VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

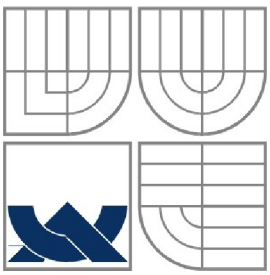
MULTIDIMENZIONÁLNÍ DATA V SYSTÉMU OLAP

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

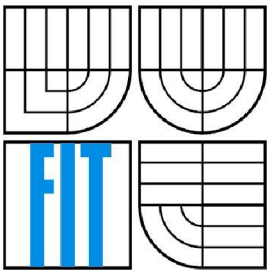
AUTOR PRÁCE
AUTHOR

JAROSLAV KUPČÍK

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

MULTIDIMENZIONÁLNÍ DATA V SYSTÉMU OLAP

MULTIDIMENSIONAL DATA IN OLAP SYSTEM

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE
AUTHOR

JAROSLAV KUPČÍK

VEDOUCÍ PRÁCE
SUPERVISOR

Prof. Ing. TOMÁŠ HRUŠKA CSc.

BRNO 2008

Zadání bakalářské práce/6692/2007/xkupci03

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2007/2008

Zadání bakalářské práce

Řešitel: **Kupčík Jaroslav**
Obor: Informační technologie
Téma: **Vícedimenzionální data v systému OLAP**
Kategorie: Databáze

Pokyny:

1. Seznamte se se systémy pro podporu rozhodování, zejména pro OLAP technologii.
2. Vyřešte problém zobrazení vícedimenzionálních dat na obrazovce.
3. Realizujte rozhraní systému pro rozhodování jako komunikační variantu v informačním systému.

Literatura:

- Lacko,L.: Datové sklady, analýza OLAP a dolování dat, Computer Press 2003, ISBN 80-7226-969-0

Při obhajobě semestrální části projektu je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Hruška Tomáš, prof. Ing., CSc.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2007
Datum odevzdání: 14. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

L.S.

doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Jaroslav Kupčík**
Id studenta: 79329
Bytem: Masarykova 955, 735 14 Orlová
Narozen: 04. 06. 1986, Karviná
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Vícedimenzionální data v systému OLAP
Vedoucí/školitel VŠKP: Hruška Tomáš, prof. Ing., CSc.
Ústav: Ústav informačních systémů
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:
tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
   - ☐ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....	.....
Nabyvatel	Autor

Abstrakt

Práce se zabývá problematikou multidimenzionálních dat v systému OLAP, zejména jejich vizualizací na dvourozměrné obrazovce. Stručně je zde rozebrán způsob uložení dat v multidimenzionální databázi a formální popis struktury dat v hyperkrychli. Následuje velmi podrobná klasifikace grafů přihlížející zejména na jejich použitelnost pro vizualizaci multidimenzionálních dat a pro demonstraci při oficiálních prezentacích, kde hraje největší roli názornost. Poslední částí práce je popis etap vývoje rozhraní informačního systému pro vizualizaci multidimenzionálních dat.

Klíčová slova

multidimenzionální data, vizualizace, OLAP, grafy

Abstract

Thesis considers the problems of multidimensional data in OLAP systems, especially its visualization at 2D computer screen. Briefly it describes the way how the data are stored in multidimensional database and formal description of hypercube structure. Next chapter contains very detailed classification of diagrams including the facts about its usability for visualization of multidimensional data and for official presentations where the high synopticity is required. Last chapter is description of single stages in development of information system interface for multidimensional data visualization.

Keywords

multidimensional data, visualization, OLAP, diagrams

Citace

Kupčík Jaroslav: Multidimenzionální data v systému OLAP.
Brno, 2008, bakalářská práce, FIT VUT v Brně.

Multidimenzionální data v systému OLAP

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Prof. Ing. Tomáše Hrušky CSc.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jaroslav Kupčík
9. května 2008

Obsah

1 Úvod

V současnosti může být člověk považován za sběratele dat. Díky moderním digitálním technologiím je možno každou sekundu získávat další a další údaje popisující jevy či vlastnosti světa kolem nás. Celosvětový objem dat zpracovávaný v běžných databázích se odhadem zdvojnásobuje každých 18 až 24 měsíců [1].

Nabízí se otázka, k čemu je vlastně sběr takového objemu dat užitečný? Jaký má toto na první pohled až bezhlavé shromažďování a klasifikace údajů smysl? Ve stálé rostoucí záplavě dat není problém se rychle ztratit, čím dál tím větším uměním se stává nalézt mezi daty požadovanou informaci, nebo ji ze správného vzorku dat vyvodit. Metody zpracování a analýzy dat a následného získávání znalostí patří mezi oblasti, které jsou intenzivně zkoumány a vyvíjeny.

Samotná data ve svém obrovském objemu vyžadují od myšlení člověka vysoký stupeň abstrakce, aby si je vůbec dokázal představit. Fyzicky se jedná pouze o odchylky magnetických vrstev na plotnách pevných disků datových skladišť, které však mohou reprezentovat složitě strukturované a nekonečně dlouhé tabulky nepřeberného množství údajů.

Práce s takovýmito daty je velice náročná, jejich prezentace ve stravitelné formě klasickým způsobem (výčtem hodnot) běžnému člověku prakticky nemožná. Nejefektivnějším řešením tohoto problému je použití paradoxně nejstaršího způsobu fyzického uchování dat – obrazu. Předmětem této práce je zkoumání možností způsobu prezentace multidimenzionálních dat zobrazením na dvourozměrné obrazovce počítače.

1.1 Výhody a nevýhody vizualizace dat

Výhody grafické reprezentace dat jsou nesporné. Je zcela evidentní, že čas potřebný pro zpracování informací lidským mozkem z obrázku je nesrovnatelně menší než jejich textový popis. Jako příklad může posloužit obrázek 1.1.

Obrázek 1.1

Porovnáním s textovou alternativou informace, která může vypadat například takto: „Zelený strom“, lze navíc zjistit, že množství předávaných informací vyplývajících z grafické reprezentace je

neskonale větší. Z obrázku je evidentní počet větví, přesný odstín zeleně nehledě na to, že z textového popisu nebyl jasný ani takový základní fakt, jako jestli se jedná o listnatý či jehličnatý strom.

Analogicky při vizualizaci velkého množství dat nejpravděpodobněji v podobě grafu si lze již z letmého pohledu na takovýto graf udělat představu o jeho maximech, minimem, monotónnosti a rychlosti stoupání či klesání hodnot v závislosti na patřičných veličinách. Zatímco porovnáváním číselných hodnot lze přehlédnutím snadno dospět k chybě, tato možnost při odezírání hodnot z grafů rapidně klesá a to i při rozsáhlém objemu dat.

Operace s daty v jejich grafické podobě mohou být v závislosti na kvalitě uživatelského rozhraní vysoce intuitivní a to i při ne příliš detailních znalostech uživatele ohledně problematiky analytického zpracování dat. Rovněž výsledky těchto operací v grafické podobě mohou být velmi názorné a přímo použitelné pro prezentace těchto znalostí i pro publikum, které o složitých technologiích nutných k dosažení těchto výsledků nemusí mít ani ponětí.

Mezi nevýhody vizualizace dat patří zejména snížení online přístupnosti. Uživatelé alternativních či mobilních zařízení mohou mít se zobrazením grafiky problémy. Pravdou ovšem zůstává, že systémy pro analýzu dat nejsou určeny k volnému přístupu pro veřejnost a proto nejsou problémy tohoto typu nikterak fatální. Je však třeba mít tento aspekt na mysli, poněvadž to v některých případech může zabraňovat univerzálnímu masovému nasazení vizualizační technologie v praxi.

Další nevýhodou grafické reprezentace dat je její náročnost na paměť. Ukládání, přemísťování a zpracování grafických dat se oproti jejich textové formě exponenciálně zesložiťuje, stoupá jak náročnost na hardware, tak komplikovanost softwaru určenému k manipulaci s daty v takovéto formě. Tyto problémy se však odsunou do pozadí, pokud se vizualizovaná data budou používat pouze pro reprezentaci výstupu analytických operací.

1.2 Formulace cílů

Práce s multidimenzionálními daty a zejména jejich vizualizace patří mezi jedny z nejintenzivněji zkoumaným problematik v současnosti. Technologii OLAP používanou pro analýzu těchto zevrubně popisuje následující kapitola OLAP. V souvislosti se samotnou vizualizací je třeba specifikovat model uložení dat, o kterém pojednává kapitola 3 Cube presentation model a následně operace nad těmito daty (viz 4 Operace nad multidimenzionální kostkou). Problematika zobrazení těchto dat nebo jejich části na dvourozměrnou plochu monitoru je řešena v kapitole 5 obsahující podrobnou klasifikaci běžně používaných grafů.

Vyjasněním těchto formálních podkladů se zjistí základní principy směrodatné pro vývoj softwarové části práce, jejíž geneze je pak strukturovaně popsána v kapitole 6 Dokumentace k projektu.

2 OLAP

OLAP z angl. *online analytical processing* je zkratkou pro analytické zpracování dat. Technologii OLAP formálně popsal E. F. Codd spolu s jeho spolupracovníky a popsal 12 základních pravidel pro analytické zpracování [2].

Popis využití můžeme najít v [3]: *Typické využití je pro analýzu velkého množství údajů. Výsledkem jsou souhrny a reporty, které slouží manažerům jako podklady pro jejich rozhodování.*

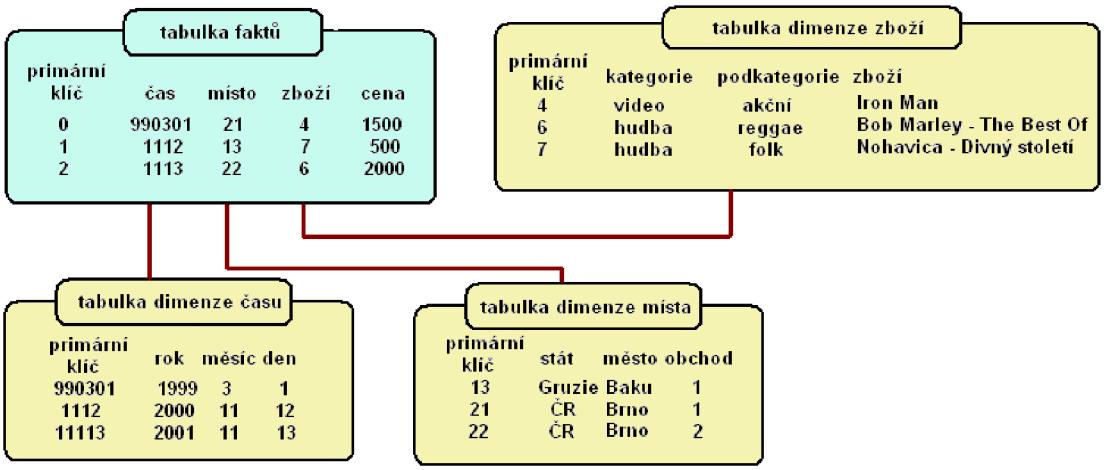
Velké množství dat potřebné pro analytické zpracování mívá často multidimenzionální strukturu.

2.1 Struktura multidimenzionální databáze

Nejčastějšími způsoby pro realizaci uložení dat v multidimenzionální databázi je hvězdicové schéma a schéma sněhové vločky [4].

2.1.1 Hvězdicové schéma

Toto schéma je tvořeno tabulkou faktů, která obsahuje cizí klíče do tabulek dimenzí, kde tento klíč odpovídá přílušným hodnotám na všech úrovních zanoření dané dimenze.



Obrázek 2.1 – Hvězdicové schéma

Tomuto schématu svou strukturou odpovídá v projektu použitý vzorek dat (viz 6.3.2 Server).

2.1.2 Schéma sněhové vločky

Je modifikace hvězdicového schématu s normalizovanými tabulkami dimenzí. Dimenze se zde mohou skládat z několika položek a tabulky dimenzí pak mohou obsahovat cizí klíče do dalších tabulek dimenzí.

Větší množství tabulek zde však má negativní vliv na dotazovací výkon.

3 CPM – Cube presentation model

První formální popis struktury uložení dat pomocí hyperkrychle (*hypercube* neboli multidimenzionální kostky) je tzv. *Cube presentation model* neboli CPM [5].

CPM se skládá ze dvou částí. První je logická vrstva, jež obstarává získávání vzorků dat na logické úrovni. Druhou je pak vrstva prezentační, jejíž součástí je datová prezentace. Toto rozdělení umožňuje odlišnou implementaci obou těchto součástí [4].

3.1 Logická vrstva

Mezi nejzákladnější prvky logické vrstvy patří [5]:

- **Dimenze** – Definovaná jako mřížka úrovní.
- **Funkce přechodu mezi úrovněmi** – Mapuje všechny prvky jedné úrovně na prvky druhé.
- **Datová množina** – Obsahuje konečný počet sekvencí prvků z množiny úrovní a mír. Tato množina vyjadřuje vztah mezi tabulkami dimenzí a tabulkami faktů. Detailní datová množina vyjadřuje tento vztah na nejnižší úrovni, tedy pro všechny datové dimenze.
- **Podmínka výběru** – Zahrnuje atomy a jejich logická spojení (konjunkce, disjunkce, negace). Atomem může být úroveň, hodnota či funkce přechodu. Detailní podmínka výběru zahrnuje úrovně na nejnižší pozici v jejich hierarchii.
- **Primární kostka** – se skládá z detailní datové množiny, detailní podmínky výběru, množinou mír (faktů), množinou úrovní a rovněž množinou operací (součet, minimum, maximum a počet).

3.2 Prezentační vrstva

Prezentační vrstva je určena pro zobrazování dat na obrazovce. Využívá k tomu několik entit:

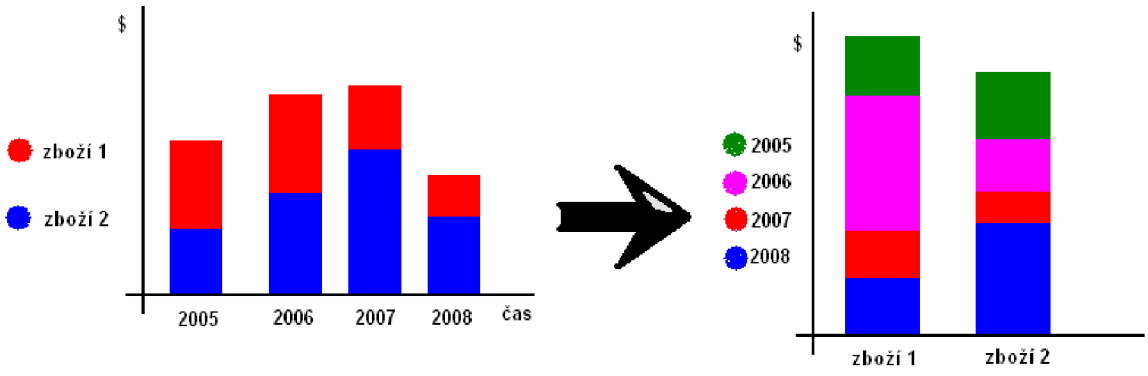
- **Bod** – Formálně se jedná o pár tvořený množinou skupin atributů (jedním z nichž je primární klíč) a množinou podmínek výběru rovnosti přislušných těmto atributům.
- **Osa** – Je množina bodů. V CPM se používají dva druhy os, *neviditelná* a *obsahová*. Neviditelná osa určuje místo pro úrovně zobrazovaných množin dat, zatímco bsahová role určuje umístění pro agregovaná data z fakt tvořících obsah hyperkrychle.
- **Hyperkrychle** – Má několik definic, pokaždé je však tvořena množinou os. Buď tyto osy obsahují všechny úrovně dané dimenze, přičemž se zde mohou vyskytnout i neviditelné a obsahové osy, nebo ji můžeme považovat za množinu fakt označených agregátovací funkcí, nebo v posledním případě jsou v definici této kostky zahrnuty všechny dimenze dat.

- **2D řez** – Představme si hyperkrychli s *K* osami. 2D řez touto hyperkrychlí je pak množinou (*K-2*) bodů, každého z jinačí osy. Jedná se vlastně o projekci os hyperkrychle do jednotlivých bodů, kromě dvou os, které pak tvoří X-ovou a Y-ovou osu grafu.
- **Pásek** – Jedná se vlastně o promítnutí jedné osy 2D řezu do jediného bodu. Výsledkem je jednorozměrná množina bodů.
- **Křížení** – je definováno jako buňka, příp. skupina buněk tvořících průnik dvou neparalelních pásků.

4 Operace nad multidimenzionální kostkou

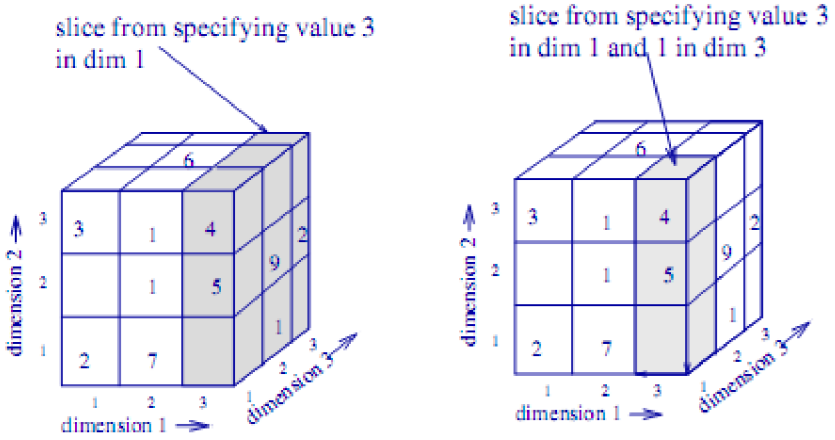
Při analytickém zpracování multidimenzionálních dat se tyto samotná data nikterak nemění. Nejpoužívanějšími operacemi jsou pak *roll-up*, *drill-down* a *pivoting*. Dalšími uživanými operacemi jsou pak např. *slice* a *dice* [6].

1. **Roll-up** – *Roll-up* znamená přesun do hrubější granularity neboli posun výše v hierarchii příslušné dimenze. Například z přehledu o údajích za jednotlivé měsíce dojde k přesunu k přehledu údajů za jednotlivé roky.
2. **Drill-down** – *Drill-down* je operace reverzní k *roll-up*. Znamená to přesun do nižší úrovně dimenze, zvětšení zanoření. Tato operace se používá k zjištění více detailních informací o datech.
3. **Pivoting** – *Pivot* je operace úzce související se zobrazením dat ve dvourozměrném prostředí. Jedná se vlastně o výběr (změnu) dimenzí zobrazovaných např. na osách grafu.



Obrázek 4.1 – Názorná transformace grafu pomocí pivotingu

4. **Slice** – Vstupem této operace je hyperkrychle o *D* dimenzích. Výstupem je hyperkrychle o (*D*-*K*) dimenzích. Je třeba specifikovat *K* hodnot (jednu pro každou z *K* dimenzí). Všechny prvky, které se v dané dimenzi nacházejí mimo zpecifikované hodnoty jsou ořezány, zbývající prvky tvoří výsledek operace.



Obrázek 4.2 – Slicing (převzato z [6])

5. **Dice** – Tato operace bývá někdy považována za synonymum pro *slice*. Dle pojetí [6] se zde vyskytuje rozdíl, a to ten, že při operaci *dice* se pro dané množství dimenzí nedefinuje jedna hodnota, nýbrž interval. Nedochází tedy zde k redukování počtu dimenzí výsledné hyperkrychle oproti hyperkrychli vstupní. Jiné zdroje však operaci definují i jinak.

5 Klasifikace grafů

Pro grafické zobrazení dat (včetně multidimenzionálních) se používají grafy. Nikoliv však v matematickém slova smyslu, nýbrž grafy jako strukturovaná grafická znázornění (reprezentace) pojmů, myšlenek, vztahů, číselných, matematických nebo statistických údajů, prostorových nebo anatomických vztahů a podobně, sloužící názornému objasnění nebo jako pomůcka v myšlenkových postupech [7]. Synonymem tohoto pojmu je diagram.

Následuje klasifikace těchto diagramů inspirovaná volným překladem z [8]. Překlad některých pojmů a částečné doplnění této klasifikace vychází z [4]. Grafy tak můžeme rozdělit na jednoduché, skupinové, skládané, kombinované, souřadnicové a ostatní speciální případy.

5.1 Jednoduché grafy

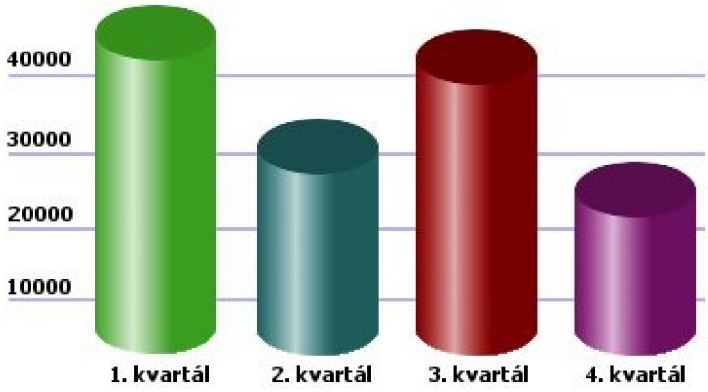
Tyto grafy vyjadřují poměr mezi dvěma veličinami. Obvykle se jedná o výčet prvků a jedné jejich vlastnosti v zájemném poměru. Použití těchto grafů při práci s multidimenzionálními daty je značně omezené, zobrazit určitou hierarchii dimenzí či více atributů jednoho prvku je prakticky nemožné.

Častokrát jsou však tyto grafy názornější než grafy složitější, poněvadž méně je někdy více a u grafických reprezentací dat to platí dvojnásobně. U těchto jednoduchých grafů se minimalizuje riziko, že při pohledu na ně se člověk „ztratí“ a vnímá pak pouze chaos barev. To je důležitý poznatek využitelný při realizace prezentací za pomoci demonstrací těchto grafů.

Jednoduché grafy můžeme rozdělit na sloupcové, řádkové, bodové, čárové a koláčové.

5.1.1 Sloupcový graf

Tyto grafy jsou všeobecně nejznámější. Obecně lze říci, že prakticky všechny jednoduché grafy zobrazují závislost veličiny na hodnotě. U sloupcových grafů to pak znamená, že na ose X můžeme najít výčet těchto veličin a na ose Y pak příslušnou hodnotu každé této veličiny. U tohoto typu grafů je grafickou reprezentací této hodnoty sloupec tvaru např. obdélníka (obdobou může být hranol, válec apod.).

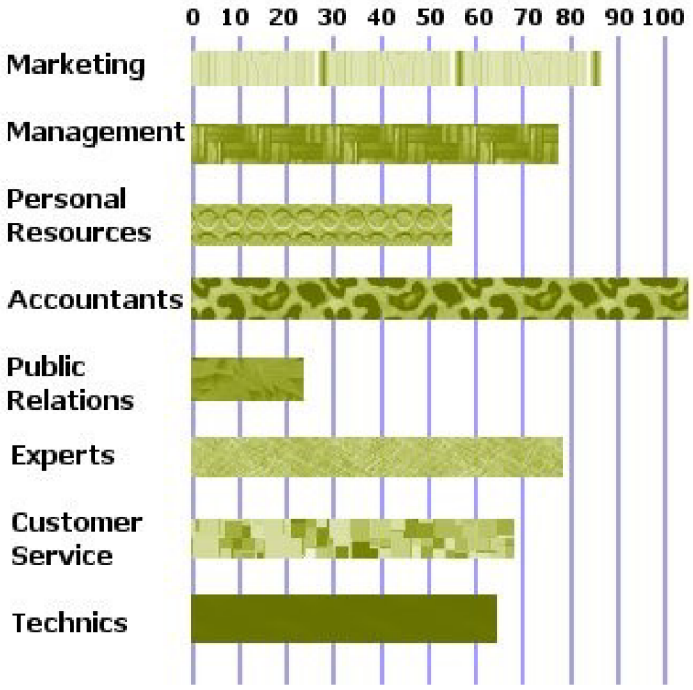


Obrázek 5.1 – Sloupcový graf

Platí tedy, že zatímco hodnoty veličin na ose Y (vyjádřené výškou sloupců) jsou většinou spojité, výčet veličin na ose X je logicky diskrétní. Samozřejmě existují grafy, kde toto neplatí a rovněž i osa X obsahuje spektrum diskrétních hodnot, v této klasifikaci však jsou takovéto grafy označovány jako souřadnicové (viz 5.2 Souřadnicové grafy).

5.1.2 Řádkový graf

V principu se jedná o grafy takříkajíc inverzní ke grafům sloupcovým v tom smyslu, že jejich osy X a Y jsou prohozeny. V praxi to znamená, že řádkové grafy obsahují na ose Y výčet diskrétních hodnot – veličin a na ose X se nachází spojitá stupnice jejich potencionálních, grafem zobrazovaných hodnot. Tato na první pohled banální záležitost se může ukázat jako fatální při řešení problému vizualizace dat, respektive při problémech spjatých s rozložením příslušných prvků vizualizace na obrazovce tak, aby se vzájemně nepřekrývali.



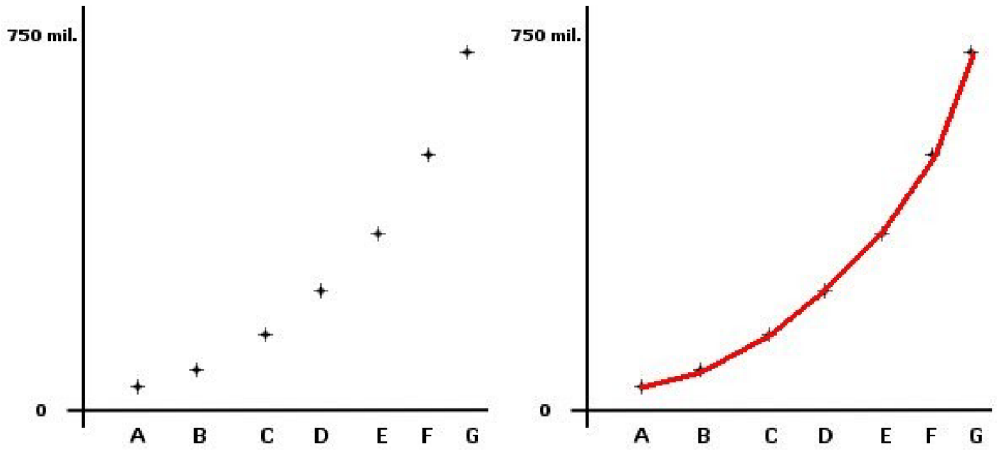
Obrázek 5.2 – Řádkový graf

Zobrazením hodnot pomocí „řádků“ (tedy ničeho jiného než obdob sloupců ze sloupcového grafu položených na bok) rozhodně usnadňuje zobrazení delších názvů diskrétních hodnot (veličin) na ose Y, jak je evidentní z obrázku 5.2.

Při inverzi os může nastat problém, kdy graf ztrácí na názornosti, poněvadž to může budit až neobvyklý až oproti každodenně běžně používaným konvencím. Problém lze proto i při použití sloupcového grafu vyřešit barevným rozlišením jednotlivých sloupců v grafu, přičemž se však k výsledné vizualizaci musí přidat jednoduchá tabulka popisující vztahy jednotlivých barev k názvům veličin. Je zde třeba však pamatovat na kontrast barev a s ním spjatou potencionální rozlišitelnost. Špatná zobrazovací schopnost monitoru či handicapovaný uživatel mohou způsobit rapidní snížení přístupnosti a tím pádem i použitelnosti softwarového produktu používajícího toto řešení.

5.1.3 Bodový a čárový graf

Tyto grafy jsou reprezentovány soustavou bodů v rovině, kde záleží na jejich souřadnicích. Každý bod svou pozicí oproti počátku v této souřadné soustavě vyjadřuje dvě věci – vzdálenost od osy X a vzdálenost od osy Y. Poněvadž jsou popisované grafy zařazeny mezi jednoduché, jedinou věcí, kterou se liší od grafů sloupcových, je grafické znázornění hodnoty veličiny. V případě bodového grafu se tedy nejedná o sloupec (hranol, válec, ...), ale pouze o bod v místě výšky potencionálního sloupce. Spojíme-li pak tyto body čarou, získáme graf čárový.



Obrázek 5.3 – Bodový a čárový graf

Čárový graf může být názornější co se týče tendence vývoje zobrazovaných hodnot, lze na něm relativně snadněji rozpoznat lokální monotónnosti včetně extrémních hodnot.

Graf spojnicový, v kterém by tyto body byly spojeny křivkou, nepatří mezi jednoduché grafy za předpokladu spojité množiny hodnot na ose X. Je-li křivka použita u diskrétní množiny hodnot na ose X, může se pak jednat o spojnicový graf, avšak jeho použití je pak zavádějící a může snadno vést k nesprávnému pochopení interpretace grafu. Proto by se měl spojnicový graf používat pouze jako graf souřadnicový.

Oproti těmto teoriím v praxi nastává často situace, kdy osa X reprezentuje čas (a to je velmi častý případ). V grafu zobrazíme hodnoty naměřené v příslušných časových intervalech a spojíme je.

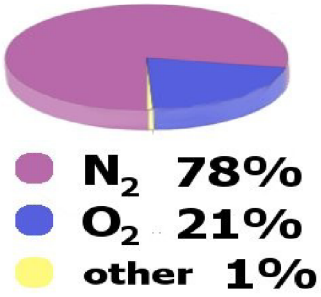
Záleží pak už jen čistě na autorovi grafu, s jakou intenzitou věří takovéto heuristice a nakolik bude považovat svůj graf za odpovídající popisu reality. Nezanedbatelné výchylky mezi ve skutečnosti diskrétními hodnotami času totiž nemusí být zdaleka nevyloučitelné. A tak se tedy rozdíl mezi jednoduchými a souřadnicovými grafy stírá a tyto dvě kategorie mohou snadno v reálném životě plynule přecházet jedna v druhou.

Variantou čárového (příp. spojnicového) grafu je graf oblastní, kdy se zobrazovanými údaji myslí celá plocha pod křivkou. Využití je zřejmé u grafů, prostřednitvím kterých chce jejich tvůrce informovat mimo jiné i o údaji o celkovém součtu hodnot za určitý časový interval.

5.1.4 Koláčový graf

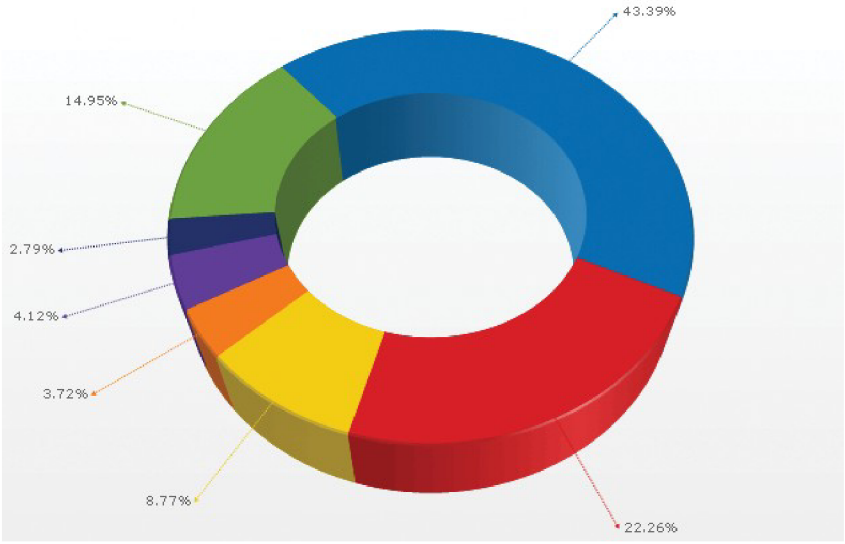
Koláčové grafy (také kruhové či výsečové grafy) jsou tvořeny kruhem či elipsou rozdělenou na jednotlivé výseče. Jsou vhodné při nutnosti poukázat na vzájemný poměr určitých hodnot, vizuálně lze totiž snadno zhruba odhadnout procentuální podíl jednotlivých složek na celku. S těmito grafy se proto lze často setkat ve statistice např. jako s výsledky průzkumu veřejného mínění či

v charakteristice podílu jednotlivých složek hospodářství či průmyslu (rovněž v kombinaci s mapovým grafem – viz níže).



Obrázek 5.4 – Koláčový graf

Variantou koláčového grafu je graf koblihový (z angl. *doughnut*), kdy graf netvoří kruh, ale mezikruží (rozdíl dvou různě velkých kruhů) opět rozdělené na tentokrát mezikružní výseče o příslušném poměru.



Obrázek 5.5 – Koblihový graf

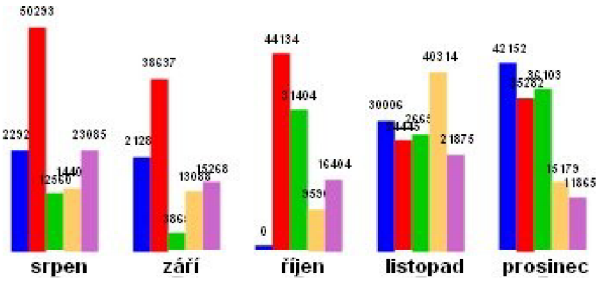
Výhody koblihového grafu oproti koláčovému nejsou na první pohled nikterak patrně zřetelné. V krajním případě by mohl být vnitřek koblihovitého grafu naplněn informacemi např. o jednotlivých zobrazovaných složkách, čímž se může ušetřit prostor kolem grafu. Přihlédnutím k praktickému využití těchto grafů např. při firemních prezentacích se však nabízí realističtější důvod pro použití koblihového grafu, a ten vyplývá přímo z jeho názvu. Při prezentaci jsou totiž posluchači častokrát více než obsahem ovlivňováni její formou. Přítomný tvar americké koblihy proto může být nezanedbatelným faktorem pro celkově kladný postoj posluchače k prezentaci.

5.2 Skupinové grafy

Velikou výhodou skupinových grafů je, že dokáží zobrazit rozdělení dat do několika kategorií. To je při práci s multidimenzionálními daty veliký přínos a proto je využití těchto grafů ve vizualizaci takovýchto dat velmi široké. Jako příklad skupinových grafů lze uvést modifikace jednoduchých grafů – sloupcového, řádkového a čárového (oblastního).

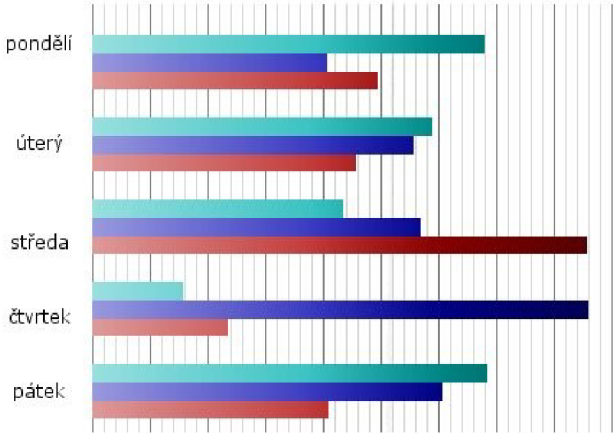
5.2.1 Skupinový sloupcový a řádkový graf

Ve skupinovém sloupcovém grafu jsou data zobrazena v podobě skupin stejného počtu sloupců, každá skupina sloupců je charakteristická pro jednu diskrétní hodnotu (veličinu) na ose X a každý sloupec reprezentuje příslušnou část (fragment) hodnoty této veličiny. Hodnotu vyjadřovanou těmito sloupci reprezentuje jejich výška odpovídající hodnotě na ose Y, která představuje jakési měřítko (osa). Lze tedy snadno porovnat jednotlivé fragmenty dat mezi sebou, celkové částky (jejich součty) odpovídající jednotlivým prvkům na ose X jsou však porovnatelné velmi obtížně. Nicméně porovnání odpovídajících fragmentů je při vhodně zvoleném barevném provedení velmi očividné a proto je využití tohoto druhu grafu při vhodných příležitostech velice účelné a efektivní.



Obrázek 5.6 – Skupinový sloupcový graf

Stejně jako v případě jednoduchých grafů, inverzí os přeměníme skupinový sloupcový graf velmi jednoduše na graf řádkový, který má stejné výhody a nevýhody oproti sloupcovému grafu jako ve své jednoduché variantě.

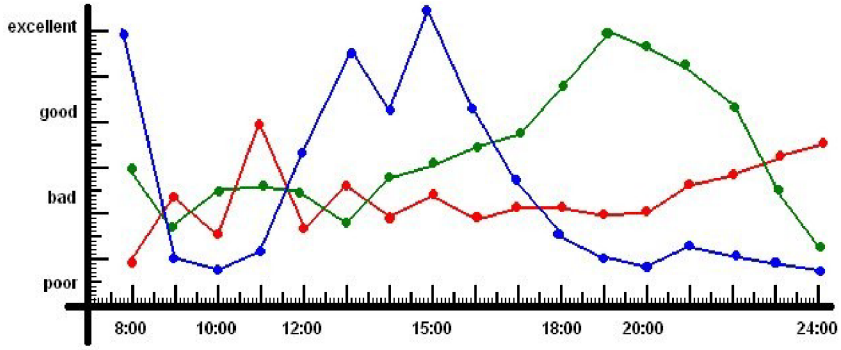


Obrázek 5.7 – Skupinový řádkový graf

5.2.2 Skupinový čárový a skupinový oblastní graf

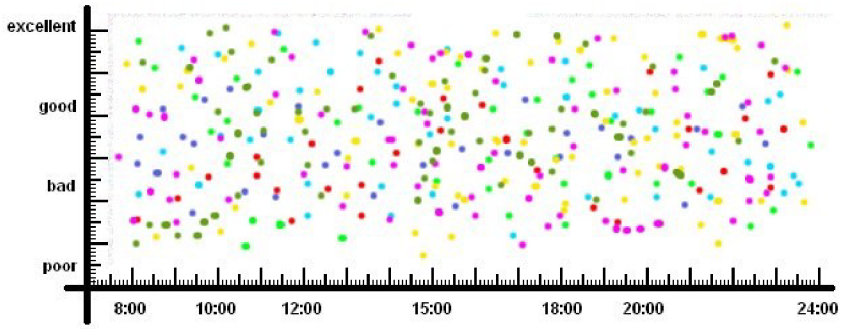
Zatímco v rámci jednoduchých grafů byl rozdíl mezi čárovým a oblastním grafem takříkajíc kosmetický, pohybujeme-li se v oblasti grafů skupinových, rozdíl mezi těmito dvěma typy grafů nabývá markantních rozdílů.

Čárový graf zde opět zobrazuje množinu veličin a hodnot jejich atributů. Rozdíl skupinové varianty oproti jednoduché spočívá v tom, že těchto atributů může být i více. Vyjádřením této hodnoty je opět pozice bodu odpovídající svou výškou příslušné hodnotě na ose Y. Spojením těchto bodů pak vznikne skupinový čárový graf.



Obrázek 5.8 – Skupinový čárový graf

Přirozeně zde existuje možnost tyto body nespojit a výsledek pak prohlásit za skupinový bodový graf. To je reálná varianta při tvorbě grafu, avšak je třeba přihlédnout ke skutečnosti, že názornost (která tvoří jeden z primárních cílů vizualizace dat) pak rapidně klesá v závislosti na zvýšování počtu zobrazovaných atributů. Proto využití skupinového bodového grafu často nemusí být v praxi příliš efektivní.

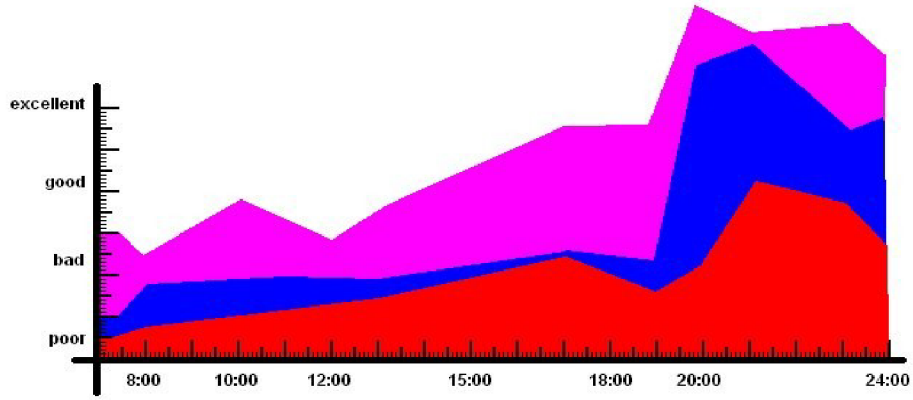


Obrázek 5.9 – Ne příliš názorný skupinový bodový graf

Skupinový oblastní graf se od skupinového čárového liší především tím, že jeho atributy jsou na sobě závislé. Pro představu stačí uvést fakt, že křivky ohraničující zobrazovanou plochu v grafu se nemohou křížit, poněvadž by došlo k jejich překrytí. Zamezit překrytí by šlo např. částečným

zprůhledněním ploch, opět při větším počtu atributů by takovýto graf ztrácel na přehlednosti a efektivitě.

Zařazením křížících se křivek do kategorie skupinových čárových grafů a z toho vyplývajícího pravidla vzájemné závislosti atributů ve skupinovém oblastním grafu pak vzniká graf uplatnitelný v praxi např. pro zobrazení vzájemných poměrů určitých veličin. Tento typ grafu pak na jednu stranu úzce souvisí s grafy skládanými (viz níže), na druhou stranu může přecházet v zobrazení spojitých souvislostí, čili v grafy souřadnicové.



Obrázek 5.10 – Skupinový oblastní graf

U skupinových grafů tvoří tedy osu X jednotlivé prvky diskrétní množiny hodnot. Grafy, které mají tuto množinu spojitou, se řadí mezi souřadnicové.

5.3 Skládané grafy

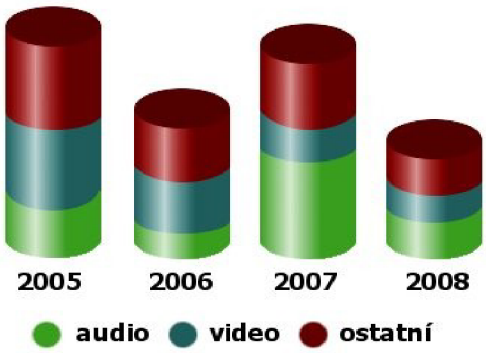
Skládané grafy umožňují zprostředkování vizualizace dat naprosto obdobných jako je tomu u grafů skupinových. Liší se snad jedině svým uspořádáním. Mezi skládanými grafy lze najít opět sloupcovou či řádkovou variantu. Navíc do této kategorie spadá specifický graf burzovní. Charakteristice zobrazení skládaných grafů se rovněž blíží nejednoduché oblastní grafy. I když byl tento graf popsán již výše jako skupinový oblastní, pravdou zůstává, že nejvalněji je používán v podobě grafu souřadnicového (viz níže).

5.3.1 Skládaný sloupcový graf

Pro představu, jak vypadá skládaný sloupcový graf, lze vycházet z jednoduchého sloupcového grafu, jehož sloupec je příčně rozdělen na několik nejlépe barevně rozlišitelných částí.

Tento graf je opět vhodně aplikovatelný pro práci s multidimenzionálními daty, může totiž velice prakticky naznačit buď strukturu dalšího zanoření v zobrazované dimenzi nebo i dimenzi zcela odlišnou od těch, které jsou reprezentovány osami X a Y. Výhodou skládaného sloupcového grafu je,

že při pohledu na něj můžeme porovnat celkový součet zobrazených atributů mezi jednotlivými prvky na ose X. Tak očividnou komparaci skupinové grafy neumožňují. Další výhodou je vizualizace vzájemného poměru těchto atributů pro konkrétní prvek. V každém sloupečku lze spatřit, zda nějaká jeho část výrazně převažuje, či zda jsou hodnoty těchto fragmentů sloupce poměrně vyvážené.

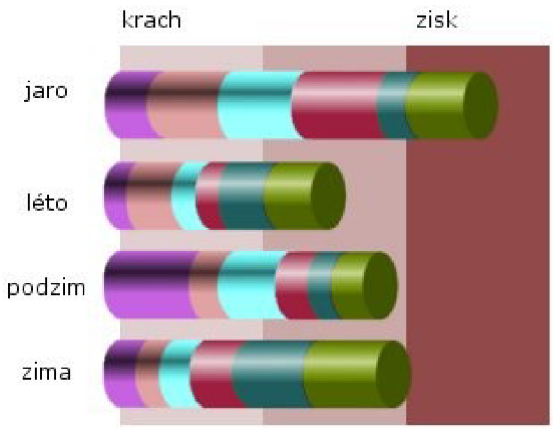


Obrázek 5.11 – Skládaný sloupcový graf

Asi největší nevýhodou skládaného sloupcového grafu je to, že jsou zde obtížně porovnatelné příslušné atributy jednotlivých prvků mezi sebou. Bezpečně porovnat při zběžném pohledu na takovýto graf lze stoprocentně pouze nejspodnější část sloupce, pozice ostatních se totiž odvíjejí právě od jeho výšky a to může jejich porovnávání značně znesnadnit.

5.3.2 Skládaný řádkový graf

Skládaný řádkový graf vznikne patřičnou modifikaci skládaného sloupcového grafu. Tento princip je objasněn u jednoduchých grafů, skupinových grafů a je v jádru identický i u grafů skládaných. Ve zkratce: Řádkový graf vznikne inverzí grafu sloupcového, tedy prohozením jeho os, přičemž hodnoty jsou reprezentovány šířkou řádků (někdy zvaných pruhů či pásků), což jsou vlastně sloupce položené na bok.



Obrázek 5.12 – Skládaný řádkový graf

Názornost tohoto grafu může oproti skládanému sloupcovému grafu kolísat případ od případu. Někdy může působit použití skládaného řádkového grafu matoucně, jindy lze pomoci jeho použití názornost umocnit. Jeho bezespornou výhodou je větší prostor pro popis veličin tvořících prvky diskrétní množiny na ose Y.

5.3.3 Burzovní graf

Moderním a v praxi používaným typem grafů svým způsobem zařaditelným rovněž mezi skládané grafy je burzovní graf.

Obrázek 5.13 – Burzovní graf (zdroj [9])

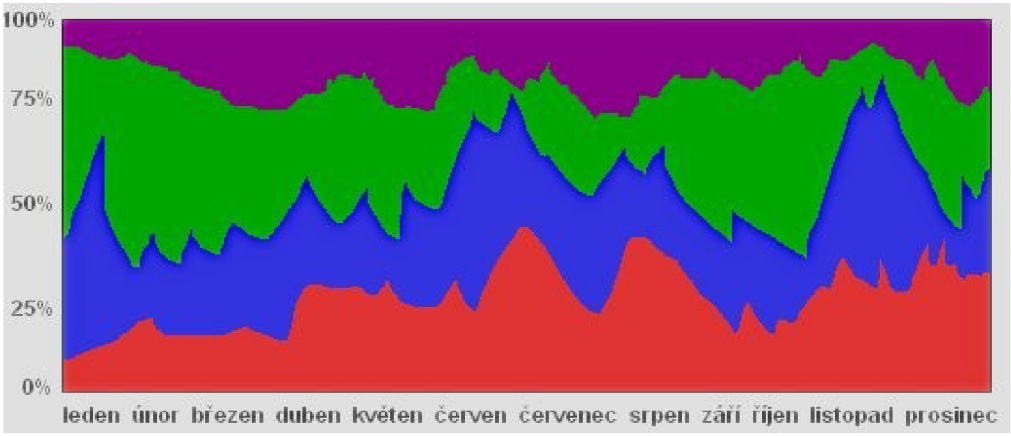
Jak je patrno z obrázku 5.13, burzovní graf zobrazuje v závislosti na čase průběh několika skutečností. Jak už název napovídá, v praxi to může být například maximánlní, minimální a výsledná cena nějakého produktu. Přirozeně použití tohoto grafu nemusí být omezenou pouze na prostředí burzy. Jedná se o graf, v kterém můžeme sledovat průběh změny několika veličin a z nich pak vyvozovat příslušné závěry ohledné tendencí a trendů, které mohou mít na tyto průběhy rozhodující vliv.

5.4 Souřadnicové grafy

Mezi souřadnicové grafy jsou obecně zařaditelné všechny grafy zobrazující vztah dvou diskrétních veličin. Lze zde proto zařadit prakticky všechny modifikace bodových či spojnicových (příp. oblastních) grafů, ale rovněž i grafy specifické pro tuto kategorii jakým je např. bublinový graf.

5.4.1 Souřadnicový oblastní graf

Princip tohoto grafu je velmi blízký skupinovému a skládanému oblastnímu grafu. V průběhu času (příp. v závislosti na jiné dimenzi) lze sledovat vzájemný poměr různých hodnot, a to buď absolutní (vyjádřený konkrétními hodnotami) či relativní (v procentech z celku).



Obrázek 5.14 – Souřadnicový oblastní graf

Osa X zde obsahuje prvky ze spojité množiny (nejčastěji času) a na nich závislé hodnoty odpovídající stupnici na ose Y (může být spojitá, ale i diskrétní). V případě, že prvky na ose X jsou z diskrétní množiny hodnot, nejedná se o graf souřadnicový, ale jednoduchý nebo skládaný (příp. skupinový).

5.4.2 Graf rozptýlení

Graf rozptýlení (z angl. *scatter chart*)vyjadřuje pomocí několika druhů bodů závislost dvou spojitých veličin, zejména pak díky koncentrací těchto bodů v určitých lze z tohoto grafu vyvozovat empirické závěry.

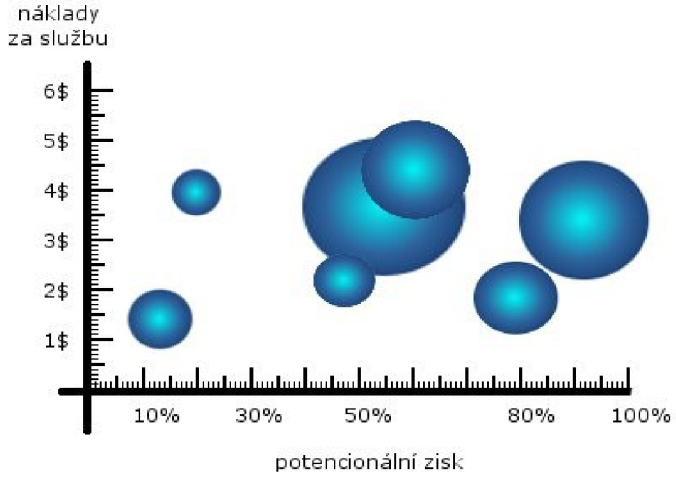
Obrázek 5.15 – Graf rozptýlení (zdroj [8])

Na obrázku 5.15 můžeme názorně vidět graf rozptýlení popisující čas odezvy serveru v závislosti na jeho zatížení. Stačí letmá znalost problematiky a pochopení významu grafu a je možné z tohoto grafu bezpečně vyvozovat závěry o výkonosti testovaných serverů.

Jak bylo uvedeno u skupinových grafů, bodový graf (jehož modifikací graf rozptýlení beze sporu je) ztrácí při velkém počtu zobrazovaných informací názornost. Proto je graf rozptýlení prakticky využivatelný pouze pro velmi omezený počet prvků (řádově 2-4).

5.4.3 Bublinový graf

Pro vysvětlení principu bublinového grafu následuje citace z výkladového slovníku [10] vysvětující pojem Metoda bublinových grafů: *Jedna z metod pro hodnocení strategického portfolia inovačních projektů, ve které se jednotlivé projekty znázorňují jako bubliny – kruhy, jejichž průměr se liší podle velikosti projektu (např. ve smyslu nákladů). Tato metoda poskytuje rychlý vizuální přehled o rovnováze projektů různé velikosti vzhledem ke kritériím v oblasti rizik a potenciálních přínosů.*



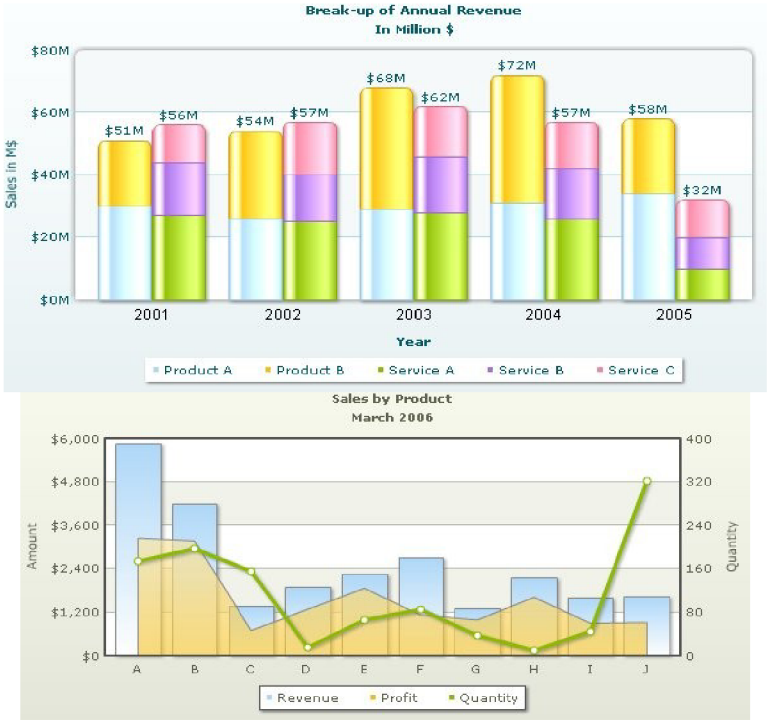
Obrázek 5.16 – Bubble chart

Jednotlivé bubliny tedy mohou svým průměrem obecně vyjadřovat jednu dimenzi, zatímco pozice jejich středu vůči osám X a Y hodnoty faktů v dalších dimenzích.

5.5 Kombinované grafy

V praxi velmi často využívanými grafy jsou grafy kombinované. Kombinovaný graf může vzniknout téměř jakoukoliv kombinací výše uvedených typů grafů. Tímto způsobem můžeme sečíst pozitiva několika různých přístupů k tvorbě grafů a na opačné straně lze takto omezit až odstranit jejich jednotlivé nevýhody.

Z toho vyplývá, že tyto kombinované grafy jsou horkými kandidáty pro jejich využití při vizuálních demonstracích, je třeba však mít na paměti, že velké množství zobrazovaných informací může mít v konečném výsledku spíše matoucí vliv.



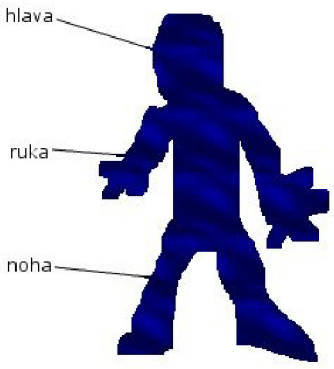
Obrázek 5.17 – Kombinované grafy (zdroj [8])

Na obrázku 5.17 nahoře lze vidět sloupcový graf vytvořený kombinací grafu skládaného a skupinového. Dole pak následuje kombinace grafů sloupcového, čárového a oblastního.

Kombinací grafů lze získat nespočetné množství různých variací, z kteréhož pak lze vybrat efektivně graf vhodný pro konkrétní prezentovanou záležitost.

5.6 Speciální grafy

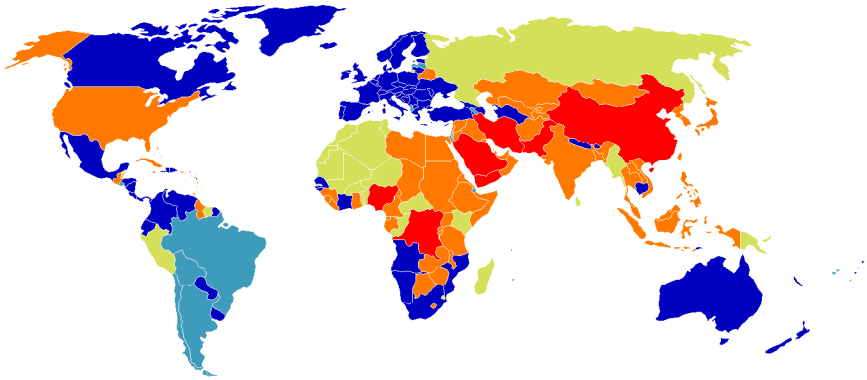
Mezi speciální grafy se řadí všechny grafy nezařaditelné do žádné z předchozích kategorií. Může se jednat o grafy mapové či stromové, ale i o další různá schémata (např. algoritmů) a diagramy včetně grafů matematických. Jelikož dle definice grafu [7] se jedná prakticky o jakékoliv strukturované grafické znázornění jakékoliv skutečnosti, může být hranice mezi grafem a jakoukoliv vizuální reprezentací dat poněkud nejasná.



Obrázek 5.18 – Strukturované grafické zobrazení - graf?

5.6.1 Mapové grafy

Mapové grafy jsou využitelné pro vizualizaci přísně teritoriálně vymezených dat vztahujících se na jistý region [4].

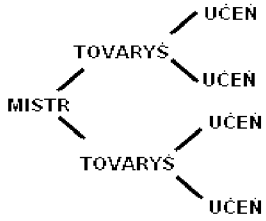


Obrázek 5.19 – Mapový graf trestů smrti ve světě. Ve státech obarvených modře byl trest smrti zrušen pro všechny delikty, ve světle modrých zrušen pro normální delikty (avšak jen mimo válečný stav), v zelenožlutých zrušen nebyl, ale za posledních 10 let nebyl uplatněn. Následují oranžové státy, v kterých je uplatňován proti dospělým a červené, kde je uplatňován i proti mladistvým.

V praxi se tyto grafy využívají velmi často v kombinaci s grafy kruhovými (ale rovněž i sloupcovými a jinými) nebo se pro reprezentaci dat využívají čísla či barvy umístěné přímo do mapy.

5.6.2 Stromové grafy

Tyto grafy zobrazují data ve tvaru stromu, což je struktura charakterizovatelná jako neorientovaný, souvislý graf (v matematickém slova smyslu), který neobsahuje žádnou kružnici [11].



Obrázek 5.20 – Jednoduchý stromový graf

Na rozdíl od matematické definice jsou stromové grafy tvořeny často orientovaným stromem. Uzly ve stromovém grafu mohou popisovat kvalitativní, kvantitativní, hierarchické či logické vazby mezi jednotlivými prvky grafu pomocí čísel, barev, symbolů či příslušným propojením hranami mezi sebou.

Stromový graf může reprezentovat schéma datové struktury, ale i např. *rozhodovací strom*, který souvisí s technologií OLAP a popisuje algoritmus pro získávání znalostí.

6 Dokumentace k projektu

Tato kapitola obsahuje slovní popis jednotlivých etap návrhu a vývoje praktické části bakalářské práce. Jednotlivé podkapitoly odpovídají příslušným etapám vývoje.

První podkapitola s názevem Specifikace požadavků charakterizuje podrobné zadání vychájící z požadavků v zadání bakalářské práce. Je zde rozebrána funkčnost programu, jeho chování, formu vstupu a výstupu, to vše s ohledem na praktickou využitelnost výsledného produktu.

Následující podkapitola Analýza se zabývá různými možnostmi, jak program realizovat. Jsou zde brána v potaz pro i proti týkající se nasazení různých technologií s ohledem na možnosti potencionálního rozšiřování a doplňování produktu v budoucnu. Kromě toho je zde zveřejněn konečný vzorec venkovního chování programu tak, jak jej vnímá uživatel. Z těchto myšlenek pak vyplývají zásady uživatelské přívětivosti směrodatné pro veškerý další vývoj.

Podkapitola Návrh řešení popisuje chování programu při pohledu zevnitř. Jedná se o slovní charakteristiku vnitřní funkčnosti programu ve stylu takovém, aby byl pochopitelný i člověku, který s technologiemi, které jsou nakonec pro realizaci projektu vybrány, nemá přílišné zkušenosti. Jinými slovy jedná se o zcela obecný popis základních algoritmů, jejichž rozšíření a implementace bude tvořit jádro programu.

Popis řešení naopak charakterizuje konečnou implementaci v použitém programovacím jazyce. Lze se zde seznámit s jednotlivými částmi programu, způsobem jejich interakce a konkrétními funkcemi, jejich vstupy a výstupy. Součástí této podkapitoly je rovněž podrobný popis způsobu práce s finálně vytvořeným uživatelským rozhraním. Zde uvedené informace odpovídají obsahu uživatelské nápovědy zabudované do programu, která se nachází v příloze.

6.1 Specifikace požadavků

Součástí zadání bakalářské práce bylo „realizujte rozhraní systému pro rozhodování jako komunikační variantu v informačním systému“. Při implementaci softwarové části práce bylo kromě této formulace přihlíženo k potencionální praktické využitelnosti projektu a funkčnosti z toho vyplývající. Z těchto údajů a dalších konzultací a indicií byl vyvozen základní požadavek na výslednou aplikaci – program musí být online dostupný.

V praxi to znamená dostupnost kdykoliv a z kteréhokoliv místa, kde se vyskytuje připojení k internetu. Tento požadavek má bezpočet dalších důvodů a výhod. Aplikace má být určena k zobrazování (vizualizaci) multidimenzionálních dat. Pokud by program pracoval v režimu offline, byla by na pracovním terminálu nutná přítomnost OLAP serveru. Vezme-li se v potaz, že v praxi se OLAP servery s multidimenzionálními daty používají pro práci s gigantickými databázovými sklady, vyskytne se hned několik aspektů, které by offline verzi tohoto produktu když už ne znehodnotili, pak

aspoň velice zdegradovali její potencionální marketingovou úroveň. U bakalářské práce možná zní tyto argumenty trochu podivuhodně, je však třeba připomenout, pro jaké účely se vůbec OLAP technologie používají. Má-li výsledný produkt alespoň budit zdání potencionální praktické využitelnosti v budoucnu (přirozeně po patřičných úpravách přibližujících projekt komerčnímu software), je třeba následnující fakta brát v potaz.

Jedním z nejpodstatnějších bodů by bylo omezení možnosti prezentace produktu. Nároky na hardware jsou u profesionálních OLAP serverů nemalé (nemluvíme nyní o nejmenovaných SQL serverech vhodných tak maximálně pro pokusy) pro většinu strojů v současnosti nesplnitelné (nebo splnitelné např. po odebrání veškerého ostatního software, což image produktu rovněž nepřispěje).

Dalším aspektem je možná potřeba možnosti využití programu na více terminálech. Stačí zde pak totiž jen jeden OLAP server zpracovávající požadavky. Omezení počtu uživatelů pak téměř mizí (v závislosti na složitosti jejich požadavků), každopádně na této hodnotě (počtu uživatelů) již nezávisí množství nutně potřebného hardware schopného provozovat jednotlivé servery. Představa, že v opačném případě by vlastně každému uživateli musel připadnout hardware dostatečně výkonný pro provoz OLAP serveru, je značně neekonomická.

Nespornou výhodou online dostupnosti je potencionální mobilní dostupnost. Moderní technologie umožňují přístup na internet prakticky odkudkoliv, přičemž v dnešní době, kdy rozdíl několika minut může na burze znamenat hranici mezi milionovými zisky a ztrátami, tvoří tento fakt další nesporný argument pro.

Všechny tyto požadavky tedy přebijí i skutečnost, že u online verze rapidně stoupají nároky na bezpečnost, roste možnost průniku neoprávněné osoby do systému a úniku citlivých dat. Proto je-li tento přístup realizován v praxi, je třeba dbát obzvláštní zřetelnost na bezpečnost, nejlépe zajistit spoluúčast odborníka na problematiku kryptologie a bezpečných přenosů dat.

Mezi další požadavky na program pro práci a vizualizaci multidimenzionálních dat přirozeně patří schopnost tyto data zobrazovat (vizualizovat pomocí grafů) a pracovat s nimi, čímž se nemyslí jejich modifikace (to není smyslem analytického zpracování), ale operace jakými jsou *roll-up* a *drill-down* (změna úrovně zanoření v jednotlivých dimenzích), příp. *pivoting* (podrobněji viz 4 Operace nad multidimenzionální kostkou).

Další požadovaná vlastnost vyplývá z předpokladu, že do budoucna může existovat určitá pravděpodobnost reálné využitelnosti programu v praxi. V tomto případě by ovšem bylo nutné provést velké množství úprav, zejména podrobněji zapracovat na uživatelském rozhraní, přenosu a exportu dat. Proto při všech dalších etapách výboje byl brán zřetel na to, aby byl výsledný produkt lehce upravitelný a modifikovatelný. Pravdou zůstává, že i v případě, kdy program zůstane jen akademickou demonstrací, tato vlastnost rozhodně neuškodí.

6.2 Analýza

Ze zadání plyne, že výsledný program tedy má být jakýmsi „rozhraním“ mezi OLAP serverem a uživatelem, které umožní uživateli na základě jeho požadavků vizualizovat patřičnou část datové hyperkostky reprezentující strukturu a obsah multidimenzionálních dat.

Data pak mohou být zaslána systému s umělou inteligencí pro dolování znalostí, což ovšem překračuje rámec této práce. Proto v tomto projektu budou data vizualizována a z nich plynoucí rozhodování bude přenecháno na empirickém úsudku uživatele (manažera či analytika). Dostatečná názornost vhodně zvoleného grafu pak bude na správnosti těchto rozhodnutí mít rozhodující vliv.

Vyskytlo se zde tedy několik spolu souvisejících otázek ohledně toho, jakými způsoby bude tedy program realizován. Je zde (jak už bylo řečeno) potřeba rozhodnout, který graf bude pro vizualizaci použit (viz 5 Klasifikace grafů). Dále je zde problém ohledně technologií určených k realizaci vizualizace, tedy vykreslení takovéhoto grafu. Nakonec je rovněž potřeba určit technologii zajišťující práci s multidimenzionálními daty, způsob jejich uložení a provádění operací s těmito daty.

6.2.1 Typ grafu

Při výběru vhodného grafu lze vycházet z kapitoly 5 Klasifikace grafů, kde je do kategorií roztříděna většina v dnešní době používaných grafů a kde je rozebrán jak jejich popis, tak klady a zápory těchto grafů a vhodnost jejich použití v konkrétních situacích. Pro efektivní zobrazení multidimenzionálních dat nejsou příliš vhodné grafy jednoduché. Naproti tomu grafy skládané či skupinové vykazují jasně, že jejich použití je více než vhodné. Jako potencionální řešení se nabízejí rovněž grafy kombinované, například graf vzniklý kombinací skupinového sloupcového a skládaného sloupcového nabývá velkého potencionálu, co se množství zobrazovaných informací týče. Obecně bychom tímto grafem mohli zobrazit až čtyři dimenze, což tomuto grafu vzbuzuje z hlediska zobrazování multidimenzionálních dat velké sympatie. Je třeba však pamatovat na názornost grafu, velké množství informací může působit matoucně. Implementace takovéhoto grafu je navíc o poznání těžší, nicméně tento graf, i pokud nebude použit, představuje do budoucna možnost rozšíření programu.

Jelikož jednu ze zobrazovaných dimenzí tvoří čas, bude zcela jistě názornější použít graf sloupcový oproti grafu řádkovému.

Sloupcový graf se jeví implementačně nejjednodušeji, proto by mu měla být dána přednost před ostatními grafy. To však nezaručuje jeho názornost. Celá otázka volby grafu by byla nejlépe vyřešena, kdyby se do implementace zahrnula možnost uživateli samotnému dát na výběr, čímž by se docílila větší univerzálnost programu a spolu s volbou barev by vznikl velice praktický a užitečný nástroj pro práci s multidimenzionálními daty.

Vzhledem k povaze a úrovni projektu však bude vhodnější tyto možnosti (*features*) zařadit do kategorie potencionálních rozšíření do budoucna.

6.2.2 Technologie vizualizace

Při tvorbě online aplikace se automaticky nabízí několik technologií schopných tvorby jakési grafiky. Jednou z nejjednodušších možností je použití hypertextového preprocesoru (PHP) pro vygenerování obrázku grafu a jeho následné zobrazení na internetové stránce. Výhoda je zřejmá, uživatel nepotřebuje mít nainstalován žádný software kromě internetového prohlížeče. Bohužel velikou nevýhodou je, že vizualizace se provádí na straně serveru. To má několik aspektů:

1. **Zátěž serveru** – Nejenom, že server je nucen provádět náročné operace s multidimenzionálními daty, ještě k tomu musí zvládat generovat obrázek, který by měl pokud možno působit uživatelsky přívětivě. S větším množstvím uživatelů pak může odezva serveru stoupat až k neúnosné míře.

2. **Zátěž linky** – Přenos obrazových dat je nesporně náročnější než přenos dat reprezentovaných textem a čísly. Takovýto přístup je v rozporu praktickým uvažováním i morálními postoji zdravého člověka, když přitom víme, že vizualizovat přenášená data je schopný i klient a nebude pak zbytečně docházet k takovému plýtvání kapacity internetového připojení.

Vizualizaci dat na klientovi lze tedy realizovat např. pomocí JavaScriptu. Ten díky DHTML (dynamickému HTML) může manipulovat s DOM (*data object model*) způsobem, jehož výsledkem může být zobrazení určité grafiky, jejíž prvky však ovšem mohou být pouze různě velké obdélníčky. Takováto grafická úprava je velice skromná. I když ji lze esteticky přiupravit různými obrázky (i průhlednými či pomocí různých dynamických efektů vylepšenými), možnost generování např. koláčových grafů se zde nikdy příliš efektivně nahradit nepodaří.

Jinou možností je použití technologie Flash schopné zobrazit vektorovou grafiku i animace. Jedná se sice o komerční software (oproti PHP či JavaScriptu), jeho přehrávač je však volně dostupný a z internetu stažitelný (krytý autoritou známé firmy Adobe, čili by neměla hrozit obava uživatele ze spyware, adware či malware), přičemž i vývojový nástroj je k dispozici zdarma (časově omezeně, ale pro vytvoření příslušné části programu dostatečně). Výhody jsou nesporné, vektorová grafika umožňuje v rukou profesionálů vytvářet doslova opravdové grafické skvosty.

Poznámka k řešení: Implementace modulu pro zobrazování grafu a komunikaci s uživatelem byla provedena ve Flashi. Přesto, že vizualizovaný graf ve výsledku postrádá avizované grafické efekty a stejného dojmu by bylo dosaženo i za použití JavaScriptu, je třeba mít na vědomí, že ve výsledku jsou potencionální možnosti rozšíření programu za použití této technologie neskonale větší.

6.2.3 Technologie OLAP

Jak už bylo řečeno výše, jednotlivé body analýzy spolu úzce souvisí a proto je zde třeba připomenout, že pro vizualizaci grafu byl použit Flash. Vyskytuje se zde totiž problematika přenosu dat mezi jednotlivými technologiemi. Na první pohled banální záležitost se může ukazát jako velice komplikovaná a chyby při návrhu funkčnosti těchto mechanik (např. předpoklad nezrealizovatelných kooperací mezi použitými technologiemi) se mohou projevit zhatěním celé dosavadní práce a nutnosti od základů strukturu programu přehodnotit.

Je tedy třeba si uvědomit, jak dostat data do Flashe. Žádný volně dostupný modul pro Flash, který ze serveru s multidimenzionálními daty na základě parametrů vytáhne požadovaná data a sestrojí z nich údaje potřebné k vykreslení grafu velice pravděpodobně neexistuje. Jelikož Flash se používá jako součást webových stránek, často generovaných pomocí PHP, nejčastějším způsobem komunikace Flashe s okolím se děje právě pomocí PHP. Proto bylo PHP zvoleno jako prostředník mezi Flashem a virtuálním OLAP serverem.

Obdobu analytické zpracování tedy může provádět samotné PHP ze svých dat uložených ve struktuře vícerozměrných polí odpovídajících hvězdicovitému schématu a Flashi pak poskytovat potřebná data. S přihlédnutím k potencionálnímu využití v praxi se tento model ukazuje jako funkční s tím, že modul PHP by výsledná data nezískával ze svých interních proměnných, nýbrž by byl odkázán na komunikaci se skutečným OLAP serverem. Podrobněji viz další kapitola Návrh řešení.

6.3 Návrh řešení

Program je tedy tvořen ze dvou součástí. Jednu tvoří kód v PHP, který na serveru simuluje analytické zpracování dat tak, že jednotlivý vstup a výstup odpovídá tomu, jako by opravdu udržoval spojení s OLAP serverem a vracel výsledky analýzy. Část druhá je ve Flashi a funguje na straně klienta, kam se stáhne v rámci internetové stránky představující potencionální rozhraní v informačním systému.

Na základě akcí uživatele se spustí ve Flashi příslušná událost, která vyšle dotaz serveru, kde se zpracuje. Navrácená data se pak ve Flashi vizualizují v podobě sloupcového skupinového grafu.

6.3.1 Klient

Flash na straně klienta zobrazuje údaje z databáze na serveru. Na ose X se nachází časová dimenze, zatímco rozdíly ve skupinách u každého prvku grafu (pro příslušný čas) odpovídají dimenzi místa (ve vizualizovaném grafu se vlastně jedná o osu Z, nicméně z hlediska implementace je odpovídající osa označována jako Y). Výška jednotlivých sloupců pak odpovídá agregaci hodnot z průsečíků těchto dvou dimenzí.

Ovládání je navrženo s ohledem na co největší jednoduchost a z toho plynoucí uživatelskou přívětivost. Prakticky se zde nachází jen dva ovládací prvky spolu s případným tlačítkem

pro nápovědu. Kromě toho se po načtení programu před prováděním samotné analýzy objeví tlačítko **start**. Kliknutím na něj se spustí vizualizace.

Prvním základním ovládacím prvkem je tedy tlačítko pro operaci *roll-up*.

Obrázek 6.1 – Tlačítko roll-up

Toto tlačítko snižuje úroveň zanoření v příslušné dimenzi, proto se v programu nachází dvakrát, jednou pro dimenzi času a podruhé pro dimenzi místa. Jelikož jsou počáteční hodnoty úrovně zanoření nulové, při startu programu jsou tyto tlačítka neviditelné a k jejich zobrazení dojde až po zanoření v příslušné dimenzi. Po klinutí na **start** dojde k vizualizaci veškerých údajů v databázi – součtu hodnot za všechny roky a ze všech míst.

Druhým ovládacím prvkem je tlačítko tlačítko pro operaci *drill-down*.

Obrázek 6.2 – Tlačítko drill-down

Toto tlačítko se zobrazí u každého prvku na ose X i na ose Z, jinými slovy umožňuje se zanořit v příslušné dimenzi a omezit zobrazená data na taková, která patří do intervalu odpovídajícího prvku, ke kterému je toto tlačítko přiřazeno. Což je vlastně popis odpovídající principu zanoření se do konkrétní dimenze. Je-li zanoření maximální, tyto tlačítka se nezobrazí.

Při kliknutí na tyto ovládací prvky je vykreslený graf smazán. Program si pamatuje parametry (čas a místo) naposled zobrazeného grafu. Vlastnostem jednotlivých tlačítek *drill-down* jsou přiřazeny hodnoty charakterizující zpřesňující parametry pro popis časových intervalů a intervalů místa faktů vizualizovaných vykreslením grafu po tom, co na ně uživatel klikne.

Naproti tomu tlačítka pro *roll-up* tyto atributy postrádají, pro snížení úrovně zanoření nejsou třeba žádné zpřesňující data. Hodnoty vyjadřující úroveň zanoření v časové dimenzi a dimenzi místa se pošlou serveru. Rovněž tak se posílají hodnoty aktuálně zobrazovaného roku, měsíce, kontinentu a státu. Tyto hodnoty jsou pro server podstatné, pouze při zanoření do příslušné úrovně (o konkrétních údajích závislosti úrovně zanoření na zobrazovaných datech pojednává následující oddíl Server). V závislosti na těchto zaslaných datech pak server vrátí klientovi údaje o počtu sloupců a „řádků“ (číslo vyjadřující počet sloupců v jedné skupině), názvy jednotlivých prvků na osách času a místa a nakonec konkrétní hodnoty sloupců v grafu. Po načtení těchto údajů klientem dojde na jejich základě k vizualizaci výsledného grafu.

6.3.2 Server

Na straně serveru je program tvořen několika soubory obsahující PHP skripty. Kromě úvodní stránky `index.php`, na které se zobrazí ve Flashi naprogramovaná klientská část, to jsou zejména pro funkčnost nezbytné 3 soubory, jeden obsahuje strukturu dimenzí (`dimenze.php`), druhý

(`fakta.php`) samotná fakta a třetí soubor (`olap.php`), který je určen pro komunikaci s Flashem, pracuje s oběma předchozími soubory. Jeho jádrem je cyklus procházející vícerozměrným polem faktů. Na základě parametrů z Flashe pak dle indexů tohoto třírozměrného pole skript roztřiďuje jeho hodnoty do dvojrozměrného pole reprezentující hodnoty jednotlivých sloupců. Třírozměrné pole faktů tedy odpovídá tabulce faktů z modelu hvězdicovitého schématu, jednotlivé indexy pak odpovídají cizím klíčům odkazujícím do tabulek dimenzí. Struktura těchto tabulek je napodobena za pomocí polí v souboru `dimenze.php`.

Pokud indexy třírozměrného pole (první index charakterizuje místo, druhý čas, třetí pak druh zboží, který však na výslednou vizualizaci nemá vliv) odpovídají parametrům zobrazovaných faktů, jsou na základě pomocných funkcí z `dimenze.php` vypočítány indexy sloupce grafu, do kterého se hodnota této položky přičte.

Názvy jednotlivých řádků, skupin sloupců spolu s jejich počtem a počtem a hodnotami jednotlivých sloupců jsou ze souboru `olap.php` předávány Flashi.

Tabulka faktů reprezentovaná třírozměrným polem v souboru `fakta.php` byla vytvořena náhodným vygenerováním pomocí skriptu, který obsahuje souboru `generator.php`. Při použití programu v praxi by samotný sběr těchto statistických dat tvořil velice komplikovanou záležitost, na hodnověrnosti těchto faktů je pak totiž závislý veškerý efekt závěrů vytvořených na základě práce s programem.

6.4 Popis řešení

Program byl implementován dle požadavků zadání s ohledem na provedenou analýzu. V této kapitole naleznete použité datové typy a stručný výpis použitých funkcí.

6.4.1 Flash

Při startu programu dojde k inicializaci pomocných proměnných, nastaví se nulová úroveň zanoření (pro jednotlivé dimenze se jedná o konkrétní proměnné `xlev` a `ylev`) a implicitní hodnoty pro rok, měsíc, kontinent a stát, které odpovídají parametrům prováděného řezu multidimenzionální kostkou. Zobrazí se tlačítko **start**. Další běh programu spočívá v událostech závislých na akcích uživatele.

Kliknutím na **start** se toto tlačítko zneviditelní a zavolá se funkce `createGraf`. Tato funkce se volá rovněž kdykoliv po stisknutí tlačítka (jedno zda *roll-up* či *drill-down*). `createGraf` vymaže graf, pokud již byl nějaký vykreslen a pomocí funkce `loadVariables` se zahájí načítání proměnných z PHP. Komunikace s PHP probíhá pomocí URL syntaxe, kdy vstupní parametry se umístí za otazník („?“), který následuje název skriptu `olap.php`. Tyto parametry (úroveň zanoření a časoprostorové intervaly) jsou ve formátu `parametr1=hodnota1&parametr2=hodnota2&parametr3=hodnota3&...`

Tento skript pak na standardní výstup vypisuje výstupní parametry (počet řádků, sloupců, jejich názvy a hodnoty sloupců) ve stejném formátu. Načítání těchto dat však Flash provádí asynchronně.

Proto je na konci funkce `createGraf` volána funkce `checkParamsLoaded`, která pomocí pomocné proměnné vypisované na závěr analytického zpracování prováděného v `olap.php` kontroluje, zda došlo již ke kompletnímu načtení dat. V opačném případě tato funkce s určitým intervalem periodicky sama sebe znovu zavolá. Jsou-li data načtena, provádí se vykreslení grafu funkcí `drawGraf`.

Tato funkce, je-li to třeba, zviditelní tlačítka pro *roll-up*, podle počtu řádků a sloupců vypočítá jejich příslušnou šířku a mezery mezi nimi. Následuje složitá kalkulace souřadnic pro popisky jednotlivých os (a pro s nimi spjaté *drill-down* tlačítka) a rovněž také pro samotné sloupečky ve grafu. Následuje vykreslení grafu, kdy se periodicky volá funkce `graf`, která navyšuje výšku příslušného sloupce a vypočítaný zlomek jeho výsledné hodnoty, dokud této hodnoty graf nedosáhne.

6.4.2 PHP

Hlavní soubor analytického zpracování dat prováděné PHP skriptem je `olap.php`. Tento soubor pro svou práci používá pomocí direktivy `include` data ze souborů `fakta.php` a `dimenze.php`. V tomto souboru se tedy prochází třírozměrné pole z `fakta.php` představující tabulku faktů. Jeho indexy odpovídají místu a času, kdy došlo k zaznamenané události (v praxi např. prodej zboží). Třetí index vyjadřuje druh zboží, na který se v této zjednodušené verzi analytického zpracování nebral ohled.

Index času představující cizí klíč do tabulky dimenze času byl zjednodušen dle vzoru YYYYMMDD (Y z ang. *year*, tedy rok. M odpovídá měsící a D dnu). Index místa se pak skládá obdobně ze tří dvojic číslic vyjadřující kontinent, stát a nakonec město, kde k zaznamenané události došlo.

Soubor obsahuje rovněž funkci *dnu_v_mesici*, která vrací počet dnů příslušného měsíce v určitém roce. Tato funkce je velmi užitečná např. při zjištění počtu skupin sloupců při maximálním zanoření v časové dimenzi.

Úroveň zanoření času (`xlev`)

hodnota	**skupiny sloupců představují**
0	žádné zanoření, 1 sloupec
1	roky, 2000-2007
2	kvartály daného roku
3	měsíce daného roku
4	dny daného měsíce

Úroveň zanoření místa (`ylev`)

hodnota	**barvy jednotlivých sloupců v každé skupině popisují**
0	žádné zanoření, 1 sloupec ve skupině („řádek“)
1	minimální zanoření, kontinenty
2	státy
3	města

Výstup z PHP vrací v URL formátu proměnné `cols` vyjadřující počet sloupců (časová dimenze), `rows`, jehož hodnota značí počet řádků (dimenze místa). Dále pak jsou to skupiny proměnných `colX` a `rowX`, kde X zastupuje index příslušného řádku či sloupce. Hodnoty těchto proměnných zde pak vyjadřují jejich názvy (důležité pro popisy os). Nakonec proměnné `cXrY` vyjadřují hodnoty konkrétních sloupců X na řádku Y (číslováno od 0).

7 Závěr

V rámci práce bylo vytvořeno rozhraní schopné online vizualizovat výsledky analytického zpracování multidimenzionálních dat. Obzvláštní ohled byl brán na co nejmenší hardwarové i softwarové požadavky uživatele. Součástí práce je rovněž výčet všech obecně používaných druhů grafů a analýza jejich vlastností, zejména vhodnosti pro reprezentaci multidimenzionálních dat. Důraz byl kladen rovněž na názornost grafů a s tím související použitelnost při prezentacích před publikem i bez hlubších znalostí databázové problematiky.

V rámci dalšího vývoje projektu je zde ponechán dostatečný prostor pro vylepšení grafické úrovně uživatelského rozhraní jakož i funkčnosti projektu s možností vizualizace více druhů grafů a změny dimenzí na příslušných osách. Pro nasazení v praxi je třeba rovněž vyřešit otázku bezpečnosti a šifrování dat přenášených po síti.

Literatura

[1] http://www.systemonline.cz/clanky/data-a-informace.htm

[2] http://en.wikipedia.org/wiki/OLAP

[3] L. Lacko: Datové sklady, analáza OLAP a dolování dat

[4] Ing. Ladislav Ruttkay: Vizualizácia dat

[5] Maniatis, A., Vassiliadis, P., Spiros Skiadopoulos, Vassiliou, Y.: CPM: A Cube Presentation Model for OLAP

[6] KASER, O. OLAP operations
http://pizza.unbsj.ca/~owen/backup/courses/OLAP-2004/olap2.pdf

[7] http://cs.wikipedia.org/wiki/Diagram

[8] http://www.fusioncharts.com

[9] http://www.mesec.cz/

[10] http://www.transcen.cz/slovnik.php?kod=M

[11] http://cs.wikipedia.org/wiki/Strom_%28graf%29

Seznam příloh

1 Manuál

Toto je uživatelská nápověda programu Vizualizace dat. Nápověda je rozdělena celkem do tří částí. Online verzi produktu najdete na http://www.hate.xf.cz. Aktuální online verzi konkrétně tohoto dokumentu pak na http://www.hate.xf.cz/help/. První část manuálu se zabývá instalací, zkopírování zdrojových kódů a Flash animace na server. V druhé je možné se dozvědět, jak program ovldat. Poslední třetina nápovědy řeší problémy, které mohou potencionálně nastat.

1.1 Instalace

Na server je třeba zkopírovat kořenový soubor `index.php`, následně pak složky `flash` (obsahující vizualizátor grafu na straně klienta), `php` (které obsahuje vzorek dat a prostředky k jeho zpracování) a `help` (pro možnost zobrazení této nápovědy).

1.2 Ovládání

Začněte kliknutím na tlačítko **start**. Následně by měl být vizualizován graf s minimální úrovní zanoření (tzn. jedním řádkem a jedním sloupcem). U každé dimenze se pak nachází tlačítko pro přechod v této dimenzi o úroveň níž.

Obrázek A.1 – Šipky pro snížení úrovně zanoření v hierarchii příslušné dimenze (vlevo) a pro zpětné vynoření (vpravo)

Kliknutí na tlačítko vyvolá vizualizaci grafu se zanořením v příslušné dimenzi a navíc i k zobrazení tlačítka pro návrat o úroveň výše. Šipka nahoru umožňuje posun v nahoru hierarchii dimenze místa, šipka vlevo pak v dimenzi času.

1.3 Problémy

Pokud se graf po kliknutí na **start** delší dobu nezobrazuje, je třeba zkontrolovat, zda je server nastaven pro správnou interpretaci php skriptů. Toho lze docílit poměrně snadno. Správnost funkčnosti php ověříme pomocí souboru `olap.php`, v kterém se náchází rozhraní pro vytvoření údajů pro graf z databáze. Stačí porovnat tento soubor ve složce php, kterému pošleme správné vstupní informace, s jeho online variantou na `hate.xf.cz`.

Příklad cesty k Vašemu souboru:

```
http://www.nazevserveru.cz/podadresar/php/olap.php?
xlev=1&ylev=1&rok=1&kontinent=1&stat=1&mesic=1
```

Adresa kontrolní online verze:

http://hate.xf.cz/php/olap.php?xlev=1&ylev=1&rok=1&kontinent=1&stat=1&mesic=1

2 Zdrojové texty

2.1 Soubor `olap.php`

```
<?php
      header("Content-Type: text/html; charset=utf-8");

      /*  Bakalarska prace - Vizualizace multidimenzionalnich dat v systemu OLAP  */
      /*  autor: Jaroslav Kupcik (xkupci03), soubor: olap.php, posl. uprava: 4.5.08  */
        include "fakta.php";
        include "dimenze.php";
        $xlev=$_REQUEST["xlev"];
        $ylev=$_REQUEST["ylev"];
        $rok=$_REQUEST["rok"];
        $mesic=$_REQUEST["mesic"];
        $kontinent=$_REQUEST["kontinent"];
        $stat=$_REQUEST["stat"];
        // funkce vraci pocet dnu v danem mesici daneho roku
        function dnu_v_mesici($mesic,$rok)
        {
          return Date("t",MkTime(0,0,0,$mesic,1,$rok));
        }
        foreach($data as $misto => $data2) // prolezeme fakta, prvni index je misto
        {
          foreach($data2 as $cas => $data3) // druhy index je  cas
          {
            $prachy=0;
            foreach($data3 as $val)
              $prachy+=$val; // hodnota konkretniho faktu v cyklu prolezani
            // podminka zda tento udaj je v pozadovanem casovem rozsahu
             if ($xlev<2 || (rok($cas)==$rok && $xlev<4) || ($xlev==4 && rok($cas)==$rok &&
mesic($cas)==$mesic)) // podminka casu
            { // nasleduje podminka zda tento udaj je v zobrazovanem prostorovem rozsahu
                 if ($ylev<2 || ($ylev<3 && kontinent($misto)==$kontinent) || ($ylev==3 &&
staat($misto)==$stat && kontinent($misto)==$kontinent))  // podminka mista
              {
                $x=x_index($cas,$xlev);   // vypocitani indexu sloupce, kam se udaj pricte
                $y=y_index($misto,$ylev);
                // echo "x: $x, y: $y <br/>";
                if (isset($sloupecek[$x][$y]))
                   $sloupecek[$x][$y]+=$prachy; // pripocitani hodnoty do spravneho sloupecku
v grafu
                else $sloupecek[$x][$y]=$prachy;
              }
            }
            //print "\$data[$misto][$cas] = $prachy<br/>"; // kontrola - zakomentovano
          }
        }
```

```
         // nasleduji nikoliv dimenze mist, jak by se na prvni pohled mohlo zdati,
         // nybrz nazvy popisu casove osy a to v zavislosti na urovni zanoreni
        if ($xlev==0) $col[0]="2000-2007";
        elseif ($xlev==1) // roky
          $col=Array("2000","2001","2002","2003","2004","2005","2006","2007");
        elseif ($xlev==2) // kvartaly
          $col=Array("leden-březen","duben-červen","červenec-září","říjen-prosinec");
        elseif ($xlev==3)  // mesice
           $col=Array("leden","únor","březen","duben","květen","červen","červenec","srpen","zá
ří","říjen","listopad","prosinec");
        elseif ($xlev==4)  // dny
          for ($i=0;$i<dnu_v_mesici($mesic, $rok+2000);$i++)
            $col[$i]=($i+1).".";
        // popisy osy mista vychazi z tabulky dimenzi, rovnez v zavislosti na urovni zanoreni
        if ($ylev==0) $row[0]="suma";
        elseif ($ylev==1) $row=$kontinenty;
        elseif ($ylev==2) $row=$staty[$kontinenty[$kontinent]];
        elseif ($ylev==3) $row=$mesta[$staty[$kontinenty[$kontinent]][$stat]];
        // vypis udaju na standardni vystup formou "promenna=hodnota&promenna2=hodnota2&..."
        // (pochopitelne pro FLASH)
        echo "cols=".count($col);
        echo "&rows=".count($row);
        // vypis popisu os
        for ($i=0;$i<count($col);$i++)
          echo "&col".$i."=".$col[$i];
        for ($i=0;$i<count($row);$i++)
          echo "&row".$i."=".$row[$i];
        // vypis hodnot sloupcu
        for ($x=0;$x<count($col);$x++)
          for ($y=0;$y<count($row);$y++)
          {
            echo "&c".$x."r".$y."=".$sloupecek[$x][$y];
            if (!$sloupecek[$x][$y])
              echo "0";
          }
        // kontrolni promenna signalizujici FLASHi kompletni nacteni dat
        echo "&done=done";
      ?>
```

2.2 Soubor dimenze.php

```
<?php
     /*  Bakalarska prace - Vizualizace multidimenzionalnich dat v systemu OLAP  */
     /*  autor:  Jaroslav  Kupcik  (xkupci03),  soubor:  dimenze.php,  posl.  uprava:
4.5.08 */

     // pole reprezentujici hierarchickou strukturu mista - reprezentace tabulky
dimenze mista
     $kontinenty=Array("Amerika","Evropa","Asie");
     $staty["Amerika"]=Array("USA","Kanada","Mexiko","Brazílie","Uruguay","Chile","
Peru");
     $staty["Evropa"]=Array("Česká
republika","Slovensko","Polsko","Německo","UK","Rusko","Francie");
     $staty["Asie"]=Array("Indie","Čína","Japonsko");
     $mesta["USA"]=Array("New  York",  "Washington",  "San  Francisko",  "Ottawa",
"Pittsburgh", "Los Angeles", "Houston", "Chicago");
     $mesta["Kanada"]=Array("Toronto",     "Québec",     "Victoria",     "Winnipeg",
"Halifax");
     $mesta["Mexiko"]=Array("Ciudad de México", "Puebla", "Veracruz", "Acapulco");
     $mesta["Brazílie"]=Array("Brasília", "Sao Paulo", "Rio de Janeiro", "Belém",
"Porto Alegre");
     $mesta["Uruguay"]=Array("Montevideo", "San José");
     $mesta["Chile"]=Array("Santiago", "Punta Arenas");
     $mesta["Peru"]=Array("Lima", "Iquitos");
     $mesta["Česká republika"]=Array("Praha", "Brno", "Ostrava", "Olomouc");
     $mesta["Slovensko"]=Array("Bratislava", "Košice", "Banská Bystrica");
     $mesta["Polsko"]=Array("Varšava", "Krakov", "Gdaňsk", "Vratislav");
     $mesta["Německo"]=Array("Berlín", "Bonn", "Frankfurt", "Hamburg", "Mnichov");
     $mesta["UK"]=Array("Londýn",     "Birmingham",     "Glasgow",     "Manchaster",
"Sheffield");
     $mesta["Rusko"]=Array("Moskva",     "Petrohrad",     "Novosibirsk",     "Omsk",
"Čeljabinsk");
     $mesta["Francie"]=Array("Paříl", "Lyon", "Le Havre", "Bordeaux", "Marseille");
     $mesta["Indie"]=Array("Dillí", "Kalkata");
     $mesta["Čína"]=Array("Peking", "Šanghaj", "Hongkong", "Macao");
     $mesta["Japonsko"]=Array("Tokio", "Jokohama", "Kóbe");
       // vraci hodnotu kontinentu faktu extrahovanou z mista
       function kontinent($misto)
       {
         return floor($misto/100);
       }
       // vraci hodnotu statu faktu extrahovanou z mista
       function staat($misto)
```

```
        {
          return floor(($misto%100)/10);
        }
        // vraci hodnotu roku faktu extrahovanou z casu
        function rok($t)
        {
          return floor($t/10000);
        }
        // vraci hodnotu mesice faktu extrahovanou z casu
        function mesic($t)
        {
          return floor(($t%10000)/100);
        }
        // vraci hodnotu dne faktu extrahovanou z casu
        function den($t)
        {
          return $t%100;
        }
         // funkce pro urceni X-oveho indexu sloupce, do ktereho se pripocte
jednotlivy fakt v zavislosti na hodnote casu tohoto faktu a urovni zanoreni
        function x_index($t,$lev)
        {
          if ($lev==0) return 0;
          elseif ($lev==1) return rok($t);
          elseif ($lev==2) return floor((mesic($t)-1)/3);
          elseif ($lev==3) return mesic($t)-1;
          elseif ($lev==4) return den($t)-1;
        }
         // funkce pro urceni Y-oveho indexu sloupce, do ktereho se pripocte
jednotlivy fakt v zavislosti na hodnote mista tohoto faktu a urovni zanoreni
        function y_index($misto,$lev)
        {
          if ($lev==0) return 0;
          elseif ($lev==1) return kontinent($misto);
          elseif ($lev==2) return staat($misto);
          elseif ($lev==3) return $misto%10;
        }
      ?>
```

2.3 Soubor generator.php

Pomoci tohoto souboru byl vygenerován 30000 řádkový vzorek dat umístěný v souboru fakta.php.

```
<?php
/*  Bakalarska prace - Vizualizace multidimenzionalnich dat v systemu OLAP  */
/*   autor: Jaroslav Kupcik (xkupci03), soubor: generator.php, posl. uprava: 4.5.08 */
  include "dimenze.php";
  // funkce vraci pocet dnu v danem mesici daneho roku
  function dnu_v_mesici($mesic,$rok)
  {
    return Date("t",MkTime(0,0,0,$mesic,1,$rok));
  }
  $pocet=$_POST['pocet'];
  if (!$pocet) // pokud nemame pocet, vytvorime formular na zadani parametru generovani
      echo 'Současná verze PHP: '.phpversion().'<br/><form method="POST" action="generator.php">
        počet položek: <input type="text" name="pocet"><br/>
        min value: <input type="text" name="min"><br/>
        max value: <input type="text" name="max"><br/>
        <input type="submit" value="OK">
        </form>';
  else for ($i=0;$i<$pocet;$i++) // v opacnem pripade generujeme
  {  // generovane hodnoty jsou plne zavisle na zadanych parametrech a udajich z dimenze.php
    $kontinent=rand(0,count($kontinenty);
    $stat=rand(0,count($staty[$kontinenty[$kontinent]])-1);
    $mesto=rand(0,count($mesta[$staty[$kontinenty[$kontinent]][$stat]])-1);
    $misto=$kontinent*100+$stat*10+$mesto;
    $rok=rand(0,7);                    $mesic=rand(1,12);
    $den=rand(1,dnu_v_mesici($mesic, $rok+2000));
    $cas=$rok*10000+$mesic*100+$den;    $zbozi=rand(1,100);
    $prachy=rand($_POST['min'],$_POST['max']);
      if (!isset($data[$misto][$cas][$zbozi])) // neopakujeme stejne polozky dvakrat
    {
      $data[$misto][$cas][$zbozi]=$prachy;
      echo "\$data[$misto][$cas][$zbozi]=$prachy;<br/>\n"; // vystup
    }
  }
?>
```

1.1 Soubor `flash.swf`

```
/* Bakalarska prace --- Jaroslav Kupcik (xkupci03)      */
/*  VIZUALIZACE MULTIDIMENZIONALNICH DAT V SYSTEMU OLAP */
/*   posledni upravy: 3. kvetna 2008                    */
maxH = 200;  // maximalni vyska zobrazovaneho sloupce
maxVal=0;    // nejvyssi hodnota (odpovidajici posleze maxH)
xlev=0;  // uroven zanoreni v dimenzi casu
ylev=0;  // uroven zanoreni v dimenzi mista
grok=1;  // aktualni rok
gmesic=1;  // aktualni mesic
gkontinent=1;  // aktualni kontinent
gstat=1;  // aktualni stat
cols=0;  // pocet sloupcu v grafu
rows=0;  // pocet radku v grafu
rozestup=80; // rozestup mezi sloupci
sirka=10;  // sirka sloupce
this.kontrola.text="Začnete kliknutím na tlačítko START"; // uvodni text
this.rollupX._visible=false; // zneviditelneni tlacitek pro rollup
this.rollupY._visible=false; // (zviditelni se pri urovni zanoreni > 0)
// pole barev jednotlivych sloupcu
colors=new        Array(0x0000FF,0xFF0000,0x00CC00,0xFFCC66,0xCC66CC,0x00CCFF,
0x996600,0xFF99FF);
// funkce pro vykreslovani grafu, zvetsuje vysku sloupce (target) az po
prislusnou hodnotu
graf = function (target) {
 pomer=maxVal/maxH; // pomer nejvyssiho sloupce a max. vysky
 //this.kontrola.text=pomer; // kontrolni text - ve vysledku zakomentovan
 if (target.sloupek._height*pomer<target.hodnota) { // pokud je sloupec nizsi
nez ma byt
        prirust= target.hodnota/pomer/30; // vypocte se prirustek
        target.sloupek._height += prirust;  // sloupec se zvetsi o prirustek
        target.sloupek._y -= prirust; // musi se posunout, aby nerostl smerem
dolu
        target.num.text = Math.round(target.sloupek._height*pomer); // zmena
hodnoty v popisku hodnoty
 }
 else target.num.text = target.hodnota; // konecna vysledna hodnota v popisku
hodnoty
 target.num._y = -1*target.sloupek._height-target.num._height;
};
xpozice = 100; // pomocna X-ova pozice, kterou vlastne na nic nepotrebujeme
xpozice_sloupcu=122; // pocatecni pozice sloupcu
```

```
     xpozice_casu=140;   // pocatecni pozice popisku casove osy (a drill down
tlacitek)
     ypozice = 300; // Y-ova pozice sloupcu
     ypozice_mista=330;  // Y-ova pozice popisku osy mista + drill down tlacitek

     // funkce, ktera vymaze predchozi graf a nacte data z PHP
     function createGraf()
     {
      done="undone"; // kontrolni promenna, zda jsou data nactena
      //   this.kontrola.text="pred:   x"+xlev+"   y"+ylev+"   r"+rows+"   s"+cols+",
rok:"+grok+"mes: "+gmesic; // kontrolni text - zakomentovan
      this.kontrola.text="Načítám...";  // nacitani
      for (x=0; x<cols; x++)  // smazani vsech sloupecku
          for (y=0; y<rows; y++)
            {
     //                   this.kontrola.text+="*";
                   eval("grafc"+x+"r"+y).sloupek._visible=false;
                   eval("grafc"+x+"r"+y).removeMovieClip();
            }
      for (x=0; x<cols; x++) // smazani osy casu
      {
            eval("textc"+x).drilldown._visible=false;
            eval("textc"+x).popisek.text="";
            eval("textc"+x).removeMovieClip();
      }
      for (y=0; y<rows; y++)  // smazani osy mista
      {
            eval("textr"+y).popisek.text="";
            eval("textr"+y).ctverecek._visible=false;
            eval("textr"+y).removeMovieClip();
      }
      // zacneme nacitat promenne z PHP, kteremu posleme prislusne parametry
      loadVariables("../php/olap.php?xlev="+xlev+"&ylev="+ylev+"&rok="+grok
+"&mesic="+gmesic+"&kontinent="+gkontinent+"&stat="+gstat,this);
      //loadVariables("../php/test.php",this); // test - zakomentovan
      // pravidelne kontrolujeme, zda jsou data nacteny
      param_interval = setInterval(checkParamsLoaded, 500);
     };

     // funkce, ktera kontroluje, zda byla nactena posledni promenna z PHP (done)
     // pokud ano, vykresli se graf, pokud ne, tato funkce se sama zavola s
prislusnym zpozdenim
     function checkParamsLoaded()
     {
         clearInterval(param_interval);
      this.kontrola.text+=".";
```

```
       done=_level0.done;
       if (done=="done")
              drawGraf(); // funkce pro vykresleni grafu
       else
              param_interval = setInterval(checkParamsLoaded, 1000);

     }

     // vykresleni grafu
     function drawGraf()
     {
       this.kontrola.text="Hotovo";//,    r"+grok+"    m"+gmesic+"    k"+gkontinent+"
s"+gstat+" x"+xlev+" y"+ylev+" cols:"+cols+" rows:"+rows;
       // zviditelni tlacitek rollup, je-li zanoreni>0
       if (xlev>0) this.rollupX._visible=true;
       else this.rollupX._visible=false;
       if (ylev>0) this.rollupY._visible=true;
       else this.rollupY._visible=false;
          // flash se zde z PHP snazi pochopit pocet sloupcu a radku v grafu, nacez
       // dojde k prislusne uprave na techto cislech zavislych sirek a rozestupu
sloupcu
       rows=parseInt(_level0.rows);
       cols=parseInt(_level0.cols);
       if (cols==12) rozestup=70;
       else if (cols>12) rozestup=50;
       if (rows==1) sirka=30;
       else if (rows<4) sirka=15;
       else if (rows<6) sirka=10;
       else sirka=6;
       y_posun=0; // pomocne promenne pro pozicovani sloupcu
       x_posun=0;
       //this.kontrola.text+=done;
       //this.kontrola.text+=";  po:  x"+xlev+"  y"+ylev+"  r"+rows+"  s"+cols;  //
kontrola - zakomentovano
       maxVal=0; // vynuluje se max. hodnota (v kazdem grafu se zjistuje znova)
       for (x=0; x<cols; x++) // vykresleni jednotlivych sloupcu
       {
              if (x>15) // graf je na dva radky
              {
                     // posun na zacatek 2. radku
                     y_posun=250;
                     x_posun=rozestup*16;
              }
              for (y=0; y<rows; y++)
           {
                 // prilepeni MovieClipu se sloupcem a jeho popiskem
```

```
                    this.attachMovie("graf",                          "grafc"+x+"r"+y,
this.getNextHighestDepth());
                    // nastaveni souradnic
                    eval("grafc"+x+"r"+y)._x  =  xpozice_sloupcu+rozestup*x-x_posun+
(-rows/2+y)*sirka;
                    eval("grafc"+x+"r"+y)._y = ypozice+y_posun;
                    // nastaveni sirky
                    eval("grafc"+x+"r"+y).sloupek._width=sirka;
                    // pripadna uprava pozice popisku
                    if (rows==1) eval("grafc"+x+"r"+y).num._x+=10;
                    // obarveni
                    barva = new Color(eval("grafc"+x+"r"+y).sloupek);
                  barva.setRGB(colors[y%8]);
                    // text popisku + pripadne nastaveni max. hodnoty
                    val=parseInt(eval("_level0.c"+x+"r"+y));
                    eval("grafc"+x+"r"+y).hodnota = val;
                    if (val>maxVal) maxVal=val;
      //this.kontrola.text+=eval("grafc"+x+"r"+y).sloupek._width;
            }
      }
      y_posun=0;
      x_posun=0;

      for (y=0; y<rows; y++) // vytvoreni osy mista
      {// prilepeni MovieClipu s popisem osy mista a drill down tlacitkem
        this.attachMovie("textik2","textr"+y,this.getNextHighestDepth());
        eval("textr"+y).popisek.text=eval("_level0.row"+y);
        // obarveni ctverecku barvou prislusnych sloupcu
        barv = new Color(eval("textr"+y).ctverecek);
           barv.setRGB(colors[y%8]);
        // nastaveni atributu tlacitka pro drill down (parametry pro PHP)
        eval("textr"+y).lkontinent=gkontinent;
              eval("textr"+y).lstat=gstat;
        if (ylev==1)
              eval("textr"+y).lkontinent=y;
        if (ylev==2)
             eval("textr"+y).lstat=y;
        if (ylev==3) // pri max. zanoreni drilldown zneviditelnime
              eval("textr"+y).drilldown._visible=false;
        // nastaveni pozice
        eval("textr"+y)._x = 70;
        eval("textr"+y)._y = ypozice_mista-35*y;
      }
      // vytvoreni casove osy
      for (x=0; x<cols; x++)
      {// popis casove osy
```

```
        if (x>15) // jsme-li na 2. radku
             {   // posuneme se na 2. radek
                    y_posun=250;
                    x_posun=rozestup*16;
             }
        // prilepeni popisu osy s drill down tlacitkem
        this.attachMovie("textik","textc"+x,this.getNextHighestDepth());
        eval("textc"+x).popisek.text=eval("_level0.col"+x);
        // nastaveni atibutu tlacitka (parametry pro PHP pri drill downu)
        eval("textc"+x).lrok=grok;
               eval("textc"+x).lmesic=gmesic;
        if (xlev==1)
               eval("textc"+x).lrok=x;
        else if (xlev==3)
               eval("textc"+x).lmesic=x+1;
        if (xlev==4) // pri max. zanoreni drilldown zneviditelnime
               eval("textc"+x).drilldown._visible=false;
        // nastaveni pozice
        eval("textc"+x)._x = xpozice_casu+rozestup*x-x_posun;
        eval("textc"+x)._y = ypozice+y_posun;
      }

    }
```

Následují úseky kódu příslušné událostem některých objektů.

Tlačítko **start**:

```
on (release)
{
       start._visible=false;
       createGraf();
}
```

Tlačítka pro *roll-up*:

```
on (release) { // po kliknuti roll upujeme
       _level0.ylev--;
       _level0.createGraf();
}
on (release) { // po kliknuti roll upujeme
       _level0.xlev--;
       _level0.createGraf();
}
```

Drill-down tlačítka:

```
on (release) { // po kliknuti nastavime parametry a drill downujeme
       _level0.xlev++;
       _level0.grok=this.lrok;
       _level0.gmesic=this.lmesic;
       _level0.createGraf();
}
on (release) { // po kliknuti nastavime parametry a drill downujeme
       _level0.ylev++;
       _level0.gkontinent=this.lkontinent;
       _level0.gstat=this.lstat;
       _level0.createGraf();
}
```

A nakonec základní prvek animace, funkce pro růst grafu:

```
this.onEnterFrame = function() { // rekurzivni volani funkce grafu pro jeho
vykresleni
 _level0.graf(this);
};
num.autoSize = "center";
```