VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

METODY PRO PODPORU ROZHODOVÁNÍ V PROSTŘEDÍ LÉKAŘSKÉ APLIKACE

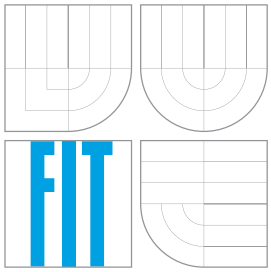
SEMESTRÁLNÍ PROJEKT
SEMESTRAL PROJECT

AUTOR PRÁCE
AUTHOR

Bc. PETR MRÁZEK

BRNO 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

METODY PRO PODPORU ROZHODOVÁNÍ V PROSTŘEDÍ LÉKAŘSKÉ APLIKACE

DECISION SUPPORT METHODS IN A MEDICAL APPLICATION

SEMESTRÁLNÍ PROJEKT
SEMESTRAL PROJECT

AUTOR PRÁCE — Bc. PETR MRÁZEK
AUTHOR

VEDOUCÍ PRÁCE — Ing. VLADIMÍR BARTÍK, Ph.D.
SUPERVISOR

BRNO 2008

Abstrakt

Cílem tohoto semestrálního projektu je navrhnout aplikaci, která by představovala rozšíření stávající lékařské databázové aplikace o možnosti provádění pokročilých analýz nad nashromážděnými zdravotnickými daty.

Klíčová slova

OLAP, dolování dat, datové sklady, analýza

Abstract

The objective of the semestral project is to design an application which would represent the widening of the recent medical database application by any possibilities to practise advanced analysis above the gathered medical data.

Keywords

OLAP, data mining, data warehouse, analyse

Citace

Petr Mrázek: Metody pro podporu rozhodování v prostředí lékařské aplikace, semestrální projekt, Brno, FIT VUT v Brně, 2008

Metody pro podporu rozhodování v prostředí lékařské aplikace

Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením pana Ing. Vladimíra Bartíka, Ph.D.

.....
Petr Mrázek
5. ledna 2008

Obsah

Kapitola 1

Úvod

Většina lékařů využívá k provozování své praxe některý z dostupných softwarových produktů určených k vedení jejich odborné agendy. Tyto produkty ovšem ve většině případů nabízejí jen velmi omezené možnosti analýzy nad zdravotnickými daty, která jsou těmito produkty spravována. Omezují se jen na poskytnutí základních statistických reportů jako je množství výkonů provedených za dané období, množství získaných bodů, nejčastější výskyt diagnóz, celkový počet pacientů v kartotéce a podobně.

Cílem tohoto semestrálního projektu je tedy navrhnout aplikaci, která by představovala rozšíření pro tyto produkty poskytující lékařům v soukromých ambulantních ordinacích možnost pokročilé analýzy nad zdravotnickými daty, která během provozování své praxe nashromáždili. V následujících kapitolách se budeme nejprve zabývat problematikou datových skladů, prostředků OLAP a dolování dat a dále pak samotnými nároky na výslednou aplikaci a jejím návrhem. Na závěr budou shrnuty dosažené výsledky a nastíněno další pokračování v projektu v rámci diplomové práce.

Kapitola 2

Datové sklady

Datový sklad [1] je možné chápat jako strukturované úložiště dat, které slouží k ukládání informačního bohatství organizací za co možná největší časové období. Níže jsou uvedeny požadavky na datový sklad podle Billa Inmona:

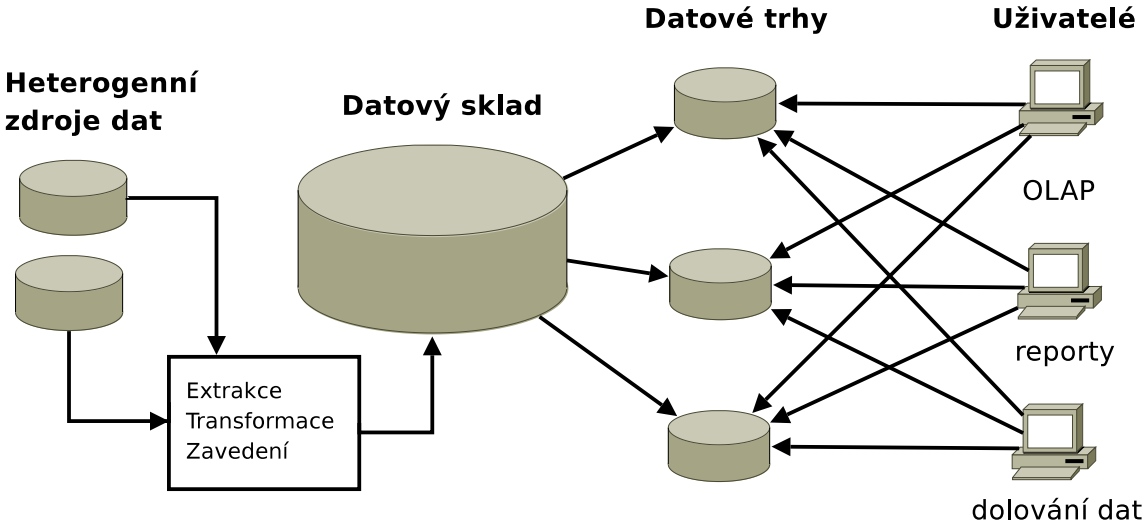
1. **Subjektová orientace** - do datového skladu jsou zanášena data spíše podle předmětu zájmu, než podle konkrétní aplikace, ve které byla tato data vytvořena.
2. **Integrovanost** - datový sklad musí být integrovaný a jednotný. Data týkající se konkrétního předmětu jsou do skladu uložena pouze jednou. Data přicházejí do datového skladu z nekonzistentního a neintegrovaného operačního prostředí, proto musí být tato data v etapě přípravy a zavedení upravena, vyčištěna a sjednocena.
3. **Časová variabilita** - data jsou do datového skladu zanášena jako série snímků, kde každý představuje jistý časový úsek. V operačním databázovém prostředí jsou data ukládána za kratší časové období dní, maximálně měsíců, zatímco v datovém skladu jsou data většinou i za několik let chodu organizace.
4. **Neměnnost** - Na rozdíl od operačních transakčních databází, kde jsou data vkládána, modifikována i mazána, v datovém skladu se většinou nemění ani neodstraňují, jen se v pravidelných intervalech přidávají nově získaná data.

2.1 Datový trh

Z hlediska investic i času je budování datového skladu poměrně náročný projekt a proto se v některých případech přistupuje k budování datového skladu po menších částech - datových trzích. Datový trh lze tedy chápat jako podmnožinu datového skladu, která je určena pro menší organizační části organizace (např. prodej, marketing atd.). Datové trhy mohou vzniknout i opačným způsobem, kdy je nejprve vybudován centrální datový sklad a z něj je pak vytvořeno několik datových trhů. Výsledkem jsou pak menší nároky na provoz a údržbu.

2.2 Způsoby budování datového skladu

Existují dvě nejčastěji používané metody budování datového skladu:



Obrázek 2.1: Architektura datového skladu s datovými trhy

- Metoda velkého třesku
- Přírůstková metoda

2.2.1 Metoda velkého třesku

Tato metoda staví na předpokladu, že je možné realizovat implementaci datového skladu v rámci jediného projektu. Metoda je složena ze tří následujících etap:

1. **Analýza požadavků organizace**
2. **Vytvoření datového skladu**
3. **Vytvoření přístupu k datovému skladu buď přímo nebo pomocí datových trhů.**

Jedinou výhodou této metody je fakt, že je možné celý projektový plán vypracovat ještě před jeho samotnou realizací. Mezi její hlavní nevýhody naopak patří jednak velké riziko změny požadavků a poměrně dlouhá doba, než se dostaví první výsledky investic.

2.2.2 Přírůstková metoda

Tato metoda předpokládá budování datového skladu po jednotlivých etapách - postupně přibývají nová rozšíření, která zapadají do celkové architektury datového skladu. Typicky se začíná s budováním několika málo oblastí, které se implementují ve formě rozšiřitelného datového trhu a tento trh je následně poskytnut koncovým uživatelům. První podsystémy tedy začnou pracovat a přinášet výhody krátkém čase po spuštění projektu.

Přírůstková metoda se skládá z následujících kroků(viz obrázek 2.2):

1. **Strategie** - Definování jak krátkodobých, tak dlouhodobých cílů strategie zahrnujících účel datového skladu, jeho vybudování, dokumentaci, školení uživatelů atd. V této fázi je definován i základ architektury datového skladu organizace.

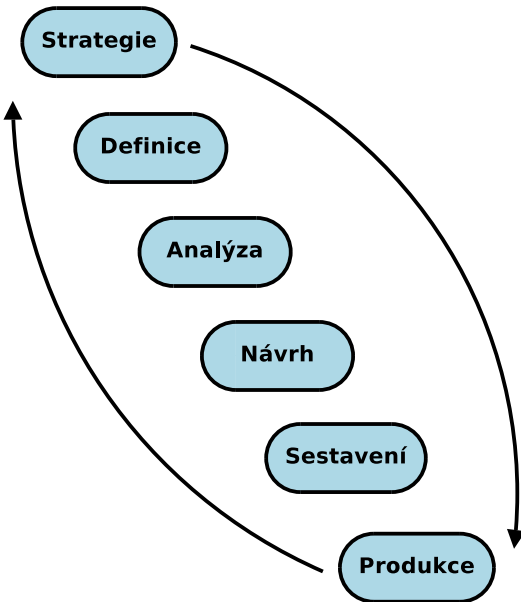
2. **Definice** - Definování cíle a rozsahu přírůstkového vývoje. Fáze zahrnuje jak návrh architektury datového skladu, tak architekturu technických prostředků. Vytvoří se počáteční přírůstek, zdokumentují se zdroje dat a vymezí se rozsah kvality těchto údajů.

3. **Analýza** - Získávání dat a požadavků na přístup k datům, informace o uživatelích skladu, vytvoření relačních a multidimenzionálních modelů a metadat.

4. **Návrh** - Požadavky, které byly získané během analýzy jsou transformovány do detailních podmínek návrhu.

5. **Sestavení** - Během této fáze se vytvoří a otestují navržené databázové struktury a moduly pro práci s datovým skladem.

6. **Produkce** - Datový sklad je nasazen do provozu, začíná se řídit růst a správa datového skladu.



Obrázek 2.2: Iterační cyklus přírůstkové metody

K hlavním výhodám této metody patří:

- **rychlejší návratnost vložených investic**
- **možnost implementace škálovatelné architektury**
- **tvorba datového skladu po částech udržuje kontinuitu projektu s požadavky uživatelů**

2.3 Proces ETL

Poté, co je datový sklad, či datový trh naimplementován, je nutné ho naplnit daty. Data jsou většinou umístěna v různých nehomogenních operačních prostředích, databázových

systémech, hardwarových platformách, archivních systémech, podnikových systémech a v různých dokumentech. Kromě interních dat v rámci organizace je také možné získat data analýzou konkurenčního prostředí, od obchodních partnerů, z internetu, nebo jejich zakoupením, což samozřejmě jen zvyšuje jejich různorodost. Takováto data je důležité před zavedením do datového skladu nejdříve vyextrahovat, vyčistit a pak ve vhodné formě zavést do datového skladu.

Proces ETL se skládá z následujících tří etap:

- **Extrakce** - Cílem této etapy je získání dat z výše uvedených zdrojů. Pro extrakci jsou k dispozici různé nástroje a technologie.
- **Tranformace** - Jedná se o soubor úkonů, které vedou ke zkvalitnění dat získaných v předchozí etapě. Během této etapy tedy dochází k sjednocování formátování dat, sjednocení přiřazení datových typů, peněžních jmen atd. Během transformace se většinou vyskytují následující typy problémů, které je nutné vyřešit:
  - chybějící hodnoty
  - duplicitní záznamy
  - nejednoznačnost dat
  - nejednoznačnost názvů pojmů
  - nejednoznačnost peněžních měn
  - chybějící datum
  - nejednotnost formátů čísel a textových řetězců
- **Zavedení(Loading)** - V této poslední fázi procesu ETL dochází k samotnému fyzickému přesunutí extrahovaných a transformovaných dat do datového skladu.

2.4 Porovnání rozdílů mezi datovým skladem a operační databází

Následující tabulka shrnuje rozdíly mezi typickou operační databází a datovým skladem.

Vlastnost	**Operační databáze**	**Datový sklad**
čas odezvy	zlomky vteřin až vteřiny	vteřiny až hodiny
operace	DML	primárně jen pro čtení
původ dat	30-60 dní	série snímků za časový okamžik
organizace dat	podle aplikace	podle předmětu, času, atd.
velikost	malá až velká	velká až velmi velká
zdroje dat	operační, interní	operační, interní, externí
činnosti	procesy	analýza

Kapitola 3

Úvod do problematiky OLAP

Termín OLAP(Online Analytical Processing)[1] zavedl Dr. E. F. Cood k popisu technologie, která by pomohla překlenout mezery mezi využitím osobních počítačů a řízením podnikových dat.

Níže je uvedeno dvanáct původních pravidel, které by měl splňovat OLAP od Dr. E. F. Codda:

1. **Multidimenzionální konceptuální pohled** - OLAP by měl poskytovat uživateli multidimenzionální model odpovídající jeho podnikatelským potřebám tak, aby takový model mohl využívat pro analýzu shromážděných dat.

2. **Transparentnost** - Technologie systému OLAP, podřízená databáze a architektura výpočtu by měly být pro uživatele transparentní, proto aby mohl naplno využívat svoji produktivitu a odbornost použití front-end nástrojů a prostředí.Důležitá je samozřejmě i heterogennost vstupních dat, kterou zajistíme v procesu ETL.

3. **Dostupnost** - Systém OLAP by měl přistupovat jen k těm datům, která jsou potřebná pro analýzu. Systém by měl být také schopen přistupovat ke všem datům potřebným pro analýzu, nezávisle na tom, ze kterého heterogenního podnikového zdroje tato data pocházejí, jak často jsou obnovována a podobně.

4. **Konzistentní vykazování** - I když počet záznamů a tedy i velikost databází postupem času roste, uživatel by neměl poznat žádné podstatné snížení výkonu.

5. **Architektura klient-server** - Systém OLAP musí odpovídat principům architektury klient-server s přihlédnutím na maximální cenu a výkon, flexibilitu a interoperabilitu.

6. **Generická dimenzionalita** - Každá dimenze dat musí být ekvivalentní ve struktuře i operačních schopnostech.

7. **Dynamické ošetření řídkých matic** - Systém OLAP by měl být schopen adaptovat svoje fyzické schéma na konkrétní analytický model, který optimalizuje ošetření řídkých matic, přičemž dosáhne a udrží požadovanou úroveň výkonu.

8. **Podpora pro více uživatelů** - Systém OLAP musí být schopný podporovat pracovní skupinu uživatelů pracujících současně na konkrétním modelu.

9. **Neomezené křížové dimenzionální operace** - Systém OLAP musí dokázat rozeznat dimenzionální hierarchie a automaticky vykonat asociované kumulované kalkulace v rámci dimenzí a i mezi nimi.

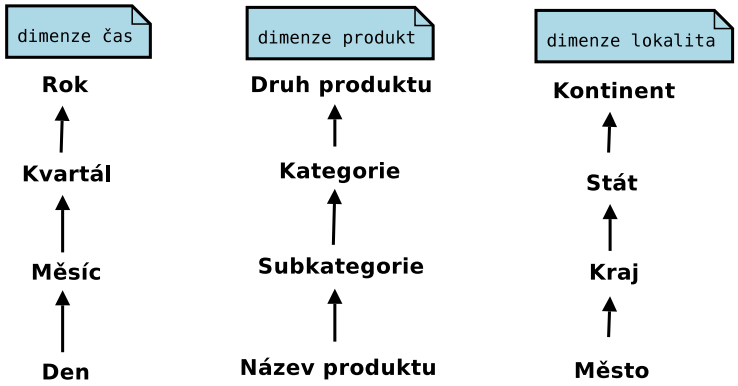
10. **Intuitivní manipulace s daty** - Pravidlo definuje konsolidované přeorientování cest na detailní úroveň a zpět *drilldown*, *drillup*. Uživatelské rozhraní by mělo umožňovat všechny manipulace způsobem "ukázat a klepnout, případně zachytit a přemístit" v buňkách kostky.

11. **Flexibilní vykazování** - Musí existovat schopnost uspořádat řádky, sloupce a buňky způsobem, jenž umožní analýzu intuitivní vizuální prezentací analytických sestav.

12. **Neomezené dimenze a úrovně agregace** - V závislosti na požadavcích podnikání muže mít analytický model více dimenzí, přičemž každá z nich muže mít vícenásobné hierarchie. Systém OLAP by neměl zavádět žádné umělé omezení počtu dimenzí nebo úrovní agregace.

3.1 Multidimenzionální datový model

Multidimenzionální datový model je možné zobrazit ve formě vícerozměrné kostky. Takovou kostku lze pak chápat jako jistý ekvivalent tabulky v relační databázi. Každá kostka je složena z několika dimenzí. Dimenze obsahují organizačně nebo logicky hierarchicky uspořádaná data(obrázek 3.1). V podstatě se jedná o textové popisy obchodování.

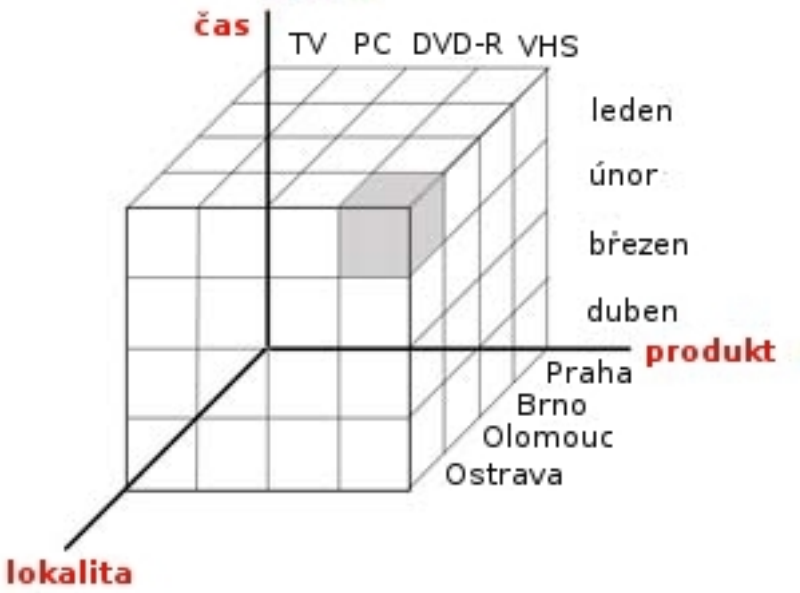


Obrázek 3.1: Příklad hierarchických úrovní pro dimenze čas, produkt a lokalita

Data(označovaná jako **fakta**) se v kostce nacházejí na průsečících jednotlivých dimenzí. Fakta jsou numerické měrné jednotky obchodování představující množství, na jehož základě se analyzují vztahy mezi dimenzemi. Pomocí vhodně zvolených kombinací dimenzí a úrovní jejich abstrakce lze získat informace např. o celkovém objemu prodeje za jisté časové období, zjistit, jaké zboží šlo v jednotlivých obdobích nejvíce na odbyt, prodej zboží v různých lokalitách atd. Pro výpočet OLAP kostek je nutné provádět velké množství výpočtů a to téměř v reálném čase.

Uveďme si jednoduchý příklad třídimenzionální datové kostky(obrázek 3.2). Tato kostka obsahuje následující dimenze:

- čas
- lokalita
- produkt



Obrázek 3.2: Příklad třídimenzionální datové kostky.

Předpokládejme, že zkoumanou měrnou jednotkou(faktem) je objem prodeje. Například následujícími kombinacemi dimenzí lze pak získat tyto informace:

- **čas - produkt** - informace o objemu prodeje jednotlivých druhů produktů za určité období
- **čas** - informace o celkovém objemu prodeje za určité období
- **čas - lokalita - produkt** - informace o objemu prodeje jednotlivých druhů produktů za určité období v jednotlivých lokalitách

3.1.1 Srovnání relačního a multidimenzionálního modelu

Níže jsou uvedeny základní výhody a nevýhody obou modelů.

- **Relační model**
  Mezi hlavní výhody relačního modelu patří:
  - značné množství odborníků, kteří s tímto modelem roky pracují
  - existence velkého množství vývojových prostředků pro vývoj aplikací a generování reportů
  - uplatnění v transakčních databázích a datových skladech

  Nevýhody:

  - naprostý nedostatek komplexních analytických nástrojů
  - objem dat, k nimž lze v rozumném čase přistoupit je omezen

- **Multidimenzionální model**
  Výhody:

  - schopnost rychlého komplexního přístupu k velkému objemu dat
  - přístup k relačním a multidimenzionálním datovým strukturám
  - možnost provádění komplexních analýz
  - značné schopnosti pro modelování a prognózy

  Nevýhody:

  - při změně dimenzí bez přizpůsobení časové dimenze nastávají komplikace
  - klade vyšší požadavky na velikost úložiště

3.2 Operace nad OLAP kostkou

Jak již bylo zmíněno výše, data v multidimenzionálním modelu jsou organizována do určitého počtu dimenzí. Každá taková dimenze má několik úrovní abstrakce, díky čemuž lze na data nahlížet z různých pohledů. Pro definování požadovaných pohledů jsou k dispozici následující základní OLAP operace:

- **Roll-Up** - Posun v hierarchii dimenze o jednu úroveň výše. Pokud bychom byly například v dimenzi čas na úrovni den a provedli operaci Roll-up, pak bychom se posunuli na úroveň měsíc(viz příklad hierarchií na obrázku 3.1). Nyní by tedy byla data v kostce seskupena nikoliv podle dnů, ale podle jednotlivých měsíců.

- **Drill-Down** - Jedná se o opak předchozí operace Rool-up. V hierarchii dimenze se posuneme o jednu úroveň hlouběji a získáme tak detailnější výsledky. V případě dimenze lokalita bychom se posunuli např. z úrovně stát na detailnější úroveň kraj.

- **Slice & Dice** - Operace Slice slouží k provedení selekce nad jednou dimenzí. Výsledkem operace je podkostka obsahující pouze data, která splňující danou podmínku. Příkladem takové podmínky může být např: (lokalita = Praha). Ve výsledné podkostce by tedy byla jen data vztahující se k dané lokalitě. Naproti tomu operace Dice umožňuje selekci nad více než jednou dimenzí. Podmínka pro selekci by mohla vypadat následovně: (lokalita = Brno) and (cas = duben) and (produkt = TV). Výsledkem bude opět podkostka splňující danou podmínku.

- **Pivot** - Rotace os v 2D nebo 3D kostce, nebo zobrazení 3D kostky jako kolekce 2D kostek za účelem jiné prezentace dat v kostce.

3.3 Typy OLAP

OLAP systémy mohou být podle typu úložiště multidimenzionálních dat rozděleny do následujících kategorií:

- **Multidimenzionální OLAP(MOLAP)** - Data se získávají buď z datového skladu nebo z operačních zdrojů. Analytická data jsou pak uložena ve speciálních datových strukturách a sumářích. Během tohoto procesu se vypočítá tolik předběžných výsledků, kolik je z časového i kapacitního hlediska možné - data se tedy v úložišti ukládají jako dopředu vypočítaná pole. Data jsou organizována takovým způsobem, který poskytuje maximální výkon vzhledem k požadovaným dotazům uživatelů, nevýhodou je však redundance takových dat, protože jsou uloženy jak v relační databázi, tak v multidimenzionální databázi. Další značnou nevýhodou je fakt, že v případě vyššího počtu dimenzí může velikost takové databáze extrémně narůstat.
- **Relační databázový OLAP(ROLAP)** - Data pro analýzu jsou v tomto případě získávána z relačního datového skladu. Po zpracování jsou data z relačních databází předkládána uživatelům jako multidimenzionální pohled. Data a metadata jsou ukládány jako záznamy v relační databázi. Metadata jsou pak systémem ROLAP následně využívána ke generování SQL dotazů, kterými jsou získána analytická data požadovaná uživateli. Výhodou úložiště typu ROLAP je, že nedochází k redundanci, protože data zůstávají uložena v relační databázi, nevýhodou je oproti systémům MOLAP nižší dotazovací výkon.
- **Hybridní OLAP(HOLAP)** - Jedná se o kombinaci úložišť MOLAP a ROLAP. Využívají se výhody jednotlivých typů úložišť a eliminují se nevýhody. Samotná data zůstávají uložena v relačních databázích a do multidimenzionálních struktur se ukládají pouze vypočítané agregace.

3.4 Schémata tabulek dimenzí

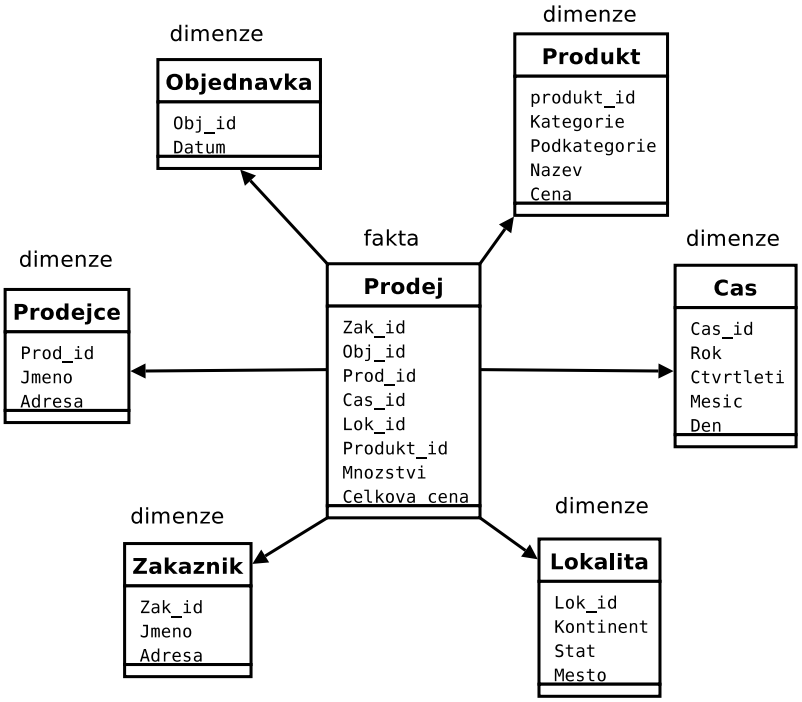
V případě systémů typu ROLAP, které využívají relační datový sklad je nutné namapovat multidimenzionální model a OLAP operace na relační tabulky a SQL dotazy. Mezi nejpoužívanější schémata datového skladu patří schéma hvězdy a schéma sněhové vločky.

3.4.1 Schéma hvězdy

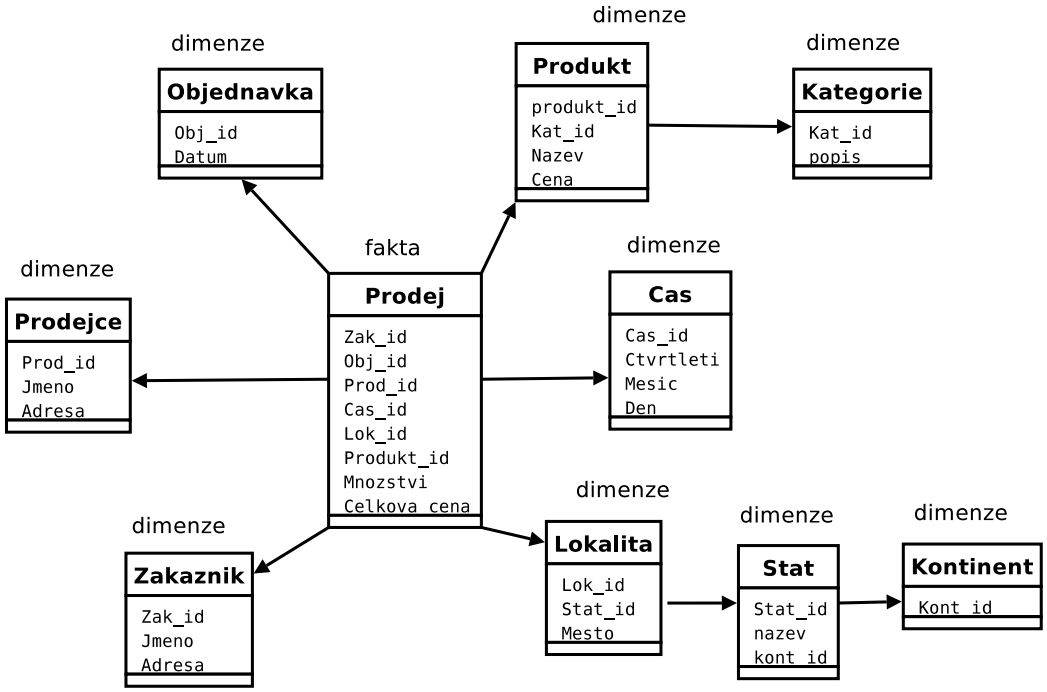
Schéma hvězdy(obrázek 3.3) se skládá z jedné centrální tabulky faktů, která obsahuje cizí klíče, které se vztahují k primárním klíčům v jednotlivých tabulkách dimenzí, z nichž každá obsahuje informace o jedné z dimenzí. Schéma nemá normalizované dimenze ani relační propojení mezi tabulkami dimenzí. Jeho výhodou je vysoký dotazovací výkon, nicméně jeho vytvoření je díky nenormalizovaným dimenzím poměrně pomalé.

3.4.2 Schéma sněhové vločky

Schéma sněhové vločky(obrázek 3.4) je podobné jako schéma hvězdy s tím rozdílem, že některé tabulky dimenzí jsou normalizované.To znamená, že některé dimenze jsou složeny z více tabulek. Výhodou tohoto modelu je díky normalizovaným dimenzím snížení redundance, nicméně při dotazovaní je nutné provést více tabulkových spojení, což negativně ovlivňuje dotazovací výkon.



Obrázek 3.3: Příklad schématu hvězdy



Obrázek 3.4: Příklad schématu sněhové vločky

Kapitola 4

Úvod do problematiky dolování dat

Proces dolování dat(Data mining) [1][2][3] je zaměřen na analýzu velkého množství dat z různých perspektiv a jejich přeměnu na užitečné informace, které jsou následně využity k podpoře rozhodování. Z matematického hlediska se jedná o hledání korelací, tedy vzdájemných vztahů nebo vzorů v datech. V následujících sekcích jsou uvedeny nejčastější typy analýz v rámci procesu dolování dat.

4.1 Asociační pravidla a frekventované množiny

Analýza se z hlediska asociačních pravidel zaměřuje na hledání různých souvislostí nad množinami datových položek. Z matematického hlediska se jedná o zkoumání korelace, ať už pozitivní, nebo negativní. Pozitivní korelace znamená, že vysoká úroveň jedné proměnné je doprovázena vysokou úrovní korelační proměnné. Negativní korelace naopak udává, že vysoká úroveň jedné proměnné bude provázena nízkou úrovní korelační proměnné.

Typickým příkladem využití asociačních pravidel je analýza nákupního košíku, kdy se snažíme zjistit, jaké zboží si zákazníci nejběžněji kupují dohromady. Na základě objevených vztahů mezi jednotlivými položkami v nákupním košíku je pak možné rozmístit zboží v obchodě efektivněji, popřípadě vhodně sestavovat nabídkové akce atd. Níže je uveden možný tvar asociačního pravidla:

$$tiskarna \Rightarrow papir\ do\ tiskarny\ (s = 2\%, c = 60\%)$$

kde hodnoty s a c představují metriky asociačního pravidla - podporu a spolehlivost. Podpora udává v kolika transakcích z celkového počtu všech transakcí se vyskytly tyto dvě položky společně a spolehlivost udává v kolika transakcích z celkového počtu transakcí obsahujících první položku se zároveň vyskytla i druhá položka.

4.1.1 Typy asociačních pravidel

Asociační pravidla je možné rozdělit podle níže uvedených kritérií:

- **Podle typu hodnot v pravidlech** - Pokud pravidla popisují asociace mezi kvantitativními atributy nebo položkami, pak se jedná o kvantitativní asociační pravidla. V případě, že nás zajímá pouze přítomnost nebo nepřítomnost jisté položky, pak se jedná

o booleovská asociační pravidla. Níže je uveden příklad kvantitativního asociačního pravidla:

$$vek = 30..50 \wedge prijem = 30000 - 48000 \Rightarrow koupi\,(tiskarna)$$

- **Podle počtu dimenzí obsažených v pravidlech** - Asociační pravidla se označují jednodimenzionální, jestliže obsahují pouze jednu dimenzi. V opačném případě se jedná o pravidla vícedimenzionální.
- **Podle úrovní abstrakce v pravidlech** - Existují metody umožňující získání asociačních pravidel nad různými úrovni abstrakce. Tato pravidla se označují jako víceúrovňová.

$$\begin{aligned} vek &= 30..50 \Rightarrow koupi\,(tiskarna) \\ vek &= 30..50 \Rightarrow koupi\,(multifunkcni\ tiskarna) \end{aligned}$$

4.2 Klasifikace a predikce

Klasifikací rozumíme proces umožňující přiřazení vstupních objektů do konečného počtu jistých tříd na základě jejich vlastností. Predikce je naproti tomu proces přiřazující datům hodnoty obecně spojitého charakteru - tyto hodnoty jsou předpovídany na základě vlastností daného objektu.

Proces klasifikace je rozdělen na dvě fáze:

- **Fáze učení** - Z databáze je vybrána trénovací množina dat. U těchto dat je nutné znát přesně třídu, do které jsou přiřazeny. Klasifikátor pak na základě těchto dat generuje klasifikační pravidla, prostřednictvím kterých je pak možné jednotlivé objekty přiřazovat s jistou přesností do konkrétních tříd.
- **Fáze testování daného klasifikátoru** - Jakmile jsou vytvořena klasifikační pravidla, je nutné tyto pravidla otestovat. Je vybrána tzv. testovací množina dat, u které je nutné, stejně jako v případě trénovací množiny, znát klasifikační třídy, do kterých tato data náleží. Data by neměla pocházet z trénovací množiny kvůli možnému zkreslení testovacích výsledků. Data jsou na základě klasifikačních pravidel rozdělena do jednotlivých tříd. Následně se určí procentuální úspěšnost klasifikátoru a na základě této úspěšnosti je pak rozhodnuto, zda je tento klasifikátor možno použít ke klasifikaci operačních dat, či nikoliv.

4.2.1 Klasifikace pomocí rozhodovacích stromů

Klasifikace je založena na grafu stromové struktury. Každý vnitřní uzel grafu obsahuje podmínku pro hodnotu jistého atributu a koncové uzly představují třídy, do kterých jsou objekty klasifikovány. Takovýto graf je pak možné snadno přepsat do tvaru klasifikačních pravidel.

4.2.2 Bayesovská klasifikace

Bayesovská klasifikace je založena na statistice. Pro každý nový vzorek dat je na základě statistických metod vypočtena pravděpodobnost náležení prvku do jednotlivých tříd. Vzorek dat je následně přiřazen do třídy s největší pravděpodobností náležení.

4.2.3 Klasifikace pomocí neuronových sítí

Dalším klasifikačním modelem jsou neuronové sítě. Tyto sítě jistým způsobem simulují strukturu lidského mozku. Základním logickým prvkem neuronových sítí je neuron. Do neuronu vstupují vstupní data z vnějšího vstupu sítě nebo z výstupu jiného neuronu. Tato data jsou neuronem částečně zpracována pomocí sumace a aktivačních funkcí a následně se přesouvají na vstup dalšího neuronu, nebo na výstup neuronové sítě.

Zpracování pomocí neuronových sítí nevychází z žádného statistického hlediska, nýbrž pracuje na principu rozpoznávání vzorů a minimalizace chyb. Neuronová síť je tvořena uzly uspořádanými do vrstev. Data se nejprve rozdělí do trénovací a testovací množiny. V každé iteraci jsou vstupy zpracovány systémem a následně porovnány se skutečnou hodnotou. Změřená chyba je předána systému k upravení původní váhy. Proces je ukončen v okamžiku dosažení určené minimální chyby.

4.3 Shluková analýza

Shlukování je rozdělování objektů na základě jejich podobností do tříd. Jednotlivé třídy pak obsahují navzájem hodně podobné objekty, které zároveň nejsou příliš podobné objektům z jiných tříd. Podobnost objektů je hodnocena podle hodnot jednotlivých atributů objektů a často se k tomuto účelu využívají vzdálenostní funkce. Proces shlukování nevyžaduje na rozdíl od klasifikace žádnou trénovací množinu ani žádné předdefinované třídy.

4.3.1 Typy shlukovacích metod

Shlukovací metody lze rozdělit do následujících kategorií:

- **Metody založené na rozdělování** - Metody rozdělují n objektů do k tříd, kde pro k, n platí $k \leq n$. Každá třída obsahuje alespoň jeden objekt a každý objekt patří pouze do jedné třídy. Je nutné zadat počet tříd, do kterých se mají objekty rozdělovat. Nejprve se vybere náhodně k objektů reprezentujících jednotlivé třídy a ostatní objekty se zařadí do jednotlivých tříd na základě podobnosti. V dalším kroku se hledají interaktivně objekty nejlépe reprezentující jednotlivé třídy a objekty jsou přesouvány mezi jednotlivými třídami tak, aby jejich podobnost byla v rámci třídy maximální a podobnost s objekty jiných tříd minimální.

- **Hierarchické metody** - Vytváří se hierarchický rozklad množiny objektů - vytváří se strom shluků. Hierarchické metody se rozdělují podle průběhu rozkladu na *Shlukující hierarchické metody*, kdy se nejprve zařadí každý objekt do speciální třídy a následně se slučují nejpodobnější třídy, dokud nejsou všechny třídy spojeny do jedné třídy, nebo se nedosáhne požadované úrovně shlukování a na *Rozdělující hierarchické metody*, kdy se všechny objekty nejdříve umístí do jediné třídy a pak se jednotlivé třídy dělí na menší třídy do té doby, než je každý objekt umístěn ve zvláštní třídě, nebo se dosáhlo požadované úrovně rozdělování.

- **Metody založené na hustotě** - Za shluky se považují oblasti s velkou hustotou objektů v prostoru dat, které jsou od sebe oddělené oblastmi s malou hustotou objektů. Objekty vyskytující se v oblastech s malou hustotou se považují za šum. Výhodou těchto metod je skutečnost, že umožňují nacházet shluky různých tvarů a jsou schopné se vypořádat s odlehlými hodnotami a šumem v datech.

- **Metody založené na mřížce** - Využívá se víceúrovňová mřížková datová struktura. Prostor objektů je rozdělen na konečný počet buněk tvořících mřížku. Nad touto mřížkovou strukturou pak probíhají veškeré operace slukování. Předností tohoto typu metod je doba zpracování datové množiny, která je závislá pouze na počtu buněk struktury a nikoliv na celkovém počtu objektů.

- **Metody založené na modelech** - Snaha o optimalizaci shody mezi datovou množinou a nějakým daným matematickým modelem. Cílem je tedy nalezení takových shluků, které by co nejvíce odpovídaly danému modelu. Data jsou ve většině případů generována na základě složené pravděpodobnostní distribuční funkce. Patří zde např. metody neuronových sítí a konceptuální shlukování. Konceptuální shlukování nehledá na rozdíl od běžného shlukování jen skupiny objektů,ale snaží se pro každou skupinu objektů nalézt charakteristický popis.

- **Metody pro shlukování vysoce dimenzionálních dat** - Řada shlukovacích metod má problémy při shlukování vysoce dimenzionálních dat, jelikož byly navrženy jen pro použití s malým počtem dimenzí. U rostoucího počtu dimenzí je totiž problémem skutečnost, že jen některé z těchto dimenzí jsou relevantní pro jednotlivé shluky. Data ostatních dimenzí pak generují poměrně velké množství šumu znemožňujícího nalezení shluků.

  Pro shlukování takovýchto dat je možné využít metodu transformace rysů transformující data do prostoru s menším počtem dimenzí při zachování relevantních vzdáleností mezi jednotlivými objekty, nebo metodu výběru atributů, která je založena na nalezení nejvíce relevantních atributů pro danou úlohu a odstranění nadbytečných atributů.

Kapitola 5

Návrh rozšíření existující lékařské aplikace

5.1 Úvod

Cílem této kapitoly je shrnout nároky na aplikaci, která by představovala nejen rozšíření stávající lékařské ambulantní databázové aplikace, ale bylo by ji možné využít i spolu s ostatními podobně zaměřenými zdravotnickými produkty. Takováto rozšířená použitelnost s sebou samozřejmě přináší jeden zásadní problém - jakým způsobem extrahovat z libovolné aplikace data, nad kterými bude následně prováděna pokročilá analýza.

Přestože si mohou být jednotlivé aplikace velmi podobné a z hlediska funkcionality a ovládání si opravdu podobné jsou, tak každá takováto aplikace ukládá ambulantní data svým vlastním způsobem, ať už do svých vlastních datových struktur, nebo za použití různých typů databázových systémů.

Pokud by mělo navrhované rozšíření pracovat přímo s interními daty všech těchto aplikací, bylo by nutné znát dopodrobna způsob uložení dat pro každou z nich, což by bylo bez intenzivní spolupráce se softwarovými společnostmi, které tyto aplikace vytvořily nereálné. Řešením tohoto problému je využití jedné vlastnosti, kterou mají všechny použitelné ambulantní aplikace společné. Touto vlastností je generování datových dávek pro pojišťovny. Každá datová dávka má dopodrobna definované datové rozhraní, které musí všechny tyto aplikace dodržovat. Navrhovaná aplikace tedy bude pracovat nad importovanými záznamy z vygenerovaných datových dávek, tzn. nad stejnými daty, jaká jsou ambulantními lékaři předávaná pojišťovnám ke zpracování.

5.2 Pojem dávka

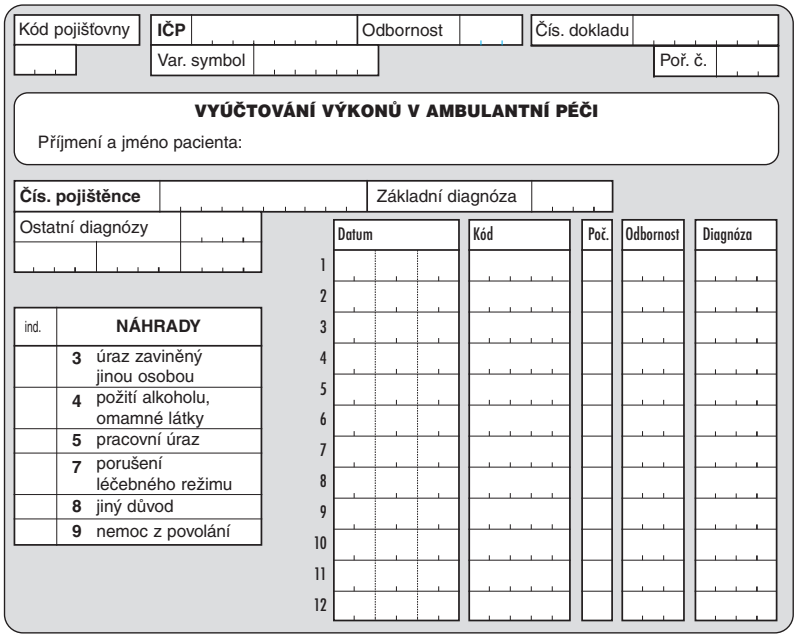
Dávka[4][5] je vyvrcholením lékařovi práce, týkající se pojišťovny. Jsou do ní vtěsnány všechny účty, které lékař za poslední období nashromáždil a které budou odeslány pojišťovně na datovém nosiči. Dávky jsou uloženy v ASCII souboru, který má předepsané standardní jméno KDAVKA.XXX, kde XXX představuje kód pojišťovny. Platí zásada, že na každém datovém nosiči je jen jeden soubor.

5.2.1 Definice datového rozhraní dávky

Dávka začíná úvodní větou dávky, která obsahuje základní informace o zdravotnickém zařízení a dávce. Následující věty reprezentují jednotlivé doklady. Doklady se skládají z typů vět pevné délky řazených za sebou. Jednotlivé věty jsou od sebe odděleny znaky "posun vozíku a nová řádka" (CRLF). Atributy věty nejsou odděleny delimitory.

5.2.2 Dávka ambulantní smíšená

Jelikož je navrhovaná aplikace určena pouze pro využití v soukromých ambulantních ordinacích, bude pracovat jen se záznamy získanými z dávek typu ambulantní smíšené. Tento typ je určen pro vykazování výkonů v ambulantní péči. V podstatě se jedná o elektronickou verzi klasického papírového formuláře uvedeného na obrázku 5.1.



Obrázek 5.1: Papírový formulář pro vyúčtování výkonů

Typy vět Doklady se v této dávce skládají z následujících čtyř typů vět:

1. **záhlaví** – vyskytuje se pro každý doklad jednou a obsahuje převážně informace o pacientovi, jemuž je doklad přiřazen, počtu vykazovaných výkonů, pořadí dokladu v dávce a další. Níže jsou uvedeny její atributy.

   Popis typů v tabulkách atributů: C -znakový atribut, N - numerický atribut, D - datum ve tvaru "DDMMRRRR", $ - peněžní atribut, formát x.y (x míst včetně desetinné tečky, z toho y desetinných míst).

Název	Typ	Délka	Začátek	Popis
TYP	C	1	0	typ věty “A“ - záhlaví
HCID	N	7	1	číslo dokladu
HSTR	N	1	8	Pořadové číslo listu dokladu
HPOC	N	1	9	Celkový počet listů dokladu
HPOR	N	3	10	Pořadové číslo listu v dávce. Povolený rozsah od 1-999.
HCPO	C	3	13	číslo pojišťovny
HTPP	C	1	16	Typ připojištění
HICO	C	8	17	Identifikační číslo pracoviště - IČP
HVAR	C	6	25	Variabilní symbol
HODB	C	3	31	Smluvní odbornost pracoviště
HROD	C	10	34	Číslo pojištěnce
HZDG	C	5	44	Číslo základní diagnózy, pro kterou byl pacient v tomto období léčen
HKON	C	1	49	nevyplňuje se
HICZ	C	8	50	nevyplňuje se
HCDZ	N	7	58	nevyplňuje se
HREZ	C	10	65	nevyplňuje se
HCCEL	$	10.2	75	Cena celkem - nepovinné
HCBOD	N	7	85	Body celkem - nepovinné
DTYP	C	1	95	Doplněk typu věty záhlaví - nevyplňuje se
Celková délka věty: 93 znaků				

2. **výkony** - prostřednictvím této věty se zadávají jednotlivé výkony, které byly na pacientovi během daného zúčtovacího období provedeny. Maximálně 99 výkonů.

Název	Typ	Délka	Začátek	Popis
TYP	C	1	0	Typ věty ”V” - výkony
VDAT	D	8	1	Datum provedení výkonu. Uvedení je povinné u prvního výkonu v daném dnu
VKOD	C	5	9	Číslo výkonu
VPOC	N	1	14	Počet provedení výkonu
VODB	C	3	15	Odbornost - uvádí se povinně v případě, kdy byl výkon proveden na pracovišti jiné smluvní odbornosti, než je uvedena v záhlaví dokladu
VDIA	C	5	18	Diagnóza - je-li uvedena základní diagnóza, uvádí se jen u výkonů, které se k základní diagnóze nevztahují. Nevyplněná diagnóza se považuje za diagnózu základní.
VBOD	N	5	23	Body za výkon - nepovinný údaj
VTYP	C	1	28	Doplněk typu věty výkony. Rezerva, nevyplňuje se
Celková délka věty: 29 znaků				

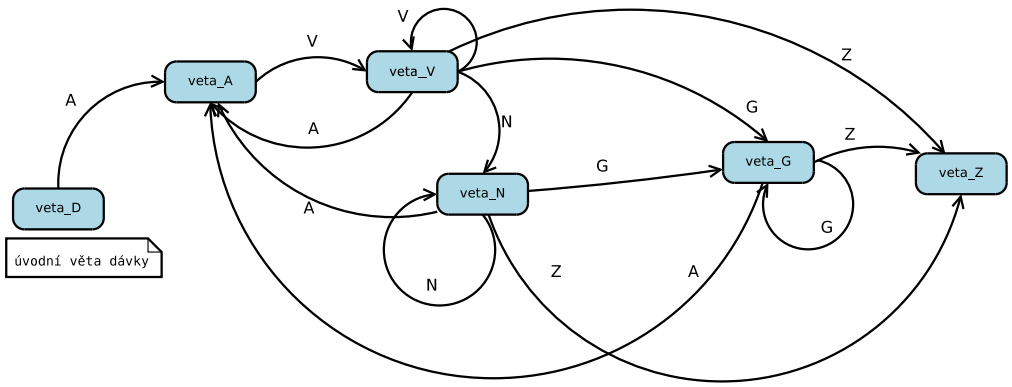
3. **náhrady za zdravotní péči** - výskyt podle počtu náhrad, maximálně 2

Název	Typ	Délka	Začátek	Popis
TYP	C	1	0	Typ věty “N” náhrady za zdravotní péči
NTYP	C	1	1	Typ náhrady
TYPN	C	1	2	Doplněk typu věty náhrady. Rezerva, nevyplňuje se
Celková délka věty: 3 znaky				

4. **ostatní diagnózy** - výskyt podle počtu diagnóz, maximálně 4

Název	Typ	Délka	Začátek	Popis
TYP	C	1	0	Typ věty “G“ - ostatní diagnózy
GCIS	C	5	1	Číslo další vyskytující se diagnózy související s poskytovanou péčí
GTYP	C	1	6	Doplněk typu věty ostatní diagnózy. Rezerva, nevyplňuje se
Celková délka věty: 7 znaků				

Věty uvedené výše mají v dávce přesně definované pořadí výskytu. Na obrázku 5.2 je znázorněn stavový diagram fiktivního automatu, který by byl schopen ověřit správné pořadí jednotlivých vět.



Obrázek 5.2: Stavový diagram znázorňující posloupnost jednotlivých vět v dávce

5.3 Požadavky kladené na aplikaci

Aplikace je učena k provádění analýz nad zdravotnickými daty získanými prostřednictvím datových dávek, které jsou generované ambulantními aplikacemi za účelem vyúčtování výkonů jednotlivým pojišťovnám. Analýza dat bude uživatelem prováděna prostřednictvím OLAP operací nad multidimenzionálním modelem, který bude namapován na relační tabulky datového skladu.

Aplikace tedy musí umožňovat zpracování jednotlivých dávkových souborů, které je složeno z následujících kroků:

1. nalezení všech výskytů dávek typu ambulantní smíšené v těchto souborech
2. selekci dat, která jsou požadována pro analýzu
3. jejich transformaci do vhodného tvaru
4. zavedení transformovaných dat do relačního datové skladu

5.3.1 Analyzovaná data

Z jednotlivých záznamů uvedených v dávce lze pro každého pacienta u kterého bylo za dané zúčtovací období provedeno alespoň jedno vyšetření extrahovat následující, z hlediska analýzy zajímavá data přestavující v navrhovaném multidimenzionálním modelu jednotlivé dimenze:

- Pacient - identifikace pacienta na základě rodného čísla, ze kterého je tudíž možné odvodit další dvě dimenze: jak věk, tak pohlaví pacienta
- Pojišťovna, u které je daný pacient veden
- Výkon - typ výkonu provedeného na pacientovi během návštěvy
- Diagnóza - informace o hlavní diagnóze pacienta na kterou byl během daného období léčen a diagnóze vztahující se k provedenému výkonu
- Čas - datum návštěvy, v rámci které byl proveden některý z výkonů

Hlavní měrnou jednotkou(faktem), nad kterou bude prováděna analýza je množství bodů získaných za provedení výkonů. Vhodným aplikováním OLAP operací nad výše uvedenými dimenzemi pak bude možné získat například následující informace:

- celkové množství získaných bodů za dané časové období
- množství získaných bodů podle jednotlivých pojišťoven v rámci daného období
- vliv věku na výskyt jednotlivých diagnóz
- selekce pacientů podle dané diagnózy a další

5.4 Návrh schématu datového skladu

Na obrázku 5.3 je znázorněn návrh datového tržiště založeného na schématu hvězdy, do kterého budou zaváděna importovaná data z dávek, nad kterými bude následně prováděna OLAP analýza.

Obrázek 5.3: Návrh datového skladu

Kapitola 6

Závěr

V rámci tohoto projektu byly definovány požadavky na výslednou aplikaci a vytvořen návrh datového skladu pro uložení importovaných zdravotnických dat, nad kterými bude aplikace následně provádět s využitím OLAP operací analýzu. Dalšími kroky v rámci diplomové práce bude volba vhodných nástrojů pro implementaci aplikace, její samotná realizace, následné shrnutí dosažených výsledků a návrh dalších rozšíření funkcionality aplikace, která by výrazně přispěla k její použitelnosti.

Literatura

[1] Lacko, L.: *Business Inteligence v SQL Serveru 2005*. Computer Press, 2006.

[2] Zendulka, J. aj.: *Studijní opora k předmětu Získávání znalostí z databází*. FIT VUT, 2006.

[3] Han, J., Kamber, M.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2006.

[4] Mrázek, P.: *Databázová aplikace pro lékaře*. Bakalářská práce. FIT VUT, 2006.

[5] Všeobecná zdravotní pojišťovna ČR. *Datové rozhraní Všeobecné zdravotní pojišťovny ČR* [online]. verze 6.2. c2007 [cit. 10.12.2007]. Dostupné z WWW: <http://www.vzp.cz/cms/internet/cz/Lekari/metodika/MDR/DR2_v62o.pdf>