Digital Libraries

Repositories

W. Arms, Cornell
Repositories

Definitions

A repository is any computer system whose primary function is to store digital material for use in a library.

An archive is a repository that is organized to emphasize the long-term preservation of information.
Information hiding

Internal organization should be hidden from client computers.
Repository layers and interfaces

Persistent Store

Object Management Layer

Interface

Clients

External Interface

Store API

Shell API
Requirements 2

Object models

• Support for a flexible range of object models.

• Few restrictions on data, metadata, external links, and internal relationships.

• New categories of information do not require fundamental changes to other aspects of the digital library.
Multiple disseminations

Client can access a choice of forms of digital object:

- Format -- PDF or HTML
- Performance -- 8 bit/pixel or 24 bit/pixel color
- Content -- thumbnail, medium-resolution, high-resolution

Repository might store alternative disseminations or derive them when requested.
Dissemination is produced by executing code at time client makes request

- Real-time sensor, e.g., traffic camera, satellite picture
- User characteristics, e.g., location, user profile
- Dissemination is intrinsically dynamic, e.g.,
  - simulation
  - virtual reality
  - computer program
  - Java applet
Metadata can be linked to digital object:

- external catalog or index
- embedded in the digital object
- generated at run time

Granularity of metadata

- collection of digital objects
- digital object
- element of digital object
Requirements 3

Open protocols and formats
• Clients use well-defined protocols, data types, and formats.
• Architecture must allow incremental changes of protocols.

Access management
• Allow a broad set of policies
• All levels of granularity
• Prepared for future developments.

Reliability and performance
• Very large volumes of data
• Absolutely reliable in retention of data
• Good performance
Repository systems

Core
Repository
Repository systems

Load Services  Core Repository  Presentation Services
Common repository systems

**Web server**
- File-based object model plus hyperlinks
- Good tools for access
- Weak on long-term preservation

**Relational database**
- Table-based object model -- schema and data dictionary
- Good tools for data management
- Used for long-term preservation in data processing
Dumb and smart objects

**Smart repositories objects**
- behaviors provided by the repository
- e.g., relational database

**Smart clients**
- behaviors provided by the client
- e.g., web server

**Smart objects**
- repository is very simple
- digital objects provide their own behaviors
- compare with object-oriented programming (data + code)
Example: CNRI repository

Dumb repository for access to digital objects

- All information stored as typed data in digital objects.
- A single digital object has both data and metadata.
- Identification of digital objects is by location independent, persistent URNs.
- Access controls built into methods for accessing digital object.
Repository Access Protocol (RAP)

RAP is a simple protocol with two main groups of commands:

- Deposit digital object
- Verify digital object
- Delete digital object
- Edit digital object
- Access digital object
- Access metadata
Repository layers and interfaces

- Persistent Store
- Object Management Layer
- RAP Interface

- Store API
- Shell API
- RAP Command
Client and repository architectures

- End Client
  - Digital Object Processing
  - Object Management
  - RAP Interface
  
  RAP Requests

- Store
  - Object Persistence
  - Object Management
  - RAP Interface
  
  RAP Replies

- ORB
Components

**Hardware**
- Repository: Sun Sparc with Solaris or IBM RS/6000 with AIX.

**Software**
- Communications: CORBA/IIOP distributed object system.
- Repository shell and object management layer: CORBA and Python.
- Persistent store: Unix file system, Oracle, Shore.
- Client: CGI scripts, Java applets.