DLF Forum

Repository Stuff

Bernie Hurley
UC Berkeley Library
(bernie@library.berkeley.edu)
Agenda

• Today’s Agenda: Repositories
  – MoA II Background
  – The MoA II Digital Library Service Model
  – Berkeley’s Current Thinking on Repositories
  – MoA II Architecture: Theory and Model Stuff

• Tomorrow’s Agenda: Page Turning
  – Encoding Digital Objects
  – Client/Repository Interactions
  – Page Turning with Standard Objects and Methods
    Calls within the MoA II Architecture
The Making of America II Project

MoA II’s Goal is to Create Community Standards for Digital Library Objects

• MoA II Partners
  – Participants: UC Berkeley, Cornell, NYPL, Penn State and Stanford
  – Funding: DLF and NEH

• Digital Library Objects Encapsulate
  – Content (e.g., digitized page, text transcription)
  – Metadata: Descriptive, Structural, Administrative & Technical
  – Methods (e.g. Repository access, Page turning)
The Making of America II Project

• Standardized Objects Need a Standard Encoding Scheme
  – The MoA II XML DTD

• Why Do We Need DL Object Standards?
  – Interoperability
  – Scalability
  – Digital Preservation

• The MoA II Testbed
The MoA II Service Model

The goal of the MoA II Model is to develop suites of tools for specific audiences that integrate the discovery, display, navigation and manipulation of standardized objects across distributed repositories.
Service Model: Assumptions

1) A National DL will be made up of many different classes of objects
   – library, archival, museum, GIS, numeric datasets

2) These objects will populate distributed repositories

3) Scholars and students will require coherent and integrated access to these objects (i.e., distributed repositories are transparent)
Assumptions (Cont.)

4) Therefore, we will need to develop *tools* that can transparently discover, display, navigate and manipulate DL objects across distributed repositories.

5) **These tools will require objects be standardized**

Note: Our interest in repositories is focused on **Client/Repository & Repository/Repository Interactions**
Repository - Defined

• Definitely Not the Penultimate Definition
  (just want to avoid confusion)

Container (database)

Services
Current Thinking on Repository Services

Conceptual View

Creation/Maint. Repository

MoA2 Objects

Archival Repository (permanent)

MoA2 Objects

Access Repository (transient)
MoA II Access Repository Architecture

- Class Tools
- Objects
- Repository

Client/Server, Object Oriented

Workstation

Network

Union Index

Repository

Repository

Repository
Implementation Technologies

• Minimize the use of HTML/CGI scripting

• Investigate
  – *Object Oriented Design*
  – *The Java Enterprise Model*
  – *Distributed Object Architectures*
    • *RMI, CORBA, DCOM*
  – *The Object Web*
Object Oriented Design

• **Definition of a Digital Object**
  – Digital objects have content
  – Digital Objects have metadata
  – Digital objects have behaviors (methods)
  – Encapsulate content, metadata and methods together in a digital object
The Java Enterprise Model

**Workstation**

- Web Browser
- Applet

**Internet**

**Web Server**

- Java Applet
Distributed Object Architecture (RMI)

• Remote Method Invocation
  – Enables communications between the client and the repository by allowing methods called in the client to execute remotely on the server

• RMI Registry (basic naming service)
  – Client and Repository communications are bootstrapped using the RMI Registry
    • MoA2 RMI compliant repositories register with the RMI Registry when they become active
    • Clients can lookup active repositories in the registry and are returned a RMI reference to the repository
    • With the reference, the client can calls defined by the MOA2Repository Interface
Distributed Object Architecture

Workstation

Application

RMI

Internet

Repository
Incl.
Remote Methods

Repository
Incl.
Remote Methods
The Object
WEB
Benefits of the Object Web

• Load Applets on Demand (version control)

• Distributed Object Middleware...
  – Designed to support distributed repositories
  – Determine where behaviors execute - client or server side

• Doesn’t Require an Extremely Powerful Client

• Scales
Full Object Web Implementation

1) Browser: Discovery Applet
   1: Object ID
   2: Object ID
   3: Object ID
   RMI

2) Browser: Display Applet
   Display Applet?
   Repository & Object ID
   RMI

Repository

Web Server

Create HTML page w/display applet

Repository Method Layer
(server side side behaviors)

Union Catalog

DBMS
(Content)

CORBA

HTTP
Z39.50/CGI/Object Web Implementation

Browser: Search Results
1: CGI URL’s?Object’s URN
2: CGI URL’s?Object’s URN
3: CGI URL’s?Object’s URN

Repository

HTTP

Z39.50

Union Catalog

Web
Zserver

CGI Script:
Create HTML page w/display applet

Repository Method Layer

DBMS (Content)

Display Applet?
Repository & Object ID

RMI

Browser: Display Applet

Hurley -- 7/17/99