Chapter 2 - Concepts and Definitions

Introduction and Requirements

Literatur:
Database Systems

- Usage: management of large sets of data
  - well-structured
  - elimination of redundancies
  - flexible query/search support
  - flexible data usage (open for new applications)
  - concurrent data access by multiple applications
  - high data currency
  - fault tolerance

- Key concepts
  - data model and DB schema
    - data modeling (DB design) is a complex, expensive task
    - requires tight cooperation of developer and user
  - normalization
  - query language (selection, join, aggregation, ...)
  - synchronization of data access operations
  - transactions
    - "all or nothing" behavior for a sequence of operations
    - automatic recovery from inconsistent DB-states due to failure
Multimedia Databases as Content Repositories

- What can databases do for managing multimedia data?
  - data independence
  - data structures and DB schemas are application-independent
  - support for storage, search

- Fundamental service
  - infrastructure for numerous applications
  - focus on APIs, not so much on end-user interfaces

- Storage and retrieval of multimedia objects

- Key concepts (in addition to database concepts discussed above)
  - device- and format-independent access to MM objects
  - representation of relationships
    - among MM objects, between MM objects and structured data
    - used for querying and navigation
  - content-based search
    - descriptive, similarity-based
Multimedia Management and Retrieval: Challenges

- Data volume
  - storage and management of large objects
  - compression
  - efficient resource management and content processing

- Implicit semantics
  - feature extraction
  - similarity search

- Heterogeneity
  - media object types, formats
  - conversion

- Complex multimedia objects
  - combination of various MM object types
  - management of structural information

- Input/Output devices
  - dependencies regarding media object types
  - optimized data/result delivery
  - real-time aspects and requirements
Multimedia Data

- related to input/output devices:
## Multimedia Data (2)

- **comparison:**

<table>
<thead>
<tr>
<th>Medium</th>
<th>Elements</th>
<th>Order</th>
<th>Typical size</th>
<th>Time-dependent?</th>
<th>Sense</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>printable characters</td>
<td>sequence</td>
<td>10 KB (5 p.)</td>
<td>no</td>
<td>visual/acoustic</td>
</tr>
<tr>
<td>Graphics</td>
<td>vectors, polygons, meshes (3D), textures</td>
<td>set or sequence (z-order)</td>
<td>10-100 KB</td>
<td>no</td>
<td>visual</td>
</tr>
<tr>
<td>Raster Image</td>
<td>pixel</td>
<td>matrix</td>
<td>1 MB (1024X1024)</td>
<td>no</td>
<td>visual</td>
</tr>
<tr>
<td>Audio Recording</td>
<td>amplitude</td>
<td>sequence</td>
<td>600 MB (audio CD)</td>
<td>yes</td>
<td>acoustic</td>
</tr>
<tr>
<td>Video-Clip</td>
<td>Raster Image, Graphics</td>
<td>sequence (script)</td>
<td>2 GB (30 min.)</td>
<td>yes</td>
<td>visual</td>
</tr>
</tbody>
</table>
Multimedia Content - Terminology

- **Media Object**
  - a data object belonging to a *single* media, e.g., a single image or piece of text

- **Multimedia Object**
  - (or "Mixed-Mode Object")
  - aggregation (composition) of media objects having different types, z.B. Video (image + audio)

- **Multimedia Data**
  - generic term for both media objects and multimedia objects

- **Multimedia Document**
  - aggregates media objects and multimedia objects
  - defines a structure and/or layout (spatial and possibly temporal)
  - may include elements for navigation/browsing (e.g., links)
Media Object

- consists of structure *and* unstructured data
- raw data
  - unstructured (s. oben)
  - large sequence (set, ... ) of small elements
    *(bits, characters, pixel, lines, ... )* 
- registration data
  - mandatory
  - required for correct **interpretation** and **identification** of raw data
    - interpretation: what is the structure? what do the elements mean?
    - identification: to distinguish among otherwise equal objects
      *(e.g., creation/recording timestamp, ...)*
- descriptive data
  - optional
  - often redundant:
    representation of **structure** and/or **content** in a different media or form
  - structured an/or unstructured
Operations on Media Objects

- Create (insert, capture)
  - from device – in a program – from a file
- Output (show, present)
  - on device – to program – in a file
- Update (modify, edit)
- Compose
  - creation of multi-media objects
- Deliver (send)
- Archive
- Process (aggregate, derive)
  - filter, analyse, extract, create descriptive data
- Search (compare)
  - pattern matching on raw data
  - similarity
  - based on descriptive data
Example: Raster Image

- **Raw data**
  - pixel matrix (pixel = picture element, or "pel")

- **Registration data**
  - # of bits/pixel (pixel depth, usually 1, 8, or 24)
  - # of pixels/line (picture width)
  - # of lines (picture height)
  - linearization: by line or column
  - pixel semantics: grayscale, color definition, index into colormap
  - (optional) colormap with specific number of entries, length of entries (e.g., 24 bit)
  - (optional) definition of color space (RGB, IHS, ...)
  - and more

- **Descriptive data**
  - text, keywords, knowledge representation describing content
  - recognized lines, areas appearing in the image
  - resulting 2-D objects such as circles, ellipses, polygons, ...
Device Independence

- Multimedia data volume requires new storage media/devices
  - optical, video tape, ...
- Devices
  - operation of devices differs from hard disk storage
- Data independence required
  - application should be independent from the storage technology used
  - no program changes required when new devices are introduced
- Goal for content repository (MMDBMS)
  - hide all device-specific properties
Format Independence

- Storage formats for media objects
  - numerous formats exist today
  - various "standards"
    - examples (raster images): GIF, TIFF, JPEG, Sun Rasterfile, FBM, PBM, ...
  - format conversion possible (most of the time)
  - additional aspects: compression, container formats (e.g., AVI)

- Goal for content repository (MMDBMS)
  - hide internal storage format from end users and applications
  - provide conversion routines
  - support change of (internal) storage format (e.g., if new compression techniques become available) without impacting the applications
Structured vs. Unstructured Data

- **structured data**
  
  (NAME = "Miller"; BDATE = "1952-06-23", ....)
  
  - values of attributes/fields with a fixed name
  - maximal length (= finite value domain)
  - predetermined (known) semantics
  - search: (classical) data retrieval (i.e., query evaluation)
    
    ```sql
    SELECT * FROM EMPS WHERE NAME = "Miller";
    ```

- **unstructured data**
  
  "His name is Miller. He was born on June 23rd, 1952."
  
  - arbitrary length
  - mostly self-describing
  - unknown/weak semantics
  - search: information retrieval
    
    Find documents about employees with the name 'Miller'.
Information Retrieval

- **Index:** library catalog on a computer
  - availability of text documents in electronic format allows for full-text search as additional option
    - search for words appearing in the document
- **Builds on methods of library sciences**
  - classification (decimal, ACM, ...) – strictly hierarchical
  - indexing: assignment of keywords (descriptors), which describe the semantics of the document (may not even appear in the text)
    - descriptors may be weighted
- **Use of a thesaurus (keyword "dictionary")**
  - lists all keywords used for indexing
  - defines relationships among keywords
    - synonyms (DBS, database system, ...), preferred term
    - broader and narrower terms
- **Manual thesaurus construction and text indexing**
  - expensive, result influenced by human interpretation
- **Automatic indexing**
  - input: text document (complete, or abstract/title), thesaurus
  - output: keywords (index terms)
  - challenges
    - handling linguistic variations, use of synonyms
    - recognizing related words in keyword phrases
      - "DB-design"
      - "the design of high-quality database schemas is ..."
      - "... are DBs. Their design is ..."
    - ...

© Prof. Dr.-Ing. Stefan Deßloch
Information Retrieval (2)

- **Query**
  - consists of keywords, which may be weighted

- **Result documents**
  - boolean decision whether a document matches the query or not is often inadequate
  - better: similarity score as a measure for describing the relevance of the document wrt. the query

- **Retrieval model defines how relevance values are determined**
  - boolean, vector-based, probabilistic, ...

- **Presentation of search results**
  - ranking (sorted descending based on relevance)

- **Improving query results**
  - iterative query process
  - relevance feedback
Information Retrieval Process

query

preprocessing

query representation

preprocessing

media objects

preprocessing

object representation

comparison (similarity)

result objects

query refinement and relevance feedback
Data Retrieval vs. Information Retrieval

- **DBS**
  - search based on exact match, result always relevant/complete

- **IRS**
  - index terms only provide a partial, ambiguous characterization of a document
  - search based on similarity measures, result may contain irrelevant documents

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Data Retrieval</th>
<th>Information Retrieval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information</td>
<td>explicit (attribute values)</td>
<td>implicit (interpretation required)</td>
</tr>
<tr>
<td>Results</td>
<td>exact (equality)</td>
<td>approximate/imprecise (similarity)</td>
</tr>
<tr>
<td>Query process</td>
<td>single query</td>
<td>iterative refinement</td>
</tr>
<tr>
<td>Query formulation</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>tolerance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Result collection</td>
<td>set</td>
<td>list (ranked)</td>
</tr>
</tbody>
</table>
Searching for Media Objects

- Specification of search criteria
  - using properties (features, formatted)
  - verbally (unformatted)
  - query-by-example, using a comparison object (pattern)

- Media objects are usually linked to formatted data
  - application-specific (e.g., archiving number)
  - registration data
  - descriptive data (if formatted)

- Search mechanisms initially based on
  - searching formatted data (classic DB techniques)
  - pattern matching
    - using raw data (e.g., full text search), or
    - using descriptive (unformatted) data
Content-based Search

- Search based on media object content at a higher level of abstraction
  - objects, activities, situations, etc. represented or described by the media object
  - requires analysis and interpretation of raw data, content
  - hard to do, can only be partially automated

- Examples
  - search for:
    - entries in criminal database using descriptions by crime scene witnesses
    - symptoms of medical conditions in x-ray images
    - satellite images showing an airport
    - news pictures showing Angela Merkel with George W. Bush
    - text documents about digital libraries
    - radio programs about benefits of highway speed limits
  - almost impossible to do based on pattern matching!

- Application and generalization of information retrieval concepts and approaches!
Content-based Search (2)

- Preselection
  - use categories, classification scheme of a content repository (e.g., relations in an RDBMS)
  - helps reduce the search scope, but usually too coarse-grained
- Content analysis at runtime
  - for a specific query
  - methods and techniques for
    - text analysis and indexing
    - image processing/recognition
    - speech recognition
  - to be applied to a (large) set of media objects
  - too expensive!
- Browsing
  - fast browsing through preselected search scope
  - puts burden of search evaluation on the end user
    - only feasible for small sets of objects
Content-based Search (3)

- **Consequence:** Content analysis as a pre-processing step
  - creation of content description
    - automatically, if possible (e.g., automatic abstracting)
    - manually: e.g., author or librarian creates a summary, assigns keywords
- **Drawback:** search criteria at runtime are restricted to those aspects covered by the analysis, cannot be expanded dynamically
  - storage of content description together with the media objects, utilization at query processing time
Representation of Metadata / Content Description

- **Formatted Data**
  - e.g., as tuples in RDBMS tables
    - efficient search
    - may result in "undesirable" entity types (e.g., "storm", "night", ...) for detailed model of content domain
    - not powerful enough

- **Knowledge Representation**
  - logic, sem. networks, frames, scripts, ontologies, ...
  - search techniques exist
  - powerful, but required expert user (knowledge engineer) to create

- **Keywords**
  - well-known (libraries, IR)
  - easy to create (also automatically)
  - not powerful enough: relationships, dependencies, causalities (e.g., for complex description of image content) are hard to represent

- **Free text**
  - easy to create
  - full text search? dependency on formulations used

- **Captions**
  - restricts natural language syntax
  - still easy to create
  - may internally be translated to a formal representation (e.g., knowledge representation)
Alternative Classification of Metadata

- content-describing (interpreting)
  
  *expresses semantics of media object at a high level of abstraction*
  
  - context-describing (e.g., index vocabulary, ontology, thesaurus)
    *describe context of a collection of objects*
  
  - context-related (e.g., identification, spatial/temporal coordinates)
    *describe object in relation to a collection of objects*
  
  - object-describing, non-textual (e.g., objects, people, activities, title)
  
  - object-describing, textual (e.g., annotation, subtitles, script, screenplay)

- content-related (non-interpreting)
  
  *properties at a low semantic level, can be extracted automatically*

  - feature (e.g., image color distribution, texture)
  
  - segment specification (e.g., start/end of video scene, shape of image segment)

- content-independent
  
  *important for management, correct interpretation*

  - presentation-related (e.g., QoS, resolution, layout)
  
  - recording-related (e.g., copyright owner, recording device)
  
  - storage-related (e.g., media type, format, storage location)
Deriving Metadata at Various Abstraction Levels

- **Automatic steps**
  - decomposition into elementary media objects
  - normalization for internal processing
  - segmentation based on semantics (e.g., image segments, video scenes)
  - type-specific feature recognition
  - feature processing for scaling, normalization of features
Multimedia Database Techniques

- Introduction of (elementary) data types
  - TEXT, GRAPHIC, IMAGE, SOUND, VIDEO, ....
  - with functionality (→ abstract data types, userdefined types)
- Extension of existing data models
  - relational model (as domains)
  - object-relational, object-oriented model (as UDTs, classes)
- Usage of existing modeling constructs
  - relations/classes
  - attributes-instance variables
  - primary keys, object identifiers
  - methods, inheritance, ...
- Extensions of query languages
  - relational algebra, SQL
Interfaces and Functionality

- Compared to traditional DBMS, programming interface differs from interactive interface
  - view image vs. analyze image in an application
  - in traditional DBMS, the query language is the common interface
- Example: RDBMS, extended with new data types IMAGE, TEXT, GRAPHICS, SOUND etc.

  ```sql
  create table Person
  (Name char(30),
   ....,
   picture Image,
   fingerprint Image)
  ```

- Access functions for the data type:
  - dependent on the type of interface
Programming Interface

(using extended SQL)

- Read a fingerprint to analyse it in the application:
  
  ```sql
  select fingerprint.height(), fingerprint.width()
  into :height, :width
  from Person
  where Name = "Miller";
  ```

  (allocation of memory for the image)

  ```sql
  select fingerprint.pixelmatrix()
  into :pixel
  from Person
  where Name = "Miller";
  ```

  (work on pixel matrix)
Programming Interface (2)

- **direct display:**
  ```sql
  exec sql
  select fingerprint.display(:window) into :error
  from Person
  where .... ;
  if ( error!= 0 )
      .... ;
  ```

- **write to file:**
  ```sql
  exec sql
  select fingerprint.toFile(:file) into :error
  from Person
  where .... ;
  if ( error != 0 )
      .... ;
  ```

- "side effects" in SQL statements
Interactive Interface

- Same example:
  ```sql
  select fingerprint
  from Person
  where Name = "Miller";
  ```
- Result is a table displayed on the screen
  - special character or icon to indicate value of type IMAGE
  - mouse-click or special command to display image on separate screen
  - (other approaches are possible)
Summary

- Different types of multimedia data
  - media object, MM object, MM document
  - raw data vs. metadata
- Classification of metadata
  - based on representation
  - based on content aspects
    - registration data vs. content description
    - content-independent, content-related, content-describing
- Requirements for storage and retrieval of multimedia objects
  - handling structured and unstructured data
  - device- and format-independent access to MM objects
  - representation of relationships
    - among MM objects, between MM objects and structured data
    - used for querying and navigation
  - content-based search
    - descriptive, similarity-based
- Overview of information retrieval and content-based search approaches