

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

ROZŠÍŘENÁ PREZENTACE DATOVÉ KOSTKY OLAP

SEMESTRÁLNÍ PROJEKT  
TERM PROJECT

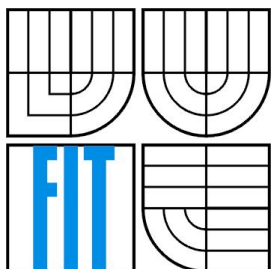
AUTOR PRÁCE  
AUTHOR

Bc. Josef Strapáč

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

# ROZŠÍŘENÁ PREZENTACE DATOVÉ KOSTKY OLAP

ADVANCED PRESENTATION OF THE OLAP DATA CUBE

SEMESTRÁLNÍ PROJEKT

TERM PROJECT

AUTOR PRÁCE

AUTHOR

Bc. Josef Strapáč

VEDOUCÍ PRÁCE

SUPERVISOR

Prof. Ing. Tomáš Hruška, CSc.

BRNO 2008

# Rozšířená prezentace datové kostky OLAP

## Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením profesora Tomáše Hrušky.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Josef Strapáč  
4. 1. 2008

## Poděkování

Na tomto místě bych chtěl poděkovat svému vedoucímu práce, prof. Tomáši Hruškovi za rady týkající se problematiky OLAP a Simonce Krčkové nejen za podporu při psaní semestrálního projektu.

© Josef Strapáč, 2008.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

## **Abstrakt**

Cílem této práce je připravení podkladů a nalezení řešení pro navazující diplomovou práci. V prvních dvou kapitolách je teoreticky objasněna problematika OLAP systémů. Ve druhé části jsou popsány a porovnány softwarové technologie.

Je kladen důraz na přívětivé a estetické řešení pro vizualizaci datové kostky.

## **Klíčová slova**

OLAP, datový sklad, dolování dat, analýza dat, vizualizace dat, datová kostka, Business Intelligence, Flash, Flex, Flare.

## **Abstract**

The aim of this work is to prepare foundation and find solution for following master's thesis. In the first two chapters is theoretically clarified problem of OLAP systems. In the second part are described and compared some software technologies.

It is insisted on friendly and aesthetic solution for visualization of data cube.

## **Keywords**

OLAP, data warehouse, data mining, data analysis, data visualization, data cube, Business Intelligence, Flash, Flex, Flare.

## **Citace**

Strapáč Josef: Rozšířená prezentace datové kostky OLAP. Brno, 2008, Semestrální projekt, FIT VUT v Brně.

# Obsah

Obsah .....	1
1 Úvod.....	2
2 Datové sklady a databáze.....	3
2.1 Relační databáze .....	3
2.1.1 12 pravidel pro relační databázové systémy .....	3
2.1.2 Popis relačních databází.....	5
2.2 Multidimenzionální databáze (MDB).....	6
2.2.1 Multidimenzionální databázový model.....	6
2.3 Business Intelligence .....	6
2.4 Datové sklady .....	7
2.4.1 Inmonova a Kimballova definice DS.....	8
2.4.2 Příprava dat.....	8
2.4.3 DW 2.0.....	9
3 OLAP systémy .....	10
3.1 OLAP.....	10
3.1.1 FASMI .....	11
3.1.2 Úložiště multidimenzionálních údajů .....	12
3.1.3 DOLAP, WOLAP, RTOLAP .....	13
3.1.4 Operace v OLAP systémech .....	13
3.2 Datová kostka .....	14
4 Vizualizace dat.....	15
5 Softwarová řešení pro grafickou reprezentaci OLAP kostky .....	16
5.1 JavaScript / AJAX .....	16
5.1.1 Google Web Toolkit (GWT).....	17
5.1.2 Ajax3D.....	17
5.2 VRML.....	17
5.2.1 X3D.....	17
5.3 Java .....	18
5.3.1 Java 3D API.....	18
5.3.2 Prefuse .....	19
5.3.3 JavaFX .....	19
5.4 Flash.....	19
5.4.1 ActionScript 3.0 .....	20
5.4.2 Flare .....	20
5.4.3 AIR .....	21
5.5 SilverLight.....	21
6 Srovnání technologií .....	22
7 Závěr .....	24
Literatura .....	25
Vysvětlení zkratk a pojmů .....	26

# 1 Úvod

Všichni jsme obklopeni daty. Daty různého charakteru a povahy – daty vědeckými, finančními, lékařskými, meteorologickými, ... Trendem současné doby je shromažďovat data. Čím více, tím lépe. Co ale udělat s takovým množstvím informací? Jak je inteligentně zanalyzovat, přechíst nebo zobrazit?

I malé množství informací se lépe chápe ve vzájemném vztahu, jsou-li zobrazena pěkným a srozumitelným způsobem. Zkuste si například představit vývoj kurzu jen na základě hodnot měny v závislosti na dnech. Spojnicový graf Vás ihned ujistí, že dlouho strádané dolary se měly vyměnit již před dvěma léty. Tento banální příklad ani v nejmenším neodráží složitost OLAP systémů, datových úložišť, ve kterých byste ručně marně hledali požadovanou hodnotu déle než pověstnou jehlu v kupce sena. Smyslem OLAP systémů ani není nalezení konkrétní hodnoty, ale zjištění souvislostí mezi daty na základě nějakých pravidel, jevů, skutečností.

Aby se analýza usnadnila, používá se pro zobrazení dat uložených v datovém modelu (v našem případě v datové kostce) vhodný graf/zobrazení. Vizualizací výsledku dotazu získáme vzhled do problému s různou úrovní zanoření (zejména v čase).

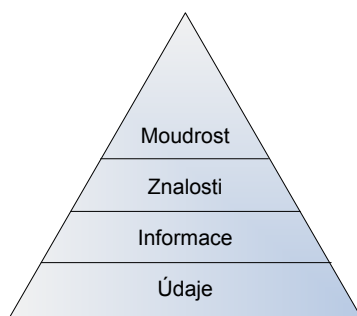
V následující kapitole si blíže objasníme problematiku datových skladů. Zaměříme se na relační databázové systémy a definici datových skladů. Po té se podrobněji podíváme na OLAP systémy. Tyto dvě kapitoly tvoří teoretický rozbor problematiky. Ve druhé části práce zanalyzujeme teoretická softwarová řešení, načež bude vše v závěru shrnuto do tabulky s porovnáním vybraných parametrů daných technologií.

Vysvětlení zkratk a pojmů na posledních stranách práce poslouží k rychlé orientaci v záplavě pojmů, zkratk a definic.

## 2 Datové sklady a databáze

Datovými sklady (angl. Data Warehouse) rozumíme komplexní data uložená ve struktuře, nad kterými je umožněno vykonávat efektivní analýzu a dotazování. Předpokládáme, že nám tato technologie nabídne prostředky nejen pro analýzu, ale i pro uložení dat. Očekáváme od datových skladů rozumnější přístup k obrovskému množství dat, která nám jsou v „surové“ podobě v podstatě bezcenná.

Tato data (údaje) je potřeba transformovat na informace, která nám jsou mnohem užitečnější. Tento proces převodu dat na informace a následně na znalosti (poznatky) prostřednictvím objevování se nazývá Business Intelligence (BI). Informace neoznačují pouze konkrétní záznam či jeho množinu, ale i trend sledované veličiny nebo vztahy mezi různými veličinami. Poznatky jsou důležitým stavebním kamenem v procesu rozhodování a jako podpora, při jejichž zrodu stojí datové sklady, analýzy OLAP (Online Analytical Processing) a dolování dat.



obr. 2.1 Hierarchie informačních úrovní

Nyní si uvedeme několik pojmů a objasníme základy, na kterých datové sklady staví, abychom se na konci této kapitoly opět vrátili k samotné definici a ozřejmění datového skladu.

### 2.1 Relační databáze

Relační databáze jsou databáze vyhovující relačnímu modelu dat. Ten je založen na matematické teorii množin a predikátové logice a definuje způsob reprezentace dat v tabulkách, integritní omezení a možné operace nad daty. Dr. Edgar F. Codd počátkem 80. let relační databázové systémy popsal pomocí následujících pravidel:

#### 2.1.1 12 pravidel pro relační databázové systémy

000: Systém musí být ohodnocen jako: relační, databázový a jako systém, jenž je možné spravovat. Aby se systém mohl označit za Systém Řízení Báze Dat (SŘBD, angl. RDBMS, dále jako systém), musí využívat relační technické prostředky, a to výhradně, k řízení/spravování databáze.

- 1: *Informační pravidlo*: Všechny údaje v relační databázi musí být reprezentovány výhradně na logické úrovni jako jediným možným způsobem – hodnotami v tabulkách.
- 2: *Pravidlo garantovaného přístupu*: Všechna data musí být přístupná bez víceznačností, tedy kombinací jména tabulky s hodnotami primárního klíče a jménem sloupce.
- 3: *Systematické zpracování hodnot typu „null“*: Nulové hodnoty (hodnota null nebo prázdná hodnota) jsou plně podporovány SŘBD a to nezávisle na datovém typu.
- 4: *Aktivní online katalog založený na relačním modelu*: Systém musí podporovat přístup autorizovaným uživatelům k online katalogu (tedy nejen k údajům, ale i k jejich popisům – metadata) pomocí stejného a jednotného jazyka.
- 5: *Úplný databázový podjazyk*: Systém musí podporovat alespoň jeden relační jazyk, který:
  - a) má lineární syntaxi
  - b) může být použit jak interaktivně, tak pomocí konzolových aplikací
  - c) podporuje definici dat (i definici pohledů – views), manipulaci s daty, bezpečnostní a integritní omezení a správu transakčních operací (begin, commit a rollback)
- 6: *Pravidlo vytváření pohledů*: Všechny pohledy, nad kterými lze teoreticky uplatnit operaci update musí být aktualizovatelné systémem.
- 7: *Schopnost vytváření (create), vkládání (insert) a mazání (delete)*: Systém musí umožňovat množinové operace s celými tabulkami při vytváření, vkládání a rušení dat.
- 8: *Fyzická datová nezávislost*: Zásah do fyzické úrovně (jakým způsobem jsou data uložena, jestli v tabulkách nebo spojených seznámech) nesmí vyžadovat zásah do aplikace založené na datové struktuře.
- 9: *Logická datová nezávislost*: Zásah do logické úrovně (tabulky, sloupce, řádky,...) nesmí vyžadovat zásah do aplikace založené na struktuře souboru.
- 10: *Integritní nezávislost*: Integritní omezení jsou definovány pomocí databázového jazyka a musí být specifikovány odděleně od aplikační logiky a uloženy v katalogu, a to i v případě změn integritních omezení nezávisle na aplikační logice.
- 11: *Nezávislost distribucí*: Distribuce oddílů databázového systému nesmí vyžadovat modifikaci aplikační logiky, musí být neviditelná pro běžného uživatele a to jak při prvním nasazení systému, tak i při jeho redistribuci.
- 12: *Pravidlo přístupu do databáze*: Jestliže má systém „nízkoúrovňový“ (procedurální) programovací jazyk, pak tento jazyk nesmí být použit k úpravě integritních omezení a musí být použit jazyk vyšší úrovně.



## 2.1.2 Popis relačních databází

### 2.1.2.1 Relace

Základem relačního modelu dat je databázová relace, tedy množina, která je vybavena pomocnou strukturou – schématem relace. Tím označíme záhlaví tabulky a jednotlivé sloupce jsou definovány pomocí atributů.

### 2.1.2.2 Integritní omezení

Správnost dat vkládaných do databáze je zajištěna integritním omezením. Může se jednat buď o slovní omezení, nebo omezení primárním klíčem.

**Primární klíč** – je jednoznačným identifikátorem každého záznamu. Buď se může vztahovat k jednomu sloupci, nebo ke kombinaci sloupců (složený klíč). Pak musí být každý řádek v rámci tabulky unikátní a jednoznačně identifikovatelný. Zároveň pole primárního klíče nesmí obsahovat prázdnou („nullovou“) hodnotu. Bez zvolení primárního klíče není možné vytvářet relace mezi tabulkami. Při výběru sloupce je nutné dbát, aby primární klíč nabýval zejména ordinálních hodnot a jeho použití bylo efektivní. Identifikace primárního klíče v jiné tabulce se značí jako cizí klíč:

**Cizí klíč** – je označení sloupce (nebo kombinace více sloupců) v tabulce a tento cizí klíč je propojen s tabulkou s primárním klíčem.

### 2.1.2.3 Normalizace

Postupnou transformací tabulky relační databáze do vhodnějšího tvaru nazýváme normalizace. Tabulky bez jakékoliv normalizace se nachází v 0NF (nulté normální formě).

**1NF (První normální forma)** – tabulka je v 1NF, pokud jsou všechny hodnoty atributů atomické (dále nedělitelné). Příklad: „Jmeno(Franta Konoupek)“ se převede na „Jmeno(Franta) a Prijmeni(Konoupek)“, kde označujeme JmenoAtributu(HodnotuAtribut).

**2NF (Druhá normální forma)** – tabulka je v 1NF a všechny atributy, kromě primárního klíče, jsou závislé na primárním klíči. Obsahuje-li tabulky pouze jeden primární klíč, je tato podmínka automaticky splněna, v opačném případě je nutná dekompozice tabulky na více tabulek, kde každá bude obsahovat alespoň jeden primární klíč a bude na atributy této tabulky odkazováno pomocí cizích klíčů.

**3NF (Třetí normální forma)** – tabulka je ve 2NF a neexistují závislosti neklíčových atributů tabulky.

Při dekompozici dbáme na následující body:

- bezztrátovost při zpětném spojení
- zachování závislostí
- odstranění redundance (opakování informace)

## 2.2 Multidimenzionální databáze (MDB)

Jsou typem databází optimalizovaných pro použití v datových skladech (DS) a OLAP (Online Analytical Processing) aplikacích. MDB jsou často vytvářeny vstupem z již existujících relačních databází, kdy se odsud data získávají například dotazy standardního dotazovacího jazyka SQL.

Do MDB se ukládají upravené a „vyčištěné“ údaje [kap. 2.4.2]. MDB na rozdíl od relačních databází používají nenormalizované tabulky. Multidimenzionální databáze zahrnují schopnost svižného zpracování dat, tím pádem je možné získat dotazované výsledky rychle. MDB využívají myšlenky datové kostky [kap. 3.2] k vyjádření dimenzí dat pro zákazníka. Klasický příklad „prodej“ může být zobrazen v závislosti na modelu, geografické poloze, čase, ceně, a mnoha jiných attributech (dimenzích).

K výhodám modelu MDB patří: rychlý a komplexní přístup k velkému objemu údajů, vytváření složitých analýz, přístup k multidimenzionálním a relačním datovým strukturám a v neposlední řadě je význačný silnou schopností pro modelování a určování prognóz.

### 2.2.1 Multidimenzionální databázový model

Agregací údajů uložených v dvourozměrné tabulce získáme multidimenzionální datovou strukturu – kostku. V podstatě jde o ekvivalent tabulky v relační databázi. Operace nad MDB modelem jsou blíže popsány v kapitole [3.1.4].

## 2.3 Business Intelligence

Technologie, praktiky, analytické a vykazovací aplikace pro sběr, analýzu a prezentaci obchodních informací. Takto heslovitě bychom mohli charakterizovat rozsáhlý záběr BI. Zabývá se zkoumáním nejen dat historických, ale i predikcí a hledáním vztahů mezi nahromaděnými daty. To umožňuje pracovníkům (zejména manažerům) velkých společností za pomoci komplexních nástrojů BI zanalyzovat data a ve správný čas rozhodnout a vykonat vhodná rozhodnutí a řešení.

Mezi produkty BI nalezneme například: Actuate server, Cognos Business Intelligence, Hyperion System, Microsoft SQL Server, Oracle Business Intelligence, MicroStrategy, SAP NetWeaver či SAS.

Základní rozdělení BI systémů je následující:

**OLTP (Online Transaction Processing)** – Termín „transakční“ je význačný. Buď se dá chápat v kontextu počítačové vědy jako transakce mezi databázemi, kdežto obchodní vyjádření pro komerci rovněž spadá do tohoto pole působnosti OLTP, které je svým způsobem neomezené.

Síla OLTP spočívá v okamžité odezvě na zákaznickovy dotazy, a to je dokázáno hlavně díky dvěma vlastnostem: jednoduchosti a efektivitě. Opačnou stranou mince je bezpečnostní riziko. OLTP

systémy jsou náchylné průniku útočníků. Čím jednodušší je systém, tím jednodušeji se dá do něj proniknout.

**MIS (Management Information System)** – Disciplína pokrývající oblasti nasazení lidských jedinců, technologií a procedur (obecně označeny jako informační systémy – IS) k řešení obchodních problémů. MIS jsou odlišné od klasických IS v tom směru, že analyzují ostatní IS v provozních aktivitách organizace. Systémy pro podporu řízení a rozhodování zajišťují poskytování kvalitních informací pro řídicí pracovníky.

**DSS (Decision-Support System)** – Třída počítačově orientovaných IS zahrnující „vědomostní databáze“ pro podporu rozhodování. DSS pracují na rozhraní taktického a strategického rozhodování.

**EIS (Executive Information System)** – Systémy pro vrcholové řízení. Jedná se v podstatě o jistý typ MIS a zpravidla se nahrazuje DSS.

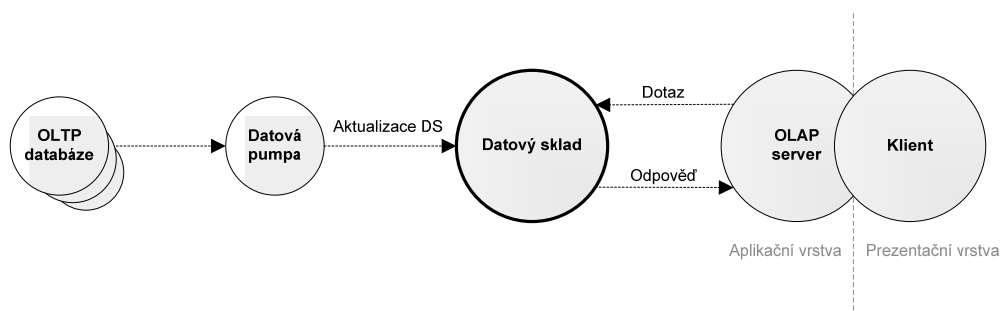
**GIS (Geographic Information System)** – Systém pro zachycení, ukládání, analýzu, správu dat a asociovaných atributů, které jsou prostorově spjaty se Zemí. V širším kontextu je GIS nástroj umožňující uživatelům vytvářet interaktivní dotazy (uživatелеm definované vyhledávání), analyzovat prostorové informace, upravovat mapy a údaje a prezentovat výsledky těchto operací.

**OLAP (Online Analytical Processing)** – OLAP je přístup k rychlému poskytnutí odpovědi analytických dotazů, jež jsou ve své podstatě vícerozměrné. Typickým použitím OLAP systémů je v business segmentu prodej, marketing, správu řízení, finančních rozpočtech a zároveň k předpovědím v tomto odvětví. Více o OLAP nalezneme v kapitole [3.1].

## 2.4 Datové sklady

Bavíme-li se o datovém skladu, myslíme tím rozsáhlou strukturu dat připravených k analýze pomocí OLAP aplikací. Někdy se oba pojmy OLAP a DS slučují a zůstává pouze OLAP, což není úplně správné, ale i my, když se budeme odkazovat na OLAP, budeme tím myslet OLAP s DS.

DS sklad slouží pro čtení (dotazy/odpovědi), jedinou výjimkou je aktualizace dat (přidávání dat nových nebo odstranění neaktuálních údajů). To je spíše možné chápat jako činnost údržbovou a provádí se v čase, kdy není DS zákazníky využíván. To je hlavní odlišnost OLAP databází od OLTP databází [obr. 2.2].



obr. 2.2 Datový sklad v rámci architektury OLAP

## 2.4.1 Inmonova a Kimballova definice DS

Existuje mnoho definic datových skladů, kde za zásadní se považuje ta od Billa Inmona z roku 1990: „*A warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process*“. Přeloženo jako: „*Datový sklad je obsahově orientovaná, integrovaná, časově rozlišená a stálá kolekce dat pro podporu rozhodovacích procesů*“.

Takto bychom mohli popsat datový sklad jako pojem, kde víceznačné termíny jsou definovány následovně:

- *Obsahově orientovaná (subject-oriented)*: data, která dávají informaci o specifickém podmětu namísto informací o nadcházejících operacích společnosti.
- *Integrovaná (integrated)*: data, jež jsou nahromaděna do datového skladu z různých zdrojů a jsou sloučena do koherentního (souvislého) celku.
- *Časově rozlišená (time-variant)*: všechna data v datovém skladu jsou identifikovatelná konkrétním časovým obdobím.
- *Stálá (non-volatile)*: data jsou trvale v datovém skladu. Další data mohou být vložena, ale nikdy nejsou odstraňována. To umožňuje managementu dosáhnout konzistentní obraz řízení společnosti.

Jinou definici datového skladu má na svědomí Ralph Kimball: „*Data warehouse is a copy of transaction data specifically structured for query and analysis*“. Tedy: „*Datový sklad je napodobení transakcí dat výslovně strukturovaných pro dotazy a analýzu*“.

## 2.4.2 Příprava dat

Jak již bylo uvedeno dříve, údaje pro DS (respektive BI/DS) jsou získávány zejména z OLTP databází ještě předtím, než se provede naplnění DS, musí se tyto údaje upravit. Tento proces se nazývá „Extrakce – Transformace – Plnění“ („Extraction – Transformation – Load“ – ETL):

### 2.4.2.1 Extrakce

Zdrojové údaje pro DS bývají často uloženy na různých hardwarových platformách, operačních systémech, ... Je nutné primární zdroje převést tak, aby odpovídaly datovému modelu. Zpravidla neodpovídají datové typy nebo je potřeba je odvodit z jiných údajů. K tomuto kroku lze využít buď aplikací vytvořených pomocí vyšších programovacích jazyků (C#, C++) nebo procedurálních nadstaveb SQL jazyka (PL/SQL, T-SQL) anebo využít tzv. gateway (přístupové brány), která převod dat řeší za běhu. Tato metoda je ale při větším množství dat nepoužitelná a zatěžuje systémové prostředky. Další možností je získat data z externích zdrojů.

Podstatnou částí této etapy přípravy dat je zhodnocení, zdali extrahovaná data odpovídají metadatům navrženým pro všechny fáze ETL. Jinak jsou data zamítnuta.

### 2.4.2.2 Transformace

Extrahovaná data jsou sérií pravidel transformovány do takové podoby, aby odpovídala datovému modelu. Některé údaje vyžadují minimální změny, u jiných je převod náročnější. Zpravidla se provádí tyto kroky:

- Vybrání jen potřebných sloupců.
- Převod hodnot typu např. „M“, „Muz“ nebo „1“ označující muže na nadefinovanou hodnotu DS, např. „M“. Jedná se o automatizované čištění dat.
- Odvození nových dat výpočtem, např. „vynos = mnozstvi \* cena“.
- Spojení dat z různých zdrojů.
- Sumarizace dat více řádků.
- Transformace nebo překlápění dat ze sloupců na řádky (i naopak).
- Aplikování dalších více či méně složitých pravidel, kde nevyhovující data jsou dle návrhu buď odstraněna, nebo ve výjimečných případech dále ručně upravena.

Nadefinování pravidel transformace závisí na cílené analýze obchodních potřeb.

### 2.4.2.3 Plnění/přenos

Závěrečná etapa ETL by měla být plánovaná (dle potřeby jednou denně, týdně,...) a automatizovaná. Dochází k přenosu dat z dočasné paměti do paměti DS a při prvotním naplnění se může jednat o velké množství.

Po zavedení údajů zpravidla dochází k indexaci dat, díky čemuž je přístup k nim ještě rychlejší.

## 2.4.3 DW 2.0

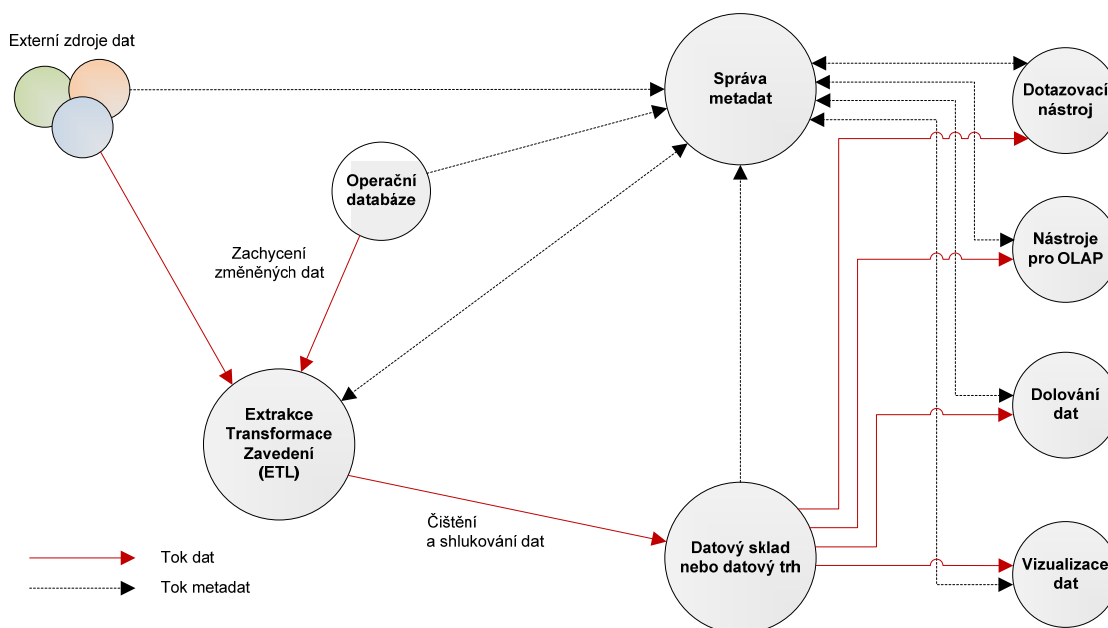
Od první definice datového skladu (DS) B. Inmonem uplynula poměrně dlouhá doba a z pouhého návrhu se DS stal běžnou součástí BI života. V roce 2006 Inmon vytvořil DW 2.0™. Obchodní značkou si Inmon chce pojistit jednoznačnou prezentovatelnost svého návrhu pro „novodobé“ DS.

DW 2.0 včleňuje některé architektonické aspekty, jenž v prvním návrhu chyběly [INM06]:

- Rozlišení životního cyklu dat v rámci datového skladu. DW 2.0 rozpozná životní cykly dat jako normální a integrovanou součást architektury.
- Začlenění nestrukturovaných dat s daty strukturovanými jako součást moderních DS. První generace DS nepovažovala nestrukturovaná data jako důležitou součást prostředí informačního procesu.
- Metadata jak na logické, tak na globální úrovni jsou integrovanou součástí prostředí moderního DS. DW 2.0 rozpoznává důležitost a rozmístění metadat. MDM (Master Data Management) je rovněž součástí prostředí DS.
- Potřeba celistvosti integrovaných dat a potřeba pro online transakci zpracovávaných dat. Oba dva požadavky je potřeba zohlednit v DS příští generace.

## 3 OLAP systémy

V předchozí kapitole jsme na obrázku [obr. 2.2] znázornili datový sklad jako zásadní prvek OLAP systémů (budeme-li dále v textu používat pojem OLAP, budeme tím myslet právě tyto systémy). Široký záběr OLAPu je možné vidět na diagramu [obr. 3.1], kdy se tímto pojmem neoznačuje pouze aplikace, nebo aplikace a datový sklad, ale celek pohánějící OLAP.



obr. 3.1 OLAP jako komplexních systém součástí

### 3.1 OLAP

OLAP je technologie uložení dat v databázi, která umožňuje uspořádat velké objemy dat tak, aby byla data přístupná a srozumitelná uživatelům zabývajícím se analýzou obchodních trendů a výsledků. Kdežto u OLTP je kladen důraz na bezpečné načítání a ukládání dat a jejich změn v prostředí více uživatelů pracujících najednou.

Základní rozdíly mezi OLTP a OLAP bychom mohli shrnout v následující tabulce [tab. 3-1], čímž si i uvědomíme zásadní rozdíly v porovnávání znacích obou technologií a hlavně bude lepší pro pochopení odlišného nasazení.

I když se Dr. Codd činil v oblasti OLAP neméně než v databázích relačních, jeho 12 pravidel (později rozšířených na 18) pro definici OLAP neměly velkou odezvu. Hlavně díky skutečnosti, že OLAP není vystaven na striktních matematických základech a mnoho poskytovatelů a výrobců si vytvářelo vlastní pravidla a směry.

Porovnávaný znak	OLTP	OLAP
Orientace	Transakční	Analytická
Orientace	Provozní zpracování	Informační zpracování
Funkce	Každodenní operace	Dlouhodobé informační požadavky, podpora rozhodování
Návrh databáze	Entitně-relační základ, aplikačně orientovaný	Hvězda/sněžná vločka, věcná orientace
Data	Současná, zaručeně aktuální	Historická
Sumarizace dat	Základní, vysoce detailní	Shrnutá, kompaktní
Náhled	Detailní	Shrnutý, multidimenzionální
Jednotky práce	Krátké, jednoduché transakce	Komplexní dotazy
Přístup	Číst a zapisovat	Většinou pouze číst
Zaměření	Vkládání dat	Získávání informací
Počet dostupných záznamů	Desítky (ale i tisíce)	Miliony
Počet uživatelů	Tisíce	Stovky
Velikost databáze	100 MB až GB	100 GB až TB
Přednosti	Vysoký výkon, vysoká přístupnost	Vysoká flexibilita, nezávislost koncového uživatele
Míry hodnocení	Propustnost transakcí	Propustnost dotazů a doba odezvy

**tab. 3-1 Rozdíly mezi OLTP a OLAP [KOU00]**

V závěru kapitoly [2.3] jsme si mohli povšimnout mnoha společných znaků BI systémů, mnoho vlastností se překrývá, jinou je daný systém specifický. Pro OLAP není žádná všemi uznávaná definice. Nigel Pendse s jeho týmem počátkem roku 1995 vytvořili test, který zkoumá vlastnosti OLAP aplikací. Nedefinuje striktně implementaci ani jiná doporučení. Pendseho tým zkoumal spjitosti a především za jakých okolností by měl být preferován jaký postup atp.

### 3.1.1 FASMI

FASMI test je zkratkou pro **F**ast **A**nalysis of **S**hared **M**ultidimensional **I**nformation [PEN05]:

**FAST** – Systém je zaměřen na doručení většiny odpovědí uživatelům v průběhu 5ti sekund, nejjednodušší analýzy by neměly zabrat více jak 1 sekundu a velice málo dotazů ne více než 20 sekund.

Nezávislý výzkum v Holandsku ukázal, že koncoví uživatelé usoudí, že v procesu došlo k chybě, pokud nebyl znám výsledek do 30ti sekund a zpravidla přecházeli k trojhmátovému řešení Ctrl+Alt+Delete pakliže nebyli upozorněni na déle trvající výpočet. Dokonce i při varování, že analýza bude trvat významně déle, uživatelé se stávali rozrušenější a jejich tok myšlenek byl náhle narušen, takže kvalita analýzy trpěla. Rychlých výsledků nelze samozřejmě dosáhnout při velkém množství dat, poskytovatelé OLAPu navrhují množství technik k zajištění rychlejší odezvy

(specializované formy úložišť, rozsáhlé „předvýpočty“ nebo hardwarová řešení na míru). I když se jedná o velký krok kupředu, kdy se dříve muselo čekat na výsledek složitě zadané analýzy dny – dnes maximálně minuty – uživatelé stejně v některých případech ztrácí pointu analýzy a pozdní odezvy OLAP dotazů jsou tak nejčastějšími „technickými problémy“.

**Analysis** – Systém by si měl poradit s jakýmkoliv obchodně logickým problémem a statistickou analýzou relevantní navržené aplikaci a koncovému uživateli by měly umožňovat zadávat tyto operace s dostatečnou dávkou jednoduchosti. Rovněž je žádané předprogramování do určité míry, ale ne pro všechny definice aplikace užitím 4GL (programovací jazyky 4. generace – Informix, ColdFusion, DataFlex,...). Je nezbytné zajistit uživatelům možnost definování účelových úvah jako části analýzy a referovat o datech určitým způsobem bez programování.

Tato analýza není podmíněna implementací v poskytovatelově nástroji nebo připojeném externím produktu, zjednodušeně řečeno, všechny potřebné funkcionality analýzy budou poskytovány koncovému uživateli intuitivním způsobem (např.: analýza časových řad, kalkulace nákladů, přepočítání měny, cílové řešení, neprocedurální modelování,...).

**Shared** – Systém poskytne všechny bezpečnostní prostředky pro diskrétnost a, pokud je nutný paralelní zápis, i uzamykání souběžných aktualizací na příslušné úrovni. Ne všichni uživatelé vyžadují zpětný zápis dat, ale pro jejich rostoucí počet, by systém měl být schopný zacházet se složenými aktualizacemi včasným a bezpečným způsobem. To je hlavní slabinou mnoha OLAP produktů, které jsou zamýšleny jen pro ke čtení se zjednodušenými bezpečnostními kontrolami.

**Multidimensional** – Klíčový požadavek. Pakliže by se měl vyzdvihnout jediný požadavek pro OLAP, byla by to multidimenzionalita. Systém musí poskytovat multidimenzionální konceptuální pohled na data, obsahující plnou podporu pro hierarchii a násobnou hierarchii, protože to je nejlogičtější způsob analýzy businessu a uspořádání. Není dáno minimální množství dimenzí, jelikož je tato vlastnost závislá na cíli aplikace.

**Information** – Veškerá data, i z nich odvozené informace, odpovídají dané aplikaci. Zpracování objemu dat, ze kterých jsou tyto informace tvořeny, se u jednotlivých produktů liší. Nemůžeme proto srovnávat produkty podle velikosti prostoru v gigabytech, ale musíme je porovnávat podle množství vstupních dat, které jsou schopny zpracovat.

### 3.1.2 Úložiště multidimenzionálních údajů

Relační databázová technologie má výsadní postavení v oblasti OLTP systémů, ta objektově-orientovaná se využívá v oblasti návrhu a implementaci aplikačních programů. Objektově-relační technologie spočívá v rozdělení návrhu aplikační logiky objektovým způsobem i s objektovými datovými typy, avšak samotná správa databáze vychází z relačního principu.

U OLAPu je to s relační technologií poměrně jinak a tak existují různé varianty: ROLAP (relational OLAP), MOLAP (multidimensional OLAP) a v praxi nejvíce využívaná varianta HOLAP



(hybrid OLAP), tedy využití obou předchozích přístupů. HOLAP vykazuje řádově rychlejší časové odezvy než základní relační struktura.

Všechny tři možnosti poskytují určité přínosy, které záleží na velikosti databáze a na způsobu, jakým budou data využívána.

#### **3.1.2.1 ROLAP**

Relační OLAP získává údaje pro analýzy z relačního datového skladu. Tyto údaje se uživateli předávají jako multidimenzionální pohled. Data a metadata jsou v úložišti ROLAP uloženy stejně jako záznamy v relační databázi. Server OLAP dynamicky využívá tato metadata na generování příkazů SQL. Data zůstávají uložena v databázi, čímž je vyřešen problém redundance.

#### **3.1.2.2 MOLAP**

Zdrojem dat multidimenzionálního OLAPu je buď datový sklad, nebo operační zdroje. Mechanismus OLAP nakopíruje data do multidimenzionálního formátu (datové struktury, sumáře). Zároveň se v průběhu tohoto procesu spočte tolik předběžných výsledků, kolik je z časového a technického hlediska možné. Lze zavést část dat ze serveru ke klientovi, čímž se dosáhne ještě rychlejší odezvy a menšího zatížení sítě.

Nevýhodou je redundance dat kvůli nakopírování do vlastních vícedimenzionálních struktur, což se negativně projeví v nárocích na úložnou kapacitu, a tím se zároveň zvyšují požadavky po technické a časové stránce. Nicméně převažující kladná vlastnost rychlé odezvy dominuje.

#### **3.1.2.3 HOLAP**

Jak již bylo řečeno, hybridní OLAP je kombinací ROLAPu a MOLAPu ve snaze využití (nebo naopak potlačení) hlavních výhod (resp. nevýhod). Údaje jsou uloženy v relační databázi a spočítané agregace se ukládají do multidimenzionálních struktur. Při dotazování se údaje vybírají do multidimenzionální paměti cache.

### **3.1.3 DOLAP, WOLAP, RTOLAP**

Jiným pohledem na celý problém při diferenciaci OLAP technologií je rozdělení na databázový OLAP (DOLAP), webový OLAP (WOLAP). DOLAP je relační SŘBD navržen k OLAP výpočtům. WOLAP se vztahuje k datům OLAP, které je možné získat přes webový server (uživateli zpravidla přes webový prohlížeč). RTOLAP je stále snem většiny analytiků. Veškerá data jsou prvně natažena do operační paměti, načež až zde jsou na ně aplikovány analytické výpočty za běhu.

### **3.1.4 Operace v OLAP systémech**

OLAP systémy napomáhají ke komplexním, datově náročným a častým dotazům typu ad hoc. Na rozdíl od OLTP systémů je OLAP vybaven předprogramovanými funkcionalitami jako například:

**Roll-up** – Někdy se tato operace nazývá „drill-up“. Data jsou sumarizována se zvyšující se generalizací na datové kostce.

**Drill-down** – Opačná operace k „roll-up“. Zvyšuje se úroveň zanoření a z méně detailního pohledu se dostáváme k pohledu detailnějšímu. Někdy můžeme při drill-down zavést další dimenzi (dimenze) kostky.

**Pivot** – Někdy označováno i jako „rotate“. Jde o vizuální operaci, kdy se zaměňují osy v pohledu za účelem znázornění dat z jiného alternativy. Například můžeme osy 3D kostky převést na sérii 2D rovin.

**Slice and dice** – Operace „slice“ provádí výběr jedné dimenze datové kostky – vzniká „podkostka“. Naopak „dice“ vybírá dimenze minimálně dvě a rovněž vytváří podkostku.

**SortingŘazení** – Data jsou seřazena dle ordinálních hodnot.

**Výběr** – Data jsou dostupná buď hodnotami, nebo rozsahem.

**Jiné operace.**

## 3.2 Datová kostka

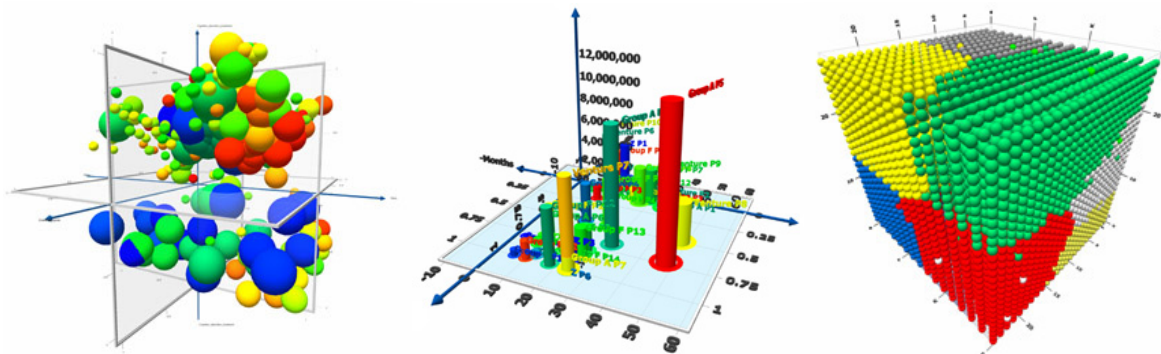
Datová, neboli OLAP kostka, je druhem vícedimenzionální matice, která umožňuje uživateli zkoumat soubor dat z různých pohledů. Obvykle se pracuje se třemi dimenzemi, to ovšem není maximum, některé systémy umí pracovat až se 64 rozměry.

V kapitole [3.1.2] jsme zmínili různé metody ukládání dat. Zvolení správné metody je klíčové pro práci s datovou kostkou.

## 4 Vizualizace dat

Jelikož data vyjadřují nějaké množství, objem, velikost,... Není vždy jednoduché zorientovat se a správně usoudit závěr z rozsáhlé, ale nepřehledné tabulky. Pro různé případy existují typizované grafy či zobrazovací metody, které uživateli zkoumaný problém dokážou prezentovat mnohem lépe, jasněji a rychleji, než pouhá čísla.

Jako příklad výstupů profesionální aplikace Miner3D mohou být následující vizualizace:



### 4.1 Různé možnosti zobrazení analýz

Jak již bylo řečeno, pro analýzu každé problematiky je vhodný jiný typ grafu. Při návrhu sady grafů bude nutné prozkoumat typické dotazy v aplikaci řešené vedoucím. Tato část však proběhne při další etapě, při samotném řešení diplomové práce.

# 5 Softwarová řešení pro grafickou reprezentaci OLAP kostky

V současné době je k dispozici poměrně široká paleta nástrojů, které je možné použít k vizualizaci výsledků (nebo pouze jejich částí). Záleží, jaké nároky klademe na kvalitu výstupního řešení a stejně tak, jaký kompromis jsme schopni učinit pro dosažení lepšího a uživatelsky příjemnějšího zobrazení.

Hledáme proto takové technologie, které nám umožní za co nejmenšího (či nejsnazšího) zásahu či nastavení shlédnout grafickou reprezentaci výsledků. Uživatel nechce být rušen doinstalováním podpůrných prohlížečů, plug-inů, či jiného softwaru pro běh aplikací, které nejsou nativně podporovány webovým prohlížečem. Proto by navržené řešení mělo odpovídat současným standardům s přihlédnutím k budoucím trendům. Přesto budeme zkoumat i ty produkty, jejichž aplikační podpora na straně uživatelů není tolik rozšířená nebo oblíbená.

## 5.1 JavaScript / AJAX

JavaScript (JS) je interpretovaný, objektově orientovaný programovací jazyk, který je zpracován na straně klienta webovým prohlížečem. Tím pádem je zajištěno zpracování skriptů psaných v jazyce JS, není-li však spouštění javascriptů zamezeno uživatelem. Statistika\* udává, že v prosinci roku 2007 mělo JavaScript povoleno asi 95 %.

Ačkoliv není AJAX samostatnou technologií, či přímým „derivátem“ JS, při vytváření „AJAXovských“ aplikací se bez JS zpravidla neobejdeme. Mimo JS jsou k vytváření interaktivních webových aplikací (RIA) použity technologie jako: (X)HTML, CSS, DOM nebo XMLHttpRequest. Kdykoliv uživatel zadá nějaký požadavek serveru (odešle formulář, přechází na novou stránku), v drtivé většině případů dochází ke znovunačtení celé webové stránky, čímž síť podstupuje opětovný přenos již jednou zaslanych dat.

Sice je tento způsob interakce mezi uživatelem a serverem brán jako běžná věc, existují však způsoby, jak načítat jen ta data, která jsou uživatelem požadována a která tak nebudou tvořit ve většině případů již jednou stažená, redundantní data (nebereme-li v potaz cílenou aktualizaci dat). XMLHttpRequest je rozhraní pro komunikaci webových aplikací mezi klientem a serverem prostřednictvím protokolu HTTP.

---

\* <http://www.thecounter.com/stats/2007/December/javas.php>

### 5.1.1 Google Web Toolkit (GWT)

Jde o framework založený na platformě Java, ve kterém si uživatel může vytvořit webovou aplikaci složenou ze znovupoužitelných UI komponent (renderovaných jako HTML) a logiky (ve formě JavaScriptu) na straně webového prohlížeče a asynchronního volání serverové logiky skrze AJAX. Hlavním prvkem GWT je kompilátor Javy do JS. Jelikož jsou zdrojové kódy aplikace v Javě, je potřeba je přeložit pro použití na webovém prohlížeči (tzv. „web mód“). Ale i přesto je možné spouštět program jako klasickou Java aplikaci v prostředí speciálně upraveného GWT prohlížeče.

Pro náš případ ovšem není tato technologie příliš vhodná.

### 5.1.2 Ajax3D

3D engine, který používá HTML tag <canvas>. V současné době není příliš rozšířen a ani podporován, i když své opodstatnění jistě najde při budoucím rozšíření HTML5.

## 5.2 VRML

VRML, neboli Virtual Reality Modeling Language, je deklarativním jazykem určeným pro popis virtuální reality (VR), ovšem VR ve smyslu poplatném roku vzniku (počáteční verze v 1994). VRML97 (VRML 2.0) je definován normou ISO, čímž je jazyk VRML zařazen mezi jeden z mála standardů. Zde je již možné specifikovat animace a programovat reakce na různé události, takže se z na první pohled běžného souborového formátu stává celá platforma pro VR.

Pro VRML scény je nutné mít na klientském počítači nainstalován prohlížeč \*.wrl souborů. Stejně tak pro zobrazování pomocí webového prohlížeče je nezbytný plug-in.

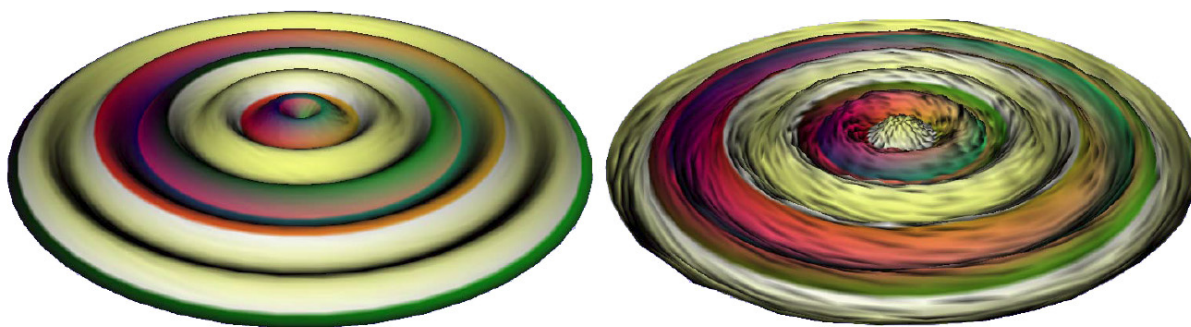
### 5.2.1 X3D

X3D v sobě kombinuje prvky VRML 2.0 (pro definici 3D scén) a značkovacího jazyka XML. ISO standard byl X3D udělen roku 2005. XML by v našem případě byl ideálním způsobem k načítání dat a následnému převedení k transformaci objektů zobrazujících výstupní informace.

X3D se v současnosti příliš neprosazuje a situace kolem tohoto projektu se dál nevyvíjí. Občas je možné nalézt zajímavé projekty založené na této technologii, ale i přesto je možné X3D (VRML) považovat spíše jako formát pro uchování informací 3D modelů nebo celých scén, než jako nástroj pro zobrazení kvalitních výstupů. Jistou výjimkou je následující rozšíření:

#### 5.2.1.1 FVRML/FX3D: Function-based extension of VRML and X3D

Toto rozšíření umožní vytvářet prakticky jakýkoliv geometrický prvek. Definováním jeho geometrických tvarů, barvy, 3D textury, transformací,... Rozšíření je navrženo jak pro navzájem spolupracující modely, tak i pro vytváření virtuálních světů.



**obr. 5.1** Netradiční zobrazení VRML pomocí FVRML/FX3D

Ovšem pro zobrazení takto navržených modelů je nutné mít nainstalován „blaxxun Contact“ (plug-in do webového prohlížeče), „BS Contact VRML/X3D“ (další VRML prohlížeč, který oproti ostatním vyniká podporou DirectX 9 a OpenGL a jinými vlastnostmi, které bychom u nekomerčních produktů marně hledali). Nakonec je nutný „FVRML/FX3D plug-in“. Výsledné zobrazení je díky softwarovému/hardwarovému renderu opravdu zdařilé, ale množství nezbytných plug-inů toto řešení činí nereálným pro běžné nasazení.

## 5.3 Java

Java je objektově orientovaný programovací jazyk firmy Sun Microsystems. Od května roku 2007 se tento jazyk dále vyvíjí jako „open source“. Jeho nejvýznamnějším kladem je přenositelnost na téměř všechna zařízení, která podporují Javu (čipové karty – JavaCard, mobilní zařízení – Java ME, desktopy – Java SE, distribuované systémy – Java EE). Nevýhodou je poměrně zdouhavé spouštění aplikace. Programy nejsou kompilovány (jako např. C++), ale je vytvořen tzv. bajtkód, který se interpretuje až za pomoci JVM (Java Virtual Machine), jenž je navržen pro každou platformu.

Statistika\* udává poměrně vysoké rozšíření Javy (asi 94 % v prosinci roku 2007). I přes svůj pomalejší start si Java nachází své uplatnění ve spoustě odvětví. V současné době existuje několik nástrojů, které za pomoci Javy umožní programovat vizuálně zajímavé aplikace. Zmíníme následující tři:

### 5.3.1 Java 3D API

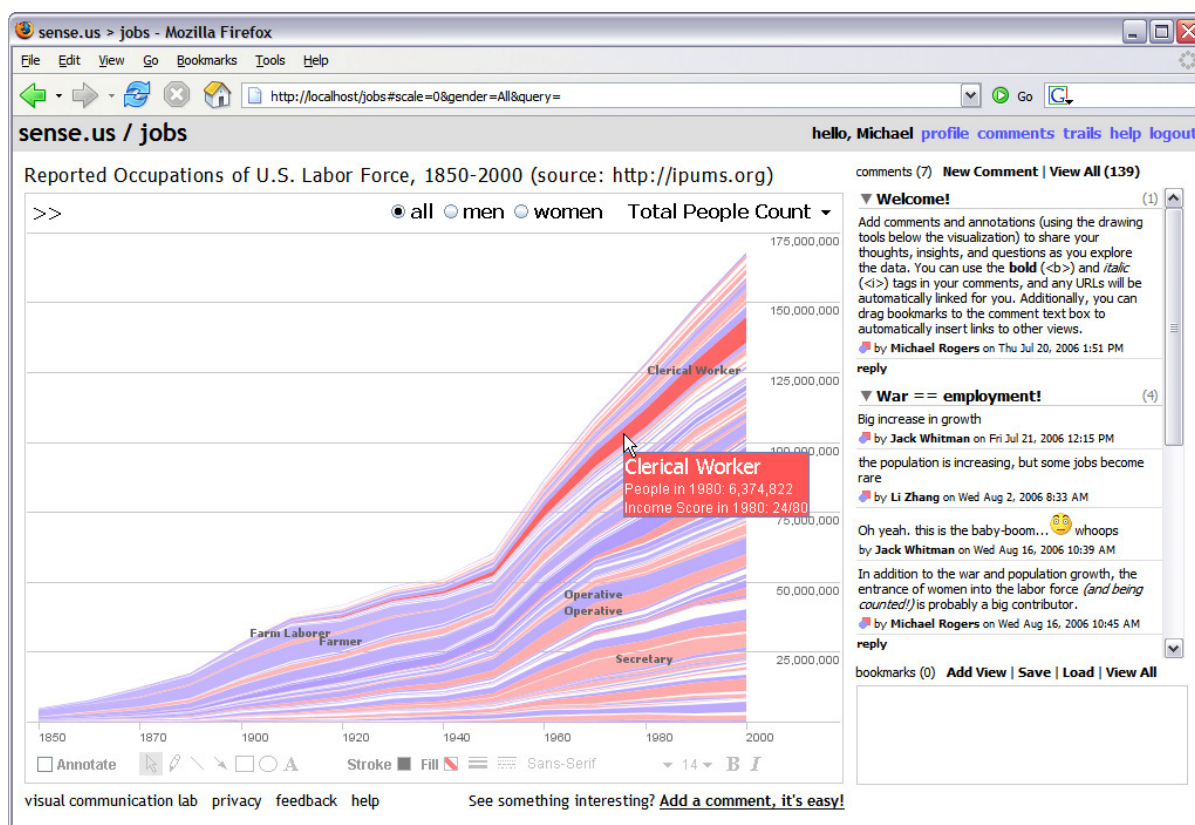
Jde o volitelný balíček pro jazyk Java, běží nad OpenGL nebo Direct3D. Java3D není pouhou obálkou pro grafická API, ale celkové rozhraní, které do sebe uzavírá grafické programování využívající reálný, objektově-orientovaný návrh. Celá scéna tvoří strom a jednotlivé objekty jsou reprezentovány uzly, které mohou být dále upravovány a definovány pomocí vlastností objektově-orientovaného programování. Java3D dále nabízí širokou podporu pro práci se zvukem.

\* <http://www.thecounter.com/stats/2007/December/java.php>

## 5.3.2 Prefuse

Platforma pro vytváření interaktivních a především vizuálně přívětivých aplikací. Prefuse podporuje bohatou sadu nástrojů pro datové modelování, vizualizaci a interakci. Poskytuje optimalizované datové struktury pro tabulky, grafy a stromy, množství návrhů a vizualizačních technik, podporu pro animaci, dynamické dotazy, integrované vyhledávání a propojení s databázi.

Prefuse je napsán v Javě a využívá 2D grafickou knihovnu. Již po zhlédnutí několika ukázkových prací, ale i produktů nasazených v reálných problémech, bychom mohli Prefuse považovat za jednu z možných voleb k řešení problematiky reprezentace datové kostky.



obr. 5.2 sense.us

## 5.3.3 JavaFX

Pokus o ryzí Java RIA nástroj je nastíněn u projektu JavaFX. Ten je zatím ve vývojové fázi, ale i tak by se mohlo jednat o výkonnou technologii a hlavně je nutné brát v potaz možnost běhu Java aplikací na všech zařízeních, kde je Java SE/ME.

## 5.4 Flash

Flash (dříve Macromedia, nyní Adobe) představuje stále populárnější software pro tvorbu interaktivních webů, her a aplikací. Ke tvorbě flashových aplikací je možné přistupovat z více stran. Základním způsobem tvorby \*.swf (od Shockwave file) souborů je vývojový nástroj Adobe Flash. Je

orientován spíše na design než na „klasické“ programování. Nicméně díky ActionScriptu (AS) je možné přistupovat ke grafickým objektům pomocí tohoto skriptovacího jazyka, založeném na JavaScriptu, jako k jakémukoliv jinému objektu, vytvořeném v kterémkoliv jiném objektově-orientovaném programovacím jazyku. Tím odpadá zdlouhavé, a většinou i nudné, definování grafických objektů v nástroji jiném než Adobe Flash.

Přístupem programátorským pro vývoj flashových aplikací je možné brát použití Adobe Flex (Flex). Jde o kolekci nástrojů pro tvorbu platformě nezávislých, RIA aplikací založených na proprietární platformě Adobe Flash. Lze použít buď Flex Builder nebo Flex SDK. Ovšem oba vývojové nástroje se v mnoha aspektech překrývají a není nic, co by se nedalo vytvořit v nástroji druhém, jen je potřeba použít odlišný přístup.

K zobrazení/přehrání flash aplikací je nutné mít nainstalován ve webovém prohlížeči tzv. „Flash player“. Díky všeobecné oblibě „flashe“ se většinou nejedná o nijak zásadní problém. Opět uvedeme statistiku\* (jistě mírně zkreslenou) společnosti Adobe: Flash Player verze 6 má nainstalováno asi 98.8 % uživatelů a poslední verzi Flash Player 9 zhruba 95 % (prosinec roku 2007).

### 5.4.1 ActionScript 3.0

ActionScript od verze 3 je zcela přepracován. Pro jeho provoz je nutné mít Flash Player 9, který obsahuje dva „kompilátory“. Jeden pro AS verze 3.0 a druhý pro AS verze 1.0 a 2.0.

AS 3.0 je zaměřen na RIA aplikace ještě více než předchozí verze a nabízí značná vylepšení. Výrobce udává až 10x rychlejší zpracování než dřívější verze AS, AS3 je plně podporován Flex Builderem 2, kód je lépe čitelnější,... V souhrnu jde o vysoce výkonný nástroj, se silnou komunitou a jistotou vývoje v následujících letech.

### 5.4.2 Flare

Flare je souborem knihoven AS 3 pro interaktivní vizualizace na webu. Celkově bychom mohli uvést totožný popis jako u Prefuse pro Javu. Zde se ovšem jedná o knihovny využívající AS 3 pro flash.

Flare aplikace mohou být vyvíjeny pomocí Adobe Flex SDK nebo Adobe Flex Builderu. Jak již bylo zmíněno dříve, pomocí nástroje Flex se vytváří klasický zdrojový kód, který se následně zkompiluje do \*.swf souboru.

Příklad kódu:

```
package {
    import flare.data.DataSource;
    import flare.vis.Visualization;
    ...
    private function buildVis(data:Data):void
    {
        vis = new Visualization(data);
        vis.bounds = new Rectangle(0, 0, 600, 500);
    }
}
```

\* [http://www.adobe.com/products/player\\_census/flashplayer/version\\_penetration.html](http://www.adobe.com/products/player_census/flashplayer/version_penetration.html)



```

        vis.x = 100;
        vis.y = 50;
        addChild(vis);

        vis.operators.add(new AxisLayout("data.date", "data.age"));
        vis.operators.add(new ColorEncoder("data.cause", Data.NODES, "lineColor",
            ColorPalette.category(data.nodes.stats("data.cause").unique)));
        vis.operators.add(new ShapeEncoder("data.race"));
        vis.data.nodes.setProperties({fillColor:0, lineWidth:2});
        vis.update();
    }
}
}

```

### 5.4.3 AIR

Adobe Integrated Runtime (dříve označován jako Apollo) je multiplatformní prostředí pro budování RIA za pomoci nástrojů Flash, Flex, HTML a Ajax, které může běžet jako klasická desktopová aplikace. Tak se webová flash aplikace může stát desktopovou a naopak.

## 5.5 SilverLight

Předchůdcem této technologie je WPF (Windows Presentation Foundation). Jde o způsob programování aplikací pro Windows Vista nebo Windows XP (je nutný SP2 a .NET 3.0 framework).

SilverLight je technologie společnosti Microsoft pro tvorbu RIA aplikací běžících pod webovými prohlížeči. Verzi 1.0 nelze považovat za adekvátního konkurenta ostatních RIA technologií a proto když budeme hovořit o SilverLight, budeme popisovat jeho vlastnosti, které budou funkční až od verze 1.1. Na rozdíl od WPF je SilverLight „provozuschopný“ již po nainstalování plug-inu do prohlížeče (úplně stejně jako je tomu u flashe).

## 6 Srovnání technologií

Do srovnávací tabulky zahrneme jen ty technologie, které mají určitý potenciál pro řešení problematiky diplomové práce. Vynecháváme tak JavaScript a Ajax, jejichž hlavní devizou není složitější vizualizace dat.

Rovněž zde nalezneme licenční ujednání jak pro IDE (vývojových prostředí), tak i pro výstupní aplikaci.

Technologie	Vývojové prostředí	Licence (IDE / aplikace)	Nutné pluginy / software	Platformy (OS: prohlížeč)	Rozšíření
VRML/X3D	Jakékoliv IDE pro VRML	? / GPL	VRML plugin	Většina OS a prohlížečů	Nízké, téměř se nepoužívá
FVRML/FX3D		? / BS Contact je komerční	Blaxxun Contact, BS Contact VRML/X3D, FVRML/FX3D plugin	Závisí na podpoře ze strany závislých pluginů	Pro studijní/testovací účely
Java3D	Jakékoliv Java IDE	? / JDL	JVM	Drtivá většina zařízení s OS podporující běh Javy	Všude, kde je nainstalován JVM
JavaFX	Zřejmě Java IDE	? / GPLv2 (?)	JVM + podpurný plugin		V raném stadiu vývoje
Prefuse	Jakékoliv Java IDE	? / BSD	JVM		Všude, kde je nainstalován JVM
Flash (AS 3.0)	Flash CS3	Komerční / EULA	Flash Player 9	Podporována většina OS a prohlížečů (problémy jsou u linuxu s amd64 jádrem)	Závisí na nainstalované verzi Flash Playeru
Flex	Flex Builder Flex SDK Eclipse plug-in	Komerční, ale pro studenty univerzit zdarma / EULA MPL / MPL Zdarma / EULA			U PPC/WinME zařízení se uvedení FP9 očekává
Flare	knihovny pro Flex	viz. Flex / BSD			
SilverLight	Microsoft Visual Studio 2008	Komerční (MSDNAA) / open-source vs. MPL	Plugin adekvátní použité verzi	Win: IE6, IE7, FF, (O v přípravě) Mac: Safari Linux (projekt Moonlight)	Stále ve vývoji

**Vysvětlivky:** ? – Závisí na konkrétním IDE s jeho licenčních podmínkách; IE – Internet Explorer, FF – Firefox, O – Opera

**tab. 6-1 Porovnání vybraných technologií**

Z možných řešení jsem vybral Flex s případným použitím knihovny Flare. I když se jedná o verzi, která stále prochází vývojem, při pohledu na aplikace založených na Prefuse (Java), se dá tušit, že se tento projekt těší velké oblibě. SilverLight je nadějnou technologií a již v alfa verzi 1.1 v některých vlastnostech předčí flash.

Proto zde uvedeme (pokud možno nezaujaté) srovnání v hlavních aspektech kladených na RIA aplikace, budeme se ptát na platformní nezávislost, podporu pro přehrávání videa (streamingu) nebo podporu grafických formátů způsob vykreslování. Nelze však tvrdit, že by pro náš problém byly všechny body srovnání důležité.

Porovnávaný znak	Flash (AS 3.0)	SilverLight 1.1
Platformní nezávislost	Téměř bez problémů	Zatím jen pro Windows a Mac OS (v blízké době se ale čeká rozšíření pro Linux a zbylé prohlížeče)
Použití dalších programovacích jazyků	Pouze ActionScript	VBScript, C#, Python, Ruby,...
Výměna dat s webovým serverem (např. pomocí Ajaxu)	Ano	Ano
Šifrovaný obsah	Ne	Ne
Podpora pro prohlížení souborů offline	Ne	Ne
Podpora grafických formátů	Podporuje většinu grafických formátů	Zatím jen PNG a JPG (SL 1.0)
Podpora pro zvuk	Ano	Ano
Rastrové filtry a efekty	Je možné aplikovat	Není možné aplikovat
Práce s fonty	Problematické	SL s fonty zachází lépe
Animační model	Založeno na jednotlivých rámečcích	Modely založeny na čase
Přehrávání videa přes celou obrazovku	S omezeními	Ano
HD video (720p)	Ne	Ano

**tab. 6-2 Porovnání Flash (AS 3.0) a SilverLight 1.1**

## 7 Závěr

V tomto semestrálním projektu, který je úvodem k samotné diplomové práci, jsme se zabývali širokou a komplexní problematikou relačních databází, datových skladů a zejména OLAP systémů. Po té jsme přešli k popisu a porovnání možných softwarových řešení.

OLAP je natolik komplikované téma, že jen samotné objasnění pojmů a pro rozlišení jemných nuancí je často potřeba použít verbální ekvilibristiky. Mnoho termínů je víceznačných a jiné termíny jsou téměř ekvivalentem pojmů tematicky podobných. Zde vidím hlavní nedostatek při popisu teorie OLAP databází a proto by bylo vhodné některé části této práce hierarchicky lépe strukturovat, potřebná data doplnit, případně podobná témata doplnit.

Návrh sady modelů bude dalším krokem v diplomové práci, je nutné této fázi věnovat adekvátní pozornost. Stejně tak bude nutné prozkoumat možnosti technologie Flex a sady knihoven Flare. I když se tento produkt zdá být z nalezených řešení tím nejpříjemnějším, na první pohled skryté svízele, se samozřejmě mohou kdykoliv vyskytnout.

Vybrat správný software byl zřejmě nejtěžší, i časově nejnáročnější, krok. Existuje obrovské množství softwarových aplikací, které mají různé možnosti. Také komunita a podpora kolem technologií je důležitá. Poměrně zajímavým zjištěním byla skutečnost, že VRML, dříve tolik opěvovaný jazyk a souborový formát, se uplatní snad ve studijních nebo testovacích účelech, ale své místo a účel si VRML formát určitě vždy najde.

Pokud bychom měli hádat, jakým směrem se současný trend vývoje webových aplikací bude ubírat, bude to právě vytváření RIA aplikací. Možnosti jsou téměř neomezené a na poli internetových technologií se objevují stále nové produkty, jen je nutné rozeznat mezi všemi „výkřiky módy“ ty perspektivní.

# Literatura

- [1] LACKO, Luboslav. *Datové sklady, analýza OLAP a dolování dat*. 1. vyd. Brno : Computer Press, 2003. 486 s. ISBN 80-7226-939-0.
- [2] JIAWEI, Han, KAMBER, Micheline. *Data Mining: Concepts and Techniques*. 2nd edition. San Francisco : Morgan Kaufmann, 2006. 743 s. ISBN 10: 1-55860-901-6.
- [3] DREWEK, Catherine. *Data Warehousing: Similarities and Differences of Inmon and Kimball* [online]. 2005-03-31 [cit. 2007-12-13].  
Dostupný z WWW: <<http://www.b-eye-network.com/view/743>>.
- [4] Chaudhuri, S. and Dayal, U. 1997. An overview of data warehousing and OLAP technology. *SIGMOD Rec.* 26, 1 (Mar. 1997), 65-74.  
Dostupný z WWW: <http://doi.acm.org/10.1145/248603.248616>
- [5] [INM06] INMON, Bill. An Introduction to DW 2.0. *Business Intelligence Network : BI & Data Warehousing Resources* [online]. March 16, 2006 [cit. 2007-12-15]. Dostupný z WWW: <<http://www.b-eye-network.com/view/2568>>.
- [6] [PEN05] PENDSE, Nigel. *An analysis of what the often misused OLAP term is supposed to mean* [online]. August 15, 2005 [cit. 2007-12-16].  
Dostupný z WWW: <<http://www.olapreport.com/fasmi.htm>>.
- [7] [KOU00] KOUBA, Z. *Datové sklady. Dobývání znalostí z databází 2000 : Sborník přednášek, 2000*
- [8] Arigon, A., Tchounikine, A., and Miquel, M. 2006. Handling multiple points of view in a multimedia data warehouse. *ACM Trans. Multimedia Comput. Commun. Appl.* 2, 3 (Aug. 2006), 199-218. Dostupný z WWW: <<http://doi.acm.org/10.1145/1152149.1152152>>
- [9] Maniatis, A. S., Vassiliadis, P., Skiadopoulos, S., and Vassiliou, Y. 2003. Advanced visualization for OLAP. In *Proceedings of the 6th ACM international Workshop on Data Warehousing and OLAP* (New Orleans, Louisiana, USA, November 07 - 07, 2003). DOLAP '03. ACM, New York, NY, 9-16. Dostupný z WWW: <<http://doi.acm.org/10.1145/956060.956063>>
- [10] TIŠNOVSKÝ, Pavel. VRML : jazyk pro popis virtuální reality [online]. Internet Info, s.r.o., 1998-2008, 8. 11. 2007 [cit. 2008-01-02]. Dostupný z WWW: <<http://www.root.cz/clanky/vrml-jazyk-pro-popis-virtualni-reality/>>. ISSN 1212-830.
- [11] Pro objasnění některých pojmů byly použity zdroje „SearchOracle.com Definitions“ nebo „Wikipedia.org“

# Vysvětlení zkratk a pojmů

**AIR** – *Adobe Integrated Runtime*.

**AJAX** – *Asynchronous JavaScript and XML*. Technologie pro vytváření RIA.

**API** – *Application Programming Interface*.

**BI** – *Business Intelligence*.

**Business Intelligence** – Obor činnosti nadřazený všem procesům zpravodajství v obchodním segmentu.

**Codd (Edgar F.)** – Britský počítačový vědec, vyjádřil se jako jeden z prvních o teorii relačních databází.

**Data cube** – viz. Datová kostka.

**Data Warehouse** – viz. Datový sklad.

**Datová kostka** – Druh vícedimenzionální matice, která uživatelům umožňuje zkoumat a analyzovat soubor dat z mnoha různých perspektiv.

**Datový sklad** – Zvláštní typ relační databáze, která umožňuje řešit úlohy zaměřené převážně na analytické dotazování nad rozsáhlými soubory dat.

**DSS** – *Decision Support System*. IS zahrnující „vědomostní databáze“ pro podporu rozhodování.

**DW, DWH** – *Data Warehouse*.

**EIS** – *Executive Information System*. Systémy pro vrcholové řízení.

**ETL** – *Extract, Transform, and Load*. *Extrakce, transformace a plnění*.

**FASMI** – *Fast Analysis of Shared Multidimensional Information*. Test OLAP aplikací.

**FLASH** – Technologie pro tvorbu (internetových) aplikací, her a prezentací. Ke svému běhu potřebují Flash player.

**FLEX** – Sada nástrojů pro vývoj aplikací, zejména RIA.

**FVRML/FX3D** – *Function-based extension of VRML and X3D*. Funkční rozšíření VRML a X3D.

**GIS** – *Geographic Information System, Geografický Informační Systém*.

**HOLAP** – *Hybrid OLAP*. Kombinace ROLAP a MOLAP.

**Inmon (Bill)** – Považován za otce problematiky datových skladů.

**JAVA** – Programovací jazyk vyvinut Sun Microsystems. Javovské aplikace jsou kompilovány do tzv. bytecodu a běží pak na jakémkoliv JVM (Java Virtual Machine).

**Kimball (Ralph)** – Autor témat datových skladů a Business Intelligence.

**MDB** – *Multidimenzionální databáze*.

**Metadata** – (z řeckého *meta-* = mezi, za + latinského *data* = to, co je dáno). Strukturovaná data o datech.

**MIS** – *Management Information System, Manažerský Informační Systém*.

**MOLAP** – *Multidimensional Online Analytical Processing*.

**MS** – *Microsoft*

**Multidimenzionální databáze** – Specializované na BI, obsahují data předpočítaná podle očekávaných požadavků uživatelů a poskytují tak vysoký výkon při dotazování a analýze.

**OLAP** – *Online Analytical Processing*.

**OLTP** – *Online Transaction Processing*.

**RDBMS** – *Relational Database Management System*.

**Relační databáze** – Databázový systém, který je založen na relačním modelu dat a relační algebře.

**RIA** – *Rich Internet application*. Webové aplikace, které mají funkcionality klasických desktopových aplikací.

**ROLAP** – *Relational Online Analytical Processing*.

**RTOLAP** – *Real Time OLAP*.

**SDK** – *Software Development Kit*.

**SilverLight** – Kódové označení: *WPF/e (Windows Presentation Foundation Everywhere)*. Plug-in společnosti Microsoft pro webové prohlížeče k vytváření RIA.

**SQL** – *Structured Query Language*. Standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích.

**SŘBD** – *Systém řízení báze dat*. Softwarové vybavení, které zajišťuje práci s databází, tzn. tvoří rozhraní mezi aplikačními programy a uloženými daty.

**VR** – *Virtuální realita*.

**VRML** – *Virtual Reality Modeling Language (Virtual Reality Markup Language)*. Formát pro reprezentování 3D interaktivní vektorové grafiky.

**X3D** – *Extensible 3D*. XML formát na ukládání 3D scén, vychází z VRML97. ISO standard.

**XML** - *Extensible Markup Language*. Značkovací jazyk konsorcia W3C. Umožňuje vytváření konkrétních značkovacích jazyků pro různé účely a široké spektrum různých typů dat.