu extraktů ze zdrojových systémů
- Identifikaci referenčních záznamů (metoda bottom-up) a atributů, které mají vstupovat do MD repository
- Identifikaci používaných číselníků a jejich konsolidaci (ve spolupráci s business analytikem)
- Návrh konceptuálního modelu MD repository (Master data model)
- Návrh mappingu datových modelů zdrojových systémů na master data model
- Kontrolu pokrytí uživatelských požadavků daty
- Návrh řešení auditování operací a historizace záznamů

- Návrh jmenných konvencí (konvence používané v datovém modelu se promítají kromě MD repository i do dalších komponent MDI hubu)

5.1.3 *Databázový specialista*

Databázový specialista je člověk s expertní znalostí databázových prostředí, zná i low-level koncepty používané v databázích. Dokáže optimalizovat fyzické uložení dat. Dokáže navrhnout optimální realizaci logického konceptu na úrovni databáze (např. přidělování náhradních klíčů při zachování referenční integrity, souběhy požadavků, obnovu po pádu).

Zodpovídá za:

- Návrh fyzického datového modelu (způsob realizace logického datového modelu v konkrétním databázovém prostředí)
- Optimalizaci databázových operací (návrh rozmístění a implementace indexů, používání proprietárních vlastností cílového DB systému pro urychlení operací, přepisování SQL dotazů)
- Optimalizace strategie uložení a přístupu k datům (partitioning, agregované uložení více databázových sloupců do BLOB a CLOB objektů, návrh dernormalizace dat z optimalizačních důvodů)

5.1.4 *Specialista datové kvality*

Specialista datové kvality má zkušenost s explorativní analýzou dat, zná best-practices pro práci s atributy běžně se vyskytujícími business entit, zná často se vyskytující problémy datové kvality. Dokáže navrhnout, jak optimálně řešit zjištěné defekty datové kvality. Umí transformovat business požadavky do instrukcí na implementační úrovni zvoleného DQ nástroje.

Zodpovídá za:

- Identifikaci podezřelých jevů v oblasti kvality dat
- Návrh pravidel pro čištění a konsolidaci dat při zohlednění best practices i zjištěných business pravidel
- Implementaci pravidel ve zvoleném DQ nástroji
- Návrh řešení zjištěných problémů datové kvality
- Vytváření podkladů pro plnění DQ datamartu
- Návrh reportů datové kvality a reportů pro manuální opravy dat

5.1.5 *Programátor*

Programátor je člověk, který umí překlomit návrh do fyzické realizace ve zvoleném jazyce (PL/SQL, shell skript, Perl, Java) a prostředí (Servlet, J2EE server). Jeho účast je omezena rozsahem na zakázku vytvářených součástí řešení.

Zodpovídá za:

- Implementaci ETL podle dodané specifikace
- Implementaci na míru navržených komponent MDI hubu podle dodané specifikace
- Self-Testování jednotlivých dodávaných komponent

5.1.6 Solution architect

Solution architect je člověk, který dokáže vidět „big picture“ výsledného řešení, vidí návaznosti řešení na okolní aplikace, dokáže řešení logicky rozdělit na samostatně vyvíjené celky a umí sestoupit až na implementační úroveň v dané dílčí oblasti, kde je v komunikaci rovnocenným partnerem např. programátorovi nebo databázovému specialistovi. Nemusí umět navrhnout low-level součásti řešení, ale musí jim rozumět. Má zkušenosti s návrhem informačních systémů a běžně používanými architektonickými návrhovými vzory. Rozumí sdíleným konceptům v návrhu informačních systémů a dokáže je komunikovat.

Zodpovídá za:

- Návrh architektury MDI hubu se zohledněním best practices a technologického prostředí u zákazníka a funkční specifikace
- Návrh vnějších rozhraní MDI hubu
- Dohled na realizaci nefunkčních požadavků (výkon, spotřeba HW prostředků, komunikace s ostatními aplikacemi/systémy, monitorování)
- Návrh workflow a orchestrace
- Integraci jednotlivých komponent MDI hubu do funkčního celku
- Metodické řízení vývoje na míru navržených komponent MDI hubu

5.2 Fáze projektu

Projekt implementace MDI hubu se skládá z typických vývojových fází:

- **analýza** – zjišťování uživatelských požadavků, zmapování prostředí
- **návrh** – popis způsobu realizace uživatelských požadavků (návrh datového modelu, návrh čistících a unifikčních pravidel, návrh workflow pro práci s navrženými datovými strukturami)
- **implementace** – vlastní realizace návrhu technickými prostředky (realizace datového modelu v databázi, realizace čistících a unifikčních pravidel ve zvoleném nástroji, implementace workflow a programování/konfigurování jeho součástí)
- **nasazení** – instalace MDI hubu v cílovém prostředí, úvodní migrace referenčních dat do MDI hubu, akceptační testy

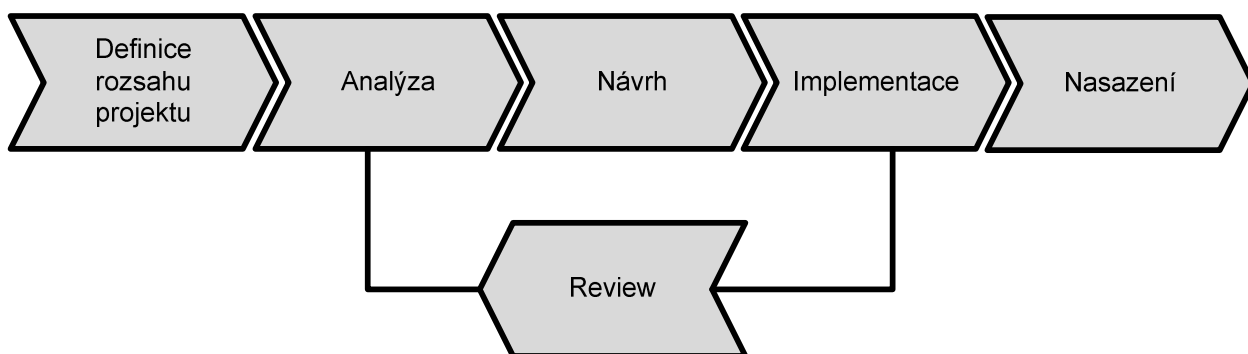
Těmito fázím předchází definice rozsahu projektu, která ale nespadá do věcné oblasti řešené v této práci. Rozsah projektu je nutné poměrně přesně omezit (co do projektu spadá a co do projektu nespadá) – z pohledu datových zdrojů a z pohledu poskytované funkcionality.

Jednotlivé fáze nejsou striktně odděleny, mohou probíhat částečně paralelně a jsou silně provázány. Důvodem je komplexita prostředí, do kterého se MDI hub nasazuje, a nemožnost navrhnout řešení jednorázově, protože požadavky zákazníka nejsou na začátku úplně přesné a v průběhu řešení se vyvíjejí. Některé zákaznické požadavky mohou být na začátku definované velmi volně a nekonkrétně a jejich upřesnění je možné až v pozdějších fázích vývoje, kdy je zákazník seznamován s konkrétními výstupy a výstupy dostávají finální podobu. Při rozhodování v analýze se může ukázat jako nezbytné provést „na zkoušku“ implementaci části řešení, aby se ověřilo, že je zvolený postup možné použít (např. z výkonnostního pohledu). Nebo je třeba ověřit dopad některých požadavků zákazníka na výsledné řešení a jediný způsob ověření je realizovat zákaznickův požadavek

v minimalistické verzi a zjistit dopady přímo. Je vhodné uplatnit některé z principů agilního vývoje:

- Několik iterací, v každé iteraci se řeší určitá definovaná sada uživatelských požadavků – každá iterace by měla být uzavřená a pokrývat jednu logickou/funkční oblast (např. systém nebo agendu)
- Uživatelské požadavky zjištěné v průběhu jiné než analytické fáze se zaevidují, klasifikují se (spadají do aktuální iterace, spadají do jiné iterace, out-of-scope) a určí se jejich priorita
- Během návrhu se jeho části prakticky ověřují implementací „nanečisto“ – vlastnímu projektu může předcházet tzv. „proof-of-concept“, ve kterém dodavatel prokazuje, že je skutečně schopen zrealizovat zadání definované zákazníkem
- Vyvíjené komponenty se průběžně testují nejen z funkčního pohledu, ale i z pohledu výkonu
- Neustále se pracuje s očekáváními zákazníka – je nutné průběžně ověřovat na reálných výstupech, že jsou v souladu se zákaznickými představami, a na případný nesoulad okamžitě reagovat. Díky intenzivnější komunikaci se zákazníkem a průběžnému review výstupů je více pravděpodobné, že zákazník obdrží výsledek, který očekává.

Agilní přístup umožňuje pružněji reagovat na nově zjištěná fakta, ale velkým rizikem pro tento přístup je nedostatečně zafixovaný rozsah (scope) projektu – zákazník generuje stále nové požadavky a vývoj na ně reaguje, protože nelze jednoznačně rozhodnout, zda jsou in-scope nebo out-of-scope.



Obrázek 10 - Schéma projektového cyklu

5.2.1 Analýza

V analytické fázi se řešitelský tým seznamuje s prostředím, ve kterém má dodávaná aplikace fungovat.

5.2.1.1 Katalogizace zdrojových systémů

Popis a motivace

Na základě dohodnutého rozsahu implementace provést soupis zdrojových systémů (datových zdrojů) a jejich charakteristik podstatných z pohledu implementace MDI hubu, a dát jejich popisu standardní strukturu. Jedná se zejména o:

- Datový model včetně business popisu uložených dat

- Formát uložení dat
- Aplikační rozhraní systému – dávková i online (webové služby)
- Závislé systémy (podklad pro impact analýzu)
- Možnosti synchronizace dat s MDI hubem (dávková, online)
- Možnost integrovat online služby MDI hubu s uživatelským rozhraním systému

Tyto informace jsou důležité pro návrh doménových pohledů, návrh ETL a návrh workflow pro synchronizaci zdrojového systému s MDI hubem.

Vstupy

- Scope projektu (seznam systémů)
- Technická i business dokumentace jednotlivých systémů
- Datové modely zdrojových systémů (i reverzované)
- Podněty vzešlé z profilingu

Výstupy

- Katalog zdrojových systémů (viz Příloha 10.3.1)

Zodpovědné role

- Business analytik, solution architekt

5.2.1.2 Katalogizace entit

Popis a motivace

Zjistit, kde a v jaké struktuře jsou uložena data o business entitách. Získat byznys popis jednotlivých entit a ověřit, jak uživatelé tyto entity chápou kontextu procesů, ve kterých aplikacích jsou data o entitách uložena a skrze jaká uživatelská rozhraní se k nim přistupuje. Při katalogizaci se rozlišuje, které entity mají charakter transakční a které referenční. Dalšími důležitými informacemi, které je potřeba získat, jsou:

- Metodické pokyny pro zadávání dat do uživatelských rozhraní
- Mapování mezi uživatelským rozhraním a datovým úložištěm systému
- Přehled validací prováděných při zadávání dat na úrovni uživatelského rozhraní a aplikační logiky
- Číselníky používané pro popis atributů entit

Tyto informace jsou důležité pro logický návrh datového modelu MDI hubu a pro návrh pravidel čištění dat. Díky katalogu entit je možné identifikovat příbuzné entity, různé role téže entity, případně vztahy mezi entitami typu generalizace-specializace.

Vstupy

- Přehled business entit získaný z procesních modelů
- Uživatelská dokumentace jednotlivých systémů
- Metodické dokumenty
- Odpovědi uživatelů získané z interview
- Sledování uživatelů při práci s aplikací

Výstupy

- Katalog entit (viz Příloha 10.3.2)
- Návrh mapování mezi reprezentací entit ve zdrojových systémech a master data modelem

- Přehled používaných číselníků a návrh jejich konsolidace

Zodpovědná role

- Business analytik

5.2.1.3 Profiling

Popis a motivace

Profiling se používá pro zjištění skutečného stavu dat uložených ve zdrojových systémech. Významné charakteristiky jsou počty záznamů (určení typu tabulky (referenční, transakční), má význam i pro sizing), vyplněnost/nevyplněnost (zjištění, zda se atribut používá, případně zda je povinný), četnost výskytu určitých hodnot (frekvenční analýza – vede k identifikaci číselníkových atributů, případně zástupných hodnot), nalezení extrémních hodnot a kvantilů (u číselných atributů velikost, u řetězcových atributů délka – vede k identifikaci nestandardních hodnot a využívá se při ověřování business pravidel), počty unikátních a distinktních hodnot (vedou k identifikaci primárních a alternativních klíčů), analýza vazeb mezi tabulkami.

Při profilingu se dále využívá:

- Analýza vzorů (pattern analysis) – kontroluje se vnitřní struktura atributu pomocí regulárních výrazů a definovaných řetězcových komponent
- Slovníková/číselníková analýza – ověřování hodnot atributů vůči číselníkům
- Kontextuální analýza – ověření, zda jsou atributy téhož záznamu ve vzájemném souladu (např. zda adresní komponenty tvoří dohromady existující adresu nebo zda název společnosti a IČO spolu souhlasí)
- Explorativní analýza („kouknu a vidím“) – slouží pro hledání podezřelých jevů, které nejsou předem známy/očekávány, jedná se v podstatě o náhodné prohlížení/zkoumání vybraných vzorků dat

Informace zjištěné při profilingu se používají při návrhu pravidel pro čištění a unifikaci a při návrhu fyzického datového modelu (velikost datových polí, datové typy).

Vstupy

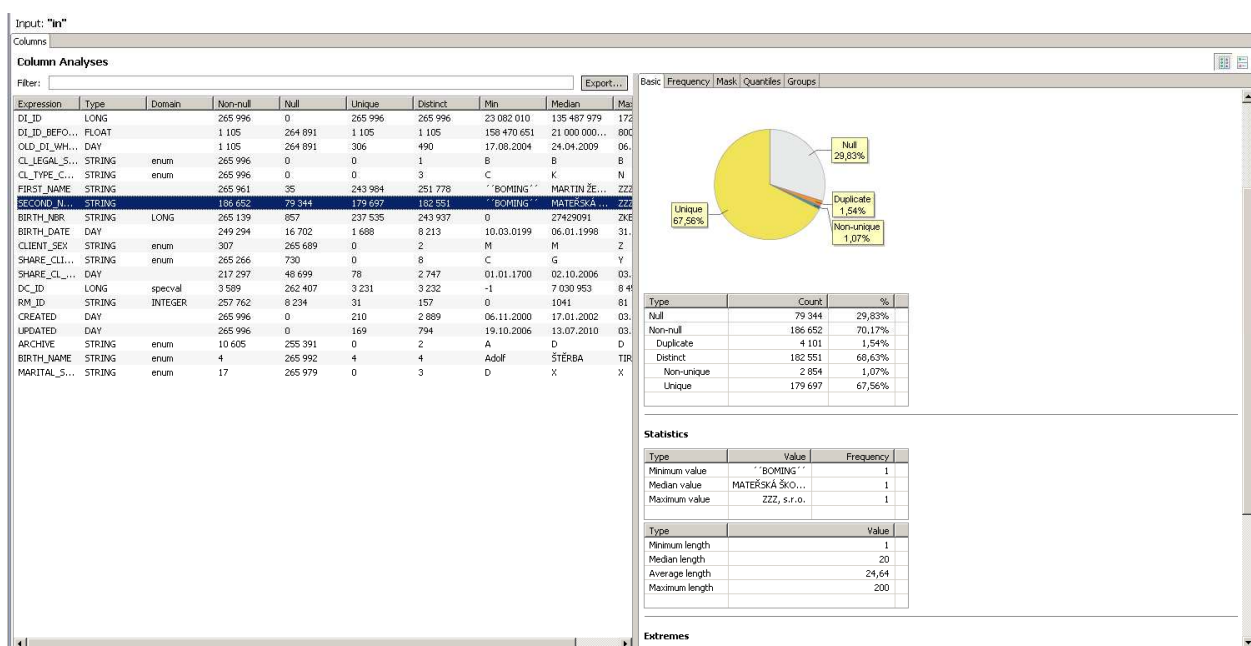
- Data zdrojových systémů
- Popis business pravidel

Výstupy

- Profilingová zjištění (viz Příloha 10.3.3)

Zodpovědná role

- Specialista datové kvality



Obrázek 11 – Ukázka výstupu profilingu v aplikaci Ataccama Data Quality Center
(www.ataccama.com)

5.2.1.4 Analýza a správa uživatelských požadavků

Popis a motivace

Uživatelské požadavky jsou základem pro funkční specifikaci, která říká, co má dodávané řešení dělat a co se od něj očekává. Tyto požadavky nemusí být všechny známy v okamžiku zahájení projektu, ale objevují se v průběhu projektu např. při interview nebo v reakci na průběžné výstupy.

Používané techniky jsou:

- Interview s business uživateli (sběr funkčních požadavků)
- Interview s technickými uživateli (sběr nefunkčních požadavků)
- Review požadavků a prioritizace

Vstupy

- Zadávací dokumentace
- Výstupy z interview
- Výstupy z workshopů

Výstupy

- Katalog uživatelských požadavků s prioritizací

Zodpovědné role

- Business analytik, solution architect

5.2.1.5 Návrh logického datového modelu master databáze

Popis a motivace

Cílem této aktivity je sjednotit různé reprezentace stejných entit do jednoho logického modelu. Z Enterprise data modelu se získává informace o vztazích mezi entitami, z katalogu entit informace o reprezentacích entit v rámci zdrojových systémů a jejich atributech. Při

vytváření logického modelu se rozhoduje, které entity budou uloženy v MDI hubu, a které jejich atributy budou uloženy v MDI hubu.

Logický návrh datového modelu je východiskem pro návrh fyzického datového modelu a pro ověření, zda jsou uživatelské požadavky saturovány z pohledu dat nutných k jejich splnění.

Vstupy

- Enterprise data model
- Katalog entit

Výstupy

- Logický datový model master databáze

Zodpovědná role

- Datový analytik

5.2.2 Design

5.2.2.1 Návrh cílové architektury MDI hubu

Popis a motivace

Návrh cílové architektury slouží pro celkový popis rámce řešení MDI hubu, jeho napojení na okolní systémy a zapojení do standardní IT infrastruktury podniku (např. ESB, monitoring). Při návrhu cílové architektury se zvažují dostupné technologické možnosti realizace jednotlivých komponent MDI hubu.

Vstupy

- Architektura podnikového IS/IT
- Referenční architektura MDI hubu
- Katalog zdrojových systémů a jejich rozhraní
- Uživatelské požadavky (nefunkční)
- Volba technologií pro realizaci jednotlivých komponent MDI hubu

Výstupy

- Cílová architektura MDI hubu
- Specifikace komponent

Zodpovědná role

- Solution architect

5.2.2.2 Návrh pravidel pro čištění dat

Popis a motivace

Na základě výsledků profilingu, best-practice pravidel pro čištění běžných entit a business pravidel se navrhují pravidla pro kontrolu, čištění a standardizaci atributů entit. Návrh pravidel se provádí iterativně a průběžně se testuje na vstupních datech. Navrhovaná pravidla, jejich výstupy a dopady jsou komunikovány se zákazníkem a je ověřeno, které transformace je možné použít automaticky a které budou podléhat manuálnímu potvrzení. Vyčištěné a standardizované atributy jsou vstupem pro unifikaci. Navržená pravidla pro čištění dat produkují při aplikaci na zdrojová data nálezy datové kvality.

Vstupy

- Best-practice pravidla čištění dat
- Výsledky profilingu
- Uživatelské požadavky

Výstupy

- Pravidla čištění dat
- Seznam indikátorů nálezů datové kvality s vysvětlením

Zodpovědná role

- Specialista datové kvality

5.2.2.3 Návrh unifikačních pravidel

Popis a motivace

Na základě best-practice pravidel pro unifikaci a výsledků profilingu jsou navržena pravidla, podle kterých bude docházet k seskupování záznamů reprezentujících stejný reálný objekt. Výsledky profilingu se používají pro výběr atributů použitelných k identifikaci konkrétní instance. Z těchto atributů se pak vytváří kombinace se silným identifikačním potenciálem (je možné s vysokou mírou spolehlivosti říci, že dva různé reálné objekty nebudou mít stejnou kombinaci těchto atributů). Výstupy unifikace je nutné průběžně testovat a konzultovat se zákazníkem. Pravidla musí být připravena i na možné řešení výjimek (zákazník si přeje nespojovat určité klienty nebo naopak spojit).

Vstupy

- Best-practice pravidla pro unifikaci
- Výsledky profilingu
- Pravidla konsolidace záznamů použita v datovém skladu
- Uživatelské požadavky

Výstupy

- Návrh pravidel pro unifikaci

Zodpovědná role

- Specialista datové kvality

5.2.2.4 Návrh fyzického datového modelu master databáze

Popis a motivace

Fyzický datový model popisuje realizaci navrženého logického modelu master databáze v konkrétním databázovém prostředí. Způsob fyzického uložení je optimalizován s ohledem na výkonnostní požadavky s použitím indexů, serializace velkých nebo málo používaných atributů do LOB, vertikálního i horizontálního partitioningu. Návrh fyzického datového modelu je iterativní a zohledňuje poznatky z testování. Změny ve fyzickém datovém modelu by neměly mít vliv na logický datový model.

Vstupy

- Logický datový model

Výstupy

- Fyzický datový model

Zodpovědná role

- Databázový specialista

5.2.2.5 Návrh rozhraní a služeb

Popis a motivace

Návrh způsobu realizace standardních rozhraní MDI hubu (dávková a online synchronizace referenčních dat, práce s výjimkami a číselný). Při zpracování uživatelských požadavků může být identifikována potřeba implementovat specializované rozhraní. Podle návrhu rozhraní se pak postupuje při vlastní implementaci rozhraní a zároveň slouží jako dokumentace pro ty, kteří budou rozhraní používat.

Vstupy

- Návrh architektury MDI hubu
- Uživatelské požadavky

Výstupy

- Popis rozhraní služeb (včetně výběru technologie a popisu způsobů užití)

Zodpovědná role

- Solution architect

5.2.2.6 Návrh ETL

Popis a motivace

Návrh ETL zahrnuje návrh transformací, které převádí zdrojová data do základní podoby. Při návrhu ETL se řeší také výpočet inkrementu. Návrh ETL může zahrnovat i návrh realizace datových toků uvnitř MDI hubu, pokud toto neřeší nějaký standardní nástroj.

Vstupy

- Logický model master databáze
- Katalog zdrojových systémů
- Katalog entit

Výstupy

- Návrh implementace ETL (způsob realizace, konkretizace na úroveň procedur, funkcí, balíčků, komponent, kroků)

Zodpovědná role

- Databázový specialista (může být nahrazen osobou specializující se na návrh a implementaci ETL – ETL programátorem)

5.2.2.7 Návrh workflow

Popis a motivace

Workflow se používá pro realizaci rutinních operací v MDI hubu. Výchozí návrh workflow vychází z referenční architektury, která obsahuje generická workflow. Pokud uživatelské požadavky obsahují něco, co je třeba opakovaně řešit a řešení lze automatizovat, navrhne se pro tento účel specializované workflow.

Vstupy

- Referenční architektura MDI hubu
- Uživatelské požadavky

Výstupy

- Návrh workflow pro implementaci

Zodpovědná role

- Solution architect

5.2.2.8 Návrh integračních testů

Popis a motivace

Integrační testy se navrhují tak, aby pokryly veškerou deklarovanou kooperaci mezi jednotlivými komponentami v rámci MDI hubu. Zároveň se také testuje komunikace MDI hubu s okolními systémy skrze definovaná rozhraní

Vstupy

- Cílová architektura MDI hubu
- Specifikace komponent

Výstupy

- Testovací scénáře

Zodpovědná role

- Solution architect (při návrhu a realizaci může být zastoupen osobou, která se specializuje na testování)

5.2.3 Implementace

5.2.3.1 Implementace ETL

Popis a motivace

ETL procesy – extrakce dat ze zdrojových systémů. Součástí implementace jsou i funkční (unit) testy, při kterých se ověřuje správnost přesunů dat a vytváření inkrementů. Při implementaci se provádí také zátěžové testy, a případné nedostatky se řeší buď optimalizací návrhu ETL nebo optimalizací fyzického modelu master databáze

Vstupy

- Návrh implementace ETL

Výstupy

- Naimplementovaná ETL komponenta řešení
- Instalační dokumentace a skripty

Zodpovědná role

- Programátor (může být nahrazen osobou specializující se na návrh a implementaci ETL – ETL programátorem)

5.2.3.2 Konfigurování DQ nástroje

Popis a motivace

Realizace čisticích a unifikačních pravidel ve zvoleném nástroji. Probíhá zpravidla souběžně s návrhem pravidel pro čištění a unifikaci, protože navržená pravidla se průběžně testují a výsledky se konzultují se zákazníkem.

Vstupy

- Návrh čisticích a unifikačních pravidel

Výstupy

- Nakonfigurovaná DQ komponenta/nástroj
- Instalační dokumentace a skripty

Zodpovědná role

- Specialista datové kvality

5.2.3.3 Implementace workflow

Popis a motivace

Workflow jsou implementována v souladu s dokumentovaným návrhem workflow. Implementace spočívá ve vytvoření konkrétního skriptu/konfiguračního souboru, který popisuje/řídí průběh vykonávání workflow. Součástí implementace je i funkční testování workflow.

Vstupy

- Návrh workflow pro implementaci

Výstupy

- Naimplementované a otestované workflow
- Instalační dokumentace a skripty

Zodpovědná role

- Programátor

5.2.3.4 Implementace rozhraní a služeb

Popis a motivace

Na základě návrhu rozhraní a služeb se realizuje implementace na zvolené technologické platformě (Java, PL/SQL, proprietární řešení MD serveru) a s použitím definované technologie rozhraní (webová služba, databázové rozhraní, souborové rozhraní). Součástí implementace jsou i funkční a výkonnostní testy.

Vstupy

- Návrh služby

Výstupy

- Realizovaná a otestovaná služba
- Instalační dokumentace a skripty

Zodpovědná role

- Programátor

5.2.3.5 Integrační testování

Popis a motivace

Pomocí integračních testů se ověřuje, že jednotlivé komponenty MDI hubu spolu korektně spolupracují. Vlastní testování dílčích komponent (unit testy) se provádí již při implementaci komponent. Při integračních testech se ověřuje také funkčnost vnějších rozhraní MDI hubu a jejich spolupráce s okolními aplikacemi.

Vstupy

- Komponenty MDI hubu
- Testovací scénáře

Výstupy

- Výsledky testovacích scénářů

Zodpovědná role

- Solution architect (při návrhu a realizaci může být zastoupen osobou, která se specializuje na testování)

5.2.4 Review

V této fázi jsou prezentovány výstupy aktuální iterace zákazníkovi a je od něj získána zpětná vazba, která se zohlední v další iteraci. Review se provádí formou workshopů a oponentur.

5.2.5 Nasazení

5.2.5.1 Instalace v testovacím/produkčním prostředí

Popis a motivace

Jednotlivé komponenty a jejich funkční celky se vyvíjejí a testují ve vývojovém prostředí. Ve chvíli, kdy je aplikace dokončena a připravena k nasazení, se nasadí nejprve do testovacího prostředí a po úspěšném dokončení akceptačního testování i do produkčního prostředí. Při nasazování do testovacího prostředí se zároveň ověřuje hladký průběh instalace a úplnost instalačních pokynů.

Vstupy

- Komponenty MDI hubu
- Instalační a konfigurační skripty
- Instalační dokumentace

Výstupy

- Nainstalovaná a provozuschopná instance MDI hubu

Zodpovědná role

- Solution architect

5.2.5.2 Úvodní load dat

Popis a motivace

Ve chvíli, kdy je instance MDI hubu zprovozněna, musí se ještě jednorázově naplnit referenčními daty zdrojových systémů. Tato operace se provádí na testovacím i produkčním prostředí. Je možné realizovat buď formou „velkého třesku“ (všechno v rámci jedné migrace) nebo formou postupného importu a připojování jednotlivých systémů. Součástí úvodního loadu dat je i úvodní vyčištění a konsolidace dat.

Vstupy

- Extrakty dat ze zdrojových systémů

Výstupy

- Instance MDI hubu naplněná referenčními daty

Zodpovědná role

- Databázový specialista

5.2.5.3 Akceptační testy

Popis a motivace

Akceptační testy slouží k tomu, aby si zákazník ověřil, že dodaná aplikace skutečně splňuje požadavky a parametry, na nichž se dohodl s dodavatelem. Dodavatel jimi dokazuje, že splnil požadavky kladené zákazníkem.

Vstupy

- Instance MDI hubu naplněná referenčními daty
- Testovací scénáře

Výstupy

- Výsledky testování

Zodpovědná role

- Zástupce zákazníka

6 Závěr

Pokračující ekonomický růst ve sféře MDM navzdory ekonomické krizi [The Information Difference, Ltd.] naznačuje, že si podniky uvědomují důležitost integrace referenčních dat napříč systémy a sdílení „jednotné verze pravdy“. Jedním z předpokladů pro zavedení správy referenčních dat podle principů Master Data Managementu v podniku je realizace sdíleného úložiště referenčních dat – MDI hubu.

Výchozí stav problémové oblasti, kterou je podniková datová základna, je popsán ve druhé kapitole. Jako významný problém současných podnikových datových základen byla indikována jejich nedostatečná integrovanost. Identifikovaný problém je možné řešit skrze integraci referenčních dat. Zvoleným přístupem k řešení problému je aplikace principů Master Data Managementu (MDM). MDM představuje koncept systematické práce s referenčními daty v podniku. Je zastřešujícím termínem pro další související oblasti (řízení datové kvality, řízení metadat, Data Governance, architektura podnikového IS/IT). Technickou realizací konceptu MDM je MDI hub – systém, který umožňuje agregovat referenční data z různých nesourodých zdrojových systémů, konsolidovat je a poskytovat je ve standardní formě zpět uživatelům i ostatním systémům. Hlavním cílem práce je navrhnout způsob realizace MDI hubu. Cíle bylo dosaženo prostřednictvím návrhu referenční architektury MDI hubu a popisu jednotlivých aktivit, které se při implementaci MDI hubu vykonávají.

Třetí kapitola, která má v práci podpůrnou roli, popisuje techniky použité a zohledněné při návrhu referenční architektury MDI hubu a při návrhu její realizace. Zde chci zdůraznit význam modelovacích technik, mapování a roli výběru správného modelovacího nástroje. Zvolený modelovací nástroj musí být dostatečně flexibilní, aby kromě databázových struktur a mapování dokázal pojmout i rozšířená (custom) metadata. Pokud se artefakty výsledného řešení mohou z metadat modelu generovat přímo (včetně dokumentace)²², významně se redukuje objem rutinních prací při implementaci a snižuje se riziko chyby zapříčiněné lidským faktorem. Techniku mapování je kromě dat vhodné použít i na kontrolu vzájemné konzistence jednotlivých částí modelu a kontrolu pokrytí uživatelských požadavků.

Ve čtvrté kapitole je popsán návrh referenční architektury MDI hubu a jednotlivé komponenty, z nichž se MDI hub skládá. Návrh je vytvořen převážně na konceptuální úrovni, a možné formy fyzické implementace jsou pouze nastíněny. Důvodem je, že výsledná podoba implementace silně závisí na konkrétním cílovém prostředí.

Pro návrh referenční architektury MDI hubu jsem zvolil perzistentní typ řešení (repository). Výhodou zvolené alternativy je to, že architekt má různé aspekty implementace více pod kontrolou než v případě virtuálního typu (registry) a řešení má řadu podobných rysů s návrhem a implementací datového skladu (DWH), takže je možné využít poznatky a zkušenosti z oblasti DWH.

²² Například pokud jsou metadata uložena v XML formátu, je možné použít pro automatizované generování artefaktů XSL transformace. Propagace případných změn je snadnější proto, že změny se provádí na jednom místě (model, resp. transformace).

Referenční architektura vychází z obecných požadavků na MDI hub. Obecné požadavky jsou pak při konkrétní implementaci doplněny o zákaznické požadavky (funkční) a požadavky infrastruktury (nefunkční). Referenční architektura je při konkrétní implementaci upravena tak, aby těmto požadavkům vyhovovala.

V rámci architektury MDI hubu chci zdůraznit význam návrhu datového modelu MDI hubu. Dobrý a dostatečně flexibilní datový model je základem pro jakýkoli MDM projekt. Algoritmy používané pro čištění a konsolidaci dat nemusí být od samého začátku optimální, mohou se měnit a vyvíjet, ale špatně navržený datový model představuje výraznou překážku v dalším rozvoji MDI hubu. Model by měl kromě cílové struktury master databáze zachycovat i datové struktury zdrojových systémů (v rozsahu nezbytném pro mapování na datové struktury MDI hubu) a definici mapování mezi zdrojovými systémy a master databází. Celkově by měl model obsahovat všechny informace potřebné pro vytvoření cílové struktury MDI hubu v databázi a její naplnění daty ze zdrojových systémů.

V páté kapitole je uvedeno rozdělení vývoje na etapy a přehled aktivit, které se odehrávají v průběhu projektu implementace MDI hubu v rámci jednotlivých etap. Nejprve jsou vymezeny role, které budou zodpovídat za vykonání jednotlivých aktivit. U každé aktivity je uvede popis (co se dělá) a motivace (proč se to dělá), vstupy a očekávané výstupy, a zodpovědná role. Aktivita jsou popsány na konceptuální úrovni, předpokládá se jistá míra zkušenosti na straně osoby v zodpovědné roli.

Při analýze je nutné uvažovat výsledné řešení v plné šíři a širších souvislostech proto, aby se řešení při dalším rozvoji nedostalo do slepé uličky. Při implementaci se v praxi osvědčil inkrementální přístup – vytváření malých, smysluplných a samostatně funkčních přírůstků. Díky tomu je možné zákazníka průběžně seznamovat s výsledky a případná rizika a problémy se daří identifikovat mnohem dříve. I zde platí pravidlo, že čím dříve se riziko podaří odhalit, tím dříve je možné mu předcházet a tím mírnější jsou případné následky selhání. Při inkrementálním vývoji je samozřejmě nezbytné neustále provádět integrační testy, aby bylo zaručeno fungování inkrementů společně s již vyvinutým celkem.

V práci není rozebíráno téma Data Governance, nicméně je nutné, aby se otázky datových standardů, odpovědnosti za data (stewardship) a řízení rozvoje datové základny (governance) řešily přinejmenším v souběhu s implementací MDI hubu.

Práce se nezabývá problematikou provozu, udržování a rozšiřování MDI hubu, což by mohlo být vhodné téma pro případnou navazující práci.

Možnost dalšího rozvoje výstupů práce dále spatřuji například v iterativním rozšiřování a zpřesňování popisu jednotlivých aktivit probíhajících při realizaci implementačního projektu MDI hubu, jejich plánování a odhady pracnosti. Další oblast, na kterou je možné navázat, je identifikace architektonických vzorů (architectural patterns) v architektuře MDI hubu, a jejich zdokumentování. Architektonický vzor je opakuující se řešení určité oblasti architektury MDI hubu, které má smysl podrobně popsat (co se dělá, proč se to dělá), zvážit alternativy řešení, vybrat optimální řešení a to zdokumentovat (jak by se to mělo dělat). Příkladem architektonického vzoru může být např. Historizace.

Další prostor pro rozvoj vidím v podrobném rozpracování metodiky pro ověření souladu návrhu a požadavků pomocí mapování požadavků na jednotlivé součásti řešení, a metodiky pro ověření vzájemné konzistence jednotlivých modelů použitých při návrhu MDI hubu.

7 Seznam obrázků

Obrázek 1 – Modely a kompromisy při jejich realizaci [Simsion, a další, 2005]	17
Obrázek 2 – Vztah mezi vrstvami modelu [Harmonizing company-wide Information Objects, 2008]	19
Obrázek 3 – Cyklus řízení datové kvality [Adastra Corporation]	23
Obrázek 4 – Referenční architektura MDI hubu.....	29
Obrázek 5 – Zdrojová, základní a kanonická struktura.....	33
Obrázek 6 – Reprezentace konsolidovaných zdrojových dat v doménovém pohledu.....	36
Obrázek 7 – Schéma fyzického uložení dat v master databázi.....	37
Obrázek 8 – Ataccama DQIT - aplikace pro řešení nálezů datové kvality[Ataccama, 2010]...	47
Obrázek 9 – Zjednodušený logický model DQ datamartu.....	48
Obrázek 10 – Schéma projektového cyklu	52
Obrázek 11 – Ukázka výstupu profilingu v aplikaci Ataccama Data Quality Center (www.ataccama.com)	55

8 Použitá literatura

Adastra Corporation. *Master Data Management Overview*. [Online] [Citace: 11. 12. 2010.] http://mdm.adastragrp.com/document.aspx?menu_id=236&id=2.

—. *Data Cleansing & Consolidation*. [Online] [Citace: 11. 12. 2010.] <http://mdm.adastragrp.com/document.aspx?id=531>.

Allen, Sharon a Terry, Evan. 2005. *Beginning Relational Data Modeling*. 2nd Edition. místo neznámé : Apress, 2005. ISBN 1-590-59463-0.

Ambler, Scott W. *Effective Practices for Modeling and Documentation. Agile Modeling*. [Online] [Citace: 11. 12. 2010.] <http://www.agilemodeling.com/>.

Ataccama. 2010. *DQIT Overview*. [Online] 2010. [Citace: 11. 12. 2010.] <http://www.ataccama.com/en/products/dqit/dqit-overview.html>.

Berson, Alex a Dubov, Lawrence. 2007. *Master Data Management and Customer Data Integration for a Global Enterprise*. místo neznámé : The McGraw-Hill Companies, 2007. ISBN 0-07-151089-3.

Cockburn, Alistair. 2000. *Writing Effective Use Cases*. místo neznámé : Addison-Wesley Professional, 2000. ISBN 0-201-70225-8.

Data Cleaning: Problems and Current Approaches. **Rahm, Erhard a Hai Do, Hong.** 2000, IEEE Data Engineering Bulletin, Sv. 23. <http://dbs.uni-leipzig.de/file/TBDE2000.pdf>.

Dyché, Jill a Levy, Evan. 2006. *Customer Data Integration: Reaching a Single Version of the Truth*. Hoboken, New Jersey : John Wiley & Sons, Inc., 2006. ISBN 0-471-91697-8.

Engels, G., a další. *Process Modeling using UML*. [Online] [Citace: 11. 12. 2010.] <http://www.cs.le.ac.uk/people/rh122/papers/2005/EFHT05PAIS.pdf>.

Eriksson, Hans-Erik a Penker, Magnus. 2000. *Business Modeling with UML: Business Patterns at Work*. místo neznámé : John Wiley & Sons, Inc., 2000. ISBN 0-471-29551-5.

Harmonizing company-wide Information Objects. **Schmidt, Alexander and Boris, Otto.** 2008. St. Gallen : GI, 2008. DW 2008 - Synergien durch Integration und Informationslogistik. Vol. 138, pp. 147-164. ISBN 978-3 88579-232-1.

Imhoff, Claudia. 2004. *How to "Excel" in Your Business Intelligence Environment*. *BeyeNETWORK*. [Online] 7. 12. 2004. [Citace: 11. 12. 2010.] <http://www.b-eye-network.com/view/467>.

Kotek, Aleš. 2008. *Metodika auditu datové kvality*. Praha : Vysoká škola ekonomická v Praze, 2008. Diplomová práce.

Kowalski, Theodore. 1999. *The Corporate Model: A Blast From the Past*. *Information Management*. [Online] 4. 1999. [Citace: 11. 12. 2010.] <http://www.information-management.com/infodirect/19990401/877-1.html>.

Loshin, David. 2009. *Master Data Management*. místo neznámé : Morgan Kaufmann Publishers, 2009. ISBN 978-0-12-374225-4.

Marco, David. 2000. *Meta Data repository: A Full Lifecycle Guide*. místo neznámé : John Wiley & Sons, Inc., 2000. ISBN 0-471-35525-2.

Novotný, Ota, Pour, Jan a Slánský, David. 2005. *Business Intelligence: Jak využít bohatství ve vašich datech*. Praha : Grada, 2005. ISBN 80-247-1094-3.

- Putchala, Surya. 2007.** MDM: A Benefits Analysis. *Information Management*. [Online] 20. 12. 2007. [Citace: 11. 12. 2010.] http://www.information-management.com/specialreports/2007_56/10000385-1.html.
- Reverse Engineering and Design Recovery: A Taxonomy.* **Chikofsky, Elliot J. a Cross II, James H. 1990.** 1, místo neznámé : IEEE Computer Society Press, leden 1990, IEEE Softw., Sv. 7, stránky 13-17. <http://www.weidl-rektenwald.at/swe/chikofsky90.pdf>. ISSN 0740-7459.
- Řepa, Václav. 2007.** *Podnikové procesy: procesní řízení a modelování*. 2., aktualizované a rozšířené vydání. Praha : Grada, 2007. ISBN 80-247-2252-8.
- Schmidt, Douglas C. 2006.** Model-Driven Engineering. [Online] únor 2006. [Citace: 11. 12. 2010.] <http://www.cs.wustl.edu/~schmidt/PDF/GEI.pdf>.
- Simsion, Graeme C. a Witt, Graham C. 2005.** *Data Modeling Essentials*. místo neznámé : Morgan Kaufmann Publishers, 2005. ISBN 0-12-644551-6.
- Summary of current work ANSI/X3/SPARC/study group: database systems.* **Bachman, Charles W. 1974.** 3, New York : ACM, červenec 1974, SIGMOD Rec., Sv. 6, stránky 16-39. ISSN 0163-5808.
- The Information Difference, Ltd.** The MDM Landscape Q2 2010. *The Information Difference website*. [Online] [Citace: 11. 12. 2010.] http://www.informationdifference.com/mdm_landscape.html.
- User requirements analysis_A review of supporting methods.* **Maguire, Martin a Bevan, Nigel. 2002.** Montreal : Kluwer Academic Publishers, 2002. IFIP 17th World Computer Congress. stránky 133-148. http://www.nigelbevan.com/papers/WCC_UserRequirements.pdf.
- Voříšek, Jiří. 2003.** *Strategické řízení informačního systému a systémová integrace*. Praha : Management Press, 2003. ISBN 80-85943-40-9.
- Wiegers, Karl E. 2003.** *Software Requirements*. 2nd edition. Redmond, Washington : Microsoft Press, 2003. ISBN 0-7356-1879-8.
- Wikipedia.** *Wikipedia - the Free Encyclopedia*. [Online] [Citace: 11. 12. 2010.] http://en.wikipedia.org/wiki/Master_data_management.

9 Terminologický slovník

Pojem/zkratka	Popis/vysvětlení
Bottom-up přístup	Analytický přístup, při kterém se z dílčích prvků odvozuje fungování systému, který tyto prvky tvoří. V přeneseném významu se jedná o přístup v modelování, kdy z detailních (fyzických) datových modelů agregují a odvozují modely vyšších vrstev (logické a konceptuální).
Business klíč	Sada významových atributů, která umožňuje s poměrně vysokou spolehlivostí spojit dva instanční záznamy reprezentující stejný reálný objekt, případně rozlišit od sebe instanční záznamy reprezentující různé reálné objekty.
Business pravidlo	Označení pro pravidla, kterými se řídí fungování podniku. Business pravidla zahrnují podnikové směrnice, nařízení ze strany autorit, standardy daného ekonomického odvětví, účetní pravidla a výpočetní algoritmy
Doménový pohled (domain view)	Specifická forma prezentace referenčních dat pro konkrétní účel nebo konkrétního uživatele. Doménový pohled zohledňuje výsledky unifikace a vazby mezi konsolidovanými a instančními záznamy
Entita	Kategorie reálných objektů, které mají společný význam, jsou popisovány stejnou sadu atributů a mohou se vyskytovat ve stejných vztazích (např. Osoba)
ESB	Enterprise Service Bus
Instance	Konkrétní výskyt dané entity (např. osoba Jan Novák, rodné číslo 700101/1234)
Kmenová data	Viz. Referenční data
Konsolidace	Proces, ve kterém je z více instančních záznamů, které reprezentují tentýž reálný objekt, vytvářen jeden záznam obsahující nejlepší dostupné informace o daném reálném objektu pocházející ze všech instančních záznamů.
Labeling datové kvality	Proces, ve kterém jsou označovány nálezy datové kvality pomocí předem definovaných kódů - "štítků". Každý atribut/entita může mít právě jeden štítek, který zpravidla reprezentuje nejzávažnější nálezy datové kvality.
LOB	Large Object, velké databázové objekty - způsob uložení nestrukturovaných dat do databáze (realizace např. v Oracle formou BLOB nebo CLOB sloupců)
Master data	Záznamy reprezentující jednotnou verzi pravdy, vznikají konsolidací a masteríngem referenčních dat.
Master data management	Souhrnný pojem pro principy, procesy, technologie a nástroje, které se uplatňují při správě master dat
MDM	Viz. Master Data Management
Nález datové kvality	Jakékoli zjištění týkající se kvality zdrojových dat, které má určitý business význam/dopad a které může vyžadovat nějakou akci/řešení ze strany uživatele (např. doplnění chybějícího údaje nebo oprava chybného údaje).
Proof-of-concept	Provedení určitého postupu nebo realizace určité myšlenky s cílem demonstrovat, že je postup proveditelný nebo myšlenka realizovatelná. Většinou se jedná malou a částečnou jednoúčelovou realizaci.

Pojem/zkratka	Popis/vysvětlení
Reálný objekt	Objekt existující v reálném světě, který je možné rozlišit od ostatních objektů reálného světa. Reálný objekt může být na úrovni systému popisován i více instančními záznamy, což je nežádoucí (nejednoznačnost, redundantní uložení dat), který řeší Unifikace
Referenční data	Data/záznamy, která reprezentují reálné objekty nebo koncepty (fyzické a právnické osoby, lokality, kontaktní údaje, produkty, hmotný investiční majetek apod.). U tohoto typu dat má smysl provádět konsolidaci a mastering (vytváření jednotné sdílené reprezentace záznamu)
Single version of truth	Jednotná verze pravdy - termín používaný pro ideální stav, kdy má každý v podniku o dané entitě stejné informace (všichni vychází ze stejných dat)
System-of-record	Systém, který slouží jako autoritativní zdroj informací pro danou entitu
Top-down přístup	Analytický přístup, při kterém se systém rozkládá na dílčí prvky a zjišťuje se, jakým způsobem spolu dílčí prvky tvoří dohromady systém. V přeneseném významu se jedná o přístup v modelování, kdy se postupuje z úrovně konceptuální (business cíle, business procesy, business entity) na úroveň logickou a fyzickou.
Unifikace	Proces, ve kterém jsou instanční záznamy seskupovány. Cílem je, aby instanční záznamy představující tentýž objekt byly ve stejné skupině. Pro realizaci unifikace se používají unifikační pravidla a business klíče.

10 Přílohy

10.1 Dodavatelé MDM software, řešení a služeb

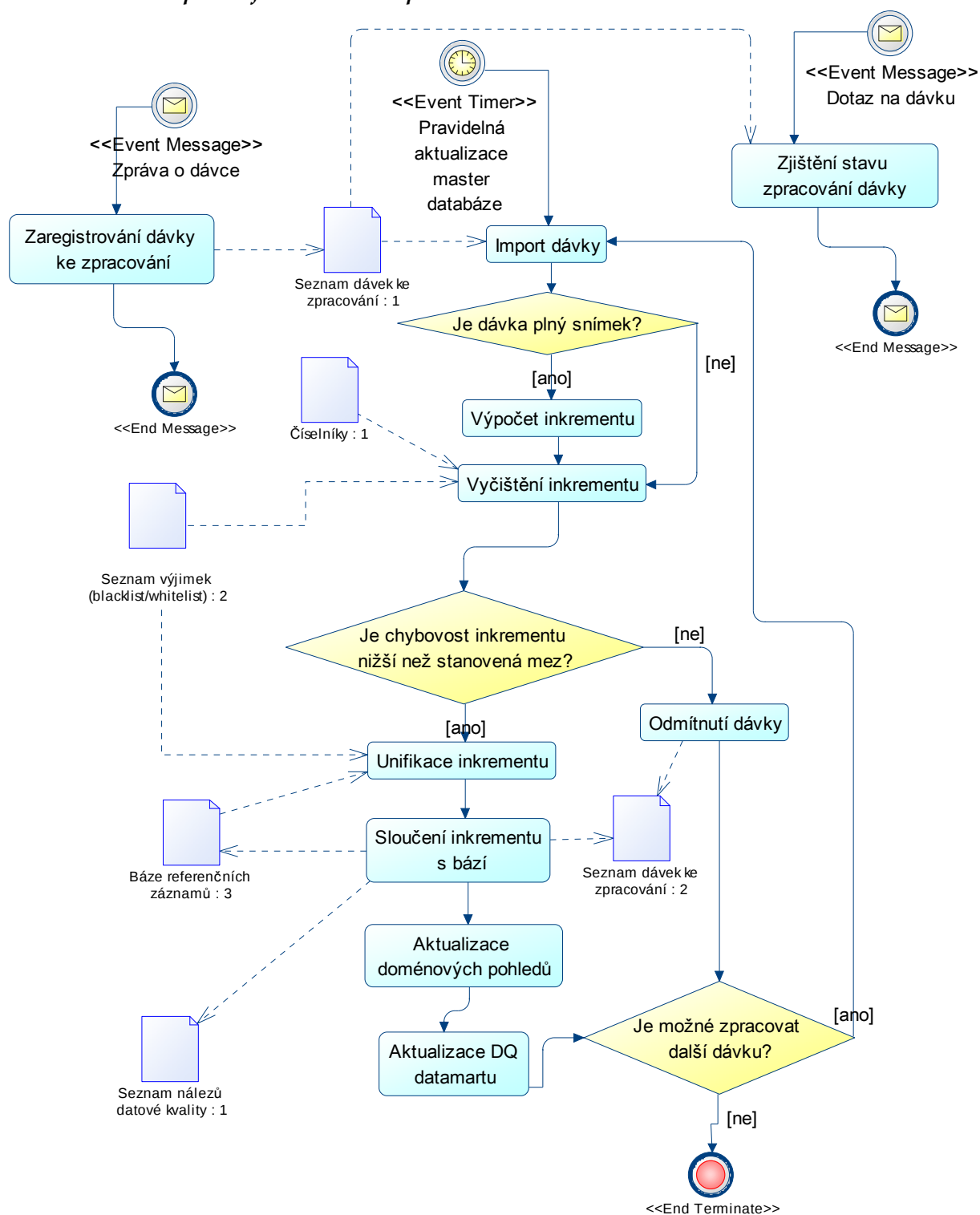
zdroj: [The Information Difference, Ltd.]

Vendor	Brief Description	Website
Ataccama	Data quality vendor who have added an MDM hub to their portfolio	www.ataccama.com
Cadis	Enterprise data management vendor specializing in financial services industry	www.cadisedm.com
Data Foundations	U. S. based enterprise data management vendor	www.datafoundations.com
D&B Purisma	Customer hub software acquired by business services giant D&B.	www.purisma.com
DataFlux	Part of the SAS Institute, DataFlux has a customer hub solution	www.DataFlux.com
Global IDs	New York-based vendor with an emphasis on data governance	www.globalids.com
Golden Source	UK vendor specializing in financial services industry	www.thegoldensource.com
GXS	Vendor specializing in product master data but which can manage other data domains.	www.gxs.com www.gxsproductmasterdata.com
Heiler	Vendor specializing in product master data	www.heiler.com
Hybris	Vendor specializing in product master data	www.hybris.com
IBM	Industry giant with comprehensive MDM offering, now including Initiate.	www.ibm.com
Informatica (Siperian)	Multi-domain MDM vendor.	www.informatica.com
Information Builders (IBI)	Business Intelligence vendor now with an MDM solution: iWay Master Data Center	www.informationbuilders.com
Kalido	Information management vendor with emphasis on data governance	www.kalido.com
Oracle	Database and applications giant with several MDM solutions	www.oracle.com
Orchestra Networks	A model-driven, multi-domain MDM vendor with an emphasis on data governance	www.orchestranetworks.com
QAD	Vendor specializing in product master	www.qad.com

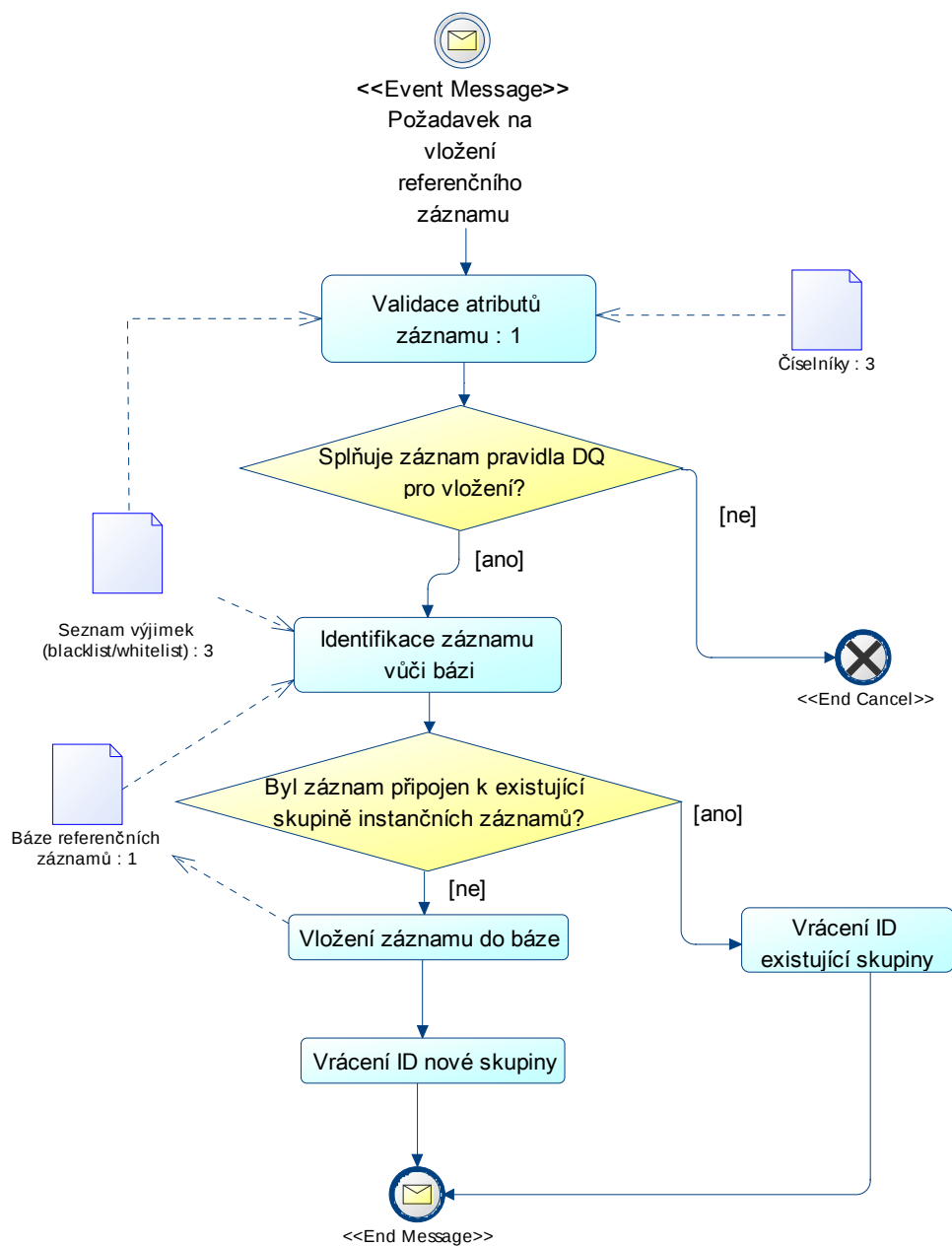
Vendor	Brief Description	Website
	data	
Riversand	Texas vendor specializing in product data	www.riversand.com
Rollstream	Vendor specializing in supplier master data	www.rollstream.com
SAP	Applications giant with its own MDM solution	www.sap.com
Smartco	Enterprise data management vendor specializing in financial services industry	www.smartco.com
Sparesfinder	Vendor specializing in the mastering of spares part data	www.sparesfinder.com
Stibo	Vendor specializing in product, supplier and location master data	www.stibosystems.com
Talend	Open source vendor, now with MDM suite	www.talend.com
Teradata	Database giant with its own MDM solution	www.teradata.com
Tibco	Integration vendor with product-oriented MDM hub	www.tibco.com
Visionware	MDM vendor based in UK, historically specializing in local government, now expanding abroad and into new markets	www.visionwareplc.com

10.2 Standardní workflow MDI hubu

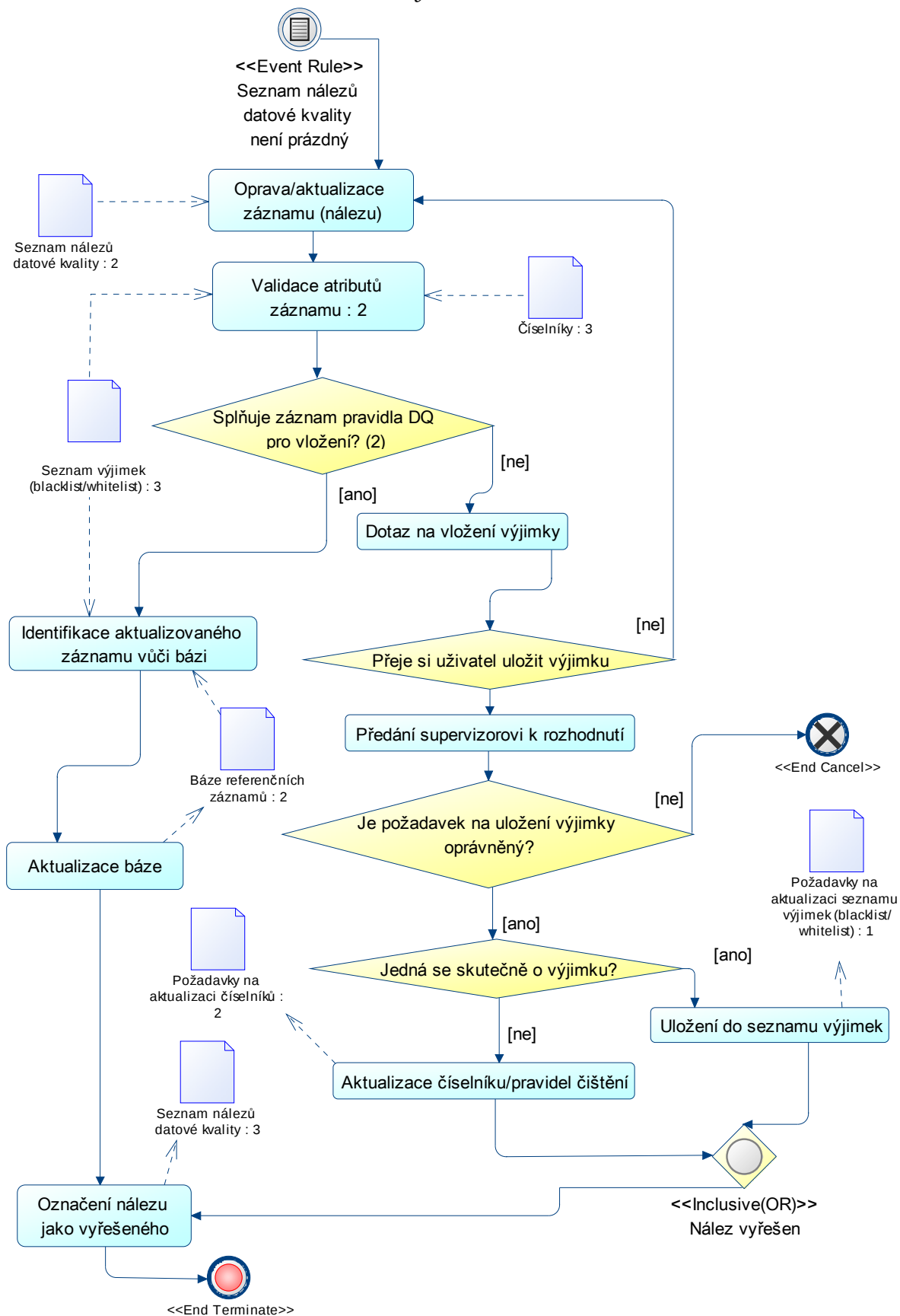
10.2.1 WF1 - Import referenčních dat přes dávkové rozhraní



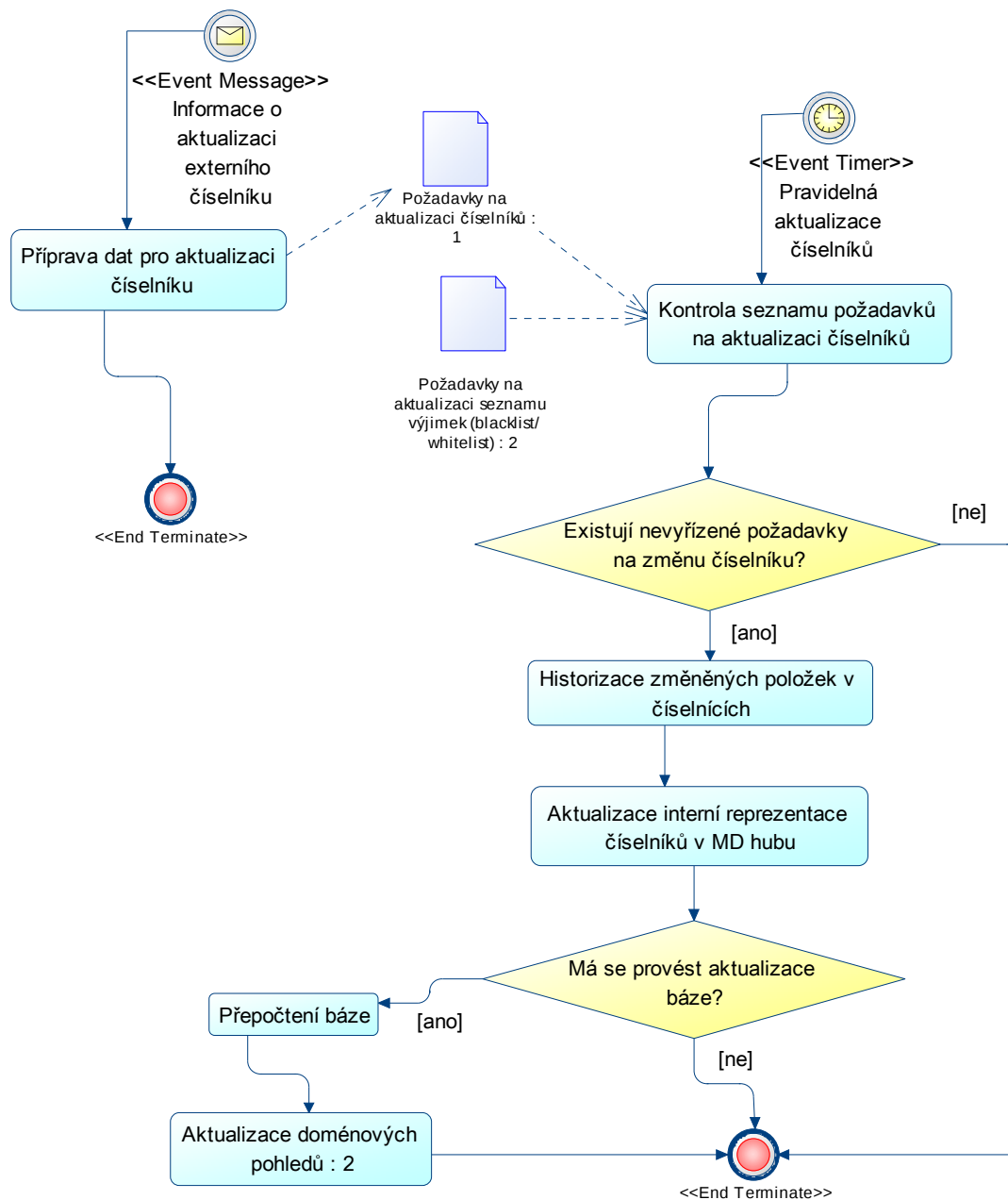
10.2.2 WF2 - Vložení referenčního záznamu přes online rozhraní



10.2.3 WF3 - Řešení nálezů datové kvality



10.2.4 WF4 - Aktualizace číselníků



10.3 Šablony výstupů

10.3.1 Katalog systémů/datových zdrojů

Název systému	• Vyplňte běžně používaný název system
Identifikátor systému	• Zkratka/kódové označení systému, pod kterým je uveden v číselníku systémů klienta (pokud existuje)
Zodpovědná osoba	• Osoba, která za system zodpovídá za daný systém a na kterou je možné se obracet s dotazy/požadavky
Zápis zhotovil	• Jméno a příjmení toho, kdo zhotovil záznam do katalogu
Verze	• Verze záznamu
Popis systému	• Stručný business popis systému • Přehled agend které systém spravuje • Odkaz na relevantní byznys dokumentaci
Závislé systémy	• Odkaz na systémy (název, identifikátor), které z daného systému přebírají data
Datový model	• Odkaz na datový model zdrojového systému nebo na jeho relevantní podmnožinu • Odkaz na dokumentaci datového modelu
Použité technologie	• Jaké technologie používá systém pro uložení dat • Způsob realizace uživatelského rozhraní • Hlavní technologie použité při realizaci systému
Přehled rozhraní	• Uveďte výčet rozhraní systému - o Typ rozhraní (souborové, databázové, webservice, jiné API) - o Umístění rozhraní – adresa, cesta - o Popis rozhraní - o Popis formátu zpráv
Přehled business entit	• Uveďte výčet entit uložených v systému - o Název entity - o Kód entity - o Odkaz do katalogu entit

10.3.2 Katalog business entit

Název entity	• Vyplňte běžně používané označení entity
Identifikátor entity	• Zkratka/kódové označení entity, která ji jednoznačně určuje
Zápis zhotovil	• Jméno a příjmení toho, kdo zhotovil záznam do katalogu
Verze	• Verze záznamu
Business popis entity	• Stručný byznys popis systému • Přehled agend které systém spravuje • Odkaz na relevantní byznys dokumentaci
Název tabulky/souboru	• Název tabulky/souboru, ve kterém jsou data entity uložena
Master entita	• Uveďte, na kterou entitu master data modelu se business entita mapuje. Pokud hraje v mapování roli generalizace/specializace, uveďte potřebné rozlišení původní entity (př. business entita Pojištěný se mapuje na master entitu Party, v typu Party se uvádí kód BEN (beneficiary))
Přehled atributů	• Uveďte výčet atributů entity ○ Název atributu ○ Název sloupce, ve kterém je atribut uložen ○ Datový typ atributu ○ Business popis atributu ○ Umístění v uživatelském rozhraní ○ Validace/pravidla vztahující se na atribut ○ Mapování na atribut master entity

10.3.3 Přehled zjištění profilingu

Popis zdroje (metadata)						Výsledky profilingu						Návrh mappingu ²³	
Zdroj (název souboru, označení databáze)	Zdrojová tabulka	Zdrojový atribut	Datový typ zdroje ²⁴	Délka atributu ve zdroji ²⁵	Číselník zdroje ²⁶	Počet hodnot (záznamů)	Počet prázdných hodnot (NULL)	Počet neprázdných hodnot (NOT NULL)	Počet unikátních hodnot	Počet distinktních hodnot ²⁷	Zjištění datové kvality ²⁸	Master entita	Master atribut

²³ Návrh mappingu na základě dostupné dokumentace a zjištění z profilingu

²⁴ Použití označení datového typu ve zdroji

²⁵ Délka pole definovaná v databázi, délka nejdelšího řetězce uloženého ve sloupci

²⁶ Pokud používá atribut explicitně číselník, uvést odkaz na zdroj

²⁷ Kolik různých hodnot se vyskytuje v atributu. Používá se k detekci atributů používajících číselník.

²⁸ Uvést jakákoli relevantní zjištění o hodnotách atributu (nalezené vzory, často se vyskytující slova, konformita atributu s regulárním výrazem apod.)

10.4 Souhrnné reporty datové kvality

10.4.1 Souhrnná statistika datové kvality za entitu

- Report obsahuje vybrané atributy entity Party (osoby, s nimiž podnik má/měl nějaký obchodní vztah)

	SYSTEM1		SYSTEM2	
Record types	Count	Percentage	Count2	Percentage3
Persons (records for processing)	1008047	99,93%	311548	99,16%
Legal entities (records excluded from processing)	690	0,07%	2624	0,84%
Total	1008737	100,00%	314172	100,00%
Birth number	Count	Percentage	Count2	Percentage3
Valid	974313	96,65%	307698	98,76%
Invalid	33734	3,35%	3408	1,09%
Empty	0	0,00%	442	0,14%
Total	1008047	100,00%	311548	100,00%
Gender	Count	Percentage	Count2	Percentage3
Valid	1001381	99,34%	309035	99,19%
Corrected	4349	0,43%	2513	0,81%
Invalid	0	0,00%	0	0,00%
Empty	2317	0,23%	0	0,00%
Total	1008047	100,00%	311548	100,00%
Birth date	Count	Percentage	Count2	Percentage3
Valid	993938	98,60%	309168	99,24%
Corrected	8849	0,88%	2209	0,71%
Invalid	2548	0,25%	0	0,00%
Empty	2712	0,27%	171	0,05%
Total	1008047	100,00%	311548	100,00%
Full name	Count	Percentage	Count2	Percentage3
Valid	959538	95,19%	297866	95,61%
Swapped FN and LN	448	0,04%	196	0,06%
Invalid	48005	4,76%	13486	4,33%
Empty	56	0,01%	0	0,00%
Total	1008047	100,00%	311548	100,00%

10.4.2 Labelling

10.4.2.1 Labelling na úrovni záznamu – souhrnná statistika

- Právnícká osoba

count	Q_FLAG_ALL	Popis	Flag_propag
186 703	CLI_PO_CORRECTED	Záznam ze vstupu byl opraven	Y
67 679	CLI_PO_NOT_OK	Záznam ze vstupu se nepodařilo opravit	N
47 353	CLI_PO_CORRECT	Záznam na vstupu je vpořádku	Y
46 183	CLI_PO_CORRECTED_BUT	Záznam na vstupu byl opraven, ale některý atribut se přenáší v původní podobě (DIČ)	Y
11 576	CLI_PO_CORRECT_BUT	Záznam na vstupu je vpořádku, ale některý atribut se přenáší v původní podobě (DIČ)	Y
359 494	SPOLU		

10.4.2.2 Labelling na úrovni atributu – souhrnná statistika

- IČO

count	Q_FLAG_ICO	Popis	Propagable
344 594	ICO_OK	Validní IČO ověřené vůči etalonu (Albertina SK/ORSR/ZRSR)	Y
8 070	ICO_NOT_FOUND	IČO má správný formát, ale nebylo ověřeno vůči etalonu	N
4 440	ICO_RC_IN_ICO	Místo IČO je vyplněno rodné číslo	N
888	ICO_MISPELLED	IČO má správný formát, ale nesplňuje kontrolní součet	N
569	ICO_BAD_FORMAT	IČO nebylo hodnoceno (interní zákazník)	N
514	ICO_NOT_SCORED	IČO má špatný formát	N
378	ICO_EMPTY_OR_DEFAULT	IČO nevyplněno nebo obsahuje defaultní hodnotu	N
41	ICO_EXTERNAL	IČO nevyplněno nebo obsahuje defaultní hodnotu	N
359 494	SPOLU		

10.4.2.3 Labelling záznamů na instanční úrovni

INSTANCE_ID	Q_FLAG_ALL	FLAG_PROPAG	IN_CLIENT_TYPE	STD_CLIENT_TYPE	Q_FLAG_ICO	IN_ICO	STD_ICO	Q_FLAG_CNAME	IN_CNAME	STD_CNAME
98901	CLI_PO_NOT_OK	N	SELF	FOP	ICO_BAD_FORMAT	3395128		NAME_NOT_SCORED	Bobot Karol	
988528	CLI_PO_NOT_OK	N	COMP	PO	ICO_BAD_FORMAT	314099038		NAME_NOT_SCORED	Nafta a.s. Gbely	
983346	CLI_PO_NOT_OK	N	SELF	FOP	ICO_BAD_FORMAT	371816111		NAME_NOT_SCORED	Brunovský Pavol	
312882	CLI_PO_CORRECTED	Y	COMP	PO	ICO_OK	31630006	31630006	NAME_ACCENTS	LESNÉ SPOLOČENSTVO ŠTIAVNÍK sro.	Lesné spoločenstvo Štiavnik, s.r.o.
271162	CLI_PO_CORRECTED	Y	COMP	PO	ICO_OK	31644147	31644147	NAME_ACCENTS	A.T.P. INTERIÉR, s.r.o.	A.T.P. INTERIER, spol. s r.o.
1455566	CLI_PO_CORRECTED_BUT	Y	COMP	PO	ICO_OK	35876115	35876115	NAME_MISSPELLED	Top Trend s.r.o.	T O P T R E N D s. r. o.
1443791	CLI_PO_CORRECTED_BUT	Y	COMP	PO	ICO_OK	36350664	36350664	NAME_MISSPELLED	GLOBAL STUDENT SERVICES SLOVAKIA s.r.o.	GLOBAL STUDENT SERVICES SLOVAKIA s.r.o.
1390353	CLI_PO_CORRECTED_BUT	Y	COMP	PO	ICO_OK	36642819	36642819	NAME_MISSPELLED	ALL Real plus, s.r.o.	ALLReal plus s.r.o.
527202	CLI_PO_CORRECTED	Y	SELF	FOP	ICO_OK	37191969	37191969	NAME_MISSPELLED_WORD_ORDER	Ševcová Simona - S.P. MÓDA	Simoma Ševcová - S.P. MÓDA
515850	CLI_PO_CORRECTED	Y	SELF	FOP	ICO_OK	37532979	37532979	NAME_MISSPELLED_WORD_ORDER	Schwarc Marián M.S.- Stav	Marián Schwarc M.S.- STAV
39029	CLI_PO_CORRECTED	Y	SELF	FOP	ICO_OK	34760300	34760300	NAME_MISSPELLED_WORD_ORDER	Pócsik Peter - O.Z.Consulting	Peter Pócsik O.Z.- CONZULTING

10.5 Report pro manuální opravu záznamů

- Fyzická osoba (atribut birth number byl z reportu skryt)

client_id	active	resident	flg_birth_number	flg_gender	flg_birth_date	flg_name	src_gender	out_gender	src_birth_date	out_birth_date	src_first_name	out_first_name	src_last_name	out_last_name
48974987	Y	1	NOT_OK	EMPTY	OK	OK			1900-01-01	1900-01-01	JADRANKA	JADRANKA	SARIC	SARIC
4320987	Y	1	NOT_OK	EMPTY	OK	OK			1900-01-01	1900-01-01	FERID	FERID	MUSANOVIC	MUSANOVIC
38273942	Y	1	NOT_OK	OK	OK	NOT_OK	F	F	1952-01-20	1952-01-20	FATIMA	FATIMA	OSMANOVIA	
26384045	Y	1	NOT_OK	OK	OK	OK	M	M	1979-01-24	1979-01-24	PREDRAG	PREDRAG	FURDULOVIC	FURDULOVIC
2004021	Y	2	N/A	EMPTY	OK	NOT_OK			1973-12-04	1973-12-04	MARKO	MARKO	KONIE	
2980211	Y	2	N/A	EMPTY	OK	NOT_OK			1933-01-21	1933-01-21	JEAN CLAUDE	JEAN CLAUDE	CARREAU	
23791021	Y	2	N/A	EMPTY	OK	NOT_OK			1964-01-15	1964-01-15	ABDULHAMID		FURJANI O A	
18763897	Y	1	OK	OK	NOT_OK	NOT_OK	M	M	2020-08-04		DUŠAN	DUŠAN	ŠIMŠIA	
19803839	Y	1	OK	OK	NOT_OK	NOT_OK	M	M	1997-09-05		ADIS	ADIS	MALJANOVIA	
19874331	Y	1	OK	OK	NOT_OK	NOT_OK	F	F	1999-09-06		MERSIHA	MERSIHA	RUVIA	