MultiMimsy database extractions and OAI repositories at the Museum of London

Mia Ridge

Museum Systems Team

Museum of London

mridge@museumoflondon.org.uk

Scope

- Extractions from the MultiMimsy 2000/MultiMimsy XG database
- The possibilities of an OAI Repository

Before I go on…

- What *is* OAI?

OAI is...

- In this context, 'OAI' is short-hand for the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)
- It supports 'verbs' – things the repository can do, like identify itself, return a list of records, tell you what metadata format its using, get individual records
- It's a big box of metadata and files that you can browse or search

If you're small or poor...

- You can use a Static OAI-PMH repository instead of a full or dynamic repository
- It's basically an XML file in a special format
- You can create this XML via a Word or Excel export with a bit of scripting (or macros)

About the Museum of London's environment

- We migrated from MultiMimsy 2000 to MultiMimsy XG last year
- We have another big database of archaeological data from MoLAS, databases for Archive Management Systems and the LAARC
- We have a Content Management System for the websites with thematic or interpretive content

Museum Systems Team

- Small permanent in-house IT team
- I design and develop bespoke database applications for recording, analysing and publishing archaeological or museum information
- We design and develop database-driven websites (with Web Developer and Content Manager) with content from various sources

Technical Infrastructure at MoL

- Desktop MultiMimsy client forms
- MultiMimsy server (database)
  - day-to-day Collections Management System, holds data in own format
- Staging server (database/web server)
  - runs extraction scripts as queries against MM database, holds development version of data structures and scripts and static pages for testing
- Live server (database and web servers)
  - Host the sites you see on the internet

Thinking about extractions?

- Work out what you need for the final product
- Reports can help you test your data is fit for purpose

Typical Site Goals

- Enable access to the collection
- Provide for specialist and general audiences
- Increase knowledge and understanding of the collection

But how do we do that with a system designed for collections management?

Typical design challenges

- Different audiences, different goals
  - General public
  - Researchers and specialists
- Dynamic content
  - Complex relationships between:
  - Categories
  - Object records
  - Other records

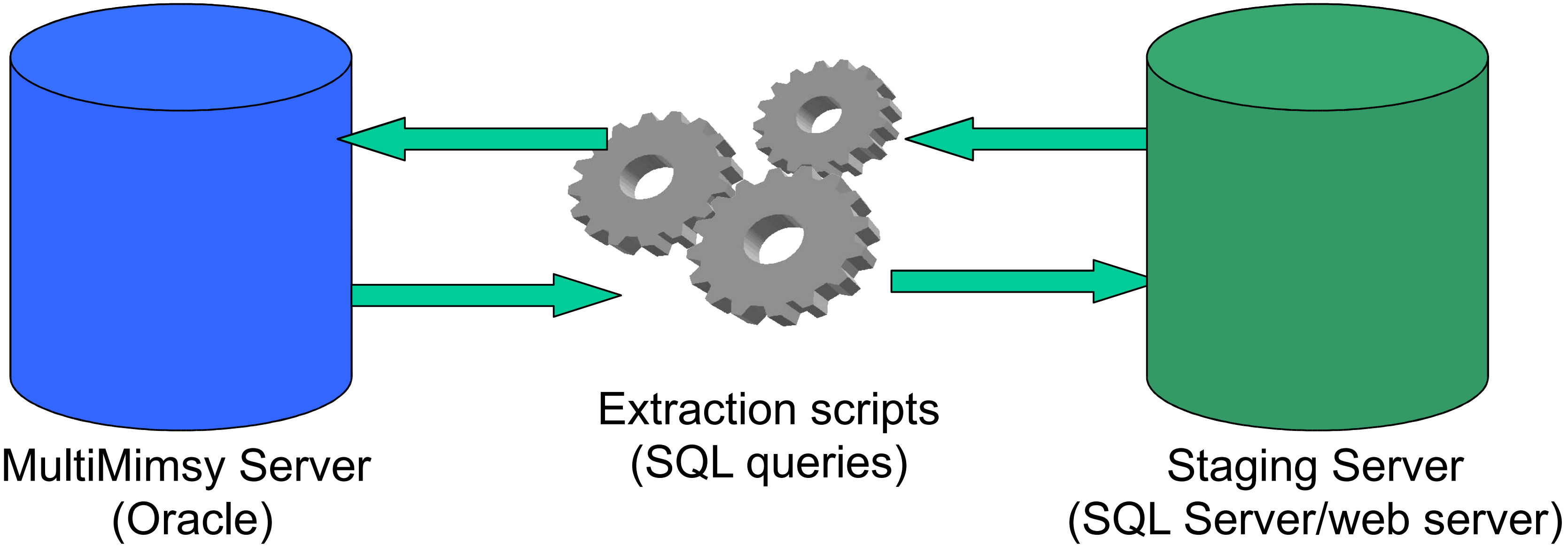
Website design and development

- Consultation with target audiences
- Consultation with cataloguing team and curators
- Design site architecture and navigation
  - Iterative process, taking into account audience needs, structure of data
- Design presentation of objects, categories, publications, images, and their relationships
- Test and re-test with your users

Typical Site Infrastructure

- Static content:
  - extended texts (stored externally)
  - Regional study
  - Site report, 'about' section
- Dynamic content from MultiMimsy:
  - object records
  - subject authority or group records
  - images
  - publications

Extraction proceses – the geeky bit



Staging server runs extraction scripts against MultiMimsy (Oracle) database server:

- Scripts are a set of SQL queries on Microsoft SQL Server, stored as ‘Data Transformation Services’ (DTS) so they can be scheduled to automatically re-run and update the web database
- Results stored in database tables on staging server

Extraction processes – the tricky bit

- Work out what schemas or data structures are needed in the final site
  - create wireframes with every possible item that might be displayed the page
  - don't forget the 'invisible' fields needed on the backend to present that data appropriately e.g. what determines which items are displayed on each page. These might include back-end fields for navigation or search.

Extraction processes – the tricky bit

- Map from publication schema to MultiMimsy fields
- Figure out how content from other sources will be linked in, e.g.:
  - string matching on authority names
  - unique IDs or accession numbers

Extraction processes

- Consistency helps – if you have used different fields in each project, you need to re-map web schemas to MultiMimsy each time

Catalogue records

- The interface you see doesn't match the backend so mapping can be tricky
- Allow time for finding these fields then working out how they're related to other tables
- Use reports and exports to generate SQL and test relationships
- Depending on your version of MM you can try and get the field name from the form.

Images

- In our implementation:
  - Image metadata is stored in the media table
  - Image files are stored on our filesystem
  - I run a SQL query to generate a list of files and their locations on the filesystem (path, image name)
  - I then run an ASP script to generate a DOS batch file which then creates the necessary directories and copies the images into them, retaining the path structure

Information records [authorities]

- More scripts to pull related authorities using links between object and information records:
  - Publications
  - People/Organisations
  - Places
  - Subjects
  - Groups

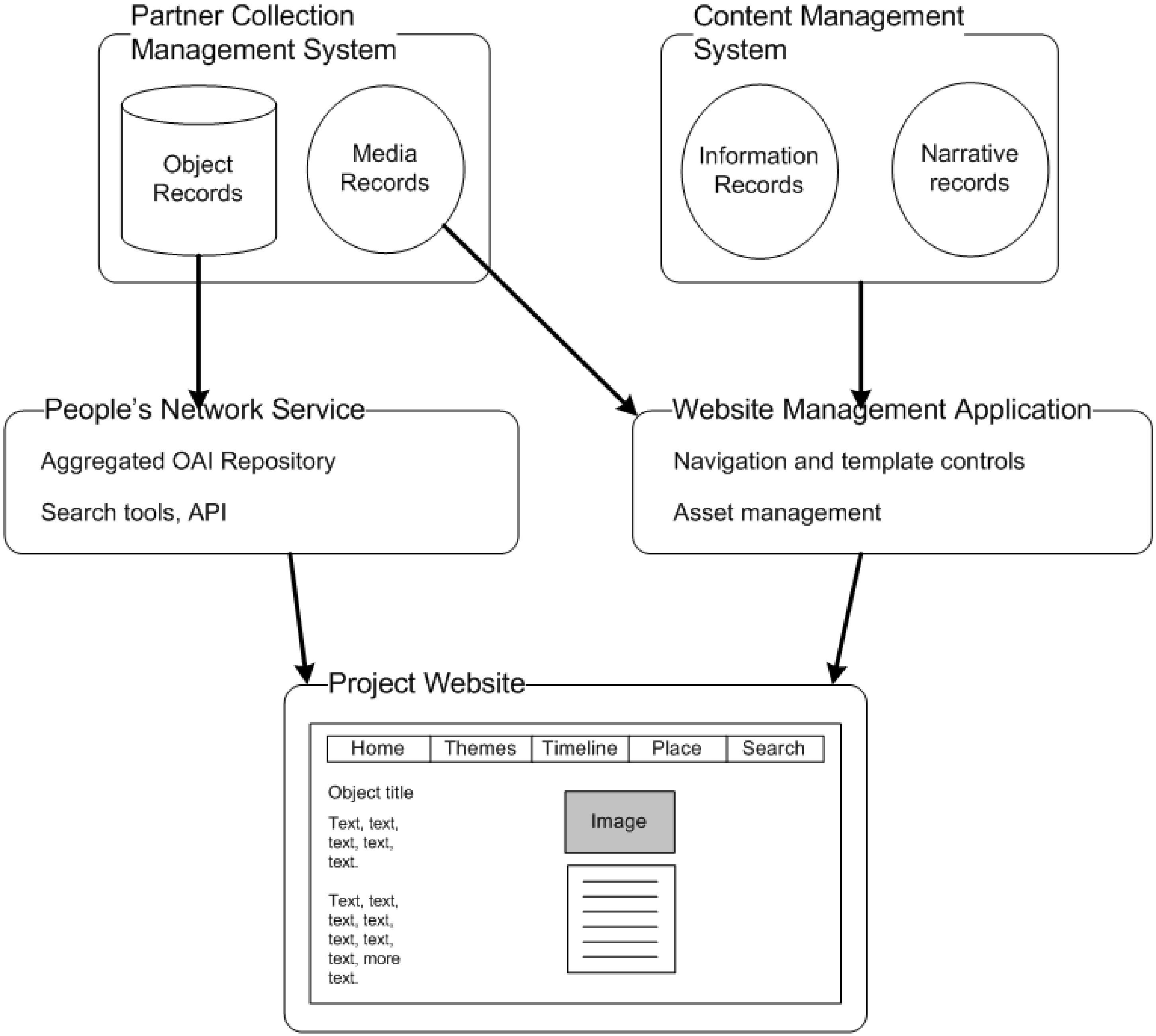
The final result

- A website!
- Hopefully you will have seen Exploring 20th Century London or another Museum of London microsite
- Here's how Exploring worked

Where does OAI come in?

- It was the model we used for the Hub partnership project, Exploring 20th Century London

Simplified System Architecture



Partner Collection Management System

Object Records

Media Records

Content Management System

Information Records

Narrative records

People's Network Service

Aggregated OAI Repository

Search tools, API

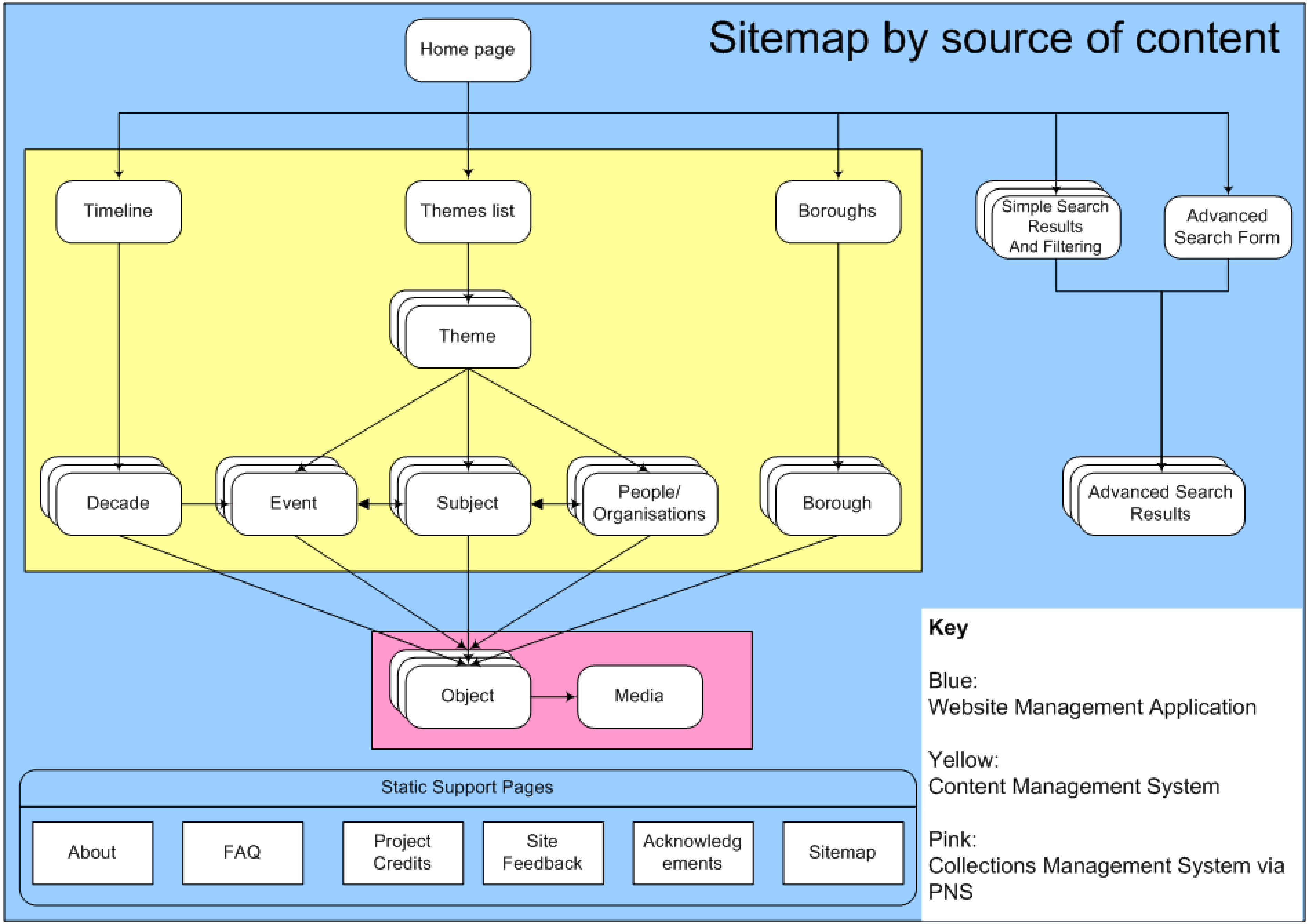
Website Management Application

Navigation and template controls

Asset management

Project Website

Home
Themes
Timeline
Place
Search
Object title
Text, text, text, text, text.
Text, text, text, text, text, text, text, more text.
Image



Sitemap by source of content

Key

Blue:
Website Management Application

Yellow:
Content Management System

Pink:
Collections Management System via PNS

Extraction models considered for X20CL

- Single static database, data loaded manually

- Automatic harvesting to central database

- Distributed System, data stored locally and queried live

Result: the harvesting model was chosen, with OAI-PMH implementation

Advantages of OAI

- More reliable and faster than querying distributed partners; saves bandwidth and processing time, as only new, updated or deleted data is moved

- Customisable project schema and data repository allow the production of re-usable and interoperable content

- Can support metadata standards such as Dublin Core and Spectrum XML

- Open standard: reduces risk of 'lock-in'

- Variety of open-source tools are available on all platforms

- Established museums, libraries, archives user base

MoL OAI repository

- The use of OAI was inherited from Exploring 20th Century London
- Since we have it, we hope that our repository will have a use beyond providing an OAI-PMH-compliant data source for partnership projects and our own internal requirements

MoL repository

- We're going to use a DSpace repository for collections data - object metadata, media files and metadata and information record (people, places, events, publications) metadata - for selected records from our Mimsy XG collections management system

Possibilities for MoL OAI repository

- Permanent, stable URI (unique address) for each object – an 'object home'
- Other people may query the repository – offers a fully-featured collections search while providing greater visibility for data
- Semantic web and other cool stuff?

Possibilities for repository

- Authoritative index into our collections database, with links to every online instance of an object, regardless of project, showing different thematic or interpretive uses of the object in other websites
- Link from object to all related information or authority records and media such as images, audio files, transcripts, object captions and descriptions; related objects

Semantic web and the repository?

- The 'object home' means our data is ready for the semantic web
- Lightweight 'semantic web' technologies can be already used with that data
- The search interface of the repository could act as an API – a 'box of tricks'

OpenSearch, RDF, feeds

- We can be fully buzzword-compliant
- Queries can be converted to RSS streams (OpenSearch, GeoRSS) or RDF
- These streams can be used by others in 'mash ups'

New uses of our data

- We aren't resourced to provide interfaces to meet every requirement
- We can provide data for others to create interfaces to browse or search data in new ways
- Mashups allow people to merge our content with other sources, e.g. online maps, other collections
- User-centric, not museum-centric

It's a bit experimental

- Working with supportive suppliers (BioMed Central)
- Currently resolving issues of how records relate to each other, as Dublin Core doesn't handle it well - possibly ORE to create 'bundles' of records
- Could we incorporate user-generated content such as links to the object from user sites, comments, tags?

Questions?