

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ÚSTAV
INFORMAČNÍCH
SYSTÉMŮ

Pokročilé databázové systémy
PDB

Multimediální databáze

Version 0.0.0.9
Evaluation.

Petr Chmelař
Lukáš Stryka

2007

Poděkování

Děkujeme všem, kteří nás při psaní této práce podporovali, především J. Zendulkovi a A. Ženíškové.

Děkujeme také za všechny připomínky.

Organizace studijní opory

Odhadovaný čas potřebný pro studium celého materiálu:

⌚ **16 hodin +-8 hodin.**

To jsou asi 2 pracovní dny (+-1den).

Následuje odhadovaný čas rozepsaný po kapitolách včetně jejich popisu.

První kapitola je věnována letnému úvodu a historii multimediálních databází,

⌚ 1 hodina.

Ve druhé kapitole jsou uvedeny základní pojmy multimediálních databází a požadované vlastnosti systému řízení báze multimediálních dat, včetně jejich vysvětlení. Dále popis standardů MPEG pro manipulaci s multimediálními daty a popis jejich obsahu,

⌚ 8 hodin.

Třetí část je věnována standardům potřebným k dobrému porozumění současným multimediálním databázovým systémům,

⌚ 5 hodin.

(tato část není oporou přednášky doc. Zendluky na téma objektově-relační databáze)

Čtvrtá část je věnována komerčním systémům, především Oracle® *interMedia*,

⌚ 1 hodina.

Závěr je věnován shrnutí a otevřeným otázkám multimediálních databází,

⌚ 1 hodina.

Součástí této práce jsou přílohy ve formě prezentace. První z nich stručně prezentuje základní poznatky této práce; druhá slouží jako jednoduchá příručka praktické realizace multimediální databáze v Oracle® *interMedia*.

Obsah

1	Úvod.....	4
1.1	Historie	4
1.2	Normalizace.....	5
2	Multimediální databázové systémy.....	6
2.1	Databázová diverzita	6
2.2	Multimediální databáze	6
2.3	Požadavky na konstrukci MMDBMS	9
2.4	Multimédia.....	11
2.5	Formáty audiovizuálního obsahu	12
2.5.1	Obrázky	12
2.5.2	Dokumenty	12
2.5.3	Zvuk.....	12
2.5.4	Video a mediální kontejnery	13
2.6	Metadata	14
2.6.1	MPEG-7.....	15
2.7	Extrakce rysů a popis obsahu	23
2.7.1	Popis obsahu audia	23
2.7.2	Extrakce rysů z obrázků	24
2.7.3	Tvar	25
2.7.4	Pohyb a umístění	26
2.7.5	Speciální aplikace.....	27
2.8	Vyhledávání.....	28
2.8.1	Podobnostní vyhledávání	29
2.9	Indexace	30
2.9.1	K-D stromy.....	31
2.9.2	Quad stromy	31
2.9.3	R-stromy	32
2.10	Ukládání.....	32
2.11	Přenos	33
2.12	Prezentace	35
2.13	Bezpečnost a ochrana	35
2.14	Architektury multimediálních systémů	35
2.15	Aplikace získávání znalostí	36
2.15.1	Klasifikace.....	36
2.15.2	Shlukování a analýza.....	39
3	Databázové technologie a standardy.....	42
3.1	Objektově orientovaný koncept.....	42
3.1.1	Objektový model ODMG.....	43

3.1.2	Definiční jazyk ODL.....	44
3.1.3	Dotazovací jazyk OQL.....	46
3.2	Objektově relační koncept.....	46
3.2.1	SQL:1999	47
3.2.2	SQL:2003	49
3.3	SQL / MM.....	50
3.4	XML databáze	50
3.4.1	Databázové systémy s podporou XML.....	52
3.4.2	Nativní XML databázové systémy.....	52
3.4.3	Hybridní XML databázové systémy	52
3.4.4	Middleware.....	52
3.4.5	XML servery	52
3.4.6	Wrappery.....	52
3.4.7	Systémy pro správu obsahu.....	53
3.4.8	Nástroje pro dotazování nad XML.....	53
3.4.9	XML data binding nástroje	53
3.4.10	Dotazovací jazyky nad XML	53
3.5	Získávání znalostí	56
3.5.1	Generalizace strukturovaných dat.....	57
3.5.2	Agregace a aproximace v generalizaci prostorových a multimediálních dat	58
3.5.3	Generalizace na multimediálních datech uložených objektově-orientovaných databázích	59
3.5.4	Dolování asociačních pravidel z XML	60
4	Komerční databázové systémy	64
4.1	Oracle interMedia	64
5	Shrnutí.....	67

1 Úvod

Požadavek multimediálních databází vznikl postupně v mnoha odvětvích. Obchodní služby a průmysl (CAD, online prodej, video a audio na přání) potřebují skladovat a distribuovat velké množství multimediálních dat. Vzdělávání (e-learning), státní správa (bezpečnost) a zdravotnictví vyžadují více. Především je to možnost dotazovat multimediální data na základě jejich obsahu.

Multimediální systémy řízení báze dat (MMDBMS¹) poskytují podporu pro správu multimediálních dat jako doplněk tradičních databázových systémů. Proto musí obsahovat obdobné prostředky pro definici multimediálních typů dat, jejich modelování, skladování, manipulaci, dotazování, řízení přístupu, organizaci a prezentaci nezávislou na typu média, i když jejich hlavním úkolem je obvykle efektivně poskytovat multimediální data a informace, které jsou v multimediálních datech obsaženy.

Pod multimediálními (MM) daty si lze představit nestruturovaná data, která dělíme na základě způsobu percepcie informace na vizuální a audio. Mezi audio patří hudba, řeč a ostatní zvuky. Vizuální informace jsou například statické obrázky, ty chápeme jako 2D grafiku (vektorová, obchodní) a další obrázky (foto), které někdy lze reprezentovat pomocí 3D modelu. Video je sekvence (pohyblivých) obrázků, volitelně doplněná o zvuk nebo text. Textová informace může představovat vlastní kategorii, nebo ji spolu s formátovanými dokumenty doplněnými o obrázky řadíme mezi vizuální data.

Multimediální databáze (MMDBs) a jejich aplikace mají v poslední době výrazně rostoucí trend. To je dáno třemi předpoklady – rozšířením počítačů, růstem jejich výpočetní síly a paměťové kapacity, rychlostí komunikačních technologií a nízkou cenou záznamových, zobrazovacích a přehrávacích zařízení (digitální fotoaparáty, audio a video přehrávače).

1.1 Historie

Počátek multimediálních databází lze stěží specifikovat přesně. Pokud bychom je chápali na určité úrovni abstrakce jako kolekci vizuálních dat, která je nějakým způsobem organizována, mohly by jejich počátky sahát až do starověkého Egypta. V případě dalších multimediálních dat do doby vynálezu záznamu zvuku a videa, konce 19. století.

S jistotou lze říci, že multimediální databáze založené na počítačích neexistovaly dříve, než byly vytvořeny první systémy pro ovládání souborů a řízení báze dat v 60. letech. Tehdy byly stanoveny také objektově orientované přístupy (Simula). Relační model dat byl představen začátkem 70. let. Organizované skladování multimediálních dat ale nebylo prakticky realizovatelné až do konce 80. let, počátku prvních objektově orientovaných systémů řízení báze dat (Graphael, Servio, Ontos). Byla to multimediální data spolu s prostorovými, která vytvořila požadavek na sblížení relačního a objektového přístupu začátkem 90. let (UniSQL, Postgres). Perzistentní existenci objektů požadovalo mnoho výrobních odvětví (simulace, CAD, CAM) nebo medicína, kartografie a také věda, zejména fyzika, chemie a biologie.

První multimediální databázové systémy začaly vznikat koncem 80. let (ORION, JASMIN, MediaWay – viz Chuckwalla, kapitola 3.4). Jejich vývoj se výrazně urychlil v 90. letech spolu s boomem multimedií, Javy, XML a internetu na osobních počítačích

¹ MultiMedia DataBase Management System

následovaný různými ad-hoc zařízeními (digitální TV, mobilní telefon, iPod) na začátku 21. století s trvalým požadavkem na multimediální komunikaci a zábavu.

1.2 Normalizace

Těžko by si mohl někdo v Japonsku poslechnout MP3 (MPEG1 Audio layer 3) skladbu Francouzské zpěvačky, nahranou v Anglii, uloženou na serveru v USA pomocí Finského telefonu, nebýt dostatečné spolupráce mezinárodních normalizačních institucí – International Organization for Standardization (**ISO**), International Electronic Commision (**IEC**) a International Telecommunications Union (**ITU**).

Společnosti a instituce, mající nějakou roli v informačních technologiích, si uvědomily, že různé inovace, vývoj novějších a kvalitnějších produktů, případně přenos aplikace na výhodnější platformu se nemusí vyplatit vzhledem k nutné investici. Proto vytvořily v průběhu 80. let konsorcia s cílem standardizovat určitou oblast IT.

Příkladem této snahy je dotazovací jazyk pro relační databáze SQL (ISO 1987), jehož standard se průběžně vyvíjí (ISO 1992, 1999, 2003) a rozšiřuje – o správu objektů, XML a funkce OLAP, nebo o multimédia a aplikační balíčky (SQL/MM, ISO 2003).

Ne vždy proces normalizace nabude schválené mezinárodní normy. Příkladem je zmíněný SQL/MM, jehož vývoj se zastavil u statických obrázků. Nebo také první standard popisující perzistentní objekty v objektových databázích skupiny Object Data Management Group (**ODMG**, 1993-2000). Ale jejich spojení se skupinou Object Management Group (**OMG**) vyústilo ve všeobecně uznávaný standard Unified Modeling Language (UML, ISO 2005).

V multimediální oblasti jsou nejznámější skupiny Joint Photographic Experts Group (**JPEG**, ISO 1994, 1999 a JPEG 2000) a Motion Picture Expert Group (**MPEG-1**, ISO 1993, MPEG-2 1999 a MPEG-4 2004 pro kompresi videa, MPEG-7 2005 pro popis MM obsahu a MPEG-21 2007 pro jejich distribuci).

Mezinárodních norem existuje velké množství, další jsou například z oblasti komunikací (zmíněné dále v textu), nebo Slovník informačních technologií (ISO: 1999). Mimo ně jsou uznávány i jiné standardy, například otevřené (**RFC** HTTP, IP) nebo firemní standardy (Adobe PDF, MS Office).

Zamyšlení k 1. kapitole

Úvod neposkytuje příliš prostoru pro doplňující otázky, slouží pouze k seznámení se se základními pojmy a zkratkami a zjištění, že nezávislý výzkum se sjednocuje prostřednictvím norem.

Jak a proč multimediální aplikace vznikly a k čemu se využívají?

Více o multimediálních datech a databázích se dozvíte v následující kapitole.

2 Multimediální databázové systémy

Velké množství dat, různých (i multimediálních) aplikací, pracujících s rozmanitými daty jednoduše požaduje databáze, protože databáze zaručují konzistenci, souběžnost, integritu, bezpečnost a dostupnost dat. Z perspektivy uživatele poskytují databáze funkce pro jednoduchou manipulaci, dotazování a získávání vysoce relevantních informací z obrovských kolekcí produkovaných a uložených dat.

2.1 Databázová diverzita

Podíváme se na databáze z hlediska jejich diverzity – typu dat, která obsahují. Předem je jednoduše rozdělíme na strukturovaná a nestrukturovaná data.

Strukturovaná data – klasická, nejčastěji časově uspořádaná (temporální) data – text a číselné informace s časovými razítky vhodné pro informační systémy obchodních společností a institucí. Často také mívají transakční charakter, například nákup nebo platební operace. Podle způsobu použití těchto databází ještě rozlišujeme:

- **Transakční zpracování (OLTP²)** je charakteristické množstvím uživatelů a operací pro manipulaci (insert, update, delete) a vyhledávání.
- **Datové sklady a OLAP³** obsahují velké objemy (strukturovaných) dat, určených primárně pro jejich analýzu a podporu rozhodování (dolování dat).

Pro strukturovaná data byly navrženy relační databáze. Tyto databáze mohou mít volitelně objektové rozšíření, případně obsahovat odkazy na externí zdroje. Pro nestrukturovaná data jsou tyto postrelační (objektově orientované) vlastnosti nezbytné.

Nestrukturovaná data jsou speciální data, jejichž struktura není známa, nebo je složitá. Jejich příkladem jsou různé fyzikální modely částic nebo vesmíru, chemické databáze molekul, medicínské záznamy DNA a podobně. Do nestrukturovaných, respektive složitě strukturovaných dat počítáme i následující obvyklé typy dat:

- **Textová data** – data s velmi složitou strukturou. Jejich příkladem je beletrie.
- **Strukturovaný text** zahrnuje strukturované (formátované) dokumenty, webové stránky, XML dokumenty nebo zdrojové kódy programů.
- **Prostorová data** – klasická data doplněná o jejich umístění, tvar a rozměry.
- **Multimediální data** – digitalizovaná audiovizuální data (viz kapitola 2.4).

Tato data se v databázích nevyskytují samostatně. Systémy řízení báze složitě strukturovaných dat poskytují podporu pro správu těchto dat jako doplněk tradičních databázových systémů. Proto musí obsahovat obdobné prostředky pro definici nestrukturovaných typů dat, jejich modelování, skladování, manipulaci, dotazování, řízení přístupu, organizaci a prezentaci nezávislou na typu, i když jejich hlavním úkolem je obvykle efektivně poskytovat požadovaná data a přidružené informace.

2.2 Multimediální databáze

Databáze není jedinou možností, jak mít data organizována. Stejně je to i s multimediálními daty. Pokud si archivujete video z dovolené na kazetách nebo DVD a vyhovuje vám to, není důvod měnit. Pokud ale máte médií velké množství (jako

² Online Transaction Processing (OLTP)

³ Online Analytical Processing (OLAP)

televize), není jejich uchovávání (ve sklepech), správa (těžko), vyhledávání (vyčerpávající) a doručování (vlakem) příliš efektivní.

Tento problém je částečně řešitelný běžnou počítačovou technikou – pomocí diskového pole, případně zveřejněné prostřednictvím WWW nebo FTP serveru. Výsledek bude uspokojivý v případě textu nebo obrázků, ale těžko v případě (real-time) proudů audia a videa. Hlavním důvodem je omezená přenosová kapacita. Proto je výhodné použít tzv. streaming server, který zajišťuje efektivní využití přenosového pásma podle požadovaného média na přání, případně (multicast) živého vysílání a některé další funkce převzaté z databázového prostředí.

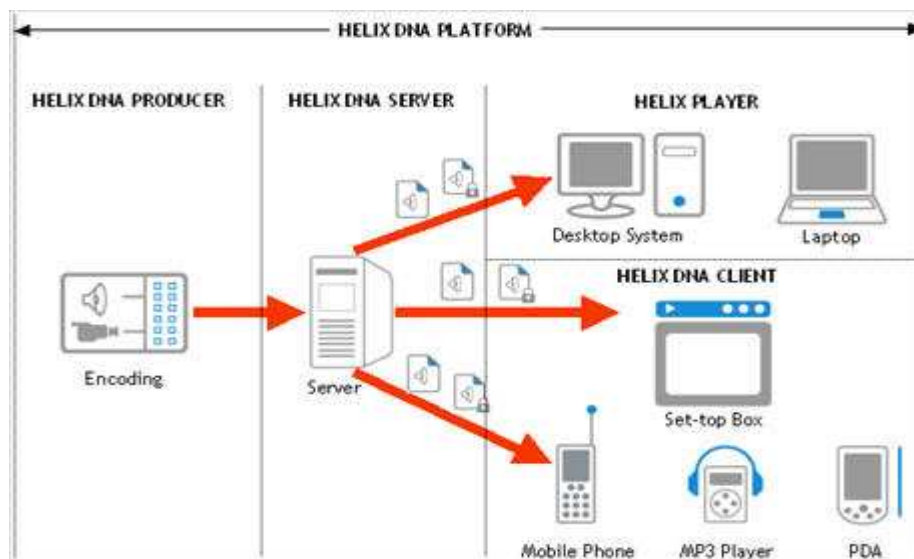


Fig. 2.1 Architektura streaming serveru (převzato z www.helixcommunity.org)

Takovýmto (Fig. 2.1) způsobem je možné provozovat televizní vysílání, video a audio na přání a podobně. Příkladem je Internetové vysílání ČT (www.ceskatelevize.cz/vysilani/), Českého rozhlasu (www.rozhlas.cz/audio/vysilani/), nebo záznamy a streaming přednášek Video serverem FIT (video1.fit.vutbr.cz) nebo množství komerčních a komunitních poskytovatelů multimediálního obsahu.

Popsané řešení se nicméně stává problematickým pro vyhledávání. Vyhledávání je stěžejní problém multimediálních dat. Vyhledání čísla, slova, případně názvu souboru je mnohem jednodušeji představitelné a realizovatelné, než vyhledání požadovaného objektu na obrázku, akce ve videu, slova nebo melodie ve zvukových datech. Vyhledávání v multimediálních datech se tedy děje výhradně pomocí metadat. Metadata jsou data o datech, jejich popis.

Říká se, že jeden obrázek vydá za sto slov. Pro popis obrázku k vyhledávání jich stačí mnohem méně. Problémem ale je, že se subjektivně liší, člověk od člověka (bracha.jpg, 8308141234.jpg). Jeho práce je navíc pro tento účel příliš drahá a neefektivní. Požadujeme tedy objektivní, automatické metody pro strukturovaný popis obsahu multimedií. Takovýto deskriptor⁴ lze získat extrakcí charakteristických rysů z média. Jak, je blíže uvedeno v kapitole 2.7.

Na základě způsobu popisu rozlišujeme dva základní typy vyhledávání:

- **Vyhledávání dle popisu dat** (description-based retrieval). Jedná se o deskripci sémantiky dat. Nějakým způsobem je nutné přiřadit název, zapsat klíčová slova, pojmenování osob, objektů, akcí, zvuků – a to nelze obecně provést žádným

⁴ Také vektor rysů nebo signatura.

automatizovaným systémem. Popis musí vytvořit člověk při jejich tvorbě (například na webu). Následný systematický popis při procházení a třídění člověkem není vůbec jednoduchý, rychlý ani levný způsob, nehledě na neúplnost a inkonsistenci způsobenou subjektivním přístupem, závislým na kontextu.

- **Vyhledávání dle obsahu dat** (content-based retrieval, CBR). Jedná se o metodu popisu syntaxe, obvykle plně automatizovatelnou. To bohatě nahrazuje její nedostatky, zejména co se týká kvality popisu sémantiky. V případě zvukových dat se může jednat o rozpoznání klíčových slov, nástrojů, not, tempa nebo i žánru. Při popisu obrazových dat, máme k dispozici dominantní barvu, histogram, hrany, textury, tvary, regiony, umístění, pohyb, popis obličeje nebo jiné biometrické informace.

Multimediální aplikace vyhledávající dle popisu dat se používají především v zábavním průmyslu pro vyhledání filmu s hledaným hercem, skladby jistého interpreta, žánru a podobně. Známé jsou Exif a ID3 popisy (kapitola 2.6). V informačním průmyslu se používá pro anotace fotografií, zvukových nahrávek a přepis mluveného slova do titulků. V průmyslu slouží například pro popis CAD/CAM výkresů, různé aplikace zasahují snad do všech odvětví. V kartografii a dalších oborech se snaží oprostit od manuální anotace snímků vzdáleného pozorování (www.mapy.cz) a používají automatickou detekci požadovaných prostorových objektů, které poté zkontrolují, případně ručně upřesní. Zajímavým projektem je NASA Earth Observing System (eosps0.gsfc.nasa.gov).

Kombinace manuálního a automatického vyhledávání je dobře použitelná v případě, že známe útržky z názvu snímku, datum pořízení, nebo klíčová slova a z jeho obsahu mohou být extrahovány další vlastnosti, například diagnóza.

Vyhledávání zaměřené na obsah má v reálném světě stále více aplikací. Důležité je v medicíně, například pro detekci anomálních tkání v různých typech snímků (kapitola 2.8), v meteorologii pro zmapování a následnou předpověď počasí. Vyhledávače obrázků na Internetu dle obsahu jsou v současné době ve stádiu vývoje, speciální aplikace ale existují v mnoha průmyslových a vědních odvětvích. Například systémy pro nalezení nových vesmírných těles fungují spolehlivě. Stejně tak rozpoznání textu (Optical Character Recognition, OCR), za předpokladu rozumné kvality média.

V případě mluveného slova – jeho přepis, identifikace mluvčího, jazyka a podobně je na světové špičce skupina Speech@FIT. V Hudebně například [Fraunhofer IDMT](http://www.fraunhofer-idmt.de).

Stále důležitější jsou také bezpečnostní aplikace. V současné době jsou schopny identifikovat člověka podle obličeje, oční duhovky, otisku prstu a dalších antropometrických vlastností. Často mají za úkol sledovat pohybující se objekty a analyzovat jejich počet, stav a chování. Ať už jde o osoby, nebo automobily – jízda na červenou, překročení rychlosti, nedovolené předjíždění, případně odcizená vozidla nebo značky včetně identifikace pachatele za volantem.

Popis na základě obsahu využívá také vlastností lidského vnímání. Zejména pro nalezení podobnosti, což je v mnoha aplikacích žádoucí. Ve vyhledávání zaměřeném na obsah se obvykle používají dva typy dotazů:

- **Předložení vzorového obrazu** (similarity search). Využívá se při tom porovnání deskriptorů extrahovaných z obrázku dotazu s deskriptory obrazů, které byly předem extrahovány a uloženy do databáze. Výsledek dotazu je například několik obrázků z databáze, jejichž deskriptory jsou dotazovanému nejbližší, a tedy by měly být tomu dotazovanému nejpodobnější.

- **Specifikace vlastností** – prvků, ze kterých je tvořen deskriptor. Jsou porovnány s těmi v databázi. Můžeme například zadat klíčové slovo, barvu, tvar (náčrtkem), specifikovat texturu, nebo parametry melodie (zanotovat).



Fig. 2.2 Dotaz pro podobnostní vyhledávání a specifikace vlastností Mony Lizy

Procesy v multimediální databázi lze zjednodušeně pochopit z ilustrace Fig. 2.3. Před uložením MM dat do databáze je provedena extrakce rysů (Feature extraction), které tvoří deskriptor (AV Description) a jsou společně uložena do databáze (Storage). Uživatelé a jiné služby mohou tuto databázi dotazovat (Search / query), výsledky dotazů prohlížet (Browse) a případně nějak filtrovat.

A to není zdaleka vše, co musí použitelný MMDBMS zajistit.

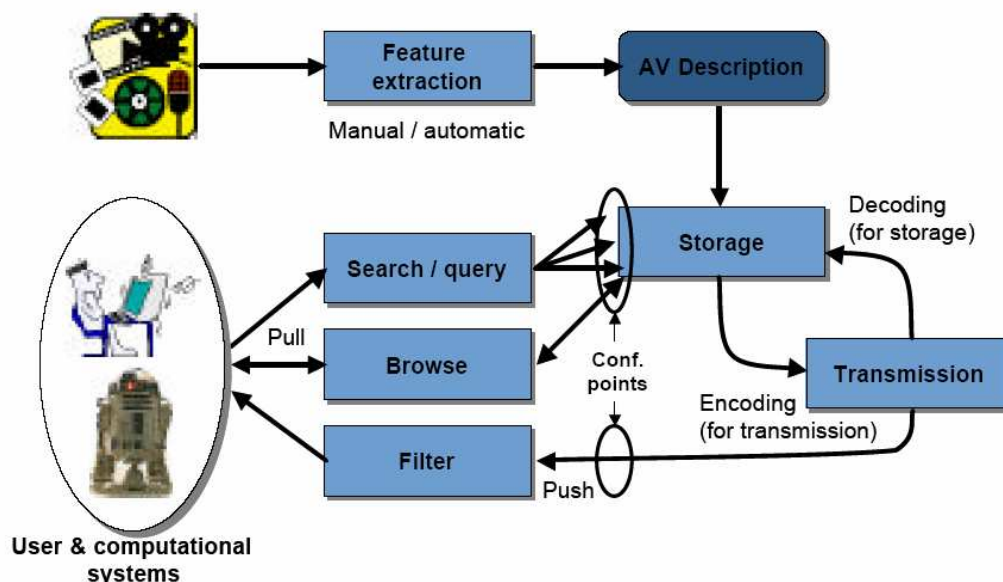


Fig. 2.3 Schéma procesů MMDB (převzato z [MP704])

2.3 Požadavky na konstrukci MMDBMS

Základní požadavky na MMDBMS jsou de facto zděděny z povahy databázových systémů a multimediálních dat. Mezi požadavky na funkčnost DBMS patří modelování,

definice a vytváření slovníku dat, manipulace s nimi, zajištění nezávislosti, bezpečnosti a integrity dat, zotavení po chybách, souběžný přístup, distribuované zpracování a zajištění co nejvyšší výkonnosti [Zen05].

Za vlastnosti zděděné z audiovizuálního povahy MM informací, které mají přímý efekt na konstrukci MMDBMS patří především rozmanitost obsahu, typů a formátů, složitost reprezentace a subjektivní interpretace, rovněž jejich velikost a temporální povaha.

Návrh kvalitního MMDBMS plyne ze splnění následujících požadavků:

- Správa různých typů vstupních, výstupních a úložných zařízení. Vstupem může být mnoho typů dat – WWW, rozmanité dokumenty, obrázky ze skenerů a fotoaparátů, pohyblivé obrázky z kamer. Zvuku z mikrofónů, syntezátorů a MIDI zařízení. Výstupem mohou být jak kvalitní obrazovky s vysokým rozlišením a hi-fi⁵ zvukem, tak jednoduché, mobilní hračky a šumítka. U některých aplikací je ještě více problémů. Například „stejně“ snímky vyhotovené pomocí magnetické rezonance, počítačové tomografie, rentgenu nebo ultrazvuku a k nim náležící textová data.
- Manipulace s množstvím různých datových a kompresních formátů. Jednotlivé MM aplikace obvykle podporují široké spektrum různých formátů. Potřebná komprese způsobuje ztrátu redundantních informací. Problémem je, že se ztratí pokaždé jiná část, respektive množství této informace – barevnost, vyšší frekvence. Například odlišné algoritmy pro změnu rozlišení nebo kompresi obrazu, zvuku a videa.
- Integrace a koexistence datových modelů. Některá data – datum, název, cena a podobné, je výhodnější uchovávat prostřednictvím relačního modelu. Oproti tomu pro dokumenty a MM data je téměř nezbytný objektově orientovaný model. Data je také často nutné uchovávat na externích úložištích.
- Podpora různých platforem a operačních systémů pro koncové uživatele. Pro vývojáře aplikací navíc plnohodnotné rozhraní pro několik programovacích jazyků pro tvorbu uživatelských funkcí (extrakce metadat, podobnostní vyhledávání), vývojových prostředí a typů aplikací (podle módy) – binární, webové atp.
- Datové proudy audia a videa mají temporální charakter. Zatímco obraz je nutné prezentovat minimálně 25 (rámců) za sekundu (lépe 30), pro zvuk je to minimálně 8 000 vzorků za sekundu (lépe 22 – 44 kHz) [MP402]. Menší rychlost způsobuje výpadky, „robotické“ zvuky a pohyby, případně zrychlený film jako za Charlieho Chaplina. Proto je nutné splnit daná QoS a real-time omezení.
- Jednotná prezentace výsledků (výsledkem může být libovolný AV obsah nebo jeho popis). Je nezbytná synchronizace z různých typů dat náležejících stejnému MM. Přehrávání pohyblivých obrázků, zvuku a titulků se musí pravidelně synchronizovat, aby se nepředcházeli (vtipné pro prezentace, příjemné pro horory).
- Ochrana autorských práv.
- Nabídka množství jednoduchých dotazů vhodných pro různé typy dat, rychlé a přesné vyhledání požadované informace. Procházení pro začínající uživatele, sofistikované dotazy pro zkušené. Stejně dotazy navíc mohou mít různou podobu – textové (název, žánr), obrazové (foto, náčrtek) nebo zvukové a to specifikací vlastností (melodie) nebo podobným příkladem (ukázka jiné písně).
- Musí být odlišena exaktní povaha parametrů klasických dat (rodné číslo) se subjektivní povahou multimediálních dat (obličej) a to pro každý tip audiovizuálních dat i jejich popis. Liší se navíc podle aplikace. Výsledek podobnostního hledání na dotaz (náčrtek): Klíčové slovo „Zub.“ je pro zubaře vyloženě nanic.

⁵ High fidelity – vysoká věrnost zvuku

- Vestavěné podobnostní vyhledávání (podle obsahu) musí korespondovat s lidským vnímáním a pojetím podobnosti. To je ale pro každý typ dat odlišné, například barva, tvar, textura, tón, barva hlasu a podobně. Je nutné stanovit jejich vzdálenost.
- Podobnostní vyhledávání ve smyslu časoprostorových dat. Prostorové – topologické a vzdálenostní dotazy (2m vlevo od), temporální dotazy (úsek, kdy zmizel) a jejich kombinace (kde se vyskytoval objekt, který se právě objevil v jiné kameře).
- Je nutné nalézt model dat, tj. vztah mezi syntaxí (popis obsahu, low-level) a sémantikou (význam, high-level) audiovizuálních dat. Také uchování modelů a jejich automatická extrakce z low-level vlastností musí být obsažena v databázi.

Zatímco některé problémy jsou v současné době uspokojivě vyřešeny, jiné jsou stále ve stádiu výzkumu, nebo pouhé vědecké fantazie (sci-fi). Například pro poslední bod platí, že čím speciálnější je aplikace, tím je výsledek uspokojivější. Vyhledání snímků s planetou Saturn, pořízených kupříkladu v srpnu 2006, v databázi hvězdárny, může mít 100% úspěšnost. Oproti tomu nalezení jediného snímku, podobného nalezenému, je na Internetu takřka nemožné.

2.4 Multimédia

Pod multimediálními (MM) daty si lze představit nestrukturovaná data, která dělíme na základě způsobu percepce informace na typ vizuální a audio. Člověk ale nevnímá pouze očima a ušima, ale především mozkiem. Z toho plyne jejich další dělení.

Vizuální data – související se zrakovým vnímáním, jsou následující:

- 2D (planární)
 - Vektorová grafika – data je možné převést na křivky, plochy a podobně (linky, loga, náčrtky, mapy), i když mohou být uchována v podobě bitmapy.
 - Text je zvláštní druh vektorové grafiky. Každý znak má svůj význam, je nutné pracovat s kontextem. Někdy bývá zařazen jako vlastní kategorie, ale to porušuje konzistenci s formátovanými dokumenty.
 - Rastrové obrázky (fotky) nemá smysl reprezentovat pomocí vektorů, ale pomocí (3D) objektů a vztahů mezi nimi.
- 3D (prostorové)
 - Modely – geometrické modely reálných objektů (koule, postava).
 - Objekty – reálná instance modelu s jeho vlastnostmi získanými z (2D) rastrů.
 - Ne-objekty – některé oblasti obrázků nejsou objekty (moře, obloha).
- 3-4D (časoprostorové)
 - Pohyblivé obrázky – časový sled obrázků vytváří iluzi pohybu (video bez zvuku).

Audio data se týkají slyšitelnosti:

- Hudba – odlišujeme speciální harmonické vlastnosti zvuků (zvonění).
- Řeč (howk).
- Zvuky nezařazené modelujeme podobně jako obrázky (výbuch).

Pokud pomineme některé speciální případy (genom), pak každý multimediální obsah spadá plně, nebo po částech, do jedné nebo více uvedených kategorií. Například webová stránka obsahuje formátovaný text, (vektorovou) grafiku, rastrové obrázky, ale i videosekvence – pohyblivé obrázky se zvukem (kombinace řeči a hudby).

Jejich kombinace se nazývají audiovizuální (AV) data.

2.5 Formáty audiovizuálního obsahu

Každý digitalizovaný (AV) obsah potřebuje (standardně) definovaný formát pro správnou interpretaci digitálního kódu, který tento kód nemůže(?) nést sebou. Jejich dělení vyplývá ze způsobu (jednoduché) reprezentace pomocí digitální techniky. Často je jeho prezentace nejednotná – různé programy pro každý typ, někdy i formát. Nicméně pro univerzální klienty MMDBMS umožňující prohlížení a tvorbu MM dat, by se měly rozlišovat pouze dva typy – vizuální a audio. Metadata je případně možné reprezentovat jako vizuální data.

2.5.1 Obrázky

Statické 2D obrázky se pro jednoduchost reprezentace dělí na bitmapové (rastrové) a vektorové.

V bitmapě grafice je obraz popsán jednotlivými barevnými, respektive jasovými, hodnotami bodů (pixelů), uspořádaných do mřížky. Jejich zástupci jsou soubory JPEG, PNG, GIF, TIFF, BMP.

Tato reprezentace způsobuje značnou redundanci, proto je výhodné využít nějaký typ komprese. Komprese by měla být bezztrátová, například PNG používá DEFLATE ([RFC 1951](#)). Nicméně lidský mozek citlivý více na jas než na barvu a nižší frekvence než vysoké, proto je možné použít (do jisté míry) ztrátovou kompresi. Jejím zástupcem je především JPEG (Joint Photographic Experts Group), využívající dominantní koeficienty diskrétní kosinové transformace (DCT) makrobloků (8x8 px) obrázku v barevném prostoru $YCbCr$ a jejich následnou bezztrátovou RLE (Huffmanovu) kompresi. Další kompresní algoritmy a bitmapové formáty jsou ISO/IEC/ITU-T: JPEG, JPEG 2000, JBIG, nebo PNG, GIF, TIFF, PCX, TGA, BMP, WMP, ILBM, PCX, PSD, ICO.

Vektorová grafika je reprezentována geometrickými útvary, nejčastěji přímkami, křivkami a plochami. Při změně měřítka se neztrácí jejich kvalita. Například SVG, PS, EPS, WMF, AI, CDR, DWG, DXF, HPGL, DVI.

2.5.2 Dokumenty

Dokumenty obsahují více než samotný prostý text (TXT, MIME). Příkladem možných typů dokumentů jsou ODT, RTF, DOC, PDF, TeX, PS, MIF, SGML, HTML, XML. V těchto dokumentech jde kromě kódování textu (ISO/IEC 8859 ASCII, ISO/IEC 10646 Unicode) dále o formátování, strukturu nebo umístění textu, který je doplněn o další multimediální obsah. Existuje velké množství dalších, méně obvyklých dokumentů, například pro publikování (DTP), nebo vývoj (C++, CAD/CAM).

2.5.3 Zvuk

Zvuk je digitalizovaný (1D) audio signál. Je reprezentován diskrétními vzorky v časové nebo frekvenční doméně.

Také zvuk obsahuje redundantní informace, bezztrátová komprese ale dosahuje jen 30 – 50% úspory. Nicméně mozek nedokáže, nebo nepotřebuje, všechna data zpracovat. Citlivost lidských uší je (menší než) pásmo 20 Hz – 20kHz. Nosná frekvence hlasu je 50 – 250 Hz, ale v telefonních sítích se používá pouze pásmo harmonických frekvencí 300 - 3400 Hz, které v mozku vyvolávají dojem, že slyší i nosnou frekvenci [Bak87]. Obdobně je to se vzorkovacími frekvencemi, udává se, že by měla být alespoň 48 kHz (CD kvalita je 44,1 kHz), ale pro 4 kHz hlasové pásmo je, dle vzorkovacího teorému, dostatečná vzorkovací frekvence 8 000 vzorků za sekundu [Čer05].

Kodeky digitálních signálů pro telekomunikační účely jsou vyvíjeny v rámci ITU (International Telecommunication Union) od roku 1972 G.711 (Siren), následovaly G.721 (ISDN) a G.723, nahrazené G.726 (ADPCM, 1990), G.728 (od 2 kbps, zpoždění 5 vzorků) a G.729 (pro VoIP).

Populární zvukové formáty jsou většinou odvozeny od zvukových stop standardních multimediálních kontejnerů (viz kapitola 2.5.4) i když byly vyvinuty zvlášť (Fraunhofer Institute, <http://www.iis.fraunhofer.de/amm/>).

Příkladem je formát MPEG-1 Audio Layer III (MP3) je komprimován prostřednictvím rozdělení na 32 frekvenčních (MDCT) pásem, ze kterých jsou dle psychoakustických poznatků a dané vrstvy (Layer I - III) vybrány dominantní koeficienty. Kompresní poměr při vzorkovací frekvenci 44100 Hz je 1:5 – 1:11. Současné MP3 používají specifikaci z MPEG-2, resp. MPEG-2.5 (nestandardizováno) pro nízké datové toky.

Vylepšením je zvuk AAC (Advanced Audio Codec) pro datové toky od 2kbps, kompresní poměr pro hlas je tedy až 1:700. Primárně jej využívá až MPEG-4 i když jeho starší verze byla specifikována již v rámci MPEG-2.

Přehled dalších formátů je možné najít například v [Hof03]. Jsou to FLAC, iLBC, RealAudio, WMA, Speex, Vorbis, AIFF, AC3. Popis formátů je v případě volné dostupnosti na stránkách jejich tvůrců (MPEG, FLAC, iLBC). Trendem v komerčních společnostech (RealNetworks, Apple, Dolby Laboratories, Microsoft) je, mimo neustálého vylepšování nosných formátů a software (zařízení) pro jejich prezentaci, poskytovat také multimediální obsah spolu s nástroji pro jejich tvorbu a distribuci.

2.5.4 Video a mediální kontejnery

Video jsou audiovizuální data s temporálním charakterem. To znamená pohyblivé obrázky se zvukem. MM formáty, které nejsou omezeny pouze na tyto data, obsahují navíc alespoň titulky, nebo jiná MM data se někdy nazývají (multi)mediální kontejnery (media container).

Video data obsahují obrovské množství informace, z velké části nadbytečné. Základní idea komprimace je tedy odstranit redundanci v rámci (video)rámčů i mezi nimi. Přední kompresní algoritmy byly vyvinuty v rámci spolupráce skupiny odborníků na pohyblivé obrázky MPEG (Moving Picture Experts Group).

Původní formát MPEG-1 (1991) využíval pro odstranění redundance jednotlivých snímků algoritmus JPEG a mezi rámci popisuje především změny, následujícím způsobem. Každých ca. 15 snímků je plný obrázek zvaný I-Frame a mezi nimi jsou popsány případné změny obrazu pomocí pohybu makrobloků (16x16), dopředné prostřednictvím P-Frame a obousměrné B-frame. Datový tok MPEG-1 byl při rozlišení 352 x 288 px (PAL, 352 x 240 px NTSC) asi 1,5 Mbps a dosahoval kompresního poměru 1:50 - 1:100 [Kos03]. Jeho nástupce, MPEG-2 (1996) byl navržen pro digitální televizní a rozhlasové vysílání prostřednictvím satelitů, pozemních vysílačů nebo Internetu.

MPEG-4 (1998) využívá kódování objektů (object coding, jako JPEG2000). Objekty mohou být video, audio nebo 3D (mřížkové) objekty. MPEG-4 umožňuje specifikovat důležitost objektů, například pro zajištění QoS a zavádí mnoho dalších vylepšení kompresních algoritmů pro datový tok od 5 kbps po 1,5 Gbps pro studia, nebo AAC.

Standards MPEG definují pouze způsob dekomprese, ale v rámci skupiny vyvíjejí i referenční software. Návrh standardů MPEG je blízký digitálním telefonním standardům ITU, která je formálně přizpůsobuje telekomunikačnímu prostředí. H.261 je příbuzné MPEG1, H.262 MPEG-2, mírně vylepšuje H.263 a H.264 obsahuje stejné části (co se týká kompresního formátu) jako MPEG-4.

Další zástupci formátů videa a MM kontejnerů jsou AVI, AU, ASF, Indeo, Ogg Media, Matryoshka, QuickTime, RealMedia, 3GP, WMV. Mediální kontejnery obsahují mimo vlastního obsahu ještě jeho popis.

2.6 Metadata

Metadata popisují (multimediální) data. Nejen digitálních knihovny vyžadují přesný strukturovaný popis každé uchovávané jednotky [ODL05]:

- Popisná metadata – informace popisující zdroj, jsou viditelná uživateli a slouží především pro vyhledávání. Dle perspektivy uživatelů jsou to [Kos03]:
 - Tvůrce obsahu vloží bibliografické informace, jako je autor, název, shrnutí.
 - Poskytovatel zdroje vloží další metadata, která usnadní nalezení média v jejich kolekci. Například klíčová slova, popis obsahu, index a podobně.
 - Konzument může přidat vlastní poznámky, případně požadavky (na datový tok).
- Administrativní (technická) metadata jsou data nezbytná pro uchování, správu a následné zobrazení obsahu. Obsahují například způsob a čas pořízení, kódování, formát a velikost obsahu, nebo informace o jeho zabezpečení. Jsou záležitostí daného formátu, případně multimediálního kontejneru. Databázový systém by měl umět tyto data interpretovat a využít především pro svou interní potřebu. Pro vyhledávání jsou spíše doplňkové (celovečerní film).
- Strukturální metadata popisují interakci objektů. Například do kterého videa patří daná stopa nebo snímek, stránka dokumentu (kompozice), jejich kategorické uspořádání (agregace), případně jejich provázání odkazy (asociace).

Zbytek této kapitoly se bude věnovat popisným metadatům tvůrce obsahu, následující kapitoly pak i dalším.

Mimo obvyklých informací v hlavičce souboru (typ, rozlišení, kódování, barevnost, vzorkovací frekvence) mohou obsahovat MM data další informace:

- EPG a RDS (Radio Data System) – popis analogového rozhlasového vysílání (např. informace o stanici včetně alternativních frekvencí, skladbě nebo dopravním zpravodajství). Obdobným typem je u digitálního televizního (a rozhlasového) vysílání EPG (Electronic Program(me) Guide), který nabízí více informací o aktuálním i budoucím programu.
- ID3 je popis použitelný výhradně v hlavičce formátu MP3 (MPEG-1 a 2), obsahuje informace jako název, jméno umělce, skladatele, alba, rok vydání a žánr.
- Exif je standard používaný digitálními fotoaparáty při pořizování snímků vyvinutý JEIDA (Japan Electronic Industry Development Association) pro formát TIFF, následně portovaný do JPEG, ale ne už JPEG2000 nebo PNG. Exif obsahuje informace o čase pořízení, informace a nastavení fotoaparátu (značka, typ, citlivost, clonu, expoziční čas, ohniskovou vzdálenost, použití blesku nebo orientaci), informace o místě pořízení (v případě dostupnosti GPS dat), umožňuje uložit náhled fotografie a dodatečné přidání informací o autorovi.
- IPTC (International Press Telecommunications Council) v roce 1979 stanovil několik atributů metadat popisující obrázky, jako název, autor, datum vytvoření, místo pořízení, kategorie, důležitost nebo klíčová slova. Adobe je převzalo, nejprve (1994) vložilo do hlavičky JPEG a TIFF, ale v roce 2005 nahradilo XMP.
- XMP (Extensible Metadata Platform) je open source formát vyvinutý Adobe Systems pro popis obrázků a dokumentů PDF, založený na XML. Zahrnuje informace převzaté z, Exif, IPTC i Dublin Core a doplňuje některé další.

- Dublin core je otevřený standard pro nalezení elektronických zdrojů. Obsahuje základní informace o zdroji (title, creator, subjekt, description, publisher, contributor, date, type, format, identifier, source, language, relation, coverage, rights). Vzhledem k tomu, že byl vytvořen pro anotaci videa, obsahuje rozšíření, které umožňuje sémanticky anotovat jednotlivé scény a podobně. Bývá implementován pomocí XML schématu.
- DICOM (Digital Imaging and Communications in Medicine) vychází ze standardu ACR-NEMA, který popisuje binární formát medicínských souborových dat. Metadata se mohou týkat pacienta (jméno, narozen), jeho stavu, dále zařízení, na kterém byl snímek vytvořen (CT, MR, XR, US⁶, výrobce, nastavení přístroje), poloha pacienta při snímání, a technické informace (barev, rozlišení) a mnohé další. Formát DICOM je kontejner snímků. Umožňuje navíc jednoznačně identifikovat pacienta a uchovává jeho snímky v hierarchii – případ (study, např. rozbitá hlava) obsahuje série (vyšetření), obsahující požadované snímky ARC-NEMA.
- MPEG-7

V přehledu je uvedeno několik běžných formátů pro popis některých typů MM dat. Moderní formáty jako JPEG-2000, PNG, MPEG-4, AAC mají možnost vložit pouze XML metadata (problém s ukazateli Exif), tedy Dublin Core nebo XMP. Ty jsou ale zaměřeny výhradně na (sémantický a technický) popis média ze strany tvůrce a neobsahují žádnou možnost vyhledávání dle obsahu (CBR). Na XML je ale založen i popis dle MPEG-7, který předchází nedostatky doplňuje.

2.6.1 MPEG-7

ISO/IEC 15938: Multimedia Content Description Interface [MP704] je rozhraní pro popis obsahu multimedií. Bylo sice vytvořeno skupinou MPEG, Motion Pictures Expert Group, tedy skupinou expertů na pohyblivé obrázky, ale neslouží pro kompresi audiovizuálních dat.

MPEG-7 poskytuje množinu standardizovaných deskriptorů pro popis obsahu různých druhů médií – statické obrazy v tištěné podobě, 3D grafika a její modely, zvuk, řeč a video nebo biometrie lidského obličeje – s cílem efektivního vyhledávání informací ve velkém množství multimediálních dat.

Popis nesouvisí se způsobem uložení médií. Je možné přiřadit deskriptor MPEG-7 k obrazu, článku v časopise, klasickému filmu, stejně jako proudu MPEG-4.

Pro uložení schémat deskriptorů je definován formát založený na XML, který lze převést do efektivní binární podoby. Norma specifikuje následující prvky:

- Deskriptory (D) reprezentují vlastnosti, rysy – atributy multimediálního obsahu založené na katalozích (název, autor, práva, popis), sémantice (kdo, co, kdy a kde – informace o objektech a událostech), syntaxi (pro CBR, barva obrazu, tón zvuku) a technologii (formát, velikost, vzorkovací frekvence).
- Popisová schémata (Description Schemes – DS) popisují strukturu a sémantiku vztahů mezi komponentami D nebo DS – typ média, jeho původ, možnosti použití, strukturální vlastnosti nebo libovolný text.
- Datové typy (DT). Zavádějí například jednoduchá pole v XML.

⁶ Počítačová tomografie, magnetická rezonance, rentgen, ultrazvuk

- Jazyk pro definici deskriptorů (Description Definition Language – DDL) definuje D, DS, DT, jejich syntax, sémantiku, možnosti jejich změny a rozšíření založené na XML, upravené za pomoci W3C pro MPEG-7. Viz. .
- Systémové nástroje (Systems tools) podporují tvorbu a přenos popisů například v binární podobě, jejich multiplexování s multimediálním obsahem, synchronizaci, formáty souborů.

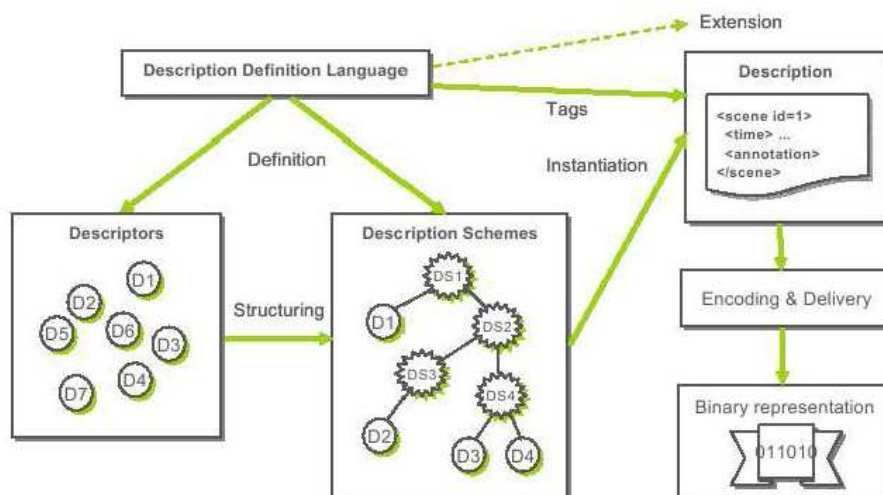


Fig. 2.4 Schéma prvků MPEG-7 (převzato z [7])

Norma MPEG-7 se dělí se na 7 tématických částí:

- Část 1 Systém: specifikuje nástroje pro přípravu deskriptorů k přenosu, ukládání, kompresi a pro synchronizaci s obsahem.
- Část 2 DDL: specifikuje jazyk pro definování standardní množiny deskripčních nástrojů (DS, D, DT) a nových nástrojů, založeno na XML.
- Část 3 Vizualní: obsahuje nástroje pro popis vizuální složky, specifikuje základní kategorie – barva, textura, tvar, pohyb, pozice, rozpoznávání obličejů.
- Část 4 Audio: definuje nástroje pro popis zvukové složky jako spektrální, časové, dynamické vlastnosti (low-level) nebo rozpoznání hlasu, barva nástroje, melodie (high-level).
- Část 5 Schémata popisující multimédia: nástroje pro popis multimediální části popis obsahu – audio i vizuální (video), použití obsahu, organizace, navigace, interaktivity.
- Část 6 Referenční software: implementace standardu se stále vyvíjí, protože norma popisuje způsob uložení popisu, nikoli způsob jeho získání.
- Část 7 Testování shody: poskytuje vodítka pro testování shody implementace deskriptorů.
- Část 8 Extrakce a použití MPEG-7 schémat. Pouze informativní, není součástí normy, například implementace experimentálního modelu.

Detailnější popis MPEG-7 Části 3 a 4 je uveden v kapitole 2.7.

⁷ Pokud by na internetu fungovalo vyhledávání podle obsahu, byla by hračka zjistit, z jakého zdroje pochází tento obrázek...

Příklad popisu obsahu video záznamu dle MPEG-7 je uveden jako Ex. 2.1, namísto manuální anotace DescriptionMetadata (...) je možné umístit Ex. 2.3, jejíž DDL definice je uvedena v příkladě Ex. 2.2

Ex. 2.1 Ilustrace anotace obsahu video dokumentu dle MPEG-7

```

<Mpeg7>
<DescriptionMetadata>...</DescriptionMetadata>
<Description xsi:type="ContentEntity">
<MultimediaContent xsi:type="VideoType">
  <Video id="video_example">
    <MediaInformation>...</MediaInformation>
    <TemporalDecomposition gap="false" overlap="false">
      <VideoSegment id="VS1">
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT2M</MediaDuration>
        </MediaTime>
        <VisualDescriptor xsi:type="GoFGoPColorType"
          aggregation="average">
          <ScalableColor numOfCoef="8"
            numOfBitplanesDiscarded="0">
            <Coeff>1 2 3 4 5 6 7 8</Coeff>
          </ScalableColor>
        </VisualDescriptor>
      </VideoSegment>
    </TemporalDecomposition>
  </Video>
</MultimediaContent>
</Description>
</Mpeg7>

```

Ex. 2.2 Ilustrace MPEG-7 DDL pro manuální anotaci video záznamu

```

<complexType name="VideoDoc">
  <element name="Title"/>
  <element name="Producer"/>
  <element name="Date"/>
</complexType>
<complexType name="NewsDoc" base="VideoDoc" derivedBy="extension">
  <element name="Broadcaster" maxOccurs="1"/>
</complexType>
<element name="VideoCatalogue">
  <complexType>
    <element name="CatalogueEntry" type="VideoDoc"/>
  </complexType>
</element>

```

Ex. 2.3 Ilustrace manuální anotace video dokumentu pomocí MPEG-7

```

<CatalogueEntry xsi:type="NewsDoc">
  <Title>"CNN 6 oclock News" </Title>
  <Producer>David James</Producer>
  <Date>1999</Date>
  <Broadcaster>CNN</Broadcaster>
</CatalogueEntry>

```

Deskriptory a popisová schémata.

Deskriptor plní funkci normativu udávajícího způsob, jak popsat obsah díla. Skládá se jednak z informací, které lze získat z AV materiálu (barva, textura, tvarové pozice, tónina, rytmus), a z informací, které již z obsahu získat nelze (autor, copyright, žánr).

Deskripty vysoké úrovně jsou výkonné, nejsou však příliš pružné. Na druhou stranu deskripty nízké úrovně jsou použitelné všeobecně, ale při vyhledávání je třeba použít výkonný vyhledávač. Deskriptor jako takový může existovat i odděleně od obsahu viz. Fig. 2.5, anebo je reprezentovaný daty, které jsou multiplexovány ve vlastním datovém toku AVM viz. Fig. 2.6.

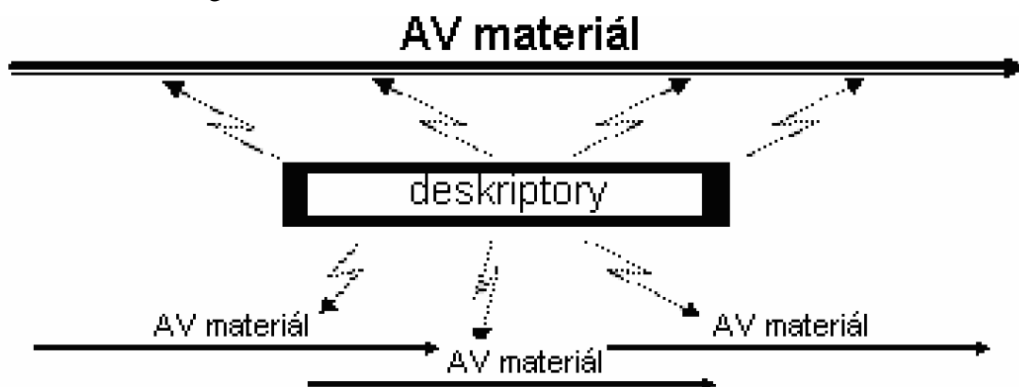


Fig. 2.5 Deskripty oddělené od obsahu (převzato z [MIS04]).

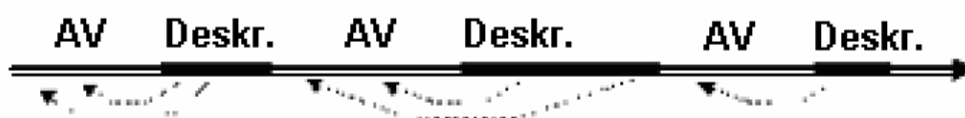


Fig. 2.6 Deskripty multiplexované s obsahem AVM (převzato z [MIS04]).

Pro vkládání popisu obsahu AVM jsou ve standardu MPEG-7 použity popisové nástroje (description tools), jejichž obsahem je deskriptor vysoké úrovně a (Ds) a popisová schémata (DSs). Při vytváření nových deskriptorů je třeba je umístit do popisového schématu, to následně vede k zefektivnění vyhledávání.

Popisové schéma (DS) určuje vztahy mezi svými součástmi (pohyb, barva,...), jež mohou být deskripty. Ty obsahují jen základní typy dat, poskytované DLL a souvislosti s jinými deskripty. Na obrázku Fig. 2.7 je ukázka popisového schématu jednoho video záběru popisujícího určitý AVM.

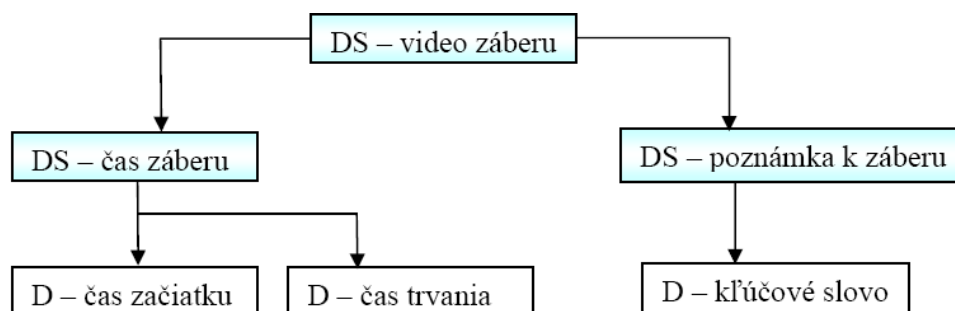


Fig. 2.7 Popisové schéma pro jeden záběr (převzato z [MIS04]).

Popisové schéma video záběru je vyjádřeno pomocí deskriptoru VideoSegment, jemuž je podřízen deskriptor času MediaTime obsahující deskriptor začátku MediaTimePoint a deskriptor doby průběhu MediaIncrDuration. Samotné popisové schéma DS představuje strukturu a obsahuje soubor hodnot deskriptorů. Jednak to mohou být deskripty vysoké úrovně, jejichž struktura je dána dalšími popisnými schématy, a jednak deskripty nízké

úrovně, jež obsahují již konkrétní hodnoty. Tento způsob popisu je jedním, ze základních stavebních kamenů MPEG-7.

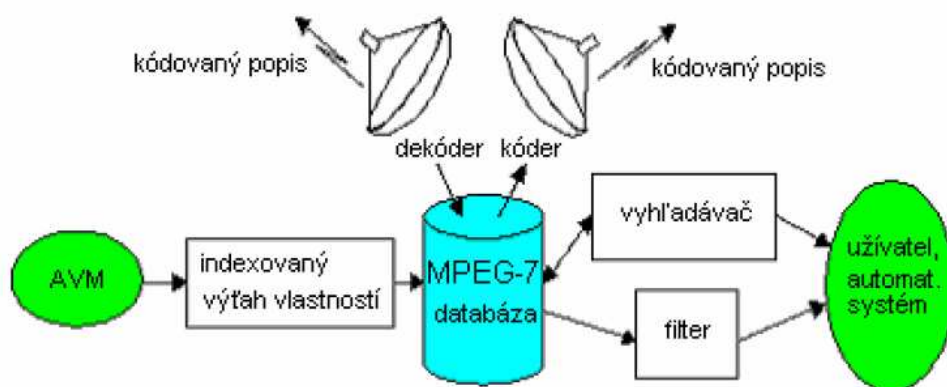


Fig. 2.8 Architektura MPEG-7 (převzato z [MIS04])

Na obrázku Fig. 2.8 je naznačena architektura MPEG-7. Ze zdroje AVM je získán popis, který je neindexován v MPEG-7 databázi. Tento indexovaný obsah pak může být zpřístupněn na základě libovolného typu dotazu. Vyhledávač pak slouží k tomu, aby porovnal daný dotaz s obsahem MPEG-7 dat a poskytl případně odkaz na požadovaný zdroj dat.

MPEG-21

MPEG-21 je nastupující standard pro komunikaci mezi všemi uživateli – tvůrci, distributory a konzumenty multimediálního obsahu v distribuovaném multimediálním systému [MP210]. Jednotkou dat je unikátní digitální položka (Item), obsahující AV obsah (JPEG2000, MPEG-4 a další) spolu s popisem MPEG-7, správou autorských práv a systémem pro jejich distribuci, prezentaci a synchronizaci.

Standard byl v době tvorby této práce (2006) za polovinou schvalovacího procesu.

DICOM

Počátečním cílem ve vývoji standardu pro přenos digitálních obrazů bylo umožnit uživatelům získat obrazy a asociované informace z digitálních snímacích zařízení ve standardním formátu, který by byl stejný napříč různými výrobci. Prvním výsledkem byl standard Americké vysoké školy radiologické ACR (American College of Radiology) a Federální asociace výrobců elektroniky NEMA (National Electronics Manufacturers Association), který specifikoval point-to-point spojení. Nicméně, rapidní vývoj počítačových sítí, archivování obrazů a komunikačních systémů ukázalo, že tento point-to-point standard by měl omezené použití. Následně bylo hlavní úsilí upřeno na přepracování ARC-NEMA standardu pomocí zabudování již existujících standardů pro sítě a pro manipulaci s informacemi v těchto sítích. Výsledkem tohoto úsilí se stal standard DICOM (Digital Imaging and Communications in Medicine).

DICOM standard je neobyčejně adaptivní, tento plánovaný rys vede k využití standardu i jinými specializacemi, které generují obrazy (patologie, endoskopie, stomatologie). Skutečnost, že mnoho výrobců medicínských snímacích zařízení jsou nadnárodními korporacemi vedla k mezinárodnímu zájmu o tento standard. Evropská standardizační společnost Comitê Européen de Normalisation používá tento standard jako základ pro plně kompatibilní standard MEDICOM. V Japonsku pak Japonská průmyslová asociace radiačních přístrojů a Vývojové centrum pro medicínské informační systémy převzaly

části DICOMu, které souvisí s výměnou obrazů na výměnných médiích, a které jsou plánovány i pro budoucí verzi Standardu pro zpracování medicínských standardů.

DICOM se stává převládajícím standardem pro předávání medicínských obrazů. Nicméně, přestože je tento standard velmi dostupný od výrobců a rychle se rozšiřuje i do neradiologického snímání, většina radiologických interpretací je limitována. Částečně je to proto, že standard byl napsán pro buď inženýry, nebo značně technicky, nebo pro administrátory.

Většina radiologů je srozuměna hlavně s klasickými rentgenovými fotografiemi, které si mohou prohlédnout kdekoliv, kde je zdroj světla. U těchto fotografií nehraje roli drobný rozdíl v expozici, zpracování. V digitálním snímání naopak může rozdíl v několika bytech znemožnit přenos z jednoho systému do jiného.

Standard DICOM se skládá z vícero dokumentů. Momentálně je publikováno 13 částí. Každý dokument tohoto standardu je identifikován názvem a číslem standardu, které je ve formě "PS 3.X-YYYY", kde X je obecně označováno jako číslo standardu a YYYY je rok standardu (např. DICOM část 2 je označena názvem „Conformance“ a číslo dokumentu je PS 3.2-1996).

Elementární jednotka DICOMu

Informační objekt a servisní třída jsou dvě základní komponenty DICOMu. Znalost těchto komponent umožňuje pochopení minimálně na funkční úrovni, co DICOM dělá a k čemu je užitečný. Informační objekt definuje jádro obsahu medicínského snímání a servisní služba definuje co se má udělat s tímto obsahem.

Servisní třídy a informační objekty jsou zkombinovány do podoby funkcionálních jednotek DICOMu. Tato kombinace je nazvána dvojice servisních objektů (service-object pair – SOP) nebo jednoduše SOP. Od doby, kdy se stal DICOM OO standardem, se to označuje jako SOP třída. SOP třída je elementární jednotka DICOMu. Všechno co se dělá v DICOMu je založeno na použití SOP tříd. Vytváření DICOM SOP třídy je vlastně analogií k vytváření věty.



Fig. 2.7 Diagram demonstrující analogii mezi vytvářením věty a vytvářením DICOM SOP třídy.

Kombinování servisních a informačních objektů není komplikované. Např. DICOM definuje řadu ukládacích SOP tříd (např. SOP třídu pro ukládání CT obrazů, SOP třídu pro ukládání MR (magnetická rezonance) obrazů). Definice CT informačních objektů a třída ukládací služby jsou zkombinovány do podoby SOP třídy pro ukládání CT obrazů. Jelikož SOP třídy jsou chápány jako způsob jak popsat funkcionalitu DICOMu, nesou UID informaci. Atributy informačních objektů a proměnné servisní třídy jsou naplněny hodnotami reálného pacienta, částečně pak snímáním zařízením a výsledným obrazem. SOP třída se stává SOP instancí a je jí přiřazeno vlastní UID.

Proces DICOM komunikace zahrnuje výměnu SOP instancí za pomoci SOP zpráv. SOP zpráva je komunikační verze SOP třídy, přičemž zahrnuje příkazy, které používají nebo poskytují konkrétní služby a datový soubor vytvořený příslušným zakódováním instance informačního objektu.

Hlavním problémem pro jakýkoliv standard je určení přizpůsobení. V mnoha situacích týkajícího se veřejného zdraví a bezpečnosti je vyžadováno zákonem přizpůsobení se standardům. Pro mnoho standardů, včetně DICOMu, je přizpůsobení se dobrovolné. Výbor pro DICOM standard (DICOM Standard Committee) nemá žádnou vynucovací autoritu. Každý kdo tvrdí, že jeho zařízení nebo software je přizpůsoben standardu DICOM musí doložit prohlášení o shodě, které popisuje přesně jak dané zařízení nebo software je přizpůsoben standardu.

Prohlášení o shodě se skládá z následujících částí: (a) implementační model aplikace, (b) prezentační kontext, který má být použit, (c) způsob, jakým je realizována asociace, (d) SOP třídy, které budou podporovány, (e) komunikační profily, které budou použity, (f) rozšíření nebo specializace, které budou použity.

Implementační model aplikace je jednoduchý diagram, který ukazuje, jak je aplikace asociována s lokálními (na navrhovaném zařízení) a vzdálenými (přes DICOM rozhraní) aktivitami. Např. lokální aktivitou může být vytvoření informačního objektu DICOM obrazu a vzdálenou aktivitou pak zobrazení tohoto objektu.

Prezentační kontext se skládá z abstraktní syntaxe (jiný výraz pro SOP třídu) a přenosových syntaxí použitých pro tuto abstraktní syntaxe. Termín abstraktní syntaxe je užit z části, protože je definován v jednom mezinárodním standardu, který se odkazuje na DICOM.

Prohlášení o shodě musí popisovat jak aktivita vykonává asociace (např. zda aktivita iniciuje asociaci a přijímá vícenásobné asociace) pro každou aktivitu v modelu. Některá zařízení, jako např. archívy pro archivování obrazů a komunikační systémy, musí podporovat vícenásobné asociace.

Seznam podporovaných SOP tříd je klíčový aspekt prohlášení o shodě. Seznam popisuje, které servisní třídy a informační objekty budou aplikací používány a akceptovány. Komunikační profil použil jednoduché stavy, které jsou podporovány komunikačním zásobníkem. Jsou k dispozici Point-to-point, ISO (implementace ISO-OSI standardu) a TCP/IP zásobníky.

Závěrečná část prohlášení o shodě detailně popisuje rozšíření nebo specializace SOP tříd. Rozšíření SOP třídy znamená přidání standardních atributů, které nejsou povinné, ale mohou být v dané aplikaci použity. Rozšířená SOP třída je nadmnožinou standardních SOP tříd a většina aplikací nebude mít problémy se spoluprací s touto třídou. Specializovaná SOP třída může rovněž obsahovat privátní atributy, které tvůrce SOP třídy považuje za povinné. Tyto specializované třídy mohou přivodit některým aplikacím komplikace během jejich interpretace. Důvod proč, jsou výrobcům takovéto třídy povoleny pro jejich vlastní potřeby je, že není záměr ani potřeba, aby tyto třídy komunikovali s jinými SOP třídami mimo dané zařízení. Specializace SOP tříd nebyla nikdy brána jako způsob jak obejít DICOM standard. Farmaceutický balíček vkládá do prohlášení o shodě rozšíření blízké lékařů. Vkládá do něj takové informace jako je chemická struktura, původ a podstata, indikované použití či vedlejší účinky.

Shrnutí

Předpokládejme, že chceme zaslat CT studii z CT skeneru do pracovní stanice a sledovat, jak by to bylo uskutečněno pomocí DICOMu. Technik nastaví na CT skeneru, že chce zaslat část CT studie. Skener by se měl dotázat na cílovou pracovní stanici. Tato interakce se děje na úrovni snímací aplikace, tedy bez potřeby DICOMu. Nicméně, skener může nasnímanou studii uložit ve své konzoli v DICOM formátu (skenovací aplikace vytvoří instance objektů přiřazením hodnot, jako jsou např. jméno pacienta, číslo medicínského záznamu, skener sám pak přidá další informace jako např. datum, čas, instituce, identifikace skeneru). Tedy zaslání studie pak vyžaduje zadání jména nebo síťové adresy cílové pracovní stanice.

Komunikační protokol (obvykle PCT/IP) pak vyřeší fyzické spojení s pracovní stanicí z CT skeneru. Výběrem funkce, která bude zasílat studii, začíná aplikační software na CT skeneru sestavování sledu SOP instancí. V tomto případě bude vytvořena jedna instance pro každý obraz, který má být zaslán. Pokud je navázáno spojení s cílovou pracovní stanicí, DICOM software začíná komunikační proces zasláním požadavku na vytvoření asociace s pracovní stanicí. Komunikační síť se ujme přecházejících kroků ustavením komunikačního kanálu. Během asociace, poskytuje CT skener svůj prezentační kontext, říká pracovní stanici, že podporuje SOP třídu pro uložení CT obrazu jako uživatel servisní třídy a jeho jednotlivé aplikace akceptují pouze třídu verifikační služby jako poskytovatele servisní třídy. CT software rovněž deklaruje, že přenosová syntaxe je implicitně VR malý endian. Pracovní stanice odpovídá, že podporuje SOP třídu pro ukládání CT obrazů jako poskytovatele servisní třídy stejně tak jako uživatele servisní třídy. Pracovní stanice odpovídá, že může implicitně použít přenosovou syntaxi VR malý endian. Notifikace akceptace asociace a způsobilosti jsou zaslány zpět příslušným zařízením.

CT skener zasílá požadavek pro ukládací službu společně s obrazem softwaru, který sestavuje DICOM zprávu skládáním nezbytných příkazových elementů a elementů datových souborů. Dále pak zašle zprávu dolů skrz komunikační zásobník. DIMSE potřebné pro třídu ukládací služby a datový soubor obsahující obraz a další data jsou uchopena komunikačním zásobníkem, kde je zpráva obvykle rozdělena do menších paketů, aby byla následně znovu poskládána do zprávy komunikačním zásobníkem pracovní stanice. Komunikační zásobník detekuje chyby, které mohou vzniknout během přenosu a je odpovědný za to, aby byly pakety zaslány na správné místo.

Znovu poskládaná DICOM zpráva je přijata aplikační vrstvou v pracovní stanici, které využívá svůj software pro lokální uložení CT obrazu. Pracovní stanice může rovněž zaslat informaci (jméno pacienta, typ studie, počet obrazů,...) jinému softwaru, který zobrazí tuto informaci do nějakého pracovního seznamu.

Tento celý proces je uskutečněn relativně rychle, obvykle přenos jednoho CT obrazu trvá od jedné do 10 sekund, v závislosti na typu softwaru a hardwaru. Jednou vytvořená asociace může být použita pro více SOP instancí.

HL7

HL7 (Health Level 7) je standardizovaný datový formát užívaný pro nemocniční textové informace - byl vyvinut pro systémy zdravotní péče. Významně pomáhá při standardizaci nezbytných rozhraní mezi systémy. Tento standard byl vyvinut v USA a nyní je oficiálním ANSI standardem. (American National Standards Institute - americká standardizační organizace sídlící ve Washingtonu. Je to nezisková organizace, která vytváří průmyslové standardy ve Spojených státech. Je členem organizace ISO a IEC.)

Již v roce 1987 byla v USA ustanovena organizace, jež usilovala o standardizaci komunikace v nemocnicích a v celém systému zdravotní péče – skupina „Health Level Seven“, zkratkou „HL7“. Jakmile jsou data zakódována do HL7, měla by být transparentní pro všechny ostatní zdravotnické systémy, které následují tento standard. Současný vývoj ukazuje, že se HL7 nejen stal důležitým komunikačním standardem ve zdravotnictví, ale že také dává impulsy pro celosvětovou standardizaci komunikace ve zdravotnických systémech.

HL7 specifikuje obsah a formát komunikace na aplikační úrovni. Ve vrstevném modelu komunikace mezi otevřenými systémy je tato úroveň nejvyšší – sedmá. Je důležité, aby řešení komunikace bylo nezávislé na použitém sw, hw a datové síti. HL7 verze 2 neřeší všechny problémy komunikace mezi dvěma systémy. Samostatná implementace rozhraní předpokládá rozsáhlé dohody o technických podrobnostech.

- Nebýt závislý na operačních systémech a programovacích jazycích.
- Zajišťovat přednostní přenos krátkých zpráv nebo zpráv s vyšší prioritou.
- Standardizace přenosu potřebných konkrétních datových elementů.
- Snadná modifikovatelnost pro realizaci požadavků vznikajících při rozvoji IS i akceptaci nových verzí operačních systémů.
- Zohlednění požadavků NIS i IS ostatních zdravotnických zařízení.

2.7 Extrakce rysů a popis obsahu

V předchozí kapitole bylo uvedeno, jak vypadá popis multimediálních dat. Tato je věnována způsobu, jak jej získat. V seznamovací kapitole (2.2 Multimediální databáze) je rozlišen manuální a automatický popis obsahu multimédií. Manuální popis není třeba příliš rozebírat. Jde o vytvoření smysluplného, objektivního názvu a popisu objektů, hovorů, scén a podobně, případně o ruční přiřazení média do jisté kategorie.

Automatický popis vytváří vektor rysů⁸ (deskriptor, signaturu) pomocí procesu známého jako extrakce rysů⁹. Rysy jsou jisté charakteristické vlastnosti obsahu média. Rysy člověka mohou být například barva kůže, vlasů, očí, jejich vzdálenost, velikost nosu, hlavy, drsnost kůže atd. Podobné jsou i rysy obrázků. Rysy pohyblivých obrázků mohou být pohyb člověka nebo jeho jiné chování. Rysy textu jsou až na výjimky (rozpoznání jazyka, autora) klíčová slova. Zvukové rysy jsou například frekvenční vlastnosti, tón, barva nástroje, hlas, rozpoznání řeči a podobné informace obsažené v MM datech.

2.7.1 Popis obsahu audia

Na rysech audia je nejlépe možné rozlišit jejich úroveň:

- **Nízká úroveň popisu** (low-level) zahrnuje množinu rysů společných pro všechny signály. Extrakce spektrálních, temporálních a dynamických (syntaktických) vlastností zvuku je plně algoritmizovatelná. Nejčastěji se používá rychlá Fourierova transformace (FFT) [Čer05] pro spektrum zvuku (Fig. 2.9), určení harmonického spektra, hustotu výkonu, základního tónu a jeho zabarvení¹⁰.
- Střední úroveň (mid-level) se někdy označují statistické modely rysů dat. Běžně se počítají statistiky (střední hodnota, korelace) nízkourovňových rysů, které mohou sloužit pro podobnostní vyhledávání, dotazem je například krátká ukázka. Je možné určit, zda se jedná o pravidelnou melodii, řeč, při absenci základního tónu je možné detekovat šum nebo ticho. Dále je možné odlišit zvukové segmenty (zprávy, hudba).

⁸ Feature vector (descriptor, signature)

⁹ Feature extraction

¹⁰ Timbre, zabarvení hlasu nebo zvuku hudebního nástroje pomocí harmonických frekvencí

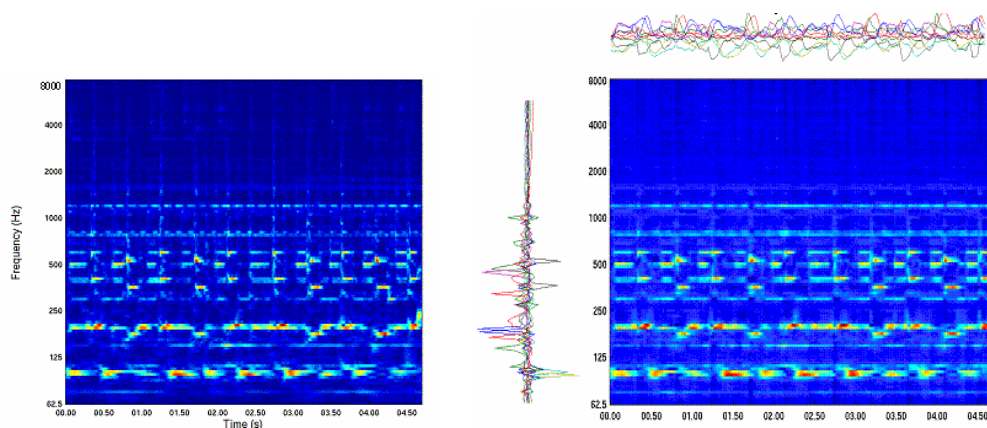


Fig. 2.9 Ukázka spektrálních (low-level) vlastností popu a rocku [MP704]

- Vysoká úroveň** (high-level, deep-level) je obvykle získána z rysů nižší úrovně. Jedná se například o sémantickou anotaci několika vzorků v databázi, ze kterých jsou extrahovány rysy, analyzovány nějakým učícím se algoritmem (Gaussovské, Bayessovské, Markovovy modely) a následně slouží pro klasifikaci (zvuku), jako v kapitole 2.15. Tyto procesy jsou poměrně dobře stanoveny pro speciální aplikace, jako je rozpoznání hudebního nástroje, melodie a především pro rozpoznání mluvčího a obsahu řeči. Dotazem tedy může být zantování melodie (query-by-humming), nebo požadovaná slova, jako by se jednalo o text.

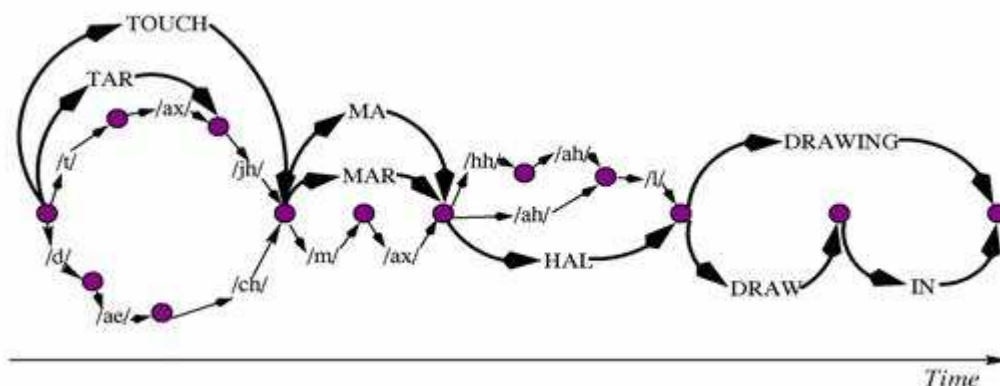


Fig. 2.10 Proces rozpoznání řeči (high-level) „Taj Mahal drawing“ [MP704]

Systémy pro automatické rozpoznání řeči (Automatic Speech Recognition, ASR) jsou schopny identifikovat „naučená“ slova pomocí jejich databáze (slovníku) a odpovídajícího repertoáru fonémů (hlásky, slabiky). V příkladě na obrázku Fig. 2.10 je ilustrován proces rozpoznání řeči s jedním slovem neznámým slovem a druhým slovníkovým. Jsou nalezeny uzly (mezi fonémy) a k nim je vytvořen (alternativní) graf fonémů a odpovídajících slov, ze kterého je vybrán nejpravděpodobnější přepis audio obsahu dle naučených, anotovaných řečových dat.

2.7.2 Extrakce rysů z obrázků

Nízkoúrovňové rysy obrazových dat jsou relativně jednoduché. Jedná se například o dominantní barvu v obrázku nebo její histogram. Pro tyto rysy není barevný model RGB ani CMYK příliš vhodný. Je rozumné použít model, který lépe odpovídá lidskému chápání barev, například HSV, HLS, nebo alespoň $YCbCr$ [Krš05].

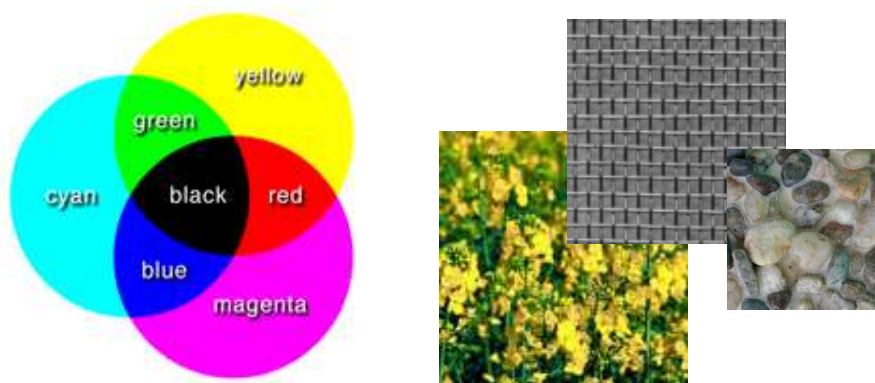


Fig. 2.11 Ilustrace barev a textur

Textura je vlastností všech reálných povrchů. Má 2D vizuální vlastnosti, například kontrast, hranová frekvence, zaměřenost, pravidelnost, nebo hmatový charakter, jako hrubost, drsnost nebo huňatost. V její počítačové reprezentaci je složená z opakujících se elementů, které se nazývají primitiva. Tato primitiva mohou být popsána jejich statistickou, geometrickou nebo frekvenční charakteristikou.

Jedním z nejefektivnějších způsobů popisu texturní informace je extrakce rysů z frekvenční oblasti, získané například DCT nebo FFT, protože odpovídá lidskému chápání. Velikost energie, rozdělené do (30) frekvenčních kanálů – dle jejich směru (6, po 30°) a velikosti (5, po oktávách), slouží jako vektor rysů (30 hodnot). Tato technika se nazývá Gaborovy filtry [Chm04], které jsou zobrazeny na obrázku Fig. 2.12.

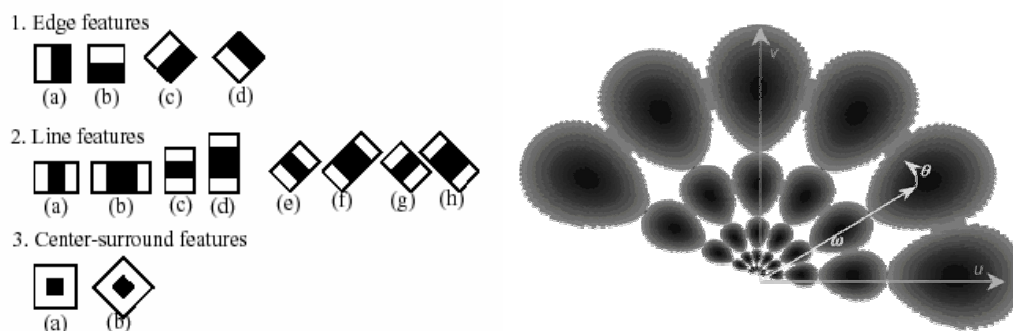


Fig. 2.12 Ukázka Haarových waveletů a Gaborových frekvenčních filtrů

Wavelety, nebo podobné rysy, jsou také běžně používané pro tvorbu deskriptorů. Mají prostorově-frekvenční charakter. Ukázka filtrů, se kterými může být obraz konvoluován je zobrazena na obrázku Fig. 2.12. Jejich výsledek je možné statisticky zpracovat a při vhodné kombinaci tvarů – rysů (například Haarovy kaskády), je možné nalézt také požadovaný objekt v obrázku.

Tyto vlastnosti jsou samy o sobě vhodné například k optické kontrole výroby pro laky automobilů nebo nanotextilie. Barva a textura jsou vhodné pro podobnostní vyhledávání objektů v databázi střední úrovně, i když nedávají vždy korektní výsledky, pokud se vztahují k celému obrázku, namísto jednotlivých objektů. Libovolný obrázek představuje mozaiku barev, lépe textur. Proto je velmi efektivní indexovat oblasti s danou texturou nebo barvou, i když by se mělo jednat pouze o čtvercové oblasti, pevné velikosti, shodné pro všechny obrázky.

2.7.3 Tvar

Hlavním problémem vizuálních dat je, že jsou příliš rozmanitá. Objekty, tak jak je vnímá člověk, jsou odlišné od způsobu jejich rozpoznání počítači, tedy oblasti s jistou barvou,

texturou, hranami a podobně. Mohou mít různou velikost, orientaci nebo barvu a přitom mají stejný význam, nebo naopak, například auto.

Existuje ale několik světlých výjimek, například analýza snímků vzdáleného pozorování země takto umožní relativně spolehlivě klasifikovat osídlené oblasti, pole i lesy a určit, zda nejsou napadeny například kůrovcem a vymežit toto napadení.

Pokud se oprostíme od způsobu, jak přesný tvar objektu získat (viz. Pohyb 2.7.4, nebo Získávání znalostí 2.15), je možné říci, že objekt se skládá z jednoho nebo více regionů v obrázku. K tomu počítáme i díry či další přerušení oblastí. Oblasti je možné popsat jako plochu, nebo její hranici:

- V případě plochy bývá oblast reprezentována jako nějaká malá bitmapa (u MPEG-7 17.5 bytů), ve které hodnoty (jasu) 1 značí příslušnost do odpovídajících oblastí, zatímco 0 k bílému pozadí. Tento způsob popisu oblasti je efektivní pro porovnávání, sledování ve videu a v případě nějakého porušení hranice.
- Hraniční reprezentace je náročnější, ale lépe odpovídá lidskému vnímání a je více odolná vůči afinním transformacím [Krš05], nehledě na možnost změny tvaru objektu ve videu, například běžícího člověka [MP704].

V současné době sice není extrakce 3D tvaru objektů ze 2D příliš rozvinutá, nicméně se hodí pro uchování modelů – předpisů, jak má reálný objekt ideálně vypadat. Jejich reprezentací je obvykle 3D mřížka.

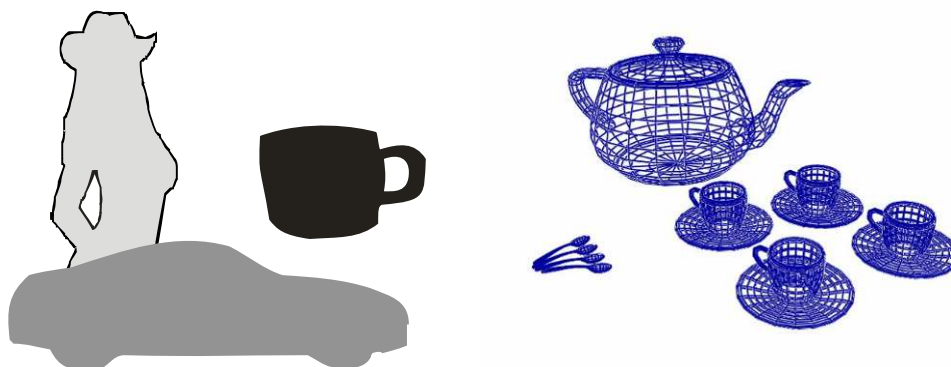


Fig. 2.13 Reprezentace tvaru plochou, hraniční křivkou a 3D mřížkou

2.7.4 Pohyb a umístění

Pohybem je myšlena změna umístění objektů v rámci následujících snímků i další transformace objektu, jako přiblížení, zmenšení, otočení, perspektivní deformace. Dalším typem pohybu, je pohyb kamery.

Kamera má obecně šest stupňů volnosti – může se pohybovat po svislé i vodorovné ose, přibližovat se a vzdalovat (zoom) a v každé z těchto os může rotovat. Pohyb kamery není vždy jednoduché zjistit (například v davu lidí), ale je to žádaná vlastnost. Pohyb kamery lze vyjádřit pomocí fundamentální matice. Tu lze odhadnout pomocí shody několika náhodných bodů v následujících snímcích, například pomocí RANSAC (Random Sample Consensus) [Chl05].

Obecným typem pohybu objektů na ploše je perspektivní transformace, která se běžně využívá například při fotbalových přenosech. Její výpočet je jen o trochu jednodušší, než výpočet fundamentální matice [Jäh99]. Argumentem složitějšího počítání, je možnost zjistit reálnou polohu objektů ve scéně, která je nezbytná například pro bezpečnostní aplikace založené na kamerových systémech.

Pokud si nepotřebujeme komplikovat život, nejjednodušším typem pohybu tuhého objektu je pohyb v rovině, kdy připouštíme pouze translaci (jako MPEG-2) a někdy i rotaci objektu. Těleso tedy příliš nemění vzhled, tvar ani velikost. Takováto tělesa lze ve videu najít pomocí metod pro estimaci (odhad) pohybu. Jednodušší metody zjistí změnu v po sobě následujících snímcích tím, že odečtou hodnoty jejich pixelů. Pak pracují buď po blocích (16 x 16 px), nebo ohraničí celou oblast a snaží se v blízkém okolí najít její nový výskyt. Tím vytvoří jednoduchý vektor pohybu. Lépe funguje například prediktivně – korekční Kalmanův filtr [Jäh99].

Když kompenzujeme pohyb kamery (odečteme vektor pohybu) a ohraničíme dále se pohybující oblasti (seskupíme bloky), jsme schopni identifikovat pohyblivé objekty a určit jejich tvar. To je velmi ceněná vlastnost při anotace pohyblivých obrázků.

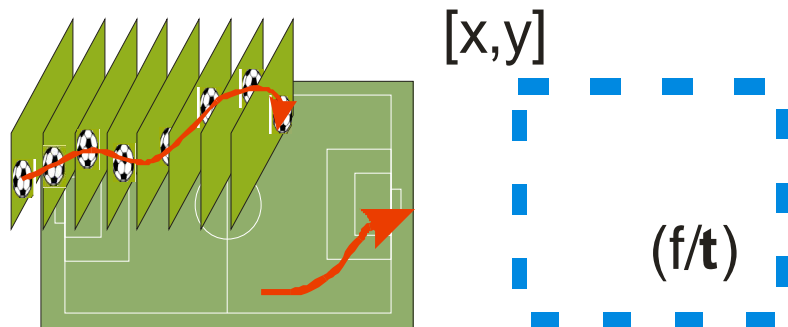


Fig. 2.14 Pohyb a umístění

Takovéto vizuální těleso je poté možné obalit konvexní obálkou (obdélníkem) a zaznamenat jeho souřadnice do databáze. Pokud doplníme časové razítko (číslo rámce), získáme časoprostorový lokátor (umístění) objektu. V případě nalezení nového objektu můžeme udělat více, než zaznamenat jeho velikost. Od manuální anotace (jméno herečky), přes extrakci rysů nižších úrovní až po automatické rozpoznání (viz. kapitola 2.15). A pokud máme k dispozici i informace o reálném prostředí, lze umístění objektu anotovat v jeho kontextu a uchovávat v (časoprostorové) databázi.

2.7.5 Speciální aplikace

Různé speciální aplikace multimediálních databází vyžadují odlišné popisy dat, ať už se jedná o jejich popis nebo o popis obsahu. Těžko může nějaká (tato) práce nebo norma obsáhnout všechny doménově specifické rysy. Z toho důvodu je doporučeno uchovávat metadata ve formátu XML (eXtensible markup language, rozšiřitelný značkovací jazyk).

Zajímavou aplikační oblastí je také nalezení a rozpoznání obličeje. Detekci je hlavy (demo) je možné udělat například dle barvy pokožky, ale ukazuje se, že texturní rysy obličeje, jako waveletová transformace nebo Gaborovy filtry dávají lepší výsledky [Hin2003]. Po ohraničení obličeje (obvykle se předpokládá pohled zepředu) jsou nalezeny dominantní části jako oči a ústa. Z těchto dat je možné určit některé biometrické informace jako poměry vzdálenosti očí, úst, nebo velikost nosu či čela. Běžnější je ale extrakce texturních rysů, nebo umístění obličeje do mřížky s pevnou pozicí očí a popis obličeje pomocí vektoru složeného z průměrné barvy (intenzity) každé buňky [MPEG-7].

Jako další příklad uvedu bezpečnostní aplikaci, která má za úkol měřit průměrnou rychlost jízdy pomocí dvou kamer, na začátku a konci měřeného úseku. V tomto případě není třeba složitě popisovat pohyb a umístění automobilu, protože ze 2 uložených fotografií s časovým razítkem je vše zřejmé. Automobily, které nepřekročí rychlost (o hodně), nás nezajímají a nejsou zaznamenány. Nejvýše jejich počet.

Obdobné je to v případě dalších aplikací (ukázka na obrázku Fig. 2.15), kde je možné obecné informace analyzovat a ihned zahodit, protože pro danou aplikaci nejsou třeba a běžné podobnostní vyhledávání by stejně nedávalo relevantní výsledky. Budeme především chtít vyhledávat podle značky vozidla, které rychlost překročilo.

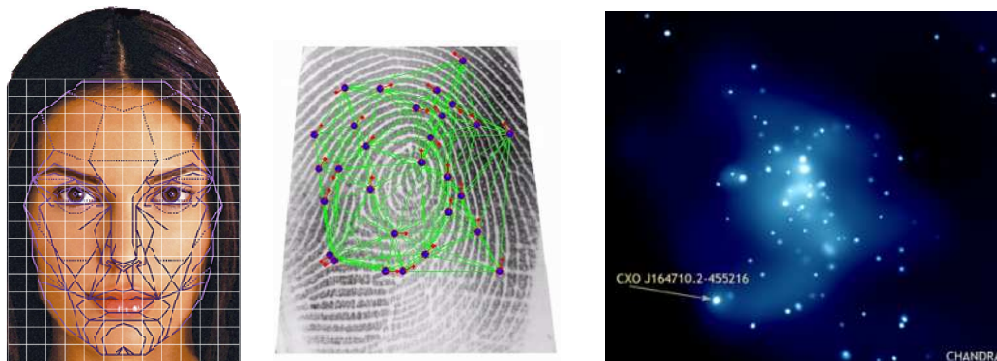


Fig. 2.15 Speciální aplikace – rozpoznání obličeje, otisku prstů, nalezení (neuronových) hvězd

2.8 Vyhledávání

Vyhledání relevantní informace v audiovizuálních datech je značně náročný, ale elementární úkol každého MMDBMS. Z toho důvodu požadujeme, aby bylo vyhledávání co možná nejefektivnější. Proto je v naprosté většině případů omezeno na metadata (obsahující popis i rysy), která musí být předem extrahována a vhodně indexována, na čemž výrazně závisí efektivita vyhledávání (Fig. 2.17). Vyhledávací algoritmus je také důležitý, ale především musí umět interpretovat správně daná metadata. Například hodnoty 0 a 1 představují v masce tvaru různé objekty, kdežto v případě jasu jsou to de facto stejné hodnoty (z 256).

Metadata jsou relativně běžná, strukturovaná (nejhůře XML) data. Je tedy možné použít libovolný strukturovaný dotaz. Tak se tomu děje v případě vyhledávání dle textového popisu dat, které je v současné době asi nejběžnější (Obrázky na Google). V příkladě Ex. 2.4 se oprostíme od prezentace obsahu.

Ex. 2.4 Dotaz dle popisu dat

```
select i.content
from Images i, Description d
where i.id = d.image and
      d.annotation LIKE '%Mars%';
```

Druhým způsobem je vyhledávání dle obsahu audiovizuálních dat. Pro něj je výše popsaný způsob krajně nepraktický. Například při dotazu na nalezení jisté textury by bylo nutné znát všech 30 koeficientů deskriptoru. Takovýto způsob se nazývá vyhledávání specifikací vlastností.

Některé MMDBMS dokáží specifikaci vlastností výrazně zjednodušit. Je možné například dotaz načrtnout, tj. specifikovat barvu a tvar objektu (query by sketch), nebo zanotovat melodii (query by humming). Příkladem je systém QBIC (<http://www.qbic.almaden.ibm.com/>)¹¹. Přestože je tento způsob pro lidi velmi intuitivní, netěší se takové podpoře, jako podobnostní vyhledávání, se kterým má mnoho společného (ilustrace Fig. 2.2).

¹¹ <http://www.hermitagemuseum.org/cgi-bin/db2www/qbicSearch.mac/qbic?selLang=English>

2.8.1 Podobnostní vyhledávání

Člověk umí vyjádřit podobnost jistým „nativním“ způsobem. Ta může být velmi subjektivní (myš), ale i více objektivní (zebra). Podobnost proto musí brát v úvahu především modely lidského vnímání [Hig06].

Princip podobnostního vyhledávání je relativně jednoduchý. Předpokládá, že pro vyhledávání požadované informace jsme schopni dodat nějaký příklad požadovaného audiovizuálního obsahu. Z tohoto média MMDBMS extrahuje stejné rysy, jako by jej chtěl uložit do databáze. Tato metadata dotazu jsou poté využita pro vyhledávání (viz. schéma Fig. 2.3).

Běžně se používají první dvě podobnostní funkce:

- Pomocí tříd podobnosti. Máme-li množinu podobných objektů, pak relace podobnosti na této množině je reflexivní (soběpodobnost) a symetrická (jako dvojčata). Ale objekt této množiny není podobný objektu jiné množiny. Formálně viz [Češ06]. Objekty mohou být označeny (tag, label) například ručně. Přiřazení do těchto množin, případně způsob jejich automatické tvorby je naznačen v kapitole 2.15.
- Pomocí (Eukleidovské) vzdálenosti (rysů). Máme-li spojitě (číselné) intervaly, a je-li možné mezi nimi měřit vzdálenosti (měla by platit symetrie, nezápornost a trojúhelníková nerovnost), pak můžeme tuto vzdálenost označit jako míru podobnosti. Nicméně záleží pouze na návrhu vektoru rysů, jestli bude tato podobnost odpovídat také jejímu lidskému pojetí. Např. frekvenční spektrum ($\alpha_i = 1$) versus délka záznamu ($\alpha_j = 0$). Pro vektory v_1, v_2 o N rysech lze vyjádřit podobnost jako váženou (α) vzdálenost d :

$$\text{Eq. 2.1} \quad d(v_1, v_2) = \sqrt{\sum_{i=1}^N \alpha_i (v_2[i] - v_1[i])^2}$$

- Podobnost odvozená od (čas)prostorových dat [Kol05]. Prostorové pro topologické a vzdálenostní dotazy (2m vlevo od). Temporální dotazy (úsek, kdy zmizel). A jejich kombinace (sledování více kamerami, kde se vyskytoval objekt dříve).
- Transformační vzdálenost¹².

Podobnostní vyhledávání v (čas)prostorových datech má několik úskalí. Tím prvním je jejich spojitý charakter – objekty je nutné vyhledávat pomocí plovoucího okna. Druhým je, že objekt může být různě velký a znamenat to stejné (portrét, dav) nebo naopak. Proto se používají algoritmy s adaptivní velikostí okna, které jsou invariantní velikostně (adaptive scaling) [Gra04] a časově (dynamic time warping) [Kad02].

Příkladem vyhledávání v obsahu audio dat obsahujících muziku je http://www.music-ir.org/mirex2006/index.php/Main_Page. V případě řečových dat je to například skupina Speech@FIT (<http://www.fit.vutbr.cz/research/groups/speech/>), nebo software <http://www-306.ibm.com/software/voice/viavoice/>. Pro obsah videa jsou to například produkty: <http://www.almaden.ibm.com/projects/cuevideo.shtml>, <http://www.virage.com/>, <http://persia.ee.columbia.edu:8080/>.

• ¹² Je popsána v literatuře. Podobnost dvou obrázků je vyjádřena cenou transformace prvního na druhý nebo naopak [Sub98]. Transformace pracují přímo s obsahem pro porovnání každé dvojice obrázků, což je silně neefektivní. Výhodou je možnost vyjádřit subjektivitu. Jedná se pouze o akademickou práci, realizace mi není známa.



Fig. 2.16 Ukázka podobnostního vyhledávání (<http://corbis.ltutech.com/>)

Příklad podobnostního vyhledávání obrazových dat (jako Fig. 2.16) je možné nalézt v následujících lokacích: <http://www.tiltomo.com/>, <http://www.pyxor.com/>. Software pro podobnostní vyhledávání je například: <http://sourceforge.net/projects/brisc>, <http://octagon.viitala.eu/>. Demonstrace podobnostního vyhledávání pomocí MMDB rozšíření Oracle Database je uvedena jako Příloha 2. Oracle interMedia aplikace.

2.9 Indexace

Index je prvek databáze, který má na starost rychlý přístup k datům a jejich vyhledání. Indexovat lze popis i obsah. Na obrázku Fig. 2.17 je znázorněna motivace využití indexu při dotazování obsahu obrázkových dat.

Použití indexu (pokud existuje) určuje plánovač na základě optimalizovaného dotazu. Optimalizátor a plánovač je v databázových systémech odvozen od standardních. Vzhledem k velkému množství typů MM dat a metadat, jejich indexů, dotazovacích jazyků (relační, objektově-relační, objektové) a množstvím různých produktů, nicméně komplikovaný. Základem jejich MM přizpůsobení je především nezpracovávat data, která (v daný moment) není třeba, nebo nelze prezentovat.

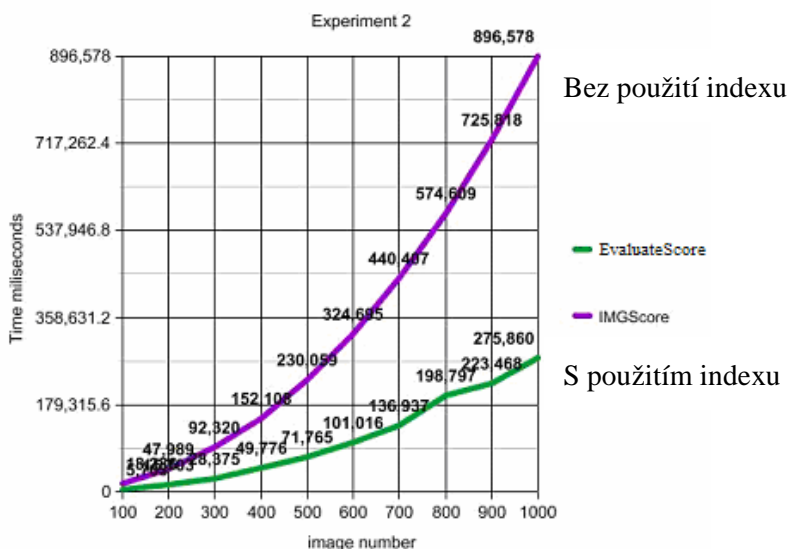


Fig. 2.17 Motivace využití indexu v databázi Oracle interMedia [Gir06]

Nejjednodušším způsobem indexování obsahu je užití předepsaných klíčových slov (label), například hokej, rock. V běžném textu se slova mohou opakovat a je nutné uvažovat jak synonyma, tak homonyma (řeší se stromem).

Nejznámější strukturou pro indexování textových dat, je frekvenční tabulka. Řádky představují jednotlivá klíčová slova (terms), což jsou všechna slova vyskytující se v uložených dokumentech (wordlist), často bez nejběžnějších slov (a, to) zvaných stoplist, nebo je použit jen kmen slova (slov). Sloupce pak tvoří jednotlivé dokumenty, respektive počet klíčových slov v daném dokumentu. Je možné použít její modifikaci pomocí řídké tabulky, jak je naznačeno v tabulce Tab. 2.1.

Tab. 2.1 Ilustrace řídké frekvenční tabulky

(Kmen) slova	Dokumenty obsahující slovo (počet)
Myš	Veterina.odt (39), Deratizace.pdf (19), Periferie.xls (3)

Indexovat lze i další obsah, například prostřednictvím vektoru rysů. Pokud jsme schopni vytvořit vektor rysů s konstantním počtem číselných parametrů (barevný histogram, textura), je možné použít klasické indexové struktury, jako jsou B^+ stromy nebo bitmapový index [Zen05].

Tradičním požadavkem MMDBs je indexování (anotovaných) objektů. Problémem multimediálních dat je automatické nalezení těchto objektů (nastíněno v kapitole 2.15) a především jejich vícerozměrná povaha. Nelze tedy použít klasické metody, ale je možné použít struktury založené na dělení prostoru.

2.9.1 K-D stromy

K-D stromy slouží pro ukládání bodových k-dimenzionálních dat. Dělí prostor podle vkládaných bodů. Při stavbě stromu indexu vložíme první bod. Do levé části stromu dáme všechny další body, které mají menší hodnotu souřadnice x a do pravé větvi. Při vkládání dalšího bodu se prostor rozdělí podle další souřadnice viz Fig. 2.18. Nevýhoda K-D stromů je špatné pořadí při vkládání, výhodou je snadná implementace.

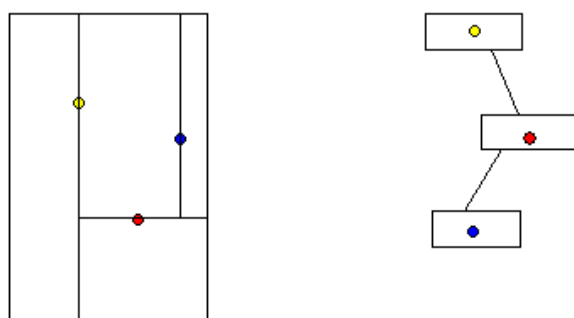


Fig. 2.18 Dělení 2D prostoru pomocí K-D stromů [Nos06].

2.9.2 Quad stromy

Quad stromy vycházejí z K-D stromů, ale vkládané body rozdělují prostor ve všech směrech, viz Fig. 2.19. Quad stromy umožňují oproti K-D rychlejší vyhledávání, ale operace rušení je náročnější, kvůli nalezení uzlu, který má být nahrazen.

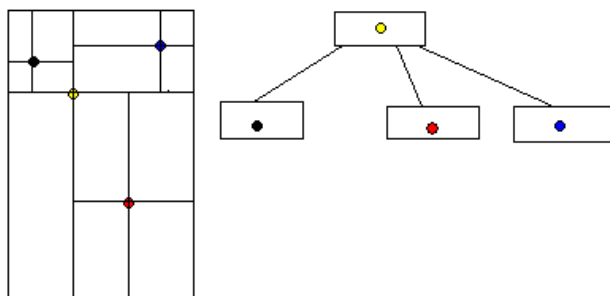


Fig. 2.19 Dělení 2D prostoru pomocí Quad stromů [Nos06].

Jejich modifikací jsou MX-quad stromy. Také dělí prostor podle všech souřadnic, ale jen tak, aby vzniklé části byly stejně velké. Dále dělí rekurzivně. V těchto stromech je pevně určena granularita, problémem je její efektivní určení.

Zobecnění K-D a Quad stromů jsou BSP (Bojary Space Partition), které obdobně rozdělují prostor libovolnými rovinami na konvexní oblasti, obsahující maximálně jeden objekt, případně jeho část.

2.9.3 R-stromy

R-stromy jsou založeny na ukládání obdélníkových struktur. Některé obdélníky představují reálné objekty vložené do databáze. Další sdružují menší obdélníky do větších, viz Fig. 2.20. Všechny uzly stromu, mimo kořenového, musí být alespoň z poloviny zaplněné. Pro neobdélníkové objekty se hledá minimální ohraničující obdélník (Minimum Bounding Rectangle, MBR). Modifikací R-stromů jsou M-stromy s kruhovými (kulovými) oblastmi, což je výhodnější pro určování eukleidovských vzdáleností, ale výpočetně náročnější.

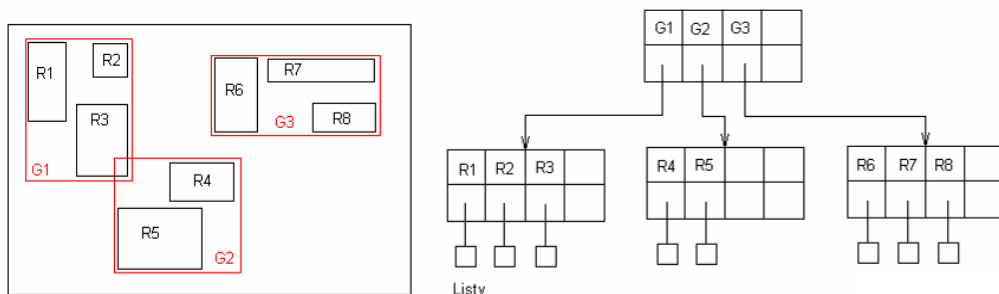


Fig. 2.20 R-stromy a dělení 2D prostoru z [Zen05].

R-stromy jsou nejvhodnějším algoritmem pro velké objemy dat. Problémem je, že se obdélníky mohou překrývat. Při vyhledávání potom nelze vyloučit určité větve, ale musí se prohledávat větší část stromu.

To řeší lépe až jejich modifikace. R^+ stromy vylučují překrývání, ale obdélník reálného objektu může být zařazen do několika listů. Běžnější jsou R^* stromy, které se při výběru uzlu pro zařazení a štěpení snaží překrytí minimalizovat [Sub98].

2.10 Ukládání

Základní vlastnost databáze je uchování dat. Pro obrovské objemy multimediálních dat to představuje značné problémy. Zatímco pro klasická strukturovaná data je jejich umístění v datovém souboru s transakční kontrolou samozřejmé, pro multimedia je kvůli

zvýšeným výkonnostních, fyzickým a finančním důvodům situace složitější. Mohou být uloženy uvnitř databáze, na streamovacím serveru anebo jinak distribuovaně:

- Uvnitř databáze jako objekt (OO DB), nebo jako BLOB (Binary large Object) v relační databázi s transakční kontrolou.
- V externím binárním souboru (BFILE), na diskovém poli nebo streamovacím serveru, který spravuje pouze DBMS, s transakční kontrolou.
- V externím souboru (souborech), ke kterým nemá MMDBMS výlučný přístup, je možné transakční kontrole podrobit pouze ukazatele (jména souborů).
- Na libovolném HTTP, FTP nebo RTSP serveru. Uchovává se jen URL.
- Na jiném (distribuovaném, streamovacím) zařízení, např. kamera s dočasnou pamětí.
- Na „pružném“ či optickém disku, pásce, nebo jiné paměti, bez možnosti okamžitého vyhledání a doručení.

2.11 Přenos

Z hlediska lidského vnímání je 1/4s ztráta (zamrznutí) video signálu omluvitelná, zvukového už ale těžko. V reproduktorech nepříjemně zapraská nebo se začne cyklicky opakovat poslední přijatá frekvence. Fakt, že MM SŘBD musí zajistit přenos multimedií pro plynulé přehrávání s dostatečnou kvalitou je dnes samozřejmostí.

V tomto ohledu je transport dokumentů a statických obrázků jednoduchou záležitostí (prostřednictvím HTTP, FTP). Nicméně i jejich přenos je vhodné optimalizovat. Příkladem je progresivní kódování JPEG, kde jsou nejprve zasílány hrubé informace o vzhledu celého obrázku a postupně upřesňovány.

Na distribuci dalších (real-time, continuous) multimediálních dat bývají kladeny různé požadavky, proto je jejich obsah distribuován více způsoby. Obecně je možné je zařadit dle míry jejich interakce:

- Žádná interakce. Do této kategorie spadá televizní a rozhlasové vysílání v libovolné formě – pozemní, satelitní vysílání, obecní rozhlas nebo internetové stanice. Jediná možnost interakce je stanici (program, kanál, stream) vypnout, resp. přepnout. Proto se používá plošné vysílání (broadcast), v Internetu multicast.
- Obsah přístupný na přání (na vyžádání, on demand). Řeší MMDBMS např. vyrovnávací paměť. Pokud nepožadujeme vyhledávání, lze použít streaming server.
- Interaktivní komunikace. Příkladem jsou telefonní hovory nebo videokonference. Jsou vyžadovány speciální protokoly, komunikační aplikace a (DB) servery pro navázání spojení, které může být přepínané, ale i paketové.

S přenosem multimedií souvisí nutnost v síťovém prostředí zabezpečit dostupnost dat využitím vyrovnávacích pamětí a dostatečné přenosové kapacity. Především z důvodu real-time přenosu multimediálních dat spolu s klasickými byla vyvinuta přepínaná síť ATM. Vzhledem k vyšší ceně ale nebylo její nasazení široce realizováno. Pro úsporu přenosového pásma byly vyvinuty také protokoly nad IP, zejména pro multicast, viz předměty PDS [Švé05] a PSI. V současné době je jejich hlavním problémem absence univerzálního mechanismu pro zajištění dostatečné přenosové rychlosti (Quality of Service, QoS).

Audiovizuální obsah je nutné často přenášet prostřednictvím nejrůznějších technologií s různou přenosovou kapacitou (DVB, LAN, DSL, 3GP nebo GPRS¹³). Je tedy nutné

¹³ Digitální televizní vysílání, počítačové a mobilní sítě

multimédia překódovat na kvalitu – datový tok – kvalitu, dle rychlosti dostupného připojení. Je možné data uložit vícekrát v různých formátech s různou kvalitou. Rozumnější někdy může být užití jediného formátu pro uskladnění všech typů médií, ze kterého se překóduje požadovaný výstup dle požadavku přehrávacích zařízení (3GP od 176 x 144 px, oproti HDTV až 1920 × 1080 px), odpovídající přenosové rychlosti a požadované kvality služeb QoS [Kos03].

Prvním (1996) a v současné době ve verzi 5 (2003) nejpoužívanějším real-time protokolem pro Voice over IP (VoIP), video a dokumenty je International Telecommunication Union (ITU) H.323. Pokrývá jak paketovou komunikaci (IP), tak přepínanou (ATM, telefoní síť). Je navržen pro univerzální video konference (point to point i multipoint) [ITU06].

Video kodekem je standard příbuzný MPEG-2, H.261 (1990), s rozlišením CIF (352 x 288 px) nebo QCIF (176 x 144 px), jeho datový tok se pohybuje v násobcích 64kbps. Novější verze je označena jako H.263 a H.264 (digitální vysílání HDTV), využívá MPEG-4. Audio kodeky jsou: G.711 s datovým tokem 64 kbps (PCM), G.723.1 s 5-6 kbps (ACELP) nebo GSM 06.10 s tokem 13 kbps (RPE-LTP). Kontrolní zprávy jsou zde zprostředkovány speciálním kanálem H.245.

Protokol H.323 je použit například sítěmi ISDN¹⁴, nebo mobilními sítěmi 3. generace (3GP, UMTS). Ilustrace je uvedena na následujícím obrázku Fig. 2.21.

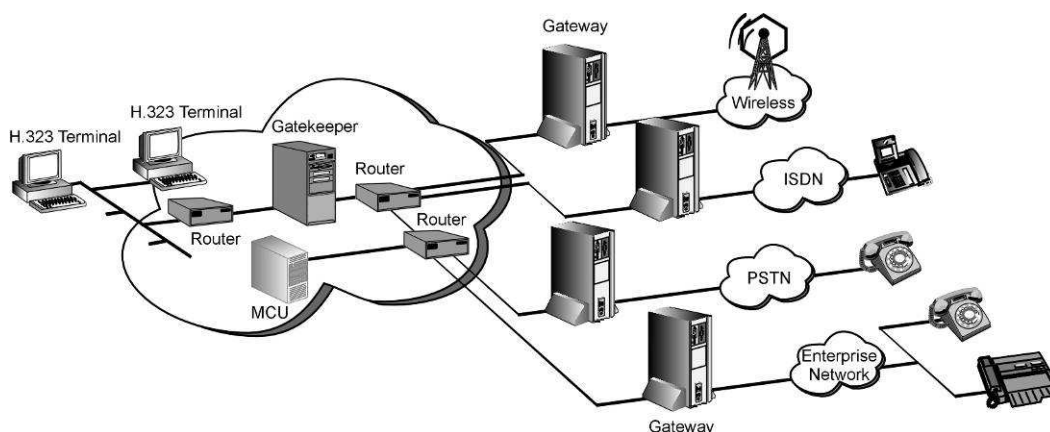


Fig. 2.21 Packet-based multimedia communications systems H.323 [RC006]

Jednodušší alternativou H.323 pro přenos hlasu (Voice over IP, VoIP) a videa je Session Initiation Protocol (SIP, RFC3261) [TIS02], skupiny Internet Engineering Task Force (IETF), která stojí za internetovými standardy IP, TCP, UDP, HTTP, SMTP a podobně. Firemní (Avaya) verze SIP je použitý například v telefonní síti VUT.

Existují i další IP protokoly pro real-time přenos MM dat, například Real-time Transport Protocol (RTP), který zajišťuje synchronizaci (ne QoS) pro streamování například MPEG-4. Používá se ve spojení s protokolem RTPS například pro streamování přednášek na FIT (rtps://video1.fit.vutbr.cz:5555/zaznamX).

Digitální televizní vysílání – DVB (Terestriální, Satelitní i Kabelové) je plně v režii MPEG-2. MPEG-1 a MPEG-2 interně obsahují obdobný předpis pro synchronizaci těchto (MPEG) dat. MPEG-4 využívá část 1. (systémy, ISO/IEC 14496-1), část 6. (MM distribuční integrační rámec) pro synchronizaci a část 8. pro přenos přes paketové sítě. Pro HDTV digitální televizní vysílání se MPEG-4 používá vestavěně v H.264, protože se čeká na nový standard MPEG-21.

¹⁴ Integrated Services Digital Network, digitální přepínané telefonní služby

2.12 Prezentace

Prezentace je jednou z nejlépe vyřešených oblastí v MM, které technologicky předbíhá ostatní. Příkladem je schopnost libovolného kodeku MPEG-4 zobrazit 3D objekty, jejichž automatický popis v průběhu komprese zřejmě ještě neexistuje. Prezentace se obvykle rozděluje na audio a vizuální část, do které mohou spadat i metadata.

Pokud máme zajištěno médium, jsme schopni AV obsah s dostatečnou kvalitou doručit, navzájem synchronizovat, pak jeho přehrávání můžeme přenechat klientským zařízením (telefon, televize, přehrávače) nebo knihovnám a standardním aplikacím operačního systému nebo webového prohlížeče.

Jediným problémem může být, pokud daný systém nezná potřebný kodek nebo protokol. Problém řeší známé programy pro zobrazení a přehrávání, jako jsou <http://www.xnview.com/>, <http://www.winamp.com/>, <http://www.videolan.org/>, <http://www.real.com/>, <http://www.apple.com/quicktime/>, nebo jejich alternativy.

2.13 Bezpečnost a ochrana

Bezpečnost je nezbytným prvkem každého počítačového systému. Aplikace multimediálních systémů jsou citlivé například v oblastech zdravotnictví, průmyslu, vývoje a výzkumu nebo při tvorbě jiných autorských děl. Pro bezpečnostní MM systémy, například letiště, je tento bod stěžejním. Proto musí také MMDBMS zajistit ochranu dat způsobem, [Han00]:

- aby k nim měly přístup pouze oprávněné osoby (autentizace a autorizace),
- aby se zpracovávaly nefalšované informace (integrita a autenticita),
- aby se dalo zjistit, kdo je vytvořil, změnil nebo odstranil (nepopiratelnost),
- aby nebyly nekontrolovaným způsobem vyraženy (důvěrnost),
- aby byly dostupné vždy, když jsou potřeba (dostupnost).

Pod první bod spadá také ochrana intelektuální vlastnictví a nelegální kopírování obsahu. Pro některé osoby, společnosti a organizace jsou znalosti, patenty, nebo autorská práva jediným zdrojem příjmů.

Tyto vlastnosti MM databází zřejmě nelze garantovat obyčejným souborovým systémem, ale je nutné použít adekvátní MMDBMS.

Navíc je třeba dát pozor na sociální dopad výsledků z bezpečnostních AV dat (Velký bratr, 1984). To platí i pro návrháře, protože tato oblast je legislativně regulována – Zákon č. 101/2000 Sb., o ochraně osobních údajů, v aktuálním znění, Stanovisko č. 4/2004 ke zpracování osobních údajů prostředky kamerového sledování, Úmluva Rady Evropy č. 108/1981 o ochraně jednotlivců se zřetelem na automatizované zpracování osobních dat a směrnice 95/46/ES.

2.14 Architektury multimediálních systémů

Architektura libovolného systému je závislá především na konkrétní požadované aplikaci a je velmi podobná klasickým databázím, pokud jsou podobné požadavky. Ilustrace obecné architektury je uvedena na obrázku Fig. 4.1.

Může se jednat o obyčejný prohlížeč fotek na lokálním počítači a nepříliš vzdálených (optických) médiích. Obecně je ale možné říci, že se jedná o aplikaci typu klient-server. V nejjednodušším případě je klient tak tlustý, jak to jen jde – zprostředkovává všechny operace. Toto řešení ale není příliš univerzální a snažíme se přesunout co nejvíce operací na server obsluhující požadavky více klientů, lépe ale na systém řízení báze dat.

Proto je často médii a společnostmi poskytujícími MM obsah použita nejméně třívrstvá architektura systému – webový, aplikační nebo streaming server (front end) a MM DBMS – pro správu metadat a dat, které mohou být uloženy i na file serveru (FTP). Klienti této architektury jsou poté producenti i konzumenti multimedií. Příkladem je Oracele *interMedia* na obrázku Fig. 4.2.

V poslední době je nicméně začínají vytlačovat distribuované systémy, s mnoha zdroji a poskytovateli obsahu, založené například na MPEG-21, které umožňují přístup mnoha zařízení (televize, mobilní telefony apod. viz Fig. 2.21) k mnoha zdrojům včetně nějakých neexplicitních informací, získaných například z bezpečnostních kamer nebo medicínských snímků.

2.15 Aplikace získávání znalostí

Získávání znalostí¹⁵ je proces, jehož vstupem jsou běžná (AV) data a výstupem nové, zajímavé a potenciálně užitečné informace. Získávání znalostí je interaktivní, což znamená, že si uživatel vybírá požadované znalosti podmínky, při jejichž změně se proces opakuje. Je tedy iterativní. Proces se skládá z několika částí, jako čištění, integrace a transformaci dat, což zahrnuje extrakci rysů, ale především dolování dat a jejich následnou evaluaci a prezentaci.

Dolování dat¹⁶ je souhrnný název skupiny metod založených na umělé inteligenci, které data klasifikují, shlukují, objevují zajímavé vlastnosti, vztahy, změny a vybírají z nich jen to podstatné, co nás zajímá nebo překvapí.

Důležitou oblastí získávání znalostí je nalezení vztahu mezi syntaxí (černý flek) a sémantikou (pes). Z tohoto pohledu se dělí dle rovin aplikace:

- Obecné aplikace se snaží získávat znalosti ze vstupních dat, o kterých nemají žádnou apriorní znalost a nejsou schopny odvodit mnoho o sémantice.
- Speciální aplikace využívají účelových zdrojových dat, o kterých je známo kde byly získány a stanovit, jaké znalosti lze dolovat zvoleným přístupem. Je možné alespoň přibližně sestavit znalostní bázi, použitelnou algoritmy pro učení (jak vypadá pes).

2.15.1 Klasifikace

Klasifikace je proces, který hledá modely, popisující a odlišující známé třídy dat, tak aby bylo možno zařadit i neznámé objekty. Z tohoto důvodu není použitelná pro všechny roviny aplikace, ale pouze pro ty, kde jsme schopni určit znalostní bázi v podobě tříd.

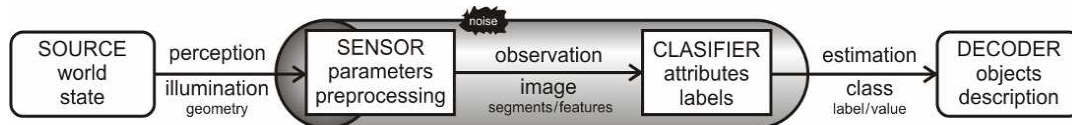


Fig. 2.22 Kanál klasifikace v počítačovém vidění [Chm006]

Pro ilustraci se nejlépe hodí naivní Bayesovská klasifikace, odvozená od počítačového vidění. Její proces je ilustrován na obrázku Fig. 2.23, formalizujícím Fig. 2.22. Zde y je skrytý stav, který si přejeme zjistit. Jediná informace, se kterou o daném objektu klasifikátor disponuje, jsou naměřená x , označovaná jako pozorování $\mu(w)$, které je ale zatíženo chybou. Obecně platí:

$$\text{Eq. 2.2} \quad \text{kompletní informace } (w) = \text{pozorovatelná } (\hat{w}) + \text{ztracená informace (šum } \tilde{w}).$$

¹⁵ Knowledge discovery in databases, KDD.

¹⁶ Data Mining, DM. Přesněji by se měl nazývat dobývání (informací) v datech.

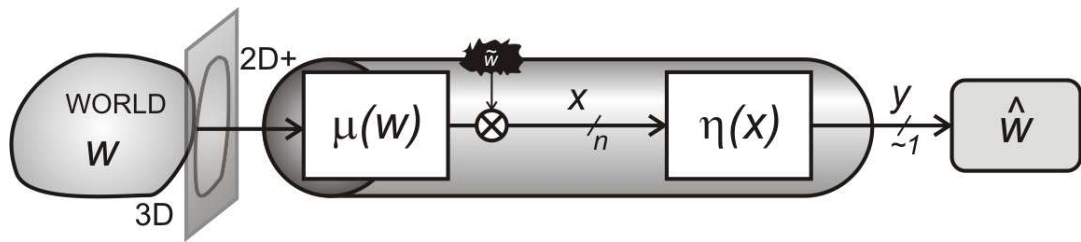


Fig. 2.23 Formální ilustrace klasifikace [Chm006]

Klasifikační funkce $\eta(x)$ obsahuje bázi dat, na základě které se rozhodne pro zařazení neznámého objektu, což je ilustrováno na následujícím příkladě.

Ex. 2.5 Příklad klasifikace neznámého hnědého, kulatého objektu

Tvar (x_1)	Barva (x_2)	Třída (y)	$P(y)$	$P(x_1/y)$	$P(x_2/y)$
<i>čtverec</i>	<i>hnědá</i>	<i>Čokoláda</i>	<i>0,33</i>	<i>0,90</i>	<i>0,90</i>
<i>kruh</i>	<i>červená</i>	<i>Jablko</i>	<i>0,33</i>	<i>0,45</i>	<i>0,70</i>
<i>kruh</i>	<i>oranžová</i>	<i>Pomeranč</i>	<i>0,33</i>	<i>0,45</i>	<i>0,70</i>
<i>?</i>	<i>?</i>	<i>?</i>	<i>0,01</i>	<i>0,10</i>	<i>0,10</i>

Před možností klasifikovat, neboli zařadit neznámý objekt do odpovídající třídy, je nutné znát charakteristické vlastnosti objektů této třídy. To je v případě *Čokolády* její *čtvercový* tvar a *hnědá* barva. V případě naivní Bayesovské klasifikace je proces učení jednoduchá statistika – *Čokoláda* se v trénovacích datech vyskytuje v 33% případů. To se označuje jako apriorní pravděpodobnosti výskytu objektu, $P(y)$. Při trénování se mohou vyskytovat další objekty, které nelze (nebo není radno) klasifikovat – (?).

Podmíněná pravděpodobnost $P(x_1/y)$ říká, že když se jedná o *Čokoládu*, bude pozorován *čtverec* s 90% pravděpodobností. Tato pravděpodobnost se nazývá likelihood.

Naivní Bayesovská klasifikace, ve variantě maximalizace posteriorní pravděpodobnosti (Maximum A Posteriori, MAP), se snaží maximalizovat pravděpodobnost výskytu dané třídy pomocí její likelihood a pravděpodobnosti výskytu prostým násobením:

$$\text{Eq. 2.3} \quad \eta^*(x) = \arg \max_x P(x | y)P(y)$$

Tedy v našem příkladě se pokusíme určit třídu *hnědého kruhu* objektu pomocí MAP:

$$P(\text{kruh} | \text{Pomeranč}) \cdot P(\text{hnědá} | \text{Pomeranč}) \cdot P(\text{Pomeranč}) = 0,45 \cdot 0,10 \cdot 0,33 = 0,0149$$

$$P(\text{kruh} | \text{Jablko}) \cdot P(\text{hnědá} | \text{Jablko}) \cdot P(\text{Jablko}) = 0,45 \cdot 0,10 \cdot 0,33 = 0,0149$$

$$P(\text{kruh} | \text{Čokoláda}) \cdot P(\text{hnědá} | \text{Čokoláda}) \cdot P(\text{Čokoláda}) = 0,1 \cdot 0,9 \cdot 0,33 = 0,0297$$

Maximalizací výsledku zjistíme, že se jedná o kulatou *Čokoládu*.

Bayesovská klasifikace se pro svoji jednoduchou interpretaci využívá také v teorii řízení. Její regresní (odhad spojitě hodnoty) varianta je ale výpočetně složitější a teoreticky se neobejde bez složité mnohorozměrné integrace [Chm006]. Ta se v praxi nicméně obchází – vypočítá se střední hodnota a směrodatná odchylka diskretizované veličiny.

Pro klasifikaci a regresi existuje množství modifikací, které neobsahují naivní předpoklad, například Gaussovské, pracující s Eukleidovskou vzdáleností a

Gaussovským (rovnoměrným) rozložením pravděpodobnosti, původně využívané pro pozorování hvězd [Sor70].

SVM Obecným algoritmem pro tyto aplikace jsou kernelové metody, například Support Vector Machine. SVM se snaží generalizovat data a lineárně separovat jednotlivé třídy (bodů v mnohorozměrném prostoru – vektorů) od sebe separační hyper-rovinou Fig. 2.24, kterou je možné representovat skalárním součinem¹⁷ vektorů:

Eq. 2.4 $\langle w, x \rangle + b = 0$, pak klasifikace lze vyjádřit jako $f(x) = \sum \alpha_i y_i \langle x_i, x \rangle + b$

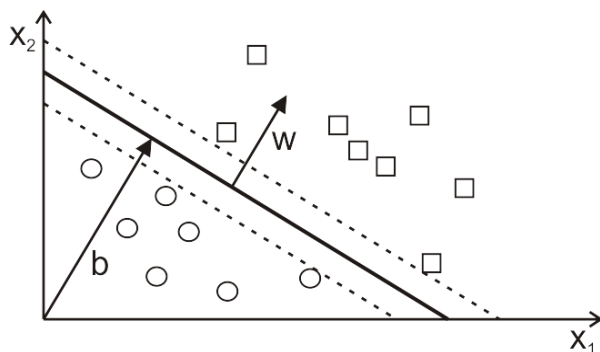


Fig. 2.24 Ilustrace lineární separace tříd pomocí SVM

V rovnici Eq. 2.4 jsou jednotlivé třídy y (čtverečky, kolečka) reprezentovány hodnotami $\{-1, 1\}$. Vektory nejbližší separační linii se nazývají support vectors, platí $\alpha = 1$.

Nalezení rovnice rozhodovací funkce, která maximalizuje vzdálenost mezi třídami, je závislá pouze na skalárním součinu vektorů, nikoli jejich poloze [COV95]. To je velice výhodná vlastnost, pokud není problém lineárně separovatelný.

V případě, že nelze třídy dat separovat, je nutné použít transformační funkci, zvanou kernel. Kernel K je skalární součin vektorů, které jsou transformovány funkcí $\phi(y)$ tak, aby byly lineárně separovatelné, jak je naznačeno v následující rovnici a obrázku.

Eq. 2.5 $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, pak klasifikace lze vyjádřit jako $f(x) = \sum \alpha_i K(x_i, x)$

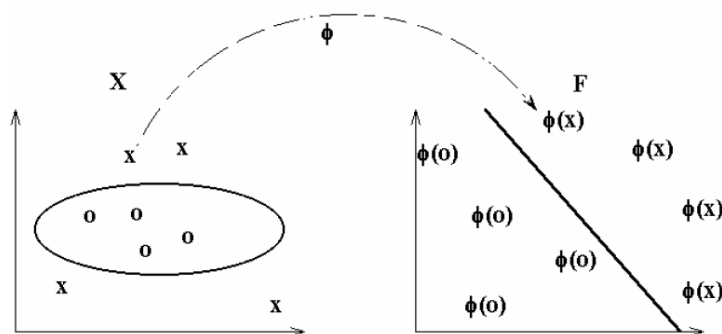


Fig. 2.25 Ilustrace transformace v kernelu pro SVM [predelat]

Příklady kernelů jsou například polynomiální funkce a RBF (Radial Basis Function):

Eq. 2.6 $K(x, z) = \langle x, z \rangle^n$, $K(x, z) = e^{-\|x-z\|^2/2\sigma}$

¹⁷ anglicky inner product, vektorů a, b : $\langle a, b \rangle = a^T b = \sum a_i b_i$ (v ortogonální bázi)

Učení SVM a další více či méně známé algoritmy jako rozhodovací stromy, hrubé množiny, fuzzy logika, Markovovy skryté modely (Markov Hidden Models, HMM) nebo Kalmanův filtr, je možné najít v [Fis06].

Vhodné je také využití neuronových sítí [Zbo05].

2.15.2 Shlukování a analýza

Shlukování (segmentace) je technika, která umožňuje seskupit podobné objekty dle míry jejich podobnosti, kterou je možné měřit za stejných podmínek jako v kapitole 2.8.1. Shlukování se snaží data zařadit do skupin, které předem nezná, respektive je vytvořit.

To, co brání shlukování v její naprosté autonomii, jsou dvě věci. První z nich je počet tříd shluků, který je možné stanovit i z maximální možné vzdálenosti dvou shluků, ale ta už je třeba nastavit manuálně, dle požadované aplikace. Zásadním problémem je dále inicializace těchto shluků. Metoda K-Means [Zbo04] určuje jejich prvotní rozmístění náhodně, což není ideální řešení, ale někdy se může podařit. Pro vizuální data bych raději volil například velikost gradientu (detekce hran) – místa s velkým gradientem mohou oddělovat oblasti. Obdobně i místa s nízkou hustotou prvků pro jiný typ dat.

Asociační analýza se snaží nalézt vztahy mezi analyzovanými daty. Ty se vyjadřují asociačními pravidly tvaru implikace $A \Rightarrow B$. Společně s metrikami podpora $S(A \Rightarrow B) = P(A \cup B)$ a spolehlivost $C(A \Rightarrow B) = P(B / A)$ určují, které hodnoty atributů se vyskytují v dané množině dat s jistou pravděpodobností společně. Více v [Han01].

Dolování asociačních pravidel z multimedií lze chápat jako automatizovanou klasifikaci se stanovenou mírou zajímavosti. Každý objekt lze považovat za transakci pro nalezení frekventovaně se nacházejících vzorů dat, nahrazujících položky. Zaměřím se na vizuální data:

- Vztahy mezi obsahem a popisem multimédia jsou pravidla typu: „Pokud je značná část horní poloviny modrá, zřejmě se jedná o oblohu.“ Tato kategorie je sblížením obsahově a popisově zaměřeného získávání informací, protože z obsahu obrázku nebo scény usuzuje na klíčové slovo a tím je obloha. Nebo naopak při klíčovém slově „ne“ lze z videa vypořadovat mírný pohyb hlavou.
- Asociace obsahu dat bez určení jejich umístění – do této kategorie náleží například výroky jako: „Pokud obrázek obsahuje dva modré čtverce, pak je zde obvykle i žlutý kruh.“ Protože se jedná o popis obsahu obrázku, ale žádný objekt zde není lokalizován.
- Asociace obsahu dat se specifikací jejich lokalizace: „Pokud je červený trojúhelník mezi dvěma modrými čtverci, pak jsou všechny umístěny ve žlutém kruhu,“ náleží k této kategorii, objekty jsou asociovány pomocí jejich prostorových vztahů.

Evoluční analýza popisuje „zvyky“ objektů, jejich změnu v průběhu času a snaží se odhalit jejich trendy. Okrajová analýza se naopak snaží hledat nezvyklé vzory v chování objektů, které jiné metody označí jako šum. Používají se metriky pro měření odchylky od tříd, například pro odhalení nekalého záměr nebo činu.

Získávání znalostí demonstruji prostřednictvím následujícího příkladu.

Ex. 2.6 Příklad získávání znalostí aplikované na fotbalový zápas [Chm06]

Získávání znalostí z videozáznamu fotbalového utkání je poměrně speciální záležitost, do jisté míry je možné popsat fotbalový zápas jednoduchými pravidly, kterým je schopen porozumět jak fanoušek Baníku, tak i počítač. Kamery zabírají reálnou zelenou plochu, která je rozdělena a ohraničena bílými liniemi. Mimo ně nás záznam obvykle nezajímá. Na hřišti je několik typů objektů – hráči (2x10 odlišné barvy), statické branky

s brankáři libovolné (šedé) barvy a míč se specifickými vlastnostmi. Rozhodčí, případně ostatní postavy klasifikovat nebudeme. Ústřední událostí je míč v brance. Předmětem získávání znalostí z multimediálních a obecně časoprostorových databází je například to, kdo se snaží dostat míč kam, jakým způsobem se mu to daří, kdo přihrává komu, jak dlouho drží míč, kde a vyvodit z toho jisté závěry.

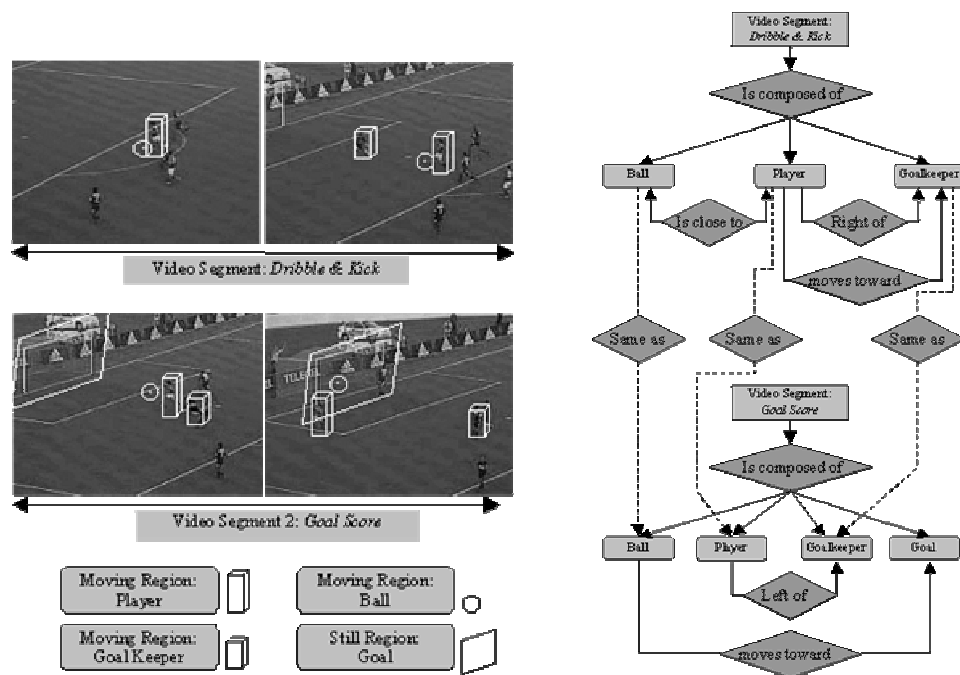


Fig. 2.26 Příklad rozpoznání objektů videa a jeho reprezentace z [M704]

„Jak lze dosáhnout popisu jako na obrázku Fig. 2.26?“ Extrakci těchto rysů lze rozložit do několika částí:

- Definice zdrojů a cílů, je nutné vytvořit znalostní bázi obsahující syntaxi (jaké objekty se v obraze mohou vyskytovat, jak vypadají, vytvořit jeho modely) a navázat na ně sémantiku (co daný objekt reprezentuje) informací, které se v obraze mohou vyskytovat. Příkladem je popis o odstavce výše.
- Extrakce objektů – každý pohyblivý region může být označen za objekt a/nebo rozdělen do podmnožin, které budou analyzovány samostatně, ať už se týká vektoru pohybu (prolínání hráčů, ruce, nohy) nebo jejich popisu jako statických obrázků. MPEG-7 poté nabízí rámec pro uložení těchto časoprostorových dat a jejich vzájemné závislosti.
- Transformace do souřadnic reálného světa – definice a vytvoření snímaného prostoru, specifikace polohy a pozice senzorů (kamer) a transformace detekovaných objektů v obraze do reálných souřadnic za pomoci stacionárních bodů na hřišti (lajny, branky).
- Vytvoření databáze pohybujících se objektů na různých úrovních – dle MPEG-7, ale je možné vytvořit i časoprostorovou databázi a do ní ukládat pohyby jednotlivých objektů tak, aby z nich bylo možné snadno vyvodit další informace.

Z této databáze lze vyvozovat znalosti následujícími přístupy:

- Klasifikace a predikce. Může sloužit na jedné úrovni pro estimaci pohybu a jeho vyhodnocení například v okamžiku, kdy kamera nezabírá dané objekty, nebo na globální úrovni může odhadnout výsledek zápasu Most – Liberec na 0 : 2.

- Shlukování má obecně široké uplatnění, může například rozřazovat hráče podle jejich příslušnosti do týmů nebo skupin (útok, obrana). Na nižší úrovni slouží pro extrakci pohyblivých regionů.
- Asociační analýza dokáže například odhadnout, že pokud při zápasu Sparta – Arsenal nahraje míč Pirés (7) Henrymu (14), pak padne do soupeřovy branky s pravděpodobností 66%.
- Evoluční analýza popisuje obvyklé chování objektů a mohla by například být schopná odhadnout, komu nahraje daný hráč a s jakou pravděpodobností nebo odhadnout výsledek utkání podle první poloviny.
- Okrajová analýza by naopak měla být schopná rozpoznat různé nestandardní situace, jako je zranění hráčů nebo změna taktiky.

Zamyšlení ke 2. kapitole

Jaká data se vyskytují v databázích?

Pro jaká data není vhodné použít relační databázi?

Jak je to s jejich strukturou? Představte si data například pro medicínu.

Znáte operace, které se provádějí na těchto datech?

Pro jaké účely jsou vhodné (multimediální) databáze?

Popište vyhledávání. Dle čeho je možné vyhledávat?

Co je nutné udělat před vyhledáváním?

Znáte popis multimediálních dat? Jaký existuje a jak lze získat?

Stručně shrňte požadavky na systém řízení báze multimediálních dat.

Jaká znáte multimediální data?

Jaké formáty se používají pro obraz, zvuk, video. Jaké pro dokumenty?

Co je to mediální kontejner?

Co jsou to metadata? Jaké typy a metadat znáte?

K čemu slouží MPEG-7? Co definuje?

Jaká média popisuje MPEG-7? Jak mohou být uložena?

K čemu slouží MPEG-21?

Co jsou to rysy, jak je lze získat a k čemu slouží?

Jaké rysy je možné extrahovat z audia, obrázků, videa nebo textu? Pro jaké aplikace?

Vysvětlete rozdíl ve vyhledávání dle obsahu a popisu.

Co je to podobnostní vyhledávání, vysvětlete jeho principy.

Co je to index, k čemu slouží a jaké techniky se využívají.

Pokuste se popsat možnosti ukládání a jejich výhody.

Jak je možné přenášet média? Jakým prostředím a protokolem?

Popište bezpečnostní opatření potřebná pro kritické aplikace a jiné zvláštnosti.

Zkuste načrtnout architektury multimediálních systémů.

Naznačte typy znalostí extrahovatelné z multimedií.

Jaký je rozdíl mezi klasifikací a segmentací?

Jaký je rozdíl mezi Bayesovskou klasifikací a SVM?

3 Databázové technologie a standardy

Následující kapitola se zabývá databázemi obsahujícími komplexní typy dat, jako jsou složitě strukturované objekty, prostorová, temporální, hypertextová a především multimediální data a také technikami pro přístup k těmto databázím.

Relační model je založen na teorii množin. Entita reálného světa je reprezentována relací, podmnožinou kartézského součinu domén, množin atomických hodnot. Data jsou uložena jako n-tice hodnot těchto domén. Více v [Zen05].

Stávající rozšíření relačních systémů řízení báze dat (DBMS) dokazuje, že jsou pro většinu aplikací, především obchodních, stále plně dostačující. Nicméně pro složitější aplikace je tato „plochá“ organizace jednoduchých dat velmi omezující:

- Omezení první normální formou (atomičnost) nedovoluje odvozené, komplexní a vnořené (nested) struktury jako pole nebo množiny.
- Oddělená data a operace pro jejich manipulaci.
- Omezený počet vestavěných datových typů a složité dotazy SQL.

Z těchto důvodů je nutné adoptovat koncepty objektově orientovaného (OO) paradigmatu, které umožňuje lepší korespondenci mezi složitě strukturovanými reálnými daty. To je vyžadováno v mnoha oblastech, jako je grafika, kartografie, medicína, fyzika nebo chemie [Cha01]. Hlavní předností objektových a objektově-relačních databází pro multimediální databázové systémy je možnost specifikovat komplexní datové typy a operace pro jejich manipulaci nezávislé na dotazovacím jazyce.

Objektový model dat v databázových systémech vychází ze známých principů objektově orientovaného modelování a programování. Objekty jsou použity pro modelování entit reálného světa. Mají svoji unikátní identitu a tvoří kolekce. Třídy jsou využívány pro definice množin objektů. Objekty téže třídy mají stejnou strukturu i operace (procedury a funkce se nazývají metody). Třídy jsou organizovány ve zájemně provázané hierarchii a nahrazují databázové schéma.

Objektově orientovaný přístup je však dále obohacen o techniky perzistence, reprezentace vztahů, dotazování, transakčního přístupu a tak podobně.

Persistentní znamená (objekt) existující bez časového omezení, do té doby než je úmyslně zničen. Úmyslné zničení je referenční nebo kaskádová akce (mazání, destrukce), nikoli ukončení transakce, sezení nebo aplikace [SQL03].

3.1 Objektově orientovaný koncept

Požadavek na perzistentní existenci komplexních datových typů a vztahů mezi nimi, jako rozšíření objektově orientovaného programování, byl zprvu řešen velkým množstvím vzájemně nekompatibilních řešení. To výrazně ovlivnilo jejich nasazení v praxi. Proto byla malými výrobci objektových databází založena skupina Object Database Management Group (ODMG), jejímž cílem bylo vyvinout nezávislý standard pro manipulaci a uchování objektů v databázích. Tato snaha vedla k výsledku v podobě knihy Object data standard: ODMG 3.0 [Cat00], v současné podobě třetí verze tohoto standardu. Skupina ODMG se v roce 2001 přidružila ke skupině Object Management Group (OMG), tvůrci architektury CORBA a správci UML.

Implementace je definována pro objektové programovací jazyky C++ a Smalltalk. Standard ODMG pro jazyk Java – Java Data Objects (JDO) je spravován společností Sun Microsystems. JDO je navrženo tak, aby bylo možné se věnovat objektovému modelování a programování a starosti s persisterací přenechat implementaci JDO.

Object Data Standard sestává ze tří částí – referenční objektový model (ODMG Object Model), definiční jazyk Object Definition Language (ODL) a dotazovací jazyk Object Query Language (OQL).

3.1.1 Objektový model ODMG

Specifikuje standardní model dat pro objektově orientované databáze. Je definován jako nadmnožina objektového modelu OMG (Object Management Group). V následujícím přehledu jde hlavně o popis perzistentního rozšíření pro objektové databáze.

Objekt je jednoznačně identifikovatelná entita reálného světa, má stav a chování. Stav sestává z hodnot vlastností – atributů a vazeb mezi objekty, chování je specifikováno pomocí operací (metod). Ne všechno v objektové databázi je objekt – literály (např. čísla) nemají, na rozdíl od objektů, jednoznačný Object Identifier (OID), jsou reprezentovány vlastní hodnotou [Cha01]. Množina objektů stejného typu – kolekce, také není objekt [Hru05]. Objektové databáze, na rozdíl od relačních, umožňují odkaz na základě identity OID. Z toho plyne také dvojí pojem rovnosti – na základě identity a stavu – rekurzivní hodnoty vnořených atributů.

Data a operace nad nimi jsou zapouzdřeny do jediné struktury: objektu. Zapouzdření poskytuje logickou nezávislost dat a je důležité pro abstraktní třídy. Z tohoto pohledu se objekt skládá z (několika) rozhraní a (jedné) implementace. Ne všechny vlastnosti jsou viditelné i z vnějšku objektu, rozhraní zpřístupňují pouze ty veřejné (public). Chování objektu zajišťují metody, které sestávají z hlavičky, ta specifikuje jejich parametry a typ případného výsledku, včetně jejich implementace v libovolném výpočetně úplném OO programovacím jazyce. V objektových databázích tedy není dotazovací jazyk (OQL) nezbytný [Cha01].

Třída je šablona pro tvorbu objektů, specifikuje strukturu atributů a množinu metod. Typ objektu je obecnější pojem, reprezentovaný obvykle jménem třídy. Každý objekt je nějakého typu a má s ostatními objekty stejného typu stejnou množinu vlastností a operací. Objektové databáze poskytují množství předdefinovaných typů (např. pro geometrické obrazce), které je možné libovolně rozšířit. Organizace (vztahů) tříd v agregační hierarchii představuje schéma tříd, například v UML diagramu databáze.

Třídy mohou mít vlastní proměnné a operace, například počet instancí nebo operátor new(). Kolekce všech instancí generovaných jednou třídou se nazývá extent. Důležitou vlastností je druh jejich perzistence:

- Implicitní vlastnost všech instancí třídy. Objekt je automaticky uložen do databáze při jeho vytvoření. Tyto systémy mají tedy dva typy konstruktorů – pro vytváření perzistentních a temporálních instancí tříd.
- Vytvoření objektu nemá vliv na jeho perzistenci. Ta je zajištěna například vložením do perzistentní kolekce.

Obdobně je to s mazáním objektů a integritou:

- Persistentní objekt je smazán až po smazání všech referencí na něj. Například pomocí garbage collection.
- Objekt může být smazán explicitně a porušit tak referenční integritu. Ta musí být zajištěna na aplikační úrovni.

Persistentní objekty mají svoji jednoznačnou identitu OID po celou dobu existence, nemění se tedy ani v případě změny příslušnosti ke třídě. Evoluce objektu se nazývá migrace.

Dědičnost umožňuje odvodit definici stavu a chování (pod)typu z definice jiného (nad)typu. Jednoduchá dědičnost mezi třídami je používána jako rozšíření (extends).

Pokud je nadtypů více, mluví se o vícenásobné dědičnosti. Systémy, které ji podporují, musí umět řešit případné konflikty. ODMG umožňuje vícenásobnou dědičnost chování (ISA) pouze v případě, že se nedědí dvě stejně nazvané operace náležící různým typům.

Dědičnost znamená, že je možné použít (nahradit) instanci podtypu tam, kde je požadován nadtyp, čímž vytváří konzistentní hierarchii. Pro implementaci znamená výhodu sdílení a znuvupoužitelnost kódu mezi třídami. Operace mají stejné jméno, ale pro každou třídu je možné vytvořit specifickou implementaci (overriding). Výběr polymorfni nebo typově přetížené (overloading) operace se provede za běhu programu (late binding). Dědičnost má dále za následek klasifikační hierarchii mezi extenty jednotlivých tříd a využívá se pro modelování jako specializace a generalizace.

3.1.2 Definiční jazyk ODL

ODL definuje objektové databázové schéma odpovídající objektovému modelu. Součástí této části je i jazyk pro výměnu objektů a jejich zálohování na disk Object Interchange Format (OIF), ale na jeho místě se ujal univerzální jazyk XML.

Namísto referenčního přehledu ODL [Cat00] uvedu příklad multimediální objektové databáze pro řízení projektů. Její schéma tříd je uvedeno na následujícím obrázku.

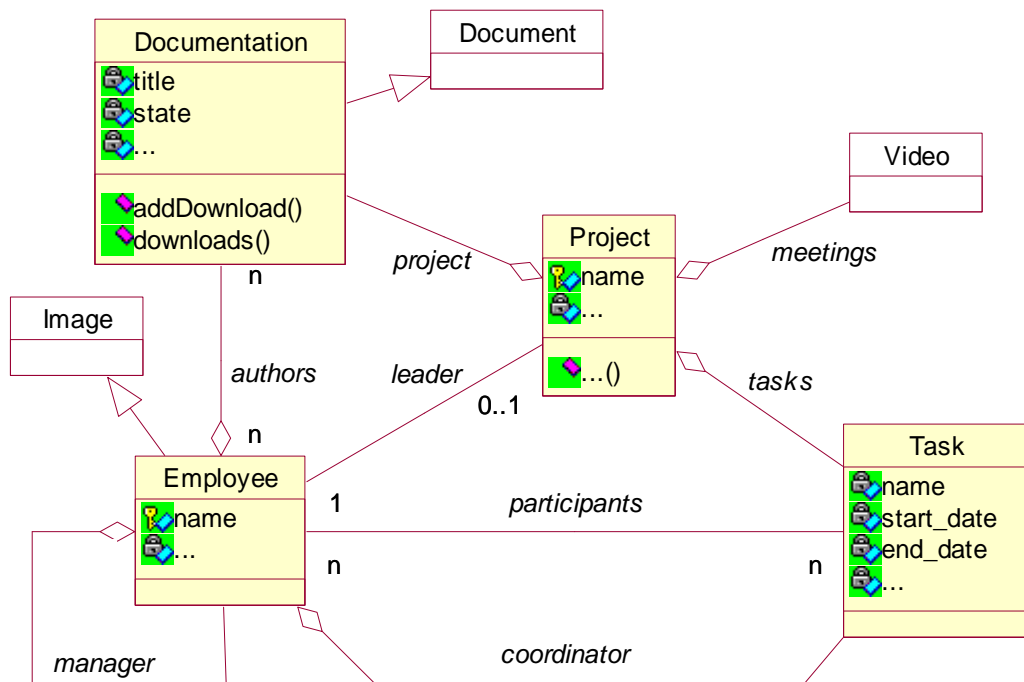


Fig. 3.1 Diagram tříd multimediální aplikace pro řízení projektů (pouze ilustraci)

Schéma databáze na obrázku Fig. 3.1 (nenapodobovat!) obsahuje 4 zjednodušené uživatelské třídy a 3 předdefinované systémem. Jsou to třídy pro uchování obrázků, videa a dokumentů. Třída zaměstnanců je odvozena od třídy obrázků pro uchování jejich fotografií, nabízí metody pro zobrazení, načtení ze souboru libovolného typu, nebo dotazování dle obsahu obrázku. To se nemusí v tomto kontextu zdát důležité, ale technika pro rozpoznání obličeje může sloužit pro kontrolu přístupu, prezence zaměstnanců na poradách, případně zabránit tomu, aby vedoucí oddělení zaměstnával pouze své příbuzné (ale už ne milenky).

Obdobně je to s třídou dokumentací, která rozšiřuje standardní třídu Document o některé atributy, například autorství a funkci která zvyšuje počítadlo při jeho stažení. Naopak video záznam porad není nutné rozšiřovat, protože metadata videa spolu s přepisem

meetingu pomocí nějakého systému pro rozpoznání řeči je dostatečné pro jeho identifikaci.

V diagramu se náhodně vyskytnou dva typy vazeb. První z nich je vztah asociace (relationship), který je obecně neorientovaný a zajišťuje referenční integritu. Pokud chceme explicitně vyjádřit orientaci vztahu, je možné použít orientovanou asociaci. V našem příkladě používám raději agregaci, která obecně neimplikuje vlastnictví a nezajišťuje integritu. To odpovídá vazbě použitím attribute v následujícím kódu.

Ex. 3.1 DDL multimediální aplikace pro řízení projektů

```
class Employee EXTENDS Image
( extent Employees
  key name )
{
  attribute string name;
  attribute Employee manager;

  relationship Project leads
    inverse Project::leader;
  relationship Set<Task> participate
    inverse Task::participants;
}

class Project
( extent Projects
  key name )
{
  attribute string name;
  attribute List<Document> documents;
  attribute Set<Task> tasks;
  attribute Set<Video> meetings;

  relationship Employee leader
    inverse Employee::leads;
}

class Task
( extent Tasks )
{
  attribute string name;
  attribute date start_date;
  attribute date end_date;
  attribute Employee coordinator;

  relationship Set<Employee> participants
    inverse Task::participates;
}

class Documentation EXTENDS Document
( extent Documentations
  key title )
{
  attribute string title;
  attribute string state;
  attribute int downloads;
  attribute Project project;
  attribute List<Employee> authors;

  void addDownload();
  int downloads();
}
```

Definiční jazyk může specifikovat mnoho dalších konceptů [Cat00]. Například rozhraní, v tomto příkladě zpřístupňující třídu zaměstnance bez jeho fotografie.

3.1.3 Dotazovací jazyk OQL

OQL je deklarativní (neprocedurální) dotazovací jazyk silně inspirovaný SQL. V relačních databázích je dotazovací jazyk jedinou možností pro přístup k datům. Objektově orientované aplikace používají spíše navigační způsob – navigace po identifikátorech objektů v agregační hierarchii tříd. Oba způsoby se ale doplňují. Dotazem je možné získat například množinu dat, se kterými dále manipulujeme navigačním způsobem. To je výhodné například pro algoritmy využívající procházení grafem (backtracking), protože ušetří velké množství náročných operací spojení (join).

Objektový dotazovací jazyk doplňuje relační o možnost spojení implicitně, pomocí hierarchické struktury objektů. Pokud vezmeme v úvahu schéma databáze z předchozí kapitoly, můžeme položit dotaz s explicitním spojením:

Ex. 3.2 Dotaz OQL s explicitním spojením

```
select d
from Documentations d, Projects p
where d.project = p.oid and
      p.name = 'Nošení dříví do lesa';
```

Nehledě na fakt, že systém nemusí poskytovat přístup k vestavěnému identifikátoru objektu, lze tento dotaz napsat mnohem úsporněji, pomocí implicitního spojení:

Ex. 3.3 Dotaz OQL s implicitním spojením

```
select d
from d in Documentations
where d.project.name = 'Vynález kola';
```

Tak, jak SQL umožňuje vytvářet libovolné nové relace (pohledy) výsledku dotazu, umí OQL vytvářet nové struktury. Samozřejmostí je také možnost volání funkcí v dotazech:

Ex. 3.4 Dotaz OQL s implicitním spojením

```
select distinct struct(nazev: d.name, stahnuto: d.downloads)
from Documentations d
where d.downloads > 0;
```

Výsledkem tohoto dotazu je množina n-tic `Set<struct(nazev: d.name, stahnuto: d.downloads)>`. Nutno poznamenat, že takto vytvořené struktury mají novou identitu a v tomto případě také ztratí informace o agregační hierarchii v databázi. Proto je využití struktur definovaných až při dotazu pro další dotazy značně omezené. OQL samozřejmě vyhodnocuje Booleovské predikáty pro objekty vyhovující zadanému dotazu, agregační funkce a podobně.

Jazyk OQL má oproti SQL několik dalších vylepšení, například také možnost dotazu na třídu, nebo i všechny její podtřídy. Ale jeho reálné implementace nebývají dokonalé a plná podpora vlastností specifikovaných ODMG často chybí nebo se liší [Cai03].

3.2 Objektově relační koncept

Začátkem 90. let si i velcí výrobci databázových systémů uvědomili nezbytnost objektového přístupu. Nicméně zjistili, že z obou přístupů je možné vybrat ty lepší

vlastnosti. Z OO konceptu zapouzdření, dědičnost a polymorfismus. Relační koncept přináší řešení persistence, integrity, řízení (souběžného) přístupu a zotavení po chybě.

Objektově relační (OR) model představuje alternativu čistě objektovému. Je, na rozdíl od OO modelu, spíše evolucí než revolucí DBMS. Integruje objektové přístupy do relačního modelu, umožňuje tak koexistenci obou typů dat a je zpětně kompatibilní. To mu zaručilo přízeň konzervativních zákazníků – přechod z relačního DBMS na objektově relační systém řízení báze dat (ORDBMS) nevyžaduje velké změny ani investice. Naopak použití OR přístupu může usnadnit vývoj aplikací i v těch oblastech, kde nejsou persistentní objekty nezbytně vyžadovány, protože reference mezi objekty výrazně usnadňují dotazování. V neposlední řadě je mapování objektových modelovacích technik (UML) výrazně přirozenější, než metody převodu diagramu tříd na relační model.

Koncem 90. let OO a OR přístup v databázových systémech do značné míry konvergoval. Například jazyk OQL je značně podobný SQL:1999. Podstatný rozdíl je jen v přístupu k objektovým datům:

- OODBMS lze chápat jako rozšíření programovacího jazyka – perzistentní objekty jsou vytvořeny v prostředí objektového jazyka (Java, C++, Smalltalk) pomocí tříd definovaných v těchto jazycích a manipulovány navigačním způsobem. Jsou tedy silně provázané (pouze objektové paradigma) a flexibilní pro aplikace na straně klienta – tyto aplikace dočasně uchovávají všechna potřebná data.
- ORDBMS oproti tomu poskytují pouze aplikační rozhraní, založené na SQL, pro manipulaci s daty, jejichž definice typů (tříd) musí být DBMS známa předem. To poskytuje možnost umístění aplikace i na straně serveru se vším co s tím souvisí, jako je možnost velkého množství současně probíhajících nezávislých dotazů a jejich případná optimalizace.

Objektových rozšíření relačních databází vzniklo mnoho, proto se podíváme na standardy, které jejich vývoj usměrňují.

3.2.1 SQL:1999

Původním záměrem SQL:1999 bylo modifikovat SQL-92 tak, aby se z SQL stal jazyk objektově relačních databází [Eis99]. Sedmiletý vývoj ale celý dokument restrukturalizoval a přidal několik základních vylepšení. K těm patří nové datové typy:

- Large Object (LOB) v binární (BLOB) a znakové podobě (CLOB). Tyto typy jsou určeny pro zpracování velkého množství dat na straně klienta, proto má jejich využití v dotazech několik omezení oproti typům z SQL-92.
- Booleovský typ (BOOLEAN) s hodnotami true, false a unknown.
- Dva složené datové typy (ARRAY, ROW). Pole je indexovaná multimnožina prvků stejného typu (konečná kolekce). Složený atribut ROW lze chápat jako běžný řádek relační tabulky, proto neporušuje 1. normální formu. Pro jeho zpřístupnění se používá tečková notace, jako `Employees.Name.title = 'Ing.'` v následujícím příkladě.

Ex. 3.5 Alternativní definice tabulky užitím ROW dle SQL:1999

```
CREATE TABLE Employees (  
  empID INTEGER PRIMARY KEY,  
  Name ROW (  
    title VARCHAR(5),  
    first VARCHAR(30),  
    mid VARCHAR(30),  
    second VARCHAR(30),  
    scitle VARCHAR(5),
```

```

),
freeDays INTEGER ARRAY[7],
inOffice BOOLEAN,
photo BLOB
);

```

SQL:1999 přidává několik predikátů (např. `Employees.Name.title LIKE18 '%Prof%'`), zavádí rekurzivní dotazy, trigger, uživatelské role a vylepšuje transakční zpracování (ROLLBACK TO SAVEPOINT). Nejdůležitější je ale objektivě orientované rozšíření.

Informace o objektech nesou v ORDBMS strukturované datové typy (User-Defined Types, UDT). Ty mají následující vlastnosti:

- Skládají se z jednoho nebo více typů, jak vestavěných (INTEGER, ARRAY), tak dále strukturovaných. Ty se mohou libovolně zanořovat. Viditelnost atributů (private, public) není v SQL:1999 řešena.
- Jsou organizovány do hierarchií. Více specializované podtypy dědí z nadtypů všechny atributy i metody a mohou být doplněny o další. Jedná se o jednoduchou dědičnost. Je ale nutné explicitně zadat, že typ může mít podtyp (NOT FINAL). Také je možné vytvořit ekvivalent abstraktní třídy (NOT INSTANTIABLE).
- Atributy jsou zapouzdřeny pomocí systémem generovaných funkcí pro čtení a zápis (set, get). Tyto funkce nemohou být přetíženy (overloading), všechny ostatní ano.
- Jejich chování je zprostředkováno pomocí metod a funkcí. Metody jsou, na rozdíl od funkcí, vázány k danému typu. Funkce mohou být přetíženy, výběr se provádí v čase kompilace. Metody mohou být polymorfní, jsou vybírány dynamicky za běhu programu (late binding).
- Porovnání hodnot objektů se děje výhradně pomocí uživatelských funkcí.

Ex. 3.6 Definice typu zaměstnance odvozeného z typu obecné osoby a typované tabulky

```

CREATE TYPE tEmployee
  UNDER tPerson
  AS ( empID INTEGER,
       manager REF(tEmployee),
       salary REAL
       photo tImage )
  INSTANTIABLE
  NOT FINAL
  REF ( empID )
  INSTANCE METHOD
    moreMoney ( amount REAL )
    RETURNS REAL
);
-- typed table Employees
CREATE TABLE Employees OF tEmployee;

```

Přístup k atributům a funkcím do strukturovaných typů se provádí pomocí tečkové notace (`Employees.salary > 10000`). Přístup k atributům je možný i funkční notací.

Reprezentace třídy (například v Javě) je možné vytvořit několika způsoby:

- Libovolný objekt se po standardní serializaci vlepí do BLOBu.
- Jednoduchý objekt je reprezentován řádem tabulky s vestavěnými typy.
- Objekt tvoří řádek s vestavěnými i strukturovanými typy, případně jediným UDT.

¹⁸ Je možné použít také SIMILAR s regulárními výrazy

- Typovanou tabulkou (viz Ex. 3.6). Každému řádku je tedy přiřazen jeden objekt. Takovýto objekt je možné jednoznačně identifikovat elementem, který se chová jako OID. Jedná se o speciální typ reference (REF). Jeho použití je jednoduché: `Employees.manager->photo`.

3.2.2 SQL:2003

V současné době nejnovější verze standardu SQL [SQL03] přináší revizi všech částí standardu SQL:1999. V našem kontextu je nejdůležitější vylepšení objektově-relačního schématu, například o zavedení násobné dědičnosti. Dále klauzule WINDOW vhodné pro zpracování a dotazování souvislých proudů dat. Pokrokem jsou standardizované funkce pro podporu On-Line Analytical Processing (OLAP) [Hru05]. SQL:2003 dále standardizuje nativní podporu pro manipulaci s XML (SQL/XML), viz [Eis02].

SQL:2003 zavádí nové datové typy BIGINT, XML, ale především je to typ kolekce (MULTISET). Její použití je obdobné jako pole až na fakt, že tato multimnožina není uspořádaná ani teoreticky konečná. Deklaruje se třemi různými způsoby – `INTEGER MULTISET` může znamenat to stejné co `MULTISET[-1, -1, 2, 3, 4, 9]` nebo `MULTISET(SELECT i FROM t)`. Operace nad multimnožinou jsou podobné jako operace nad tabulkou včetně agregačních funkcí, ale existují mírné odchylky v názvech apod., viz [SQL03].

Nově byl přidán typ TABLE, který je interně definován jako multimnožina řádků (ROW). Tento typ je používán pro konverze výsledku funkcí na tabulku a opačně. Výhodou je, že tyto funkce mohou být definovány v jazycích jako C++ nebo Java. Je zde tedy možnost, jak definovat externí (EXTERN) tabulky. Jejich použití není složité, nicméně přesahuje, spolu s dalšími souvisejícími aspekty rozsah této práce [Eis04].

Užitečná možnost je deklarovat sloupec jako generovaný. To znamená automaticky při vložení nebo aktualizaci. Buďto bez možnosti přiřadit jinou hodnotu (GENERATED ALWAYS) nebo generovaný v případě, že nebyla hodnota zadána (GENERATED BY DEFAULT). Tato vlastnost lze použít například pro obecné funkce, OLAP, nebo pro definování identity řádku, jak je uvedeno v příkladě níže.

Ex. 3.7 Definice typu zaměstnance odvozeného z typu obecné osoby (jméno, ...)

```
CREATE TABLE Photos (
  photoID INTEGER GENERATED ALWAYS
  AS IDENTITY (
    START WITH 1
    INCREMENT BY 1
    MINVALUE 1
    NO MAXVALUE
    NO CYCLE ),
  name VARCHAR(100),
  desc CLOB,
  source tImage
)
```

V tomto příkladě je generovaný sloupec ilustrován jako sekvence (SEQUENCE). Ta představuje další novinku SQL:2003. Sekvenci je možné vytvořit samostatně a při vkládání hodnoty do tabulky ji zavolat pomocí `NEXT VALUE FOR`. Protože ale její implementace, existovaly již dříve (AUTOINCREMENT v MySQL), existuje jednodušší možnost – použít sloupec s klíčovým slovem IDENTITY. Při vkládání pak není nutné hodnotu zadávat, je generována automaticky.

Další rozšíření umožňují jednodušší vytváření podobných tabulek v databázi (CREATE TABLE LIKE). Pokud požadujeme složitější operace jako změnu názvu a počtu sloupců,

stejně jako pro kopii dat, je možné použít CREATE TABLE AS. Nicméně tyto funkce nevytváří logické vazby mezi daty. Pro jejich synchronizaci je možné použít 4. klauzuli pro modifikaci dat – sloučení (MERGE).

SQL:2003 má oproti starším verzím mnoho dalších vylepšení [SQL03], jejich plná podpora ze strany tvůrců databází je nicméně, na rozdíl od ODMG, spíše otázkou času.

3.3 SQL / MM

SQL Multimedia (and Application Packages) měla původně (1999) přispět ve standardizaci abstraktních datových typů (ADT) a knihoven tříd pro vývoj a výzkum, textové databáze, zpracování dokumentů a správu multimediálních dat, jako jsou obrázky, zvuk, animace a video. Nicméně v průběhu standardizačního procesu (2001 – 2002) došlo k několika změnám a odklonu od obecných multimediálních dat, jak je vidět z následujícího přehledu částí SQL/MM:

Část 1 Framework: definuje společný koncept a předpisy.

Část 2 Full Text: zavádí typ FULLTEXT pro uchování dlouhých statí textu a některé další struktury, jako THESAURUS nebo SYNONYM. Definuje i vyhledávání odvozených slov ze stejného kmene pomocí STEMMED.

Část 3 Spatial: definuje prostorové typy a operace nad nimi.

Část 5 Still Image: zavádí dva základní typy. SI_StillImage pro uchování obrazových dat (Ex. 3.8) a SI_Feature pro uchování jejich rysů. Rysy mohou být SI_AverageColor (průměrná barva), SI_ColorHistogram (jstogram), SI_PositionalColor (rozložení barvy) a SI_Texture (texturní rysy). Všechny tyto vlastnosti obsahují metodu SI_Score pro výpočet podobnosti, která vrací hodnoty mezi 0 a 1 (Ex. 3.8). Jsou definovány i další operace nad obrázky, jako změna velikosti, ořezání nebo rotace.

Část 6 Data mining: definuje datové typy a operace pro klasifikaci, regresi, shlukování a vyhledávání (asociačních) pravidel.

Z multimediálních dat tedy zbyly pouze statické obrázky. Standard pro pohyblivé obrázky a zvuková data nebyl dokončen. Čtvrtá část měla definovat ADT pro obecné (matematické) účely, ale také nebyla dokončena.

Ex. 3.8 Datový typ SI_StillImage a podobnostní vyhledávání dle textury [Kuc04]

```
create type SI_StillImage as (  
  SI_content binary large object(SI_MaxContLength),  
  SI_contentLength integer,  
  SI_format character varying(8),  
  SI_height integer,  
  SI_width integer,  
  ...  
)  
  
SELECT *  
FROM Registrovana_Loga  
WHERE  
  SI_findTexture(Nase_logo).SI_Score(Logo) > 0.9;
```

3.4 XML databáze

XML dokumenty lze rozdělit na dvě základní skupiny. První skupinu tvoří XML dokumenty, které obsahují spíše data a druhou XML dokumenty, které v sobě ukládají spíše dokumenty (rozsáhlé texty). Na základě typu obsahu XML dokumentu se

rozhoduje, jaký typ databáze a způsob uložení bude vhodné zvolit pro takový XML dokument.

Datové XML dokumenty se označují také jako dokumenty zaměřené na data (data-centric XML documents) či XML dokumenty s datovým obsahem. Datové XML dokumenty používají XML jako vhodnou formu pro zápis dat určených k přenosu. Informace v takovýchto dokumentech jsou povětšinou určeny pro potřeby jiných programů, nikoliv pro zpracování lidmi, tedy ke čtení. XML zde slouží pouze jako obálka pro přenos a komunikaci mezi zařízeními. Životní cyklus dat je nejčastěji takový, že data jsou na počátku v nějaké databázi či aplikaci, poté jsou zabalena do XML (příp. pouze vyjmuta již v XML formátu) a přenesena do jiné databáze či aplikace. XML forma vyjádření těchto dat je v podstatě druhotná, hlavní jsou přenášená data.

Datové XML dokumenty se vyznačují poměrně pravidelnou strukturou, jemně granulovanými daty (v elementech či atributech jsou uloženy co nejmenší logické jednotky dat typu #PCDATA) a malým či žádným výskytem smíšeného obsahu v elementech. Důležitou vlastností je, že nezáleží na pořadí, v jakém jsou jednotlivá data (v elementech) seřazena, neboť charakter a typ dat určuje samotný název elementu, podle kterého lze data snadno identifikovat. Na pořadí nezáleží z pohledu aplikace, která data načítá. XML dokument nicméně stále musí být validní vůči svému XML Schema nebo DTD definici.

Ex. 3.9 Ukázka datového XML dokumentu

```
<?xml version="1.0" encoding="windows-1250"?>
<OBJEDNAVKA id="20071012/123">
  <DATUM_OBJEDNANI>12.10.2007</ DATUM_OBJEDNANI >
  <ZAKAZNIK>
    <JMENO>Tomáš Fuk</JMENO>
    <ULICE>Růžová 15</ULICE>
    <MESTO>BRNO</MESTO>
    <PSC>60200</PSC>
  </ ZAKAZNIK >
  <POLOZKA id="10203">
    <POCET_KS>1</ POCET_KS >
    <POZNAMKA>velikost L; modro-černá</POPIS>
    <CENA>200</CENA>
  </POLOZKA>
</OBJEDNAVKA>
```

Dokumentové XML dokumenty bývají v literatuře, obdobně jako datové XML dokumenty, pojmenovány různě. Například jako dokumentově zaměřené XML dokumenty (document-centric XML documents) či XML dokumenty s dokumentovým obsahem. V této budou tyto XML dokumenty reprezentovat označení dokumentové XML dokumenty. Tento typ XML dokumentů je většinou určen přímo pro čtení či zpracování lidmi. Charakterizuje je méně pravidelná struktura, hrubá granulace dat a časté používání smíšeného obsahu. Záleží na pořadí, v jakém jsou jednotlivé elementy za sebou zapsány. Tyto dokumenty jsou nejčastěji psány ručně a to přímo v jazyce XML či např. ve formátu RTF, PDF, které jsou poté do XML zkonvertovány. Data těchto dokumentů obvykle nepochází z databází. Příkladem dokumentových XML dokumentů mohou být knihy, e-maily, reklamy, (ručně psané) WWW stránky ve formátu XHTML. [Doub07]

Ex. 3.9 Ukázka dokumentového XML dokumentu

```
<?xml version="1.0" encoding="windows-1250"?>
<CLANEK>
  <NADPIS_1>Dokumentové XML dokumenty</ NADPIS_1>
```

```
<ODSTAVEC> Dokumentové XML dokumenty bývají v literatuře, obdobně  
jako datové XML dokumenty, pojmenovány různě. </ODSTAVEC>  
</CLANEK>
```

V dnešní době již existuje celá řada softwarových produktů, které pracují s XML ve spojení s databázemi. Tyto produkty se dají rozdělit do několika skupin. V následujících kapitolkách bude představeno rozdělení podle Ronalda Bourreta [Bour07].

3.4.1 Databázové systémy s podporou XML

Jedná se o databázové systémy, které pracují s XML dokumenty. Tyto databáze jsou zatím asi nejlepší variantou pro práci s datově orientovanými XML dokumenty. Databáze s podporou XML formátu se označuje jako XML enabled a dnes je XML enabled většina komerčních relačních databází. Tyto databáze poskytují uložená data ve stejném nebo i v jinak strukturovaném XML dokumentu, než v jakém byla uložena. Příčinou tohoto jevu jsou převodní algoritmy (table based, object based), které nemohou předávat informace o specifických, pro data nepodstatných elementech.

3.4.2 Nativní XML databázové systémy

Používají XML jako nativní formát pro uložení dat, jsou vhodné pro datově i dokumentově orientované XML dokumenty. Tyto systémy ukládají XML dokument v jeho přirozené podobě, tedy včetně jeho logické struktury, poznámek atd. Proto je vhodné používat nativní XML databázové systémy při práci s XML dokumenty, u kterých záleží na pořadí jednotlivých elementů. Do těchto databázových systémů se nejčastěji ukládají dokumentově orientované XML dokumenty a právě tyto dokumenty byly hlavní příčinou pro jejich vznik. Výhodou tohoto řešení je především vyšší výkon aplikací používajících XML, protože odpadá nezbytný mezikrok převodu z/do formátu XML.

3.4.3 Hybridní XML databázové systémy

Hybridní XML databázové systémy se používají jako nativní XML databázové systémy nebo jako databázové systémy s podporou XML v závislosti na požadavcích aplikace.

3.4.4 Middleware

V případě databázových systémů a formátu XML můžeme jako middleware označit nástroje umožňující transfer XML dat do databáze nebo z databáze do XML formátu. Middleware programy jsou určeny především pro datově orientované XML dokumenty.

3.4.5 XML servery

Termínem XML servery se označují aplikační servery, které poskytují svým klientům data ve formátu XML. Jedná se např. o J2EE (Java 2 Platform, Enterprise Edition) servery nebo webové servery. Příkladem jednoduchého XML serveru by mohl být webový server obsahující ASP (ActiveX Server Pages) stránky, které dle zadaných parametrů v URL načtou data z databáze a vygenerují obsah ve formátu XML – což je jednoduchá nicméně reálná situace.

3.4.6 Wrappery

Wrappery jsou programy, které s XML dokumentem nakládají tak, jako by to byl zdroj relačních dat. Wrapper obalí XML dokument a zpřístupní jeho data v relační formě (tabulkách). Nad takovým zdrojem dat lze poté provádět operace v jazyce SQL.

3.4.7 Systémy pro správu obsahu

Systémy, které podporují vytváření, správu, distribuci, publikování a zpřístupňování (firemních) informací. Jedná o aplikace vystavěné nad nativním XML databázovým systémem nebo systémem souborů na disku pro ukládání XML dokumentů.

3.4.8 Nástroje pro dotazování nad XML

Skupinu nástrojů pro dotazování nad XML tvoří samostatné programy, které umí provádět operace nad XML dokumenty. V současné době tyto programy využívá relativně mnoho XML databázových jazyků (XQL, XPath, XML-QL, XQuery), kde v podstatě standard tvoří jazyk XPath od konsorcia W3C. V podstatě nativní XML databázové systémy jsou nadmnožinou nástrojů pro dotazování, protože v sobě obsahují mechanismy pro dotazování XML dokumentů.

3.4.9 XML data binding nástroje

Programové nástroje, které z XML dokumentu vytváří objekty zapouzdřující data z toho XML dokumentu. Respektive ze schématu XML dokumentu vytváří třídy, jejichž instance jsou potom používány pro data konkrétních XML dokumentů. Objekty vytvářené XML data binding nástroji jsou přirozenější, lépe charakterizují realitu, aplikační logiku (narozdíl od objektů DOM, které popisují strukturu XML dokumentu).

3.4.10 Dotazovací jazyky nad XML

Dotazovacích jazyků nad XML daty lze vymyslet s různou syntaxí v podstatě nekonečně mnoho a toho využívají velké firmy, které často definují vlastní syntaxi programovacích jazyků podporovanou jejich produkty. Oblast dotazovacích jazyků nad XML není výjimkou (např. dotazovací jazyk SQLXML vytvořený firmou Microsoft a podporovaný pouze na MS SQL Server), i když na druhou stranu nutno říci, že dnes nejvíce používané dotazovací jazyky jsou převážně standardizovány a podporovány různými produkty a databázemi (např. XPath, XQuery, SQL/XML).

Starší dotazovací jazyky

Starší dotazovací jazyky, které dnes upadají do pozadí, buď nesplnily z hlediska funkčnosti očekávání kladené na dotazovací jazyk a nebo přirozeným vývojem byly nahrazeny výkonnějšími jazyky. Mezi dnes již nepoužívané dotazovací jazyky se řadí jazyky XML-QL, Lorel nebo jazyk XQL. Důvod, proč tyto jazyky stručně prezentuji, je takový, že splnily nezanedbatelnou roli vývoje a většina dnes používaných a preferovaných dotazovacích jazyků vychází právě z těchto starých.

XML-QL – (A Query Language for XML) je jedním z prvních dotazovacích jazyků specifikovaný v roce 1998 konsorciem W3C. Struktura jazyka je velmi jednoduchá, což ovšem ovlivnilo jeho použitelnost. Dotaz v XML-QL se skládá ze dvou částí, kde první část je dotazovací (specifikuje konkrétní tvar dotazu) a druhá část konstrukční (tvar konstrukce obsahu výstupního XML dokumentu včetně případných dalších poddotazů). XML-QL pochází z oblasti klasických relačních databází a byl zaměřen hlavně na obsluhu rozsáhlých datových souborů.

Lorel – je dotazovací jazyk vycházející z objektových databází. Jeho syntaxe připomíná jazyk OQL. Struktura dotazování je v podstatě podobná jazyku XML-QL. Využívá adresace uzlů pomocí regulárních výrazů a vázané proměnné pro značky.

XQL – (XML Query Language) je od předchozích jazyků značně odlišný. Byl specifikován konsorciem W3C opět v roce 1998. Je velice příbuzný s jazykem XPath, se kterým má i podobnou syntaxi. Pomocí XQL se provádí fulltextové prohledávání

dokumentů, které je závislé na struktuře elementů XML. Výrazy dotazu stanovují stejně jako v XPath cestu k jednotlivým elementům.

Dotazovacích jazyků, jak již bylo zmíněno, je v oblasti XML a databází celá řada. Pro úplnost výčtu uvádím ještě názvy některých ostatních jazyků jako např. XMLQL, XUpdate, YATL, XIRQL nebo Quilt, které se v literaturách objevují, ale jejich použití v praxi není tak časté.

Podle [Bour07] lze dělit dotazovací jazyky na tyto skupiny:

- *Dotazovací jazyky založeny na základě šablon (např. XSLT, XSQL)* – jazyky, které vracejí data ve formátu XML získané v relační databázi. Právě pro převod dat z relačních tabulek a transformaci do XML formátu využívají šablony. Výhodou těchto dotazovacích jazyků je flexibilita, možnost použití cyklů a podmíněných příkazů.
- *Dotazovací jazyky vycházející z jazyka SQL (SQL/XML, PL/SQL)* – jazyky, které lze poznat pomocí rozšiřujících parametrů používaných ve standardní struktuře SQL dotazů. Tyto jazyky umožňují např. přímo ve struktuře dotazu SELECT popsat výsledný tvar vrácených dat. Transformace je tedy prováděna přímo a s jazyky se nejčastěji setkáme u objektově-relačních databází.
- *Dotazovací jazyky založeny na XML (XPath, XQuery)* – jazyky vycházející z XML specifikace a používají se především nad XML dokumenty, nikoliv nad databázemi. Samozřejmě je možné jejich použití nad databázemi s rozšířenou podporou XML, tedy např. u databáze Oracle, kde se Oracle XML DB stará o nativní podporu XML.

XPath

Standard jazyka XPath je specifikací konsorcia W3C [23i] od roku 1999. V současné době se používají verze XPath 1.0 i XPath 2.0, které jsou si velmi podobné. XPath 2.0 navazuje na silné stránky XPath 1.0, ke kterým je navíc přidáno několik nových možností, např. podpora většího množství datových typů. Hlavním smyslem jazyka je identifikace a adresování různých částí XML dokumentu. XPath adresuje jednotlivé uzly dokumentu pomocí výrazů, které připomínají částečně regulární výrazy. Výrazy XPath mohou být reprezentovány datovými typy jako jsou čísla, řetězce, logické hodnoty a především podmnožiny elementů XML dokumentu. Tento jazyk stojí v pozadí většiny operací s XML, i když není používán přímo, je využit např. v XSLT pro vyhledávání a výběr elementů ve vstupním dokumentu nebo v jazyce XQuery, kterého je XPath 2.0 podmnožinou.

Princip jazyka XPath je založen na práci s abstraktním datovým modelem XML dokumentu. V paměti počítače je pro každý zpracovávaný dokument vytvořen abstraktní strom členěný do uzlů. XPath tedy nevyhodnocuje své výrazy přímo nad XML dokumentem, ale právě nad vytvořeným abstraktním stromem. Každý uzel stromu může být reprezentován jedním z následujících sedmi uzlových typů [Kos07].

- *Kořenový uzel* – umělý kořen stromové reprezentace, neodpovídá žádnému elementu v dokumentu. Synem tohoto uzlu je reálný kořenový uzel zpracovávaného XML dokumentu.
- *Uzel s elementy* – tyto uzly reprezentují všechny elementy dokumentu. Každému dokumentovému elementu odpovídá jeden stromový uzel stejného názvu, rozšiřitelný o atributy a jmenné prostory.
- *Uzel s textem* – vždy listové uzly stromu bez názvu a bez potomků. Tyto uzly zastupují konkrétní textové hodnoty XML dokumentu.

- *Uzel s atributy* – uzly atributů tvoří podstrom elementového uzlu, kterého jsou v dokumentu součástí. Jsou to listové uzly se stejným názvem jako atributy v dokumentu a již s konkrétní hodnotami.
- *Uzel s komentáři* – listové uzly bez názvu, které nesou konkrétní hodnotu pomocného uživatelského komentáře.
- *Uzel se zpracovacími instrukcemi* – každá instrukce XML dokumentu bude reprezentována jedním uzlem stromu, který bude obsahovat cíl a obsah instrukce. Je to listový uzel.
- *Uzel se jmennými prostory* – představují deklaraci jmenného prostoru uvedenou v XML dokumentu. Je to listový uzel.

XSLT

XSLT není zcela určitě dotazovací jazyk v pravém slova smyslu, přesto dotazování pomocí tohoto jazyka je možné ve spojení s jazykem XPath. Spojení obou jazyků je vhodné pro práci s více XML dokumenty. XPath umožňuje dotazování pouze nad jedním dokumentem, ale pomocí XSLT lze toto jedno-dokumentové dotazování přenést nad více dokumentů. Často se také používá vyhledání určitých hodnot z různých dokumentů a vložení výsledku, např. sečtení hodnot, do nového vstupního dokumentu.

XSLT není dotazovací jazyk bez spojení s jazykem XPath. Ovšem poměrně zajímavou a prakticky využitelnou aplikací spojenou s databázemi v XSLT je možnost vytvoření sekvence SQL příkazů z dat XML dokumentu. Příkazy lze generovat jak pro smazání, vytvoření tak i vyhledání informací. Takovýto dokument s SQL příkazy poté může sloužit např. jako dávka importující nového zákazníka nebo novou položku katalogu do relační databáze. Základem je vytvoření XSLT stylu, který bude vytvářet nový dokument, do kterého se budou vypisovat SQL příkazy zapsané v šabloně stylu.

XQuery

Dotazovací jazyk XQuery je dalším jazykem od konsorcia W3C, stejně jako XPath je založen na dotazování nad XML daty. XQuery spolupracuje s jazykem XPath 2.0 a je jeho nadmnožinou, která ovšem není zcela zpětně kompatibilní s jazykem XPath 1.0. Základy jazyka XQuery vychází hned z několika předchozích dotazovacích jazyků např. XPath, SQL a především jazyk Quilt. XQuery je hlavně dotazovací jazyk, který oproti XPath nabízí širší spektrum využití, ale do budoucna se od tohoto jazyka očekává rozšíření funkcionality o další prvky jako např. možnost aktualizovat data (zatím umožňuje jen výběrové operace). XQuery pracuje se stejnými datovými typy jako jazyk XPath 2.0.

XQuery je funkcionální jazyk, u kterého je základním stavebním prvkem výraz. Jazyk umožňuje používání různých typů výrazů, které mohou být složeny z klíčových slov, symbolů a operandů. Jednotlivé výrazy lze libovolně kombinovat dohromady, zanořovat je do sebe nebo použít výsledky výrazů jako parametry dalšího výrazu. Při práci s výrazy zachovává jazyk uzavřenost nad datovým modelem. To znamená, že výsledkem XQuery dotazu je vždy zase instance datového modelu XQuery, kde navíc použité datové typy výrazů, operátorů a funkcí musí odpovídat specifickým datovým typům datového modelu XQuery.

Další vlastností jazyka je rozlišování malých a velkých písmen (case-sensitive), silná typová kontrola zajišťující vyšší bezpečnost nebo úzké provázání s XML Schema, které popisuje strukturu výsledku a definuje datový typ výsledných dat.

XSQL

XSQL je typickým zástupcem dotazovacích jazyků založených na šablonách XML. Jazyk pracuje nad daty uloženými v relačních tabulkách, tedy nenabízí nativní podporu

XML. XSQL je většinou nazýván jako publikační nástroj, neboť hlavní účel používání XSQL je práce s XSQL webovými stránkami, které prezentují data uložená v databázi.

Jazyk XSQL vychází ze zápisu SQL výrazů do struktury XML dokumentu (XSQL stránek), kde jednotlivé elementy mají speciální význam a jsou pojmenovány speciálními klíčovými slovy. Obsahem elementů jsou SQL výrazy, které se při interpretování XSQL stránek nahrazují vrácenými hodnotami z databáze. Na takto vytvořený XML dokument se dále napojuje šablona XSLT, která určí konečný vzhled prezentovaných dat ve webovém prohlížeči, který zobrazuje webové XSLQ stránky.

Způsob dotazování jazyka XSQL dovoluje využít v obsahu elementů klasické SQL dotazy v plné šíři jejich možností, kde syntaxe je omezena pouze typem databáze, nad kterým jsou XSQL dotazy volány. Přímé volání SQL dotazů z obsahu elementů přináší ale i jistá úskalí a nevýhody. Například je obtížné zamezit uživateli aktualizovat data v databázi, není tedy možné kontrolovat uživatele (jejich přístupová práva) a obsah databázových tabulek.

SQL/XML

Jazyk SQL/XML vzniknul díky narůstající potřebě ukládat XML data uvnitř databázového relačního modelu. Z názvu tohoto dotazovacího jazyka vyplývá, že se v podstatě jedná o rozšíření klasického standardizovaného dotazovacího jazyka SQL o nové XML prvky, ať už nová klíčová slova či nové funkce, metody. SQL/XML je 14. částí standardu SQL: 2003 a za jejím vytvořením stojí skupina H2.3 Task Group, zahrnující spolek firem jako jsou např. IBM, Oracle, Microsoft, Sybase.

Jazyk SQL/XML se někdy zkráceně označuje výrazem SQLX. Na druhou stranu je potřeba rozlišovat mezi označeními SQL/XML a SQLXML (jazyk vyvíjený firmou Microsoft, není součástí standardu), kde SQLXML je zcela odlišný dotazovací jazyk a rozhodně není podporován systémem Oracle a dále nebude popisován ani rozebírán. SQL/XML je tedy typickým zástupcem jazyků, které pracují nad relačním modelem dat, ale umožňují po provedení dotazu vracet získané hodnoty ve formátu XML dokumentu.

Novým prvkem jazyka SQL/XML je především nový datový typ nazvaný XML. Tento datový typ je v principu používání stejný jako jiné datové typy jazyka SQL (VARCHAR32, INT, CLOB). Je možné jej použít pro definování typu sloupců tabulek, může obsahovat hodnotu Null a samozřejmě hodnoty XML dokument nebo i jen samotný XML element. Datový typ XML je v databázi Oracle reprezentován již zmíněným datovým typem Oracle XMLType a je nutné použít klíčového slova XMLType.

PL/SQL

PL/SQL je blokový strukturovaný jazyk, který rozšiřuje jazyk SQL v databázi Oracle. PL/SQL umožňuje v první řadě efektivnější dotazování, kdy je možné v jedné funkci blokově volat několik SQL dotazů, které nemusí být přenášeny samostatně jako u jazyka SQL. Hlavními prvky jazyka PL/SQL jsou procedury, funkce a anonymní bloky, které se mohou ještě zanořovat do dalších bloků.

3.5 Získávání znalostí

Získávání znalostí z výše uvedených databází je odlišné než získávání znalostí z uniformních struktur, které lze snadno uložit do relačních databází. Tyto databáze ukládají v sobě komplexní objekty spolu s vazbami mezi nimi. Organizují rozsáhlou množinu složitých objektů do tříd, které tvoří hierarchie tříd a podtříd. Každý objekt ve třídě je asociován s (1) identifikátorem objektu, (2) množinou atributů, které rovněž mohou obsahovat sofistikované datové struktury, hierarchie podtříd, či multimediální

data, a (3) množinu metod, které specifikují výpočetní rutiny nebo pravidla asociovaná s třídou objektu.

Jeden krok za ukládáním a přístupem k masivně škálovaným, složitým objektovým datům je systematická analýza a dolování nad těmito daty. To zahrnuje dvě hlavní úlohy: (1) konstrukci multidimenzionálních datových skladů pro uložení složitých objektových data a poskytování OLAP prostředků pro práci s těmito daty, a (2) vývoj efektivních a škálovatelných meto pro dolování znalostí z objektových databází a/nebo datových skladů. Druhá úloha je pokryta zejména dolováním konkrétních typů znalostí. Hlavním omezením pro mnoho komerčních datový skladů a OLAP nástrojů pro analýzu multidimenzionálních databází je jejich omezení na povolené datové typy pro dimenze a metriky. Většina implementací datových kostek omezuje dimenze na nenumerická data a metriky na jednoduché, agregované, hodnoty.

Pro usnadnění generalizace a indukci v objektově-relačních a objektově-orientovaných databázích je důležité vědět jak může být každá komponenta takových databází generalizována jak takováto generalizovaná data mohou být dále použita pro analýzu a dolování v multidimenzionálních datech.

3.5.1 Generalizace strukturovaných dat

Důležitým rysem objektově relačních a objektově-orientovaných databází je jejich schopnost ukládání, zpřístupnění a modelování složitě dat, jejichž hodnotou je struktura, jako jsou seznamy, množiny a data s vnořenými strukturami.

Atributy, jejichž hodnota je množina, mohou být homogenního nebo heterogenního typu. Typicky mohou být tato data generalizována pomocí: (1) generalizace každé hodnoty v množině na svoji generalizovanou úroveň, a (2) odvození z obecného chování množiny, jako je počet elementů v množině, typ nebo rozsah hodnot v množině, vážený průměr pro numerická data. Dále může být generalizace vylepšena aplikováním různých generalizačních operátorů na prozkoumání alternativních generalizačních cest. V tomto případě je pak výsledkem heterogenní množina.

Ex. 3.9 Generalizace atributů, jejichž hodnotou je množina. Generalizace atributů, jejichž hodnotou je množina. Předpokládejme, že koníček osoby je atribut, jehož hodnotou je množina {tenis, hokej, fotbal, housle, SimCity}. Tato množina může být generalizována na množinu vysokoúrovňových konceptů, jako je {sport, hudba, počítačové hry} nebo do čísla 5 (např. počet koníčků v množině). Počet může být asociován s generalizovanou hodnotou pro indikování toho, kolik elementů je generalizováno na danou hodnotu, jako např. {sporty(3), hudba(1)m počítačové hry (1)}.

Atribut jehož hodnotou je množina může být generalizován na atribut s množinovou hodnotou nebo jednoduchou hodnotou. Atribut s jednoduchou hodnotou může být generalizován na atribut s množinovou hodnotou, pokud hodnoty vytváří mřížku nebo hierarchii nebo generalizace sleduje různé cesty. Dále pak generalizace na takovýchto generalizovaných attributech s množinovými hodnotami by měla sledovat generalizační cestu každé hodnoty v množině.

Atributy, jejichž hodnotou je seznam a atributy jejichž hodnotou je sekvence mohou být generalizovány obdobným způsobem jako tomu bylo u atributů jejichž hodnotou je množina s tím rozdílem, že by mělo být bráno v potaz pořadí v jakém jsou hodnoty v seznamu či sekvenci uloženy. Každá hodnota v seznamu může být generalizována na svůj odpovídající koncept vyšší úrovně. Alternativně pak může být seznam generalizován podle svého obecného chování, jako je délka seznamu, typ jednotlivých elementů seznamu, rozsah hodnot, vážený průměr, nebo vypuštění nedůležitých

elementů v seznamu. Seznam může být generalizován na seznam, množinu nebo jednoduchou hodnotu.

Atribut, jehož hodnotou je složitá struktura může obsahovat množiny, n-tice, seznamy, stromy, záznamy a jejich kombinace, kde jedna struktura může být zanořena v jiné na libovolné úrovni. Obecně může být takový atribut generalizován několika způsoby: (1) generalizování každého atributu ve struktuře, zatímco je zachován tvar struktury, (2) zploštěním struktury a generalizováním zploštěné struktury, (3) sumarizování nízkourovňových struktur pomocí vysokoúrovňových konceptů nebo agregace a (4) návratem typu nebo přehledu struktury.

Obecně může statistická a shluková analýza pomoci při rozhodování při směrech a stupních generalizace pro určení, které generalizační procesy zachovávají hlavní rysy a odstraňují šum, izolované vrcholy nebo fluktuaci.

3.5.2 Agregace a aproximace v generalizaci prostorových a multimediálních dat

Agregace a aproximace jsou další důležité významy generalizace. Mohou být zvláště užitečné pro generalizování atributů s rozsáhlými množinami hodnot, složitými strukturami a prostorovými nebo multimediálními daty.

Když vezmeme prostorová data jako příklad. Rádi bychom generalizovali detailní geografické body do shluknutých regionů, jako jsou obchodní, obytné, průmyslové nebo zemědělské oblasti podle příslušného využití půdy. Taková generalizace často vyžaduje spojení geografických oblastí pomocí prostorových operací, jako je prostorové sjednocení nebo prostorové shlukové metody, Agregace a aproximace jsou důležité techniky pro tuto formu generalizace. V prostorovém slučování je nezbytné neslučovat pouze regiony podobného typu uvnitř stejné obecné třídy, ale rovněž spočítat celkové plochy, průměrnou hustoty nebo jiné souhrnné funkce, zatímco jsou ignorovány osamocené plochy s různými typy, pokud nejsou důležité. Jiné prostorové operátory, jako jsou prostorové sjednocení, prostorové překrytí nebo prostorový průnik (které mohou vyžadovat slučování osamocených malých regionů do velkých shlukovaných regionů), mohou rovněž použít prostorovou agregaci a aproximaci jako operátory generalizace dat.

Multimediální databáze může obsahovat složité texty, grafiku, obrazy, video fragmenty, mapy, hlas, hudbu a další podoby AV informací. Multimediální data jsou typicky uložena jako sekvence bytů s proměnnými délkami a segmenty dat jsou slinkovány dohromady nebo indexovány multidimenzionálním způsobem pro snadnou referenci.

Generalizace na multimediálních datech může být uskutečněna pomocí rozpoznání a extrakce esenciálních rysů a/nebo obecných vztahů těchto dat. Existuje mnoho způsobů jak získat takovéto informace. V obrazech mohou být pomocí agregace a/nebo aproximace zjištěny: barva, velikost, tvar, textura, orientace a relativní pozice či struktury obsažených struktur nebo oblastí. Pro segment hudba může být jeho melodie sumarizována podle jeho tónu, tempa, nebo hlavních hudebních nástrojů, které hrají. Pro článek to jsou pak jeho abstrakt nebo obecná organizační struktura (např. obsah, předmět, index remízů často se v článku vyskytujících, atd.), které mohou sloužit jako jeho generalizace.

Obecně je generalizování prostorových a multimediálních dat náročná úloha s ohledem na extrakci zajímavé znalosti implicitně uložené v datech. Technologie vyvinuté v prostorových a multimediálních databázích, jako jsou přístupy k prostorovým datům a analytické techniky, rozpoznání vzorů, obrazová analýza, textová analýza, atd., by

měly být integrovány s generalizací dat a technikami pro dolování dat pro dosažení uspořádaných výsledků.

3.5.3 Generalizace na multimediálních datech uložených objektově-orientovaných databázích

Multimediální databáze mohou obsahovat souhrn textů, grafiky, obrázků, map, hlasu, hudby a jiné formy audio či video informací. Taková multimediální data jsou typicky uložena v sekvencích bytů s proměnnou délkou. Na segmenty dat jsou provázány dohromady pro snadnější odkazování. Generalizace na multimediálních datech může být provedena pomocí rozpoznání a extrakce esenciálních rysů nebo obecných vztahů takovýchto dat. Existuje mnoho způsobů pro extrakci esenciálních rysů nebo obecných vztahů ze segmentů multimediálních dat. Pro obrázky to jsou velikost a barva obsažených objektů nebo hlavní oblasti v obrázcích získané agregací nebo aproximací. Pro segmenty hudby to může být melodie založená na aproximaci vztahů, které se opakovaně vyskytují v segmentu nebo styl, který může být založen na tónech, tempu nebo rychlosti, hlavních hudebních nástrojů atd. Pro články to může být abstrakt nebo obecná organizace jako je obsah, předmět, index výrazů často se opakující v člancích atd. Obecně je generalizace multimediálních dat pro získání zajímavých znalostí implicitně uložených v datech velkou výzvou při vývoji nástrojů pro získávání znalostí.

Generalizace na zděděných nebo odvozených vlastnostech

Objektově orientované databáze jsou organizovány do hierarchií tříd/podtříd. Některé atributy nebo metody tříd objektů nejsou explicitně specifikovány v třídě jako takové, ale jsou zděděny ze třídy vyšší úrovně. Některé OO databázové systémy umožňují zdědění vlastností z více než jedné supertřídy (vícenásobná dědičnost). Zděděné vlastnosti objektů mohou být odvozeny v OO databázích zpracováním dotazů. Z pohledu získávání znalostí není nezbytností rozlišovat, která data jsou uložena uvnitř třídy anebo zděděna ze supertřídy. Pro proces získávání znalostí jsou relevantní data získávána zpracováním dotazů a je tedy jedno, zda jsou data uložena ve třídě objektů nebo jsou zděděna.

Další důležitou OO databází jsou metody. Mnoho dat může být získáno z objektů pomocí aplikace těchto metod. Metoda je obvykle definována výpočetní procedurou nebo funkcí nebo množinou deduktivních pravidel. Je velmi obtížné generalizovat metody jako takové, proto je třeba, aby aplikační programátoři vytvořili nové metody, které jsou požadovanou generalizací daných metod. Obecně, generalizace na datech získaná aplikací metod by měla být prováděna ve dvou krocích: (1) získání relevantní množiny dat aplikací metod a (2) provedení generalizace ošetřením/úpravou získaných dat.

Generalizace nad hierarchií skládání tříd

Atribut objektu může být složen z nebo popsán jiným objektem a některé z těchto atributů mohou být dále složeny z dalších objektů, tedy jde o hierarchii skládání tříd. Generalizace nad hierarchií skládání tříd může být provedena jako generalizace na množině shluku strukturovaných dat (pokud je shluk rekurzivní, může být množina nekonečnou).

V principu je reference na složený objekt možné vyjádřit jako dlouhou sekvenci referencí odpovídajících hierarchií skládání tříd. Ve většině případů jsou dlouhé sekvence preferencí převedeny do slabších sémantických spojení mezi původním objektem a odkazujícím složeným objektem. Např. jeden atribut `vehicles owned` objektu třídy `student` může odkazovat na jiný objekt třídy `car`, který může obsahovat atribut `auto_dealer`, který může odkazovat na `manager` s atributem `children`. Obvykle je nepravděpodobné, že se najde nějaká zajímavá obecná vazba mezi `studentem` a `dítětem`

ředitele dealera jeho auta. Proto generalizace na třídě objektů je hlavně prováděna na nejbližších hodnotách popisných atributů nebo metodách atd. s omezením referencí nepřímo odkazované části hierarchie skládání tříd .

Pro objevování zajímavých znalostí by měla být generalizace utvářena pouze na objektech hierarchií složených tříd, které jsou úzce sémanticky provázány s právě testovanou třídou, ale ne s těmi, které jsou vzdáleny nebo mají slabé sémantické provázání.

Generalizace tříd

S dostupností operátorů pro generalizaci objektů může být provedena generalizace relevantních objektů založena na třídách. Vzhledem k tomu, že množina objektů ve třídě může sdílet mnoho atributů a metod a generalizace každého atributu a metody může být provedena pomocí aplikace sekvence generalizačních operátorů, které jsou popsány v předchozí kapitole, hlavní důraz je kladen na to, jak skloubit generalizační proces nad různými atributy metodami při získávání zajímavých výsledků.

Generalizace třídy objektů může být brána jako sekvence množinově orientovaných generalizačních procesů, které transformují třídu do relativně více generalizované třídy. Třída předkládaná do generalizačního procesu se nazývá pracovní třída W (working class). Počáteční třída se nazývá počáteční pracovní třída W_0 (initial working class). Třída získaná aplikací generalizačního operátoru se nazývá výsledná třída R (result class).

Atributově orientovaná indukční metoda vyvinutá při získávání znalostí v relačních databázích může být upravena pro objektově orientované databáze. Např. věk může být získán aplikací metody `compute_age` založené na aktuálním datu a hodnotě atributu `birth_date` objektu třídy `person`. Pro generalizaci založenou na třídě je vybrána a aplikována množina generalizačních operátorů na atribut v pracovní třídě s ohledem na vztahy mezi různými atributy předtím než je každý atribut generalizován na vyšší úroveň. Důvod pro ignoraci testování vztahů mezi různými atributy v prvních fázích generalizace je ten, že by se mohlo velmi mnoho nežádoucích generalizací na nízké úrovni. To by nevedlo ke generování elegantních pravidel vyjádřených ve stručné formě. Proto je indukce orientovaná na atributy, která zahrnuje i vztahy mezi atributy, použita až tehdy, kdy jsou tyto atributy generalizovány na relativně vysoké úrovni konceptu. To vede ke snížení výpočetní náročnosti generalizačního procesu databáze.

Formálně, generalizační operátor Gen může být aplikován na atribut A_i každého objektu pracovní třídy W_k , což vede na novou generalizovanou třídu R_k . Tudíž třídní generalizátor `ClassGen` je de facto aplikace objektového generalizačního operátoru Gen na atribut A_i každého objektu v pracovní třídě. Tedy

$$R_k = \text{ClassGen}\{W_k, a_i\} = \{o' : (o \in W_k) \wedge (o'.a_i = Gen(o.a_i)) \wedge (\forall (j \neq i) o'.a_j = o.a_j)\}$$

Např. pro pracovní třídu „ $W_0 = \text{Person}$ “ v univerzitní databázi, `ClassGen(W0, address)` získává výslednou třídu R_0 s jednostupňovou generalizací na atributu „address“ (např. z čísla ulice k bloku ulice) a za všech dalších atributů neměnných.

Na proces objevování znalostí může být nahlíženo jako na aplikaci sekvence operátorů generalizace databáze `ClassGen` na odlišné atributy, dokud výsledná třída neobsahuje pouze malý počet generalizovaných objektů, které mohou být uskupeny do stručného generalizovaného pravidla v termínech vyšší úrovně.

3.5.4 Dolování asociačních pravidel z XML

Z XML dat lze získávat mnoho druhu znalostí v závislosti na druhu dat a způsobu jejich reprezentace. Jako ukázka takového dolování bude představen typický způsob pro

dolování asociačních pravidel z těchto dat. Obvykle je pro vyšší efektivitu dolování nad XML daty použito strategie zpracování vstupních XML dat a jejich uložení do relačních, popř. objektově-orientovaných databází, podle charakteru těchto dat, a následné použití známého dolovacího algoritmu pro dolování nad relačními resp. Objektově orientovanými databázemi,

Obvykle je tedy pro dolování nad XML daty třeba předzpracování či následní zpracování do relačního formátu. V následující části bude popsána metoda postavená nad XMine operátorem využívající klasický přístup pracující ve třech fázích – předzpracování, dolování a následné zpracování.

XMine operator

XMine operátor je nástroj pro komplexní dolování asociačních pravidel v XML dokumentech. XMine umožňuje specifikovat komplexní dolovací úlohy efektivně a intuitivně a eviduje nejrelevantnější sémantické rysy daného problému. XMINE je založen na XPath, který je inspirovaný syntaxí XQuery v kontextu relačních databází.

Ex. 3.10 Ukázka použití XMine operátoru pro dolování asociačních pravidel z XML

Ukázka: V tomto příkladu je demonstrován problém dolování frekventovaných asociací mezi lidmi, kteří jsou označeni jako spoluautoři. Pomocí XMINE je dolovací úloha popsána následujícím způsobem.

XMINE RULE

```
IN document ("www.atlantis.edu/research.xml")
FOR ROOT IN //People/*/Publications/*
LET      BODY := ROOT/Author,
         HEAD := ROOT/Author
EXTRACTING RULES WITH
         SUPPORT = 0.1 AND CONFIDENCE = 0.2
RETURN
<RULE   support={ SUPPORT }
        confidence={ CONFIDENCE }>
<BODY>  { FOR $item IN BODY
          RETURN <Item> { $item } <Item>}
</BODY>
<HEAD>  FOR $item IN HEAD
          RETURN <Item> { $item <Item> }
</HEAD>
</RULE>
```

Definice dolovací úlohy začíná specifikací url adresy XML dokumentu.

Ve FOR sekci proměnná ROOT specifikuje fragment, který reprezentuje transakci. Další proměnné BODY a HEAD identifikují fragmenty ve zdrojových datech které by měly být v tomto pořadí v těle pravidla a v hlavičce pravidla – množina položek. SUPPORT a CONFIDENCE definují podporu a spolehlivost hledaných pravidel.

Výsledek může pak vypadat takto:

```
<RULE support="0.25" confidence="0.33">
```

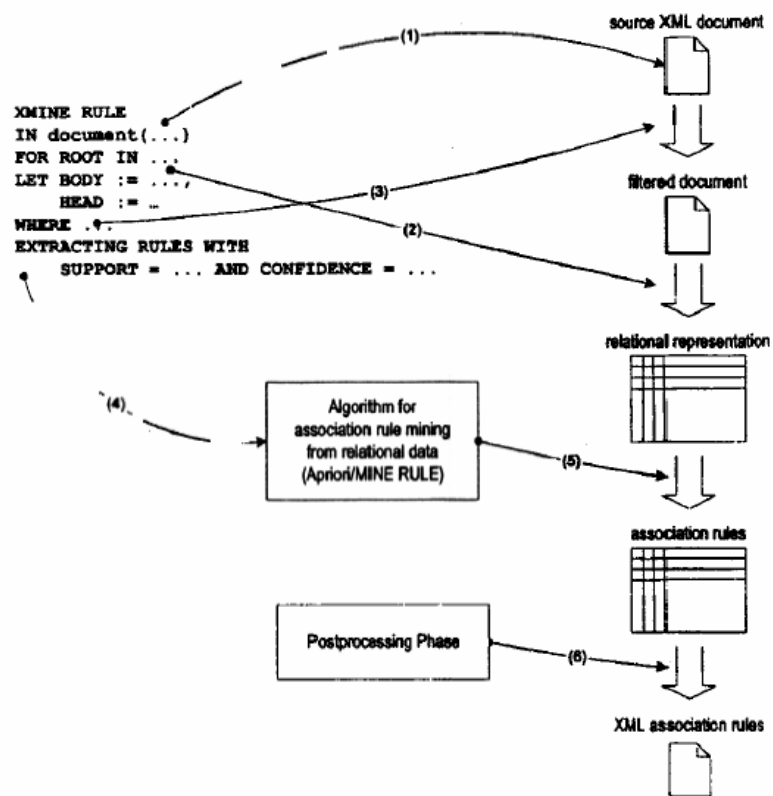



Fig. 3.3 Grafická reprezentace dokumentu, na kterým se provádí dolování

Předzpracování

Nejprve je získána množina XML fragmentů adresovaných pomocí XPath výrazů specifikovaných v klauzuli ROOT, HEAD a BODY. Tento krok může být implementován jakýmkoliv XPath interpretem.

Dolování

Dolování je prováděno nad relační tabulkou získanou v předchozím kroku s omezeními definovanými v definici úlohy. V XMINE prototypu byl využit známý algoritmus Apriori, který pracoval s danými podmínkami, jako jsou minimální podpora a spolehlivost, počet položek v části BODY nebo HEAD. Některá omezení není zatím možno při dolování nad tabulkovými daty implementovat. To bude možné až při získávání asociačních pravidel přímo z DOM (Document Object Model) reprezentace. DOM nabízí platformově i jazykově neutrální API (Application Programming Interface) poskytující programům dynamický přístup a úpravu obsahu, struktury a stylu dokumentů.

Následné zpracování

V tomto závěrečném kroku dochází k exportu získaných asociačních pravidel do XML notace.

4 Komerční databázové systémy

První MMDDBMS se začaly vyvíjet koncem 80. let. Patří mezi ně systémy ORION, ITASCA, ale první komerčně úspěšná, byla začátkem 90. let, především MediaDB společnosti MediaWay, která se v dnešní době jmenuje Chuckwalla. V současné době nabízí databázový systém pro vývojáře multimediálních aplikací na platformě Microsoft. Následující obrázek Fig. 4.1 zobrazuje její architekturu.

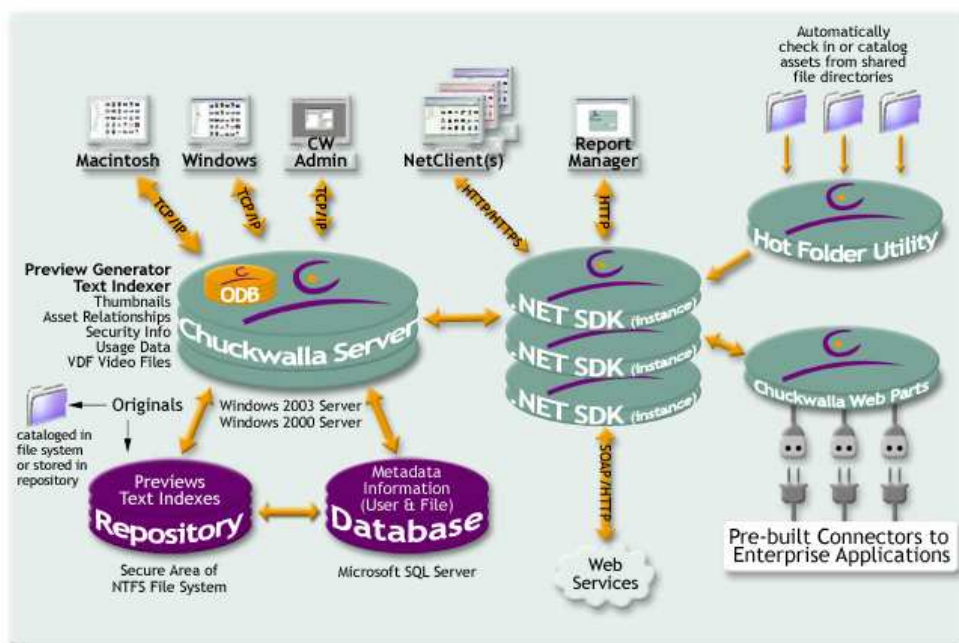


Fig. 4.1 Architektura multimediálního databázového systému (<http://www.chuckwallainc.com/>)

Dalším řešením disponuje například společnost IBM, která pro svoji databázi IBM DB2 Universal database vyvinula rozšíření (Extenders) pro audio, statické obrázky a video (<http://www-306.ibm.com/software/data/db2/extenders/aiv/>).

Nicméně zřejmě nejpropracovanějším řešením disponuje společnost Oracle.

4.1 Oracle interMedia

Oracle® *interMedia* je rozšíření databázového systému pro uchování, získávání, správu a dotazování obrázků, audia, videa, dokumentů a jim přidružených dat pro libovolné aplikace (www.oracle.com/pls/db102/portal.portal_db?selected=7).

Podporuje různé „nedatabázové“ operace (zpracování obrazu) a formáty pro uchování nestrukturovaných dat jako JPEG, MPEG, RA, QT, WMV, PDF, DOC nebo XLS. Pro ně nabízí jednoduché prostředky pro manipulaci s obsahem, například konverzi formátů, ořezání, změnu velikosti, nebo gama korekce. Pro textové dokumenty (výše uvedené včetně HTML a XML) je samozřejmostí efektivní indexace obsahu pro vyhledávání. Obdobně je tomu i pro obrázky. Možné je rozšíření o další speciální algoritmy pro zpracování multimedií, jako jsou další kodeky nebo subsystém pro rozpoznání řeči.

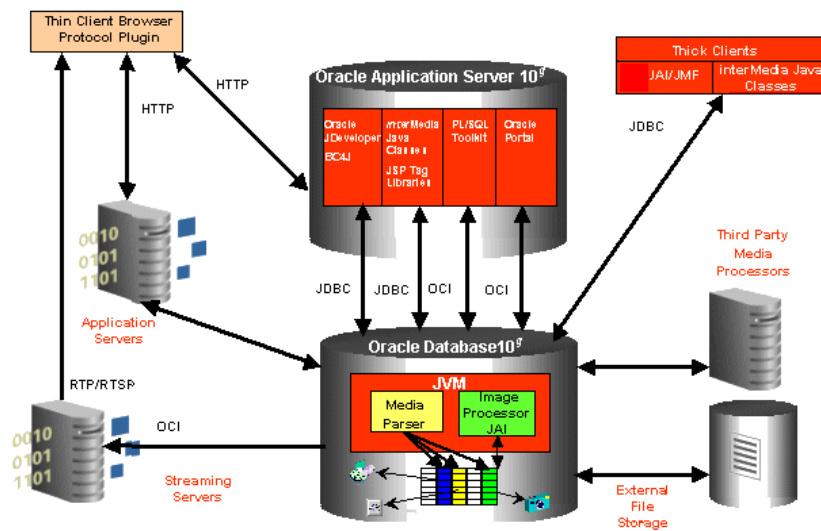


Fig. 4.2 Schéma Oracle interMedia (<http://www.oracle.com/technology/index.html>)

Oracle interMedia je založeno na objektově relačním DBMS. Obsahuje:

- Zdroj a jeho umístění (i mimo databázi) MM ve všech možných formátech.
- Formát MM a jiných, heterogenních, dat. Pro statické obrázky rozměry, rozlišení, počet barev, kompresní formát. U zvukových nahrávek použité kódování, počet kanálů, vzorkovací frekvence a doba trvání. Video má rámce popsané jako obrázky, jejich počet za sekundu, kompresní algoritmus, celkové trvání a datová propustnost, případně typ MIME.
- Metadata získaná z externích zdrojů, například autor, copyright, což jsou informace extrahované z jiných XML metadat a zpracovává zabudované informace, které jsou součástí MM jako je Exif, IPTC-IIM, XMP nebo medicínské DICOM.

Oracle ve verzi Enterprise obsahuje navíc nástroj pro popis statických obrázků dle jejich obsahu. Pro jejich indexaci a podobnostní vyhledávání. Vektor rysů (signature) obsahuje následující nízkouroňové elementy, které extrahuje při ukládání do DB:

- Barva, respektive její distribuce v obrázku – histogram.



Fig. 4.3 Obrázky podobné dle barvy [Ora05]

- Textura charakterizuje zrnitost obrázku a hranovou frekvenci.

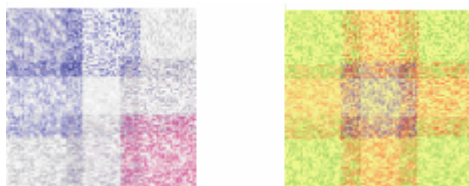


Fig. 4.4 Obrázky podobné dle textury [Ora05]

- Tvar objektů, získaných na základě segmentace obrázu, provedené podle barvy.



Fig. 4.5 Obrázky podobné dle tvaru [Ora05]

- Umístění barevných segmentů v obraze. Pomocí této vlastnosti se implicitně nevyhledává, ale slouží například k určení pozadí, jak ukazuje obrázek Fig. 4.7.



Fig. 4.6 Dva obrázky podobné dle barvy a umístění zároveň [Ora05]

Tento popis slouží pro podobnostní vyhledávání, realizované podle vážené vzdálenosti jednotlivých elementů v databázi, jejichž celková velikost je asi 3-4kB.

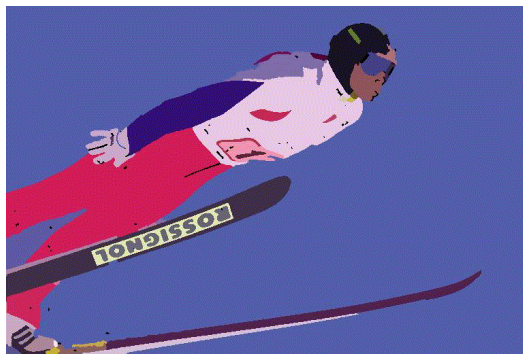


Fig. 4.7 Znázorňuje segmentaci do oblastí podle jejich barvy, z [Or05]

InterMedia aplikace je možné vyvíjet prostřednictvím jazyků Java, PL/SQL a některých dalších, příklady jejich použití jsou uvedeny v [Or05] a příloze 2 tohoto dokumentu.

Zamyšlení ke 4. kapitole

Jak by mohla vypadat nejjednodušší multimediální databáze? (Ve smyslu DVB přijímač s diskem a ethernet rozhraním nebo Galerie umění). Co taková databáze dokáže?

Jaké multimediální systémy řízení báze dat znáte?

Popište operace a data, nad kterými jsou prováděny.

Zkuste si implementaci MM databáze. Návod je uvedený v příloze 2.

5 Shrnutí

Tato práce představuje multimediální databáze jako celek. Je kompilací dostupné literatury, kterou mírně doplňuje (v oblastech, které se nám zdají zajímavé).

Navíc, k úvodní kapitole, je možné za úvod do problematiky multimediálních databází považovat kapitolu 2.2. Práce se zabývá jednotlivými požadavky multimediálních a složitě strukturovaných databází a jejich odchylkami od databází klasických, založených na relačním modelu a dalších systémů.

Práce se zaměřuje na popis obsahu založeném na primitivních vlastnostech multimédií, například barva, textura, tvar, pohyb u vizuálních, nebo frekvenční charakteristika u audio dat ve formátu XML a prostředky pro jejich manipulaci. Tyto prostředky jsou použitelné pro vyhledávání maximálně střední úrovně.

V kapitolách o získávání znalostí jsou naznačeny metody, které dokáží propojit nízkoúrovňové rysy s vysoce žádaným sémantickým popisem. Problémem je především obrovská diverzita obecných multimediálních dat, proto jsou aplikace pro nalezení informací obsažených v multimediálních a složitě strukturovaných datech úzce omezeny. V tomto vidím největší možnost rozvoje multimediálních databází.

Literatura

- [Bak87] BAKEN, R. J. *Clinical Measurement of Speech and Voice*. Taylor and Francis Ltd, 1987. 518 s. ISBN 0316078336.
- [Bour07] Bourret, Ronald: XML and Databases, [cit. 2007-10-12]. Dostupný z: <http://www.rpbouret.com/xml/XMLAndDatabases.htm>
- [Cai03] CAIXUE, Lin. *Object-Oriented Database Systems: A Survey* [online]. UCSC, 2003 [cit. 2006-05-25]. Dostupný z: <http://www.soe.ucsc.edu/~lcx/courses/cmcs277/cmcs277-project.pdf>.
- [Cat00] CATTELL, R. G. G. et al. *Object data standard: ODMG 3.0*. 2000...
- [Čer05] ČERNOCKÝ, Jan. *Stránka pro studenty kursu "Signály a systémy - ISS"* [online]. 2005 [cit. 2006-05-16]. Dostupný z: <http://www.fit.vutbr.cz/~cernocky/sig/>.
- [Doub07] Doubravec, Petr: Home pages, Z XML do PDF, [cit. 2007-10-12]. Dostupný z: <http://www.dubracec.cz/dubracecovi/docb.htm>
- [Cha01] CHAUDHRI, B. A. – ZICARI, R. *Succeeding with object databases*. J. Wiley, 2001. 442p. ISBN 0-471-38384-8.
- [Chl05] CHLUM, O – MATAS, J. Matching with PROSAC - progressive sample consensus. *Proc. of Conference on Computer Vision and Pattern Recognition* [online]. 2005. [cit. 2006-06-19]. Dostupný z: <http://cmp.felk.cvut.cz/~matas/>.
- [Chm006] CHMELARĚ, P. Bayesian Concepts for Human Tracking and Behavior Discovery, *Student EEICT 2006* [online]. VUT Brno, 2006. Dostupný z: http://www.feec.vutbr.cz/EEICT/2006/sbornik/03-Doktorske_projekty/07- Informacni_systemy/03-chmelarp.pdf.
- [Chm04] CHMELARĚ, P. *Texturní analýza - metody založené na zpracování signálů* [online]. FIT VUT Brno, 2005. Vedoucí práce Heckel, M. Dostupný z: <https://wis.fit.vutbr.cz/FIT/db/dir.php/rp/2003/RP/1704.pdf>.
- [Chm06] CHMELARĚ, P. *ZSD: Získávání znalostí z multimediálních databází*. FIT VUT Brno, 2006. Vedoucí práce Zendulka, J.
- [COV95] Cortes, C. – Vapnik, V. Support-Vector Networks, *Machine Learning*, 20. 1995.
- [Dar87] WOELK, Darrell – WON, Kim. Multimedia information management in an object-oriented database system. In *Proceedings of the Thirteenth International Conference on Very Large Databases*, Morgan Kaufmann Publishers Inc., 1987...
- [Eis02] EISENBERG, Andrew – MELTON, Jim. *SQL/XML is Making Good Progress* [online]. 2002. [cit. 2006-06-19]. Dostupný z: <http://www.sigmod.org/record/issues/0206/standard.pdf>.
- [Eis04] EISENBERG, Andrew et al. *SQL:2003 Has Been Published* [online]. 2004. [cit. 2006-06-19]. Dostupný z: <http://www.sigmod.org/sigmod/record/issues/0403/E.JimAndrew-standard.pdf>.
- [Eis99] EISENBERG, Andrew – MELTON, Jim. *SQL:1999, formerly known as SQL3* [online]. 1999. [cit. 2006-06-19]. Dostupný z: <http://dbs.uni-leipzig.de/files/lehre/sql99-eisenberg-melton-sigmod-record99.pdf>.
- [Fer06] Fernández-Dívar, I. E. *Image Data Management*. 2006. Vedoucí DP Chmelař, P.
- [Fis06] FISCHER, S. – KLINKENBERG, R. – MIERSWA, I. *YALE 3.3, Yet Another Learning Environment* [online]. 2006 [cit. 2006-09-10]. Dostupný z: http://rapid-i.com/component/option,com_weblinks/catid,24/Itemid,60.
- [Gir06] Giralda, A. R. *Image Databases Indexing*. 2006. Vedoucí MT Zendulka, J.

- [Gra04] Grandvalet, Y. – SANU, C. *Adaptive Scaling for Feature Selection in SVMs* [cit. 2006-09-10]. 2004. Dostupný z: <http://www.hds.utc.fr/~grandval/nips02.pdf>.
- [Han00] HANÁČEK, P. – STAUDEK, J. *Bezpečnost informačních systémů* [online]. 2000 [cit. 2006-09-10]. Dostupný z: <http://www.micr.cz/files/479/uvis-Bezpecnost-20000701.pdf>.
- [Han01] HAN, J., KAMBER, M. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2001. ISBN 1-55860-489-8.
- [Hig06] HIGASHI, T. – KITA, S – WATANABE, I. Texture and Objects: Interruption of Same-Object Effect in Human Vision, *IEICE Trans Inf & Syst* E89-D: 1806-1812. 2006.
- [Hin03] HINNER, J. Detekce a rozpoznávání obličejů osob a jejich identifikační význam, *Kriminalistika* [cit. 2006-09-10]. 1/2003. Dostupný z: http://www.mvcr.cz/casopisy/kriminalistika/2003/03_01/hinner.html<http://rfc.net/rfc3261.html>.
- [Hof03] HOFMAN, Jiří. *Formáty multimediálních souborů* [online]. 2006. [cit. 2006-06-19]. Dostupný z: <http://www.aldebaran.cz/studium/formaty.html>.
- [Hru05] HRUŠKA, T. *Informační systémy, Podklady pro studium* [cit. 2006-06-19]. 2005. Dostupný z: <https://www.fit.vutbr.cz/study/courses/INS/private/>.
- [ITU06] International Telecommunication Union. *H.323 : Packet-based multimedia communications systems*. 2006.
- [Jäh99] JÄHNE, Bernd – HAUSSECKER, Horst – GEISLER, Peter. *Handbook of Computer Vision and Applications*. Academic Press, 1999. 3 vol. (624, 942, 894 p.). ISBN 0-12-379770-5.
- [Kad02] KADOUS, M.W. *Dynamic Time Warping* [cit. 2006-09-10]. 2002. Dostupný z: <http://www.cse.unsw.edu.au/~waleed/phd/html/node38.html>.
- [Kol05] KOLÁŘ, D. *Postrelační databáze, Přednášky* [cit. 2006-09-06]. 2005. Dostupný z: <https://www.fit.vutbr.cz/study/courses/PRD/private/>.
- [Kos03] KOSCH, Harald. *Distributed Multimedia Database Technologies Supported by MPEG-7 and MPEG-21*. CRC Press, 2003. 280 s. ISBN 0-8493-1854-8.
- [Kos07] Kosek, Jiří: XML pro každého, [cit. 2007-10-12]. Dostupný z: <http://www.kosek.cz/xml/index.html>
- [Kra98] KRÁL, J. *Informační systémy*. Brno: Science, 1998. 357 s. ISBN 80-86083-00-4...
- [Krš05] KRŠEK, Přemysl. *Základy počítačové grafiky, přednášky* [online]. 2005. [cit. 2006-06-19]. Dostupný z: <https://www.fit.vutbr.cz/study/courses/IZG/private/>.
- [Kuc04] KUCHARĚK, T. – KRŮČEK, J. *SQL Multimedia and Application Packages* [online]. 2004 [cit. 2006-09-22]. Dostupný z: http://kocour.ms.mff.cuni.cz/~pokorny/dj/prezentace/2_51.ppt.
- [Mis04] Miškovský, M.: *Štandardy MPEG-4, MPEG-7*. Diplomová práce, Telekomunikačná katedra, Elektrotechnická fakulta, ŽU v Žiline, 2004
- [MP210] ISO/IEC JTC1/SC29/WG11. *MPEG-21 Overview* [online]. Bormans, J. – Hill K. Shanghai: 2002 [cit. 2006-09-14]. Dostupný z: <http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm>.
- [MP402] ISO/IEC JTC1/SC29/WG11. *MPEG-4 Overview*. [online]. Koenen, Rob. 2002 [cit. 2006-05-22]. Dostupný z: <http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm>.
- [MP704] ISO/IEC JTC1/SC29/WG11. *MPEG-7 Overview* [online]. Martínez, José M. Palma de Mallorca : 2004 [cit. 2005-10-22]. Dostupný z: <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>.

- [Nos06] NOSEK, F. *Podpora pro práci s multimediálními daty u serveru Oracle* [online]. 2006 [cit. 2006-05-24]. Vedoucí DP Zendulka, J. Dostupný z: <https://wis.fit.vutbr.cz/FIT/db/dir.php/rp/2005/DP/3842.pdf>.
- [ODL05] Oxford Digital Library. *Metadata in the Oxford Digital Library*. [online]. 2005 [cit. 2006-05-16]. Dostupný z: <http://www.odl.ox.ac.uk/metadata.htm>.
- [Or05] Oracle Corporation. *Oracle Database Online Documentation 10g Release2* [online]. 2005 [cit. 2005-10-27]. Dostupný z: http://www.oracle.com/pls/db102/portal.portal_db.
- [Ora06] ORACLE Corporation. *Oracle Database Online Documentation 10g Release2* [online]. 2005 [cit. 2006-05-16]. Dostupný z: http://www.oracle.com/pls/db102/portal.portal_db.
- [Pla04] PLAGEMANN, Thomas et. al. *Using Data Stream Management Systems for Traffic Analysis. A Case Study* [online]. 2004. [cit. 2006-06-19]. Dostupný z: <http://www.pam2004.org/papers/113.pdf>.
- [Raj03] RAJ, C. Pethuru. *Multimedia Database*, In *Information Technology Portal* [online]. 2003 [cit. 2006-05-25]. Dostupný z: <http://www.peterindia.net/MultimediaDatabase.html>.
- [RCo06] RAD COM. *Voice Over IP Reference Page* [online]. 2006. [cit. 2006-06-19]. Dostupný z: <http://www.protocols.com/pbook/VoIP.htm>.
- [Ros02] ROSENBERG, J., et. al. *SIP: Session Initiation Protocol* [online]. 2002. [cit. 2006-09-10]. Dostupný z: <http://rfc.net/rfc3261.html>.
- [Skri07] Skřivan, Jaromír: *Jeremyho web – Dotazovací jazyky nad XML (XML QL)*, <http://www.jeremy.cz/article.php?aid=15&cid=2&rid=3>
- [Sor70] SORENSON, H. W. *Least-squares estimation: from Gauss to Kalman*, *IEEE Spectrum* [online]. 1970. [cit. 2006-09-10] Dostupný z: <http://www.cs.unc.edu/~welch/kalman/media/pdf/Sorenson1970.pdf>.
- [SQL03] Melton, Jim. *SQL (ISO-ANSI Working Draft)*. 2003.
- [Sub98] SUBRAHMANIAN, V. S. *Multimedia database systems*. Morgan Kaufmann, 1998. 442 s. ISBN 1-55860-466-9.
- [Švé05] ŠVÉDA, Miroslav. *PDS: Osnova přednášek* [online]. 2006. [cit. 2006-06-19]. Dostupný z: <https://www.fit.vutbr.cz/study/courses/PDS/private/>.
- [Zbo05] ZBOŘIL, F. *Neuronové sítě, Obsah přednášek* [online]. 2005 [cit. 2006-09-10]. Dostupný z: <https://www.fit.vutbr.cz/study/courses/NEU/private/>.
- [Zem03] ZEMČÍK, Pavel. *Multimédia, ořednášky* [online]. 2003. [cit. 2006-06-19]. Dostupný z: <https://www.fit.vutbr.cz/study/courses/MUM/private/>.
- [Zen05] ZENDULKA, J. *IDS: Materiály k přednáškám ve formátu PDF* [online]. 2005 [cit. 2006-05-16]. Dostupný z: <https://www.fit.vutbr.cz/study/courses/IDS/private/>.

Příloha 1. Rejstřík pojmů

CORBA	(Common Object Request Broker Architecture) definuje API, komunikační protokol, objektový model a služby pro nezávislé, heterogenní a distribuované aplikace
Data	potenciální informace
Databáze	organizovaná kolekce dat
Databázový systém	systém řízení báze dat, více v [Zen05]
DBMS	(DataBase Management System) systém řízení báze dat
Formát média	informace přiřazená médiu v průběhu získávání, komprese nebo přenosu.
Informace	užitečná, použitelná, relevantní nebo zajímavá data
Klíčová slova	ruční popis média
Média	nositel informace, přeneseně multimediální data
Metadata	informace o datech, zde popis (klíčová slova, rysy) a formát média.
MM	MultiMédia
MMDB	MultiMediální DataBáze
Multimédia	digitalizovaná audiovizuální (nestrukturovaná) data; obrázky, audio, video, text
ODMG	(Object Database Management Group) skupina, sjednotila objektový přístup k DB
OLAP	(Online Analytical Processing) analytické funkce, známé jako Business intelligence, data jsou obvykle agregována (sumarizována) do podoby multidimenzionální kostky.
Popis	(description) manuálně nebo automaticky vytvořená charakteristika média, metadata.
Rysy	(features) charakteristika výzazného prvku média
SQL	(Structured Query Language) dotazovací jazyk pro definici, manipulaci, řízení integrity a (souběžného) přístupu k datům na relační bázi
XML	(eXtensible Markup Language) univerzální značkovací jazyk
ISO	(International Organization for Standardization) mezinárodní normalizační institut (v ČR zastoupen ČNI – Český normalizační institut), standard, norma (obdoba ČSN)
VoIP	(Voice over IP) hlas (a v současné době i video) přenášené protokolem IP
CBR	(Content Based Retrieval) vyhledávání dle obsahu dat.

Další bude doplněno

Příloha 2. Oracle interMedia aplikace

Pro tvorbu libovolné aplikace je stěžejní výběr prostředků. Databází zde bude Oracle Database 10g včetně rozšíření interMedia. Pro přístup Oracle doporučuje objektově-relační přístup pomocí jazyků Java a PL/SQL i když je k multimediálním datům možné přistupovat i prostřednictvím relačního modelu a tedy téměř libovolného procedurálního strukturovaného jazyka (pro seznámení s alternativami začněte na: www.oracle.com/technology/sample_code/products/intermedia/).

Pro následující ukázkou připojení, uploadu a downloadu obrázku bude použita Java a Oracle JDeveloper, který umožňuje vývoj okenních i webových aplikací.

Připojení k databázi

Nejprve je nutné se připojit se k databázi a potřebné funkce importovat:

```
// JDBC a SQLJ
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.Statement;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import oracle.jdbc.OracleResultSet;
import oracle.jdbc.OraclePreparedStatement;
// set the CLASSPATH variable
set CLASSPATH = %ORACLE_HOME%\jdbc\lib\ojdbc14.jar;
                %ORACLE_HOME%\sqlj\lib\runtime12.jar;

// interMedia
import oracle.ord.im.OrdImage;
import oracle.ord.im.OrdImageSignature;
// set the CLASSPATH variable
set CLASSPATH=%ORACLE_HOME%\ord\jlib\ordim.jar;
```

Potřebné funkce pro práci s JDBC a MM daty jsou obsaženy v balíčcích dodávaných spolu s JDeveloperem (v adresáři %ORACLE_HOME%, ve složce Q:\ORACLE\JDeveloper).

Pro připojení je nutné registrovat ovladač, připojit se k databázi – na serveru **pcuifs1** v současné době běží Oracle Database 10g. Jako jméno a heslo je obvykle přednastavený **xlogin**. Pro práci s velkými binárními daty (dvojfázový uzamykací protokol) je nutné provádět **commit** ručně(!) a inicializovat SQL statement:

```
// register the oracle JDBC driver with the driver manager
DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());

// connect pcuifs1.fit.vutbr.cz
Connection conn = DriverManager.getConnection("jdbc:oracle:thin:"
+ "login/password"
+ "@pcuifs1.fit.vutbr.cz:1521:stud");

// Note: it is CRITICAL to set the autocommit to false
conn.setAutoCommit(false);

// create a JDBC Statement object to execute SQL in the database
Statement stmt = conn.createStatement();
```

Základní datové typy

Multimediální data je možné v databázi uložit jako BLOB, ale výhodnější je využití speciálních UDT. V případě obrázku je to typ **ORDSYS.ORDImage**, jako na Fig. 5.1.

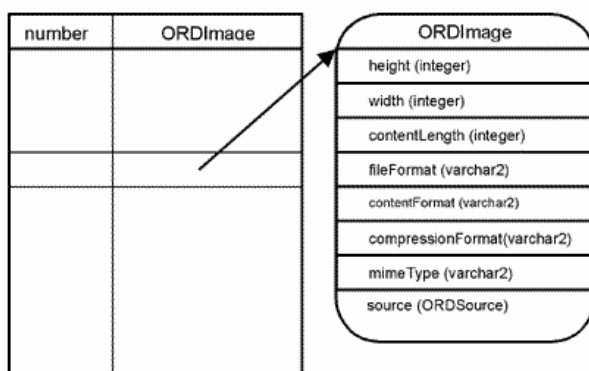


Fig. 5.1 Struktura typu ORDImage v databázi Oracle [Or05]

Pro přístup k databázi se v Javě využívá JDBC, které obecně nepodporuje objektový přístup. Proto jsou vyvinuty Java objekty pro manipulaci s těmito daty, nicméně všechny operace v objektově-relační databázi nadále vykonává DBMS, proto musí být vytvořeny jako proxy objekty (**ORDImage**) z databázového objektu, jak je uvedeno na obrázku níže.

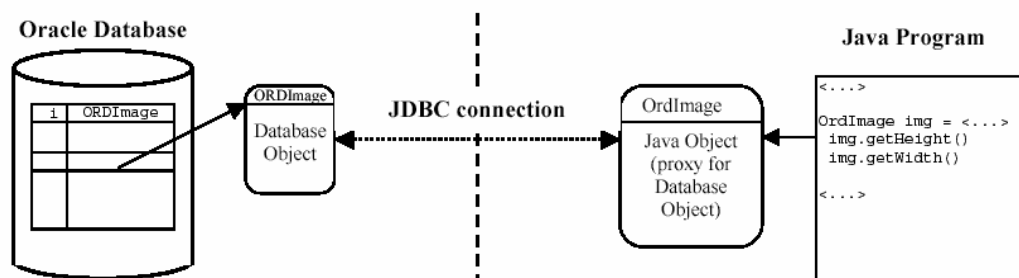


Fig. 5.2 Schéma použití ORDImage jako objektu v databázi a proxy v Javě [Or05]

Příklad vytvoření tabulky pro uchovávání obrázků:

```
CREATE TABLE image_table
(id number primary key,
 image ordsys.ordimage,
 signature ordsys.ordimagesignature
);

CREATE SEQUENCE image_sequence
INCREMENT BY 1 START WITH 1;
```

Vytvořená tabulka obsahuje sloupec pro uchování obrázku (**image**) a popisu tohoto obrázku (**signature**) pro následující vyhledávání dle obsahu a sloupec (**id**) jako primární klíč, jehož unikátní hodnoty se získávají ze sekvence definované pod ní.

Vložení obrázku

Nejprve načteme hodnotu primárního klíče ze sekvence (**nextval**) a poté jej uložíme spolu s inicializovanými objekty obrázku a jeho popisu.

```
// retrieve the next value in the sequence order
String SQLquery = ("SELECT image_sequence.nextval FROM dual");
OracleResultSet rset = (OracleResultSet)stmt.executeQuery(SQLquery);
rset.next();
int nextval = rset.getInt("nextval");

// insert a row (nextval, initialized ORDImage) into image_table
SQLquery = "INSERT INTO image_table (id, image, signature) VALUES (" +
nextval + ", ordsys.ordimage.init(), ordsys.ordimagesignature.init())";
stmt.execute(SQLquery);
```

Protože Javové proxy objekty musí být vytvořeny z databázových objektů, musíme právě vložené objekty pro obrázek a jeho popis načíst pro další úpravu:

```
// select the new ORDImage database object for update
SQLquery = "SELECT image, signature FROM image_table WHERE id = "
           + nextval + " FOR UPDATE";
rset = (OracleResultSet)stmt.executeQuery(SQLquery);
rset.next();

// select the image into a java proxy OrdImage object
OrdImage imageProxy =
    (OrdImage)rset.getORADData("image", OrdImage.getORADDataFactory());
// if not working use deprecated version of JDBC/SQLJ:
// (OrdImage)rset.getCustomDatum("image", OrdImage.getFactory());

// select the signature into a java proxy OrdImageSignature object
OrdImageSignature signatureProxy =
    (OrdImageSignature)rset.getCustomDatum("signature",
                                           OrdImageSignature.getFactory());
rset.close();
```

Až teprve teď máme inicializované objekty a obrázek se může fyzicky načíst:

```
// load the image data from file "fileName" into the ORDImage object
imageProxy.loadDataFromFile(fileName);
```

Předchozí funkce nad proxy objektem způsobí, že se obrázek načte nejen do lokálního objektu, ale také do databáze, která poté provádí všechny operace nad ním. Jsou to především operace pro inicializaci metadat (vlastností) a popis obsahu:

```
// automatically detect the image's height, width, file format...
// execute ORDImage.setProperties() on the server
imageProxy.setProperties();

// generate an image signature for the specified image on server
signatureProxy.generateSignature(imageProxy);
```

Nyní je nutné provést update a commit, aby byly provedené operace reflektovány:

```
SQLquery = "UPDATE image_table SET image=?, signature=? where id=?";
OraclePreparedStatement opstmt =
    (OraclePreparedStatement)conn.prepareStatement(SQLquery);
opstmt.setCustomDatum(1, imageProxy);
opstmt.setCustomDatum(2, signatureProxy);
opstmt.setInt(3, nextval);
opstmt.execute();
opstmt.close();

// commit all changes
conn.commit();
```

V textové reprezentaci SQL dotazu je poněkud obtížné zapsat (referenci na) obrázek a jeho popis (to lze například pomocí MIME), proto je vhodné doraz předpřipravit a poté do něj vložit uživatelská data, nahrazením (??) funkcí `setCustomDatum`.

Dotazování vlastností a manipulace s obrázky

Obrázek je nyní perzistentně uložen v databázi a je možné jej dotazovat. Nejprve si ukážeme, jak lze zjistit velikost a další vlastnosti:

```
// select the image into the proxy object (imageProxy)
String SQLquery = "select image from image_table where id = 1";
```

```

OracleResultSet rset = (OracleResultSet)stmt.executeQuery(SQLQuery);
rset.next();
OrdImage imageProxy =
    (OrdImage)rset.getORADData("image", OrdImage.getORADDataFactory());
// (OrdImage)rset.getCustomDatum("image", OrdImage.getFactory());
rset.close();

// and call getX methods ... type imageProxy. & press CTRL+SPACE
int height = imageProxy.getHeight();
int width = imageProxy.getWidth();

```

Poté, pokud například chceme uložit jeho náhled, provedeme následující zpracování:

```

SQLQuery = ("insert into image_table (id, image) " +
           "values (2, ordsys.ordimage.init())");
...
// get the newly inserted destination ORDImage object
SQLQuery = "select image from image_table where id=2 for update";
...
OrdImage dstImageProxy =
    (OrdImage)rset.getORADData("image", OrdImage.getORADDataFactory());

// call the processCopy method (processing occurs on the SERVER)
ImageProxy.processCopy(
    "maxscale=100 100 fileformat=jfif", dstImageProxy );

// update the destination image in the second row
String dstUpdateSQL = "update image_table set image=? where id=2";
opstmt.setORADData(1, dstImageProxy);
// commit

```

Uvedená funkce `processCopy` zajišťuje kopii po zpracování obrázku se zadanými parametry (viz JavaDoc [Or05]), obdobně samotné zpracování funkcí `process`.

Stažení obrázků a jejich uložení do souboru je také poměrně jednoduchá operace:

```

// export the data in row 2
String SQLQuery = "select image from image_table where id = 2";
OracleResultSet rset = (OracleResultSet)stmt.executeQuery(SQLQuery);
rset.next();
OrdImage imageProxy =
    (OrdImage)rset.getORADData("image", OrdImage.getORADDataFactory());
rset.close();

// call the getDataInFile method to write the image in row 2 to disk
imageProxy.getDataInFile("row2.jpg");

```

Dotazování jiných typů multimedií (zvuk, video, dokumenty) je obdobné, jen obsahují jiné funkce pro jejich manipulaci (v Javě i databázi):

```

import oracle.ord.im.OrdAudio;
import oracle.ord.im.OrdDoc;
import oracle.ord.im.OrdImage;
import oracle.ord.im.OrdVideo;

String SQLQuery = "select product_photo, " +
    " product_audio," +
    " product_video, " +
    " product_documents from products where product_id=3117";
PreparedStatement pstmt = conn.prepareStatement(SQLQuery);
OracleResultSet rset = (OracleResultSet)pstmt.executeQuery();
if ( rset.next() )
{

```

```

OrdImage imgProxy = (OrdImage)rset.getORAData(
    "product_photo", OrdImage.getORADataFactory());
OrdAudio audProxy = (OrdAudio)rset.getORAData(
    "product_audio", OrdAudio.getORADataFactory());
OrdVideo vidProxy = (OrdVideo)rset.getORAData(
    "product_video", OrdVideo.getORADataFactory());
OrdDoc docProxy = (OrdDoc)rset.getORAData(
    "product_testimonials", OrdDoc.getORADataFactory());
}

String audFormat = audProxy.getFormat();
String vidMimetype = vidProxy.getMimeType();

```

Mazání je provedeno obvyklým způsobem, pomocí SQL:

```

DELETE FROM image_table

// drop image table and sequence
stmt.executeQuery("DROP TABLE image_table");
stmt.executeQuery("DROP SEQUENCE image_sequence");
// commit all changes
conn.commit();

```

Vyhledávání dle podobnosti

Vyhledávání dle obsahu je jedna z věcí, které odlišují Oracle od běžných databázových producentů. Před vlastním vyhledáváním je nutné vytvořit tento popis dle obsahu (**ORDImageSignature**) pomocí funkce **generateSignature**. Vlastní dotaz je možné provést užitím spárovaných funkcí (**123**) pomocí SQL:

```

SELECT img1.id, img2.id, ORDSYS.IMGScore(123) as distance
FROM image_table img1, image_table img2
WHERE ORDSYS.IMGSimilar(img1.signature, img2.signature, 'color="0.3"
texture="0.3" shape="0.3" location="0.1"', 50, 123) = 1
AND img1.id <> img2.id
AND img1.id = 1
ORDER BY distance

```

K vlastním funkcím: **IMGSimilar** porovnává dvě signatury a vrací hodnotu (**0 / 1**), která indikuje, zda jsou obrázky dle váhy zadaných rysů (barva, textura, tvar a umístění) podobné více, než specifikovaný treshold (**50**). Spřátelená funkce **IMGScore** vrací míru podobnosti těchto obrázků, kde **0** znamená totožnost a **100** naprostou odlišnost.

Příloha 3. Práce s XML daty

Následující obrázek zobrazuje datové typy jazyka XPath 2.0 a XQuery 1.0. Obrázek je převzat z literatury [Skri07].

Identifikátor osy	Popis osy
ancestor	Rodič aktuálního uzlu a všichni jeho další předci až ke kořenovému uzlu. Uzly jsou uspořádány v opačném pořadí než v dokumentu. Zkrácená cesta nelze.
ancestor-or-self	Aktuální uzel a všichni jeho další předci až ke kořenovému uzlu. Uzly jsou uspořádány v opačném pořadí než v dokumentu. Zkrácená cesta nelze.
attribute	Uzly odpovídající všem připojeným atributům, pořadí není nijak pevně definováno. Zkrácená cesta pomocí znaku @.
child	Všechny děti aktuálního uzlu ve stejném pořadí jako v dokumentu. Zkrácená cesta implicitně vynechává tuto osu.
descendant	Všichni potomci aktuálního uzlu ve stejném pořadí jako v dokumentu. Zkrácená cesta nelze.
descendant-or-self	Aktuální uzel a všichni jeho potomci ve stejném pořadí jako v dokumentu. Zkrácená cesta pomocí znaků //.
following	Aktuální uzel a všichni jeho potomci ve stejném pořadí jako v dokumentu. Zkrácená cesta nelze.
following-sibling	Všechny následující uzly, které jsou sourozenci aktuálního uzlu, ve stejném pořadí jako v dokumentu. Zkrácená cesta nelze.
namespace	Uzly odpovídající deklarovaným jmenným prostorům, pořadí není nijak pevně definováno.
parent	Rodič aktuálního uzlu. Zkrácená cesta pomocí znaků .. (dvě tečky).
preceding	Všechny uzly, jejichž všichni potomci jsou v dokumentu ještě před aktuálním uzlem, řazeny v opačném pořadí než v dokumentu. Zkrácená cesta nelze.
preceding-sibling	Všechny předcházející uzly, které jsou sourozencem aktuálního uzlu. Uzly jsou uspořádány v opačném pořadí než v dokumentu. Zkrácená cesta nelze.
self	Aktuální (kontextový) uzel. Zkrácená cesta pomocí znaku . (tečka).

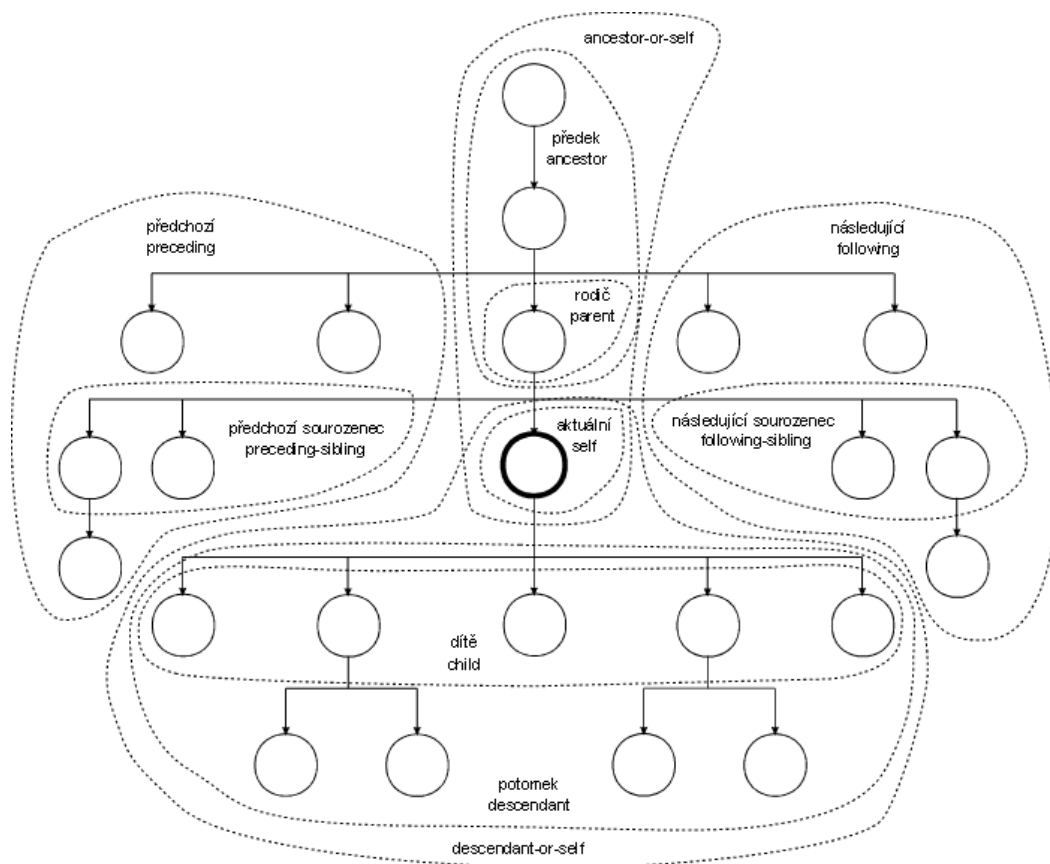


Fig. 5.4 Osy jazyka XPath [Kos07]

Ukázka použití jazyka XPath

- /OBRAZKY - výraz vybere všechny elementy OBRAZKY
- OBRAZKY/* - výraz vybere všechny synovské elementy (element OBRAZKY)
- /OBRAZKY/OBRAZEK/@id - výraz vybere všechny atributy id
- /OBRAZKY/OBRAZEK/NAZEV[1] - vybere první jméno dokumentu
- /OBRAZKY/OBRAZEK/NAZEV[last()] - vybere poslední jméno dokumentu
- //OBRAZEK[VYTVORENO/STAT='cze']/AUTOR - vybere Autora
Obrazku, který vznikl v České republice
- /self::NAZEV - vybere aktuální uzel, pokud je to element NAZEV
- following::*[1] - vybere první element, který se nachází za aktuálním uzlem

Ukázky použití XSLT

Obecná šablona spojení XSLT a Xpath

```

<?xml version='1.0' encoding='windows-1250'?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="OBRAZKY">
    <xsl:apply-templates select="OBRAZEK" />
  </xsl:template>
  <xsl:template match="OBRAZEK">
    <xsl:value-of select="@id" />
    <xsl:value-of select="AUTOR" />
  </xsl:template>

```

Výsledek vrácený po provedení předchozího příkladu:

```
1
Fuk Tomáš
2
Buben Prokop
```

Generování SQL sekvence dotazů z XSLT

```
<?xml version='1.0' encoding='iso-8859-2'?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="OBRAZKY/OBRAZEK">
    <xsl:text>INSERT INTO obrazky (id, nazev, popis, autor) </xsl:text>
    <xsl:text>values ('</xsl:text>
    <xsl:value-of select="@id"/><xsl:text>', '</xsl:text>
    <xsl:value-of select="NAZEV"/><xsl:text>', '</xsl:text>
    <xsl:value-of select="POPIS"/><xsl:text>', '</xsl:text>
    <xsl:value-of select="AUTOR"/><xsl:text>');&#xA;</xsl:text>
  </xsl:template></xsl:stylesheet>
```

Výsledkem předcházejícího příkladu budou SQL příkazy uložené v novém souboru, který je možné zaslat do databáze. Výstup je pak v následující podobě.

```
INSERT INTO obrazky (id, nazev, popis, autor)
VALUES (1,'Cyklista','Cyklista vedoucí kolo za destě','Fuk Tomáš');
INSERT INTO obrazky (id, nazev, popis, autor)
VALUES (1,'Krtek','Maskot charitativní organizace','Buben Prokop');
```

Ukázka použití XQuery funkce - doc()

```
{-- komentář--}
doc("obrazky.xml")                                {-- vrací celý dokument--}
doc("obrazky.xml")/OBRAZKY/OBRAZEK                {--vrací údaje o obrazcích--}
doc("obrazky.xml")/OBRAZKY/OBRAZEK/AUTOR/*       {--všechny položky autorů--}
```