3. Multimedia Database

Digital Asset Management

任课老师：张宏鑫
2009-10-10
Outline

1. MM content organization
2. MM database system architecture
3. MM system service model
4. Multimedia Data Storage
5. Multimedia application
3.1. Multimedia Content Organization
Metadata Model Organization

- Content-dependent Metadata
- Content-descriptive Metadata
- Content-independent Metadata
Metadata Model

• Metadata => data about data
  – forms an essential part of any database
    • providing descriptive data about each stored object, and
    • is the key to organizing and managing data objects
  – critical for describing essential aspects of content:
    • main topics, author, language, publication, etc.
    • events, scenes, objects, times, places, etc.
    • rights, packaging, access control, content adaptation, …
Metadata Model

- **Purposes** of metadata:
  - **Administrative**
    - managing and administrating the data collection process
  - **Descriptive**
    - describing and identifying for retrieval purpose, creating indices
  - **Preservation**
    - managing data refreshing and migration
  - **Technical**
    - formats, compression, scaling, encryption, authentication and security
  - **Usage**
    - users, their level and type of use, user tracking, versioning (e.g., a high resolution version and corresponding thumbnail)
Metadata Model

• Conformity with **open metadata standard** will be a vital:
  – Faster design and implementation
  – Interoperability with broad field of competitive standards-based tools and systems
  – Leveraging of rich set of standards-based technologies for critical functions
    • e.g., content extraction, advanced search, and personalization
The “role” of metadata in query processing:

- **Conceptual data view**
- **Query metadata**
- **Meta correlation**
- **Media-independent metadata**
- **Image metadata**
- **Text metadata**
- **Media dependent**
- **Media preprocessor**
- **Image**
- **Text**
# Classifying Metadata

Classification of metadata can be:

1. Specific to the **media involved**
2. Specific to the **processing**
3. **Content** specific metadata

**Image object**
- Image capture
- Image storage
- Caption
- Genre
- Period
- Subjects
- Photographer
- IP rights
- Texture

**Text object**
- title
- author
- abstract
- Full text indices

**Video**
- time based
- play rate
- camera motion
- camera lighting

*Sample Metadata*
Metadata can be classified as:

- **Content dependent** (e.g., face features; used in CBR)
- **Content-descriptive** (used in TBR)
  1. Domain-independent metadata: independent of the application or subject topic
  2. Domain-dependent metadata: specific to the application area
- **Content-independent** (e.g., photographer’s name; used in ABR)
## Metadata Classification

<table>
<thead>
<tr>
<th>Media</th>
<th>Content independent</th>
<th>Content descriptive</th>
<th>Content dependent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>status, location, date of update components</td>
<td>keywords, formats, categories, language</td>
<td>subtopic boundary word image spotting</td>
</tr>
<tr>
<td>speech</td>
<td>start, end time location confidence of word recognition</td>
<td>speakers</td>
<td>speech recognition speaker recognition prosodic cues change of meaning</td>
</tr>
<tr>
<td>Image</td>
<td>creator title date</td>
<td>keywords, formats</td>
<td>feature selection image features (e.g., histogram, segmentation)</td>
</tr>
<tr>
<td>Video</td>
<td>product title data distributor</td>
<td>camera shot action distance close-up</td>
<td>shot boundary frame features (e.g., histogram, motion lighting level, height)</td>
</tr>
</tbody>
</table>
Domain-dependent Metadata

• Standards for domain-specific metadata
  – Digital geospatial metadata
    • US Geographic Data Committee
    • http://www.fgdc.gov/metadata/metahome.html
  – Environmental data (UDK)
    • the European Environmental Catalog
  – Product data exchange (PDES)
    • an ANSI standard for the exchange of product model data
  – Rich Site Summary (RSS)
    • a lightweight XML vocabulary for describing websites, ideal for news syndication
  – Medical information (HL7)
    • provides specification for hospital records and medical information management
    • accredited by ANSI
Domain-independent Metadata Standards

  - Intended to provide:
    - conceptual framework,
    - logical explanations of the processes for an organization to describe data semantics consistently, and
    - the exchange of data and metadata across organizational units

- The standard divides data elements into 3 parts:
  - **Object class** – the thing the data describes (e.g., person, airplane)
  - **Property** – a peculiarity that describes/distinguishes objects
  - **Representation** – the allowed values and other information
**Domain-independent Metadata Standards**

- **ISO/IEC 11179**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>the label assigned to the <strong>data element (d.e.)</strong></td>
</tr>
<tr>
<td>Id</td>
<td>the unique identifier assigned to the d.e.</td>
</tr>
<tr>
<td>Version</td>
<td>the version of the d.e. (e.g., 1.1 for Dublin Core)</td>
</tr>
<tr>
<td>Registration Authority</td>
<td>the entity authorized to register the d.e.</td>
</tr>
<tr>
<td>Language</td>
<td>the language in which the d.e. is specified (e.g., English)</td>
</tr>
<tr>
<td>Definition</td>
<td>a statement representing the d.e. concept and nature</td>
</tr>
<tr>
<td>Obligation</td>
<td>indicates if the d.e. is required to be not null</td>
</tr>
<tr>
<td>Data type</td>
<td>indicates the data type that can be represented in d.e.</td>
</tr>
<tr>
<td>Maximum Occurrence</td>
<td>indicates any limit to the repeatability of the d.e.</td>
</tr>
<tr>
<td>Comment</td>
<td>a remark concerning the application of the d.e.</td>
</tr>
</tbody>
</table>
Domain-independent Metadata Standards

- The Dublin Core Metadata set
  
  http://purl.org/metadata/dublin_core

  - Originally for resource description records of online libraries over Internet
  - version 1.1
    - broaden to other media with a link to the ISO/IEC 11179 standard
      - Each Dublin Core element is defined using a set of 10 attributes from the ISO/IEC 11179
      - Six of them are common to all the Dublin Core element (3-5, 7-9)
    - 15 metadata elements (the Dublin Core) has been proposed
      - which are suggested to be the minimum number of metadata elements to support retrieval of a document-like object (DLO) in a networked environment
## The Dublin Core Metadata set

<table>
<thead>
<tr>
<th>ID</th>
<th>Core element</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Subject</td>
<td>topic addressed by the work</td>
</tr>
<tr>
<td>2</td>
<td>Title</td>
<td>the name of the object</td>
</tr>
<tr>
<td>3</td>
<td>Creator</td>
<td>entity responsible for the intellectual content</td>
</tr>
<tr>
<td>4</td>
<td>Publisher</td>
<td>the agency making the object available</td>
</tr>
<tr>
<td>5</td>
<td>Description</td>
<td>an account of the content of the resource</td>
</tr>
<tr>
<td>6</td>
<td>Contributor</td>
<td>an entity making contributions to the resource content</td>
</tr>
<tr>
<td>7</td>
<td>Date</td>
<td>associated with an event in the life cycle of the resource</td>
</tr>
<tr>
<td>8</td>
<td>Resource type</td>
<td>the nature/genre of the resource content</td>
</tr>
<tr>
<td>9</td>
<td>Format</td>
<td>physical/digital manifestation of the resource; format of the file (e.g., postscript)</td>
</tr>
<tr>
<td>10</td>
<td>Id</td>
<td>unique identifier</td>
</tr>
<tr>
<td>11</td>
<td>Relation</td>
<td>a reference to a related resource</td>
</tr>
<tr>
<td>12</td>
<td>Source</td>
<td>a ref. to a resource from which the current resource is derived</td>
</tr>
<tr>
<td>13</td>
<td>Language</td>
<td>language of the intellectual content</td>
</tr>
<tr>
<td>14</td>
<td>Coverage</td>
<td>extent/scope of the resource content; typically include location, period</td>
</tr>
<tr>
<td>15</td>
<td>Rights</td>
<td>Information about rights held in and over the resource</td>
</tr>
</tbody>
</table>
Domain-independent Metadata Standards

• Resource Description Framework (RDF)
  – Being developed by the W3C as a foundation for processing metadata
  – Allows multiple metadata schemes to be read by human and parsed by machines
  – Specific objectives include:
    • **Resource discovery** – to provide better search engine capabilities
    • **Cataloging** – for describing the content and relationships available through intelligent software agents
    • **Content rating** – describing collection of pages that represent a single logical “document”
    • **IP rights** – describing the intellectual property of web pages
    • **Privacy preferences and policies** – for users and website
    • **Digital signatures** – to create a “web of trust” for e-commerce, collaboration, and other applications
Resource Description Framework (RDF)

- The formal model of the RDF framework:
  - There is a set called Resources.
  - There is a set called Literals.
  - There is a subset of Resources called Properties.
  - There is a set called Statements, each element of which is a triple of form <pred, sub, obj>, where
    - pred is a property,
    - sub is a resource (member of Resources)
    - obj is either a resource or a literal
- The preferred language for writing RDF schemas is XML
**XML**

- Defined by the WWW Consortium (W3C)
- Originally intended as a document markup language not a database language
  - Documents have tags giving extra information about sections of the document
    - `<title> XML </title> <slide> Introduction …</slide>`
    - `<?xml … ?>` (document declaration)
    - `<!-- definition of elements -->` (comments)
  - Derived from SGML (Standard Generalized Markup Language), but simpler to use than SGML
  - **Extensible**, unlike HTML
    - Users can add new tags, and *separately* specify how the tag should be handled for display
XML
XML

- The ability to specify new tags, and to create nested tag structures made XML a great way to exchange data, not just documents.
  - Much of the use of XML has been in data exchange applications, not as a replacement for HTML
- Tags make data (relatively) self-documenting
XML

- The ability to specify new tags, and to create nested tag structures made XML a great way to exchange data, not just documents.
  - Much of the use of XML has been in data exchange applications, not as a replacement for HTML
- Tags make data (relatively) self-documenting

```xml
<bank>
  <account>
    <account-number> A-101 </account-number>
    <branch-name> Downtown </branch-name>
    <balance> 500 </balance>
  </account>
  <depositor>
    <account-number> A-101 </account-number>
    <customer-name> Johnson </customer-name>
  </depositor>
</bank>
```
Structure of XML

- **Tag**: label for a section of data

- **Element**: section of data beginning with `<tagname>` and ending with matching `</tagname>`

- Elements must be properly *nested*
  
  - Proper nesting
    
    `<account> ... <balance> .... </balance> </account>`
  
  - Improper nesting
    
    `<account> ... <balance> .... </account> </balance>`

  - Formally: every start tag must have a unique matching end tag, that is in the context of the same parent element.

- Every document must have a single top-level element
Structure of XML

– Mixture of text with sub-elements is legal in XML
  • Example:
    <account>
      This account is seldom used any more.
      <account-number> A-102</account-number>
      <branch-name> Perryridge</branch-name>
      <balance>400 </balance>
    </account>
  • Useful for document markup, but discouraged for data representation
Attributes

- Elements can have **attributes**
  
  ```xml
  <account acct-type = "checking" >
    <account-number> A-102 </account-number>
    <branch-name> Perryridge </branch-name>
    <balance> 400 </balance>
  </account>
  ```

- Attributes are specified by *name*=*value* pairs inside the starting tag of an element

- An element may have several attributes, but each attribute name can only occur once
  
  ```xml
  <account acct-type = "checking" monthly-fee="5"> 
  ```
Attributes vs. Subelements

- Distinction between subelement and attribute
  - Attributes: are part of markup
  - Subelements: contents are part of the basic document

- Some information can be represented in two ways
  - <account account-number = "A-101"> … </account>
  - <account>
    <account-number>A-101</account-number> … </account>

- Suggestion: use attributes for identifiers of elements, and use subelements for contents
More on XML Syntax

– Elements without subelements or text content can be abbreviated by ending the start tag with a `/>` and deleting the end tag
  • `<account number="A-101" branch="Perryridge" balance="200 />`

– To store string data that may contain tags, without the tags being interpreted as subelements, use CDATA as below
  • `<![CDATA[<account> … </account>]]>

Here, `<account>` and `</account>` are treated as just strings
Namespaces

- XML data has to be exchanged between organizations
- Same tag name may have different meaning in different organizations, causing confusion on exchanged documents
- Specifying a unique string as an element name avoids confusion
- Avoid using long unique names all over document by using XML Namespaces

```xml
<bank xmlns:FB='http://www.FirstBank.com'>
  ...
  <FB:branch>
    <FB:branchname>Downtown</FB:branchname>
    <FB:branchcity>Brooklyn</FB:branchcity>
  </FB:branch>
  ...
</bank>
```
XML Document Schema
XML Document Schema

- Database schemas constrain
  - what information can be stored, and
  - the data types of stored values
- not necessary in a XML document
- very important for XML data exchange
  - Otherwise, a site cannot automatically interpret data received from another site
- Two mechanisms for specifying XML schema
  - Document Type Definition (DTD)
  - XML Schema
The type of an XML document can be specified using a DTD.

DTD constraints structure of XML data:
- What elements can occur
- What attributes can/must an element have
- What subelements can/must occur inside each element, and how many times.

DTD does not constrain data types:
- All values represented as strings in XML.

DTD syntax:
- `<!ELEMENT element (subelements-specification) >`
- `<!ATTLIST element (attributes) >`
Element Specification in DTD

- Subelements can be specified as
  - names of elements, or
  - #PCDATA (parsed character data), i.e., character strings
  - EMPTY (no subelements) or ANY (anything can be a subelement)
- Example
  ```xml
  <! ELEMENT depositor (customer-name account-number)>
  <! ELEMENT customer-name (#PCDATA)>
  <! ELEMENT account-number (#PCDATA)>
  ```
- Subelement specification may have regular expressions
  ```xml
  <!ELEMENT bank ( ( account | customer | depositor)+)>
  ```
  - Notation:
    - “|” - alternatives
    - “+” - 1 or more occurrences
    - “*” - 0 or more occurrences
IDs and IDREFs

- An element can have at most one attribute of type ID.
- The **ID attribute value** of each element in an XML document must be **distinct**
  - Thus the ID attribute value is an object identifier

- An attribute of type IDREF must contain the ID value of an element in the same document.
- An attribute of type IDREFS contains a set of (0 or more) ID values.
- Each ID value must contain the ID value of an element in the same document.
Bank DTD with ID and IDREF attribute types

<!DOCTYPE bank-2[  
<!ELEMENT account (branch, balance)>  
<!ATTLIST account  
  account-number ID          # REQUIRED  
  owners                IDREFS # REQUIRED>  
<!ELEMENT customer(customer-name, customer-street,  
customer-city)>  
<!ATTLIST customer  
  customer-id        ID          # REQUIRED  
  accounts            IDREFS # REQUIRED>  
  … declarations for branch, balance, customer-name,  
customer-street and customer-city  
]>
XML data with ID and IDREF attributes

<bank-2>
  <account account-number="A-401" owners="C100 C102">  
    <branch-name> Downtown </branch-name>
    <balance> 500 </balance>
  </account>
  <customer customer-id="C100" accounts="A-401">  
    <customer-name> Joe </customer-name>
    <customer-street> Monroe </customer-street>
    <customer-city> Madison </customer-city>
  </customer>
  <customer customer-id="C102" accounts="A-401 A-402">  
    <customer-name> Mary </customer-name>
    <customer-street> Erin </customer-street>
    <customer-city> Newark </customer-city>
  </customer>
</bank-2>
Limitations of DTDs

- No typing of text elements and attributes
  • All values are strings, no integers, reals, etc.

- Difficult to specify unordered sets of subelements
  • Order is usually irrelevant in databases
  • \((A \mid B)^*\) allows specification of an unordered set, but
    - Cannot ensure that each of A and B occurs only once

- IDs and IDREFs are untyped
  • The *owners* attribute of an account may contain a reference to another account, which is meaningless
    - *owners* attribute should ideally be constrained to refer to customer elements
Domain-independent Metadata Standards

- MPEG series
  - Moving Picture Experts Group (MPEG) since 1998
  - responsible for developing standards of the coded representation of moving pictures and associated audio
Domain-independent Metadata Standards

- **MPEG series**
  - Moving Picture Experts Group (MPEG) since 1998
  - responsible for developing standards of the coded representation of moving pictures and associated audio
Domain-independent Metadata Standards

- MPEG series
  - Moving Picture Experts Group (MPEG) since 1998
  - responsible for developing standards of the coded representation of moving pictures and associated audio
# Domain-independent Metadata Standards

## Applications

<table>
<thead>
<tr>
<th>Domain-independent Metadata Standards</th>
<th>MPEG-1,-2,-4</th>
<th>MPEG-4,-7</th>
<th>MPEG-7</th>
<th>MPEG-21</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video storage</td>
<td>Video storage</td>
<td>Multimedia filtering</td>
<td>Semantic-based retrieval and filtering</td>
<td>Multimedia framework</td>
</tr>
<tr>
<td>Broadband, streaming video delivery</td>
<td>CBR</td>
<td>Content adaptation</td>
<td>Intelligent media services (iTV)</td>
<td>e-Commerce</td>
</tr>
</tbody>
</table>

## Problems and Innovations

<table>
<thead>
<tr>
<th>Compression coding communications</th>
<th>Similarity search object- &amp; feature-based coding</th>
<th>Modeling &amp; classifying, personalization, summarization</th>
<th>Media mining, decision support</th>
</tr>
</thead>
</table>

**Notes:**
- **MPEG-1,-2**
- **MPEG-4**
- **MPEG-7**
- **MPEG-21**
MPEG-7

- **Multimedia Content Description Interface**
  - Representation of information about the content
    - still pictures, graphics, 3D models, audio, speech, video & their combination
  - Goal:
    - to **support efficient search** for multimedia content using **standardized descriptions**
    - desirable to use textual information for the descriptions
Domain-independent Metadata Standards

Scope of MPEG-7

Feature Extraction → MPEG-7 Standard Description → Search Engine
## MPEG-7

### Set of description tools

#### Media

Description of the storage media: typical features include the storage format, the encoding of the multimedia content, the identification of the media. Note that several instances of storage media for the same multimedia content can be described.

#### Creation & Production

Meta information describing the creation and production of the content: typical features include title, creator, classification, purpose of the creation, etc. This information is most of the time author generated since it cannot be extracted from the content.

#### Usage

Meta information related to the usage of the content: typical features involve rights holders, access right, publication, and financial information. This information may very likely be subject to change during the lifetime of the multimedia content.

#### Structural aspects

Description of the multimedia content from the viewpoint of its structure: the description is structured around segments that represent physical spatial, temporal or spatial-temporal components of the multimedia content. Each segment may be described by signal-based features (color, texture, shape, motion, and audio features) and some elementary semantic information.

#### Semantic aspects

Description of the multimedia content from the viewpoint of its semantic and conceptual notions. It relies on the notions of objects, events, abstract notions and their relationship.
MPEG-7

Content organization

Collections

Models

User interaction

Navigation & Access
- User Preferences
- Summaries
- Views
- Variations

Content management

Media

Creation & Production

Usage

Content description

Structural aspects

Semantic aspects

Basic elements

Schema Tools

Basic datatypes

Links & media localization

Basic Tools
MPEG-7 Standard Elements

- **Descriptors** (Ds)
  - describe features, attributes, or groups of attributes of MM content

- **Description Schemes** (DSs)
  - a DS specifies the structure and semantics of the components (which may be other DSs, Ds, or datatypes)

- **Datatypes**

- **Classification Schemes** (CS):
  - lists of defined terms and meanings

- **System Tools**

- **Extensibility**
  - e.g., new DS’s and D’s; registration authority for CS
1. MM content organization
2. MM database system architecture
3. MM system service model
4. Multimedia Data Storage
5. Multimedia application
3.2 Multimedia Database System Architecture
Multimedia Architecture
Multimedia Architecture
Multimedia Architecture

Multimedia Applications
- Multimedia Documents
- MM User Interfaces
- Multimedia Tools

Database Systems, Operating Systems, Communication Systems

Computer Technology

Comprehension
- Non-Temporal Media
- Temporal Media

Applications Domain

Systems Domain

Media Domain
Multimedia Database System

Multimedia Data Management

Multimedia Database

Data Storage
Multimedia Database System

• **Multimedia database** v.s. **text database**
  – **Temporal data**: Requires temporal modeling
  – **Huge amount of data**: Compression helps get around this.
  – Data is **not easily indicative** of the information
  – Requires a lot of **pre-processing** in order to store data efficiently:
    • PCA, feature extraction and segmentation
  – **Novel Query mechanisms**
  – **Hypermedia**: The ability to interactively move around in the data.
How to Build Multimedia Database Systems?

How to build text database?  

Yahoo, Google
How to Build Multimedia Database Systems?

How to build text database? Yahoo, Google

Text document: Natural language processing

Actions

Transmission: Text database

Tree-based indexing
How to Build Multimedia Database Systems?

How to build text database?  

Yahoo, Google

Text document  
Natural language processing

Actions

Transmission  
Text database  
Tree-based indexing

Multimedia data  
Multimedia analysis

Actions

Transmission  
Multimedia database  
Multimedia Indexing
Scope

Usage
- Learning & Teaching
- Design
- User Interfaces

Services
- Content Processing
- Documents
- Security
- Synchronization
- Group Communications

Systems
- Databases
- Operating Systems
- Communications
- Media-Server
- Quality of Service
- Networks
- Opt. Memories

Basics
- Computer Architectures
- Image & Graphics
- Animation
- Video
- Audio

Applications
Scope

Usage
- Learning & Teaching
- Design
- User Interfaces

Services
- Content Processing
- Documents
- Security
- Synchronization
- Group Communications

Systems
- Databases
- Programming
- Media-Server
- Operating Systems
- Communications
- Opt. Memories
- Quality of Service
- Networks

Basics
- Computer Architectures
- Image & Graphics
- Animation
- Video
- Audio
- Compression
A Reference Architecture for MMDB System

Considerations:

- Real time aspects/constraints impose strong demands on the systems
  - Simultaneous presentation of multimedia objects may cause performance problems.

- Data Sharing
  - Due to the possibly very large multimedia data, traditional replicated data technique may not be applicable, hence data sharing is essential

- Multiple Client/ Multiple Server Architecture
A Reference Architecture for MMDB System

Considerations:
- Real time aspects/constraints
- Data Sharing
- Multiple Client/ Multiple Server Architecture
  - Many multimedia applications work with data that are stored on remote sites (e.g., VOD, tele-learning), which suggests for client / server architecture.
  - A client consists of three layers…
    - User Interaction – takes care of input and output of multimedia data
    - Server Access – allows searching of servers by the client
    - Operating System – not a real part of the MMDBS
  - A server consists of four layers:
    - DBMS Interface
    - Query Processor
    - File Manager
    - Operating System
A Generic Architecture of MMDBMS

Media objects

Users
A Generic Architecture of MMDBMS

- Media objects
  - Compression
  - MM DBMS
  - Users

Content flows from Media objects through Compression to MM DBMS, then to Users.
A Generic Architecture of MMDBMS
A Generic Architecture of MMDBMS

Media objects

Feature extraction

Indexing

Compression

MM DBMS

Query feature construction

Users

query

metadata

content
A Generic Architecture of MMDBMS

- Media objects
  - Feature extraction
  - Indexing
  - Compression
  - metadata
  - content

- MM DBMS
  - Query feature construction
  - Search Engine
  - query

- Users
A Generic Architecture of MMDBMS

- Media objects
- Feature extraction
- Indexing
- Compression
- Metadata
- Query feature construction
- Search Engine
- Query
- Results
- Users

Content
A Generic Architecture of MMDBMS

Media objects

- Feature extraction
- Indexing
- Compression

content

metadata

MMDBMS

Query feature construction

Search Engine

Feedback Query construction

Users

query

results

feedback
A Generic Architecture of MMDBMS

- Media objects
  - Feature extraction
  - Indexing
  - Compression

- MM DBMS
  - Metadata
  - Content

- Query feature construction
- Search Engine
- Feedback Query construction

- Users
  - Query
  - Results
  - Feedback
MMDB Reference Architecture: “Simplified View”

Multimedia network

CLIENT

SERVER

DBMS Interface
Query Processor
File Manager
Operating System

User Interaction
Server Access
Operating System
Detailed View of MMDB Architecture

- MM Client
- Traditional LAN / MAN
- MM Client
- Conventional data
- MM Capable LAN / MAN
- DBMS Interface, API
  - Query Processor
  - Script Generator
  - Retrieval Engine
  - Transaction Manager
  - Object Manager
  - Ext. Media Server
  - Continuous Obj. Mgr.
- MMDBMS Server
MMDBMS Development

Major steps in developing MMDBMS

1. **Media acquisition:**
   - collect media data from various sources, such as WWW, CD, TV, etc.

2. **Media processing:**
   - extract media representations and their features, including noise filtering, rendering, etc.

3. **Media storage:**
   - store the data and their features in the system based on application requirement.

4. **Media organization:**
   - organize the features for retrieval. i.e., indexing the features with effective structures.

5. **Media query processing:**
   - Accommodated with indexing structure, efficient search algorithm with similarity function should be designed.