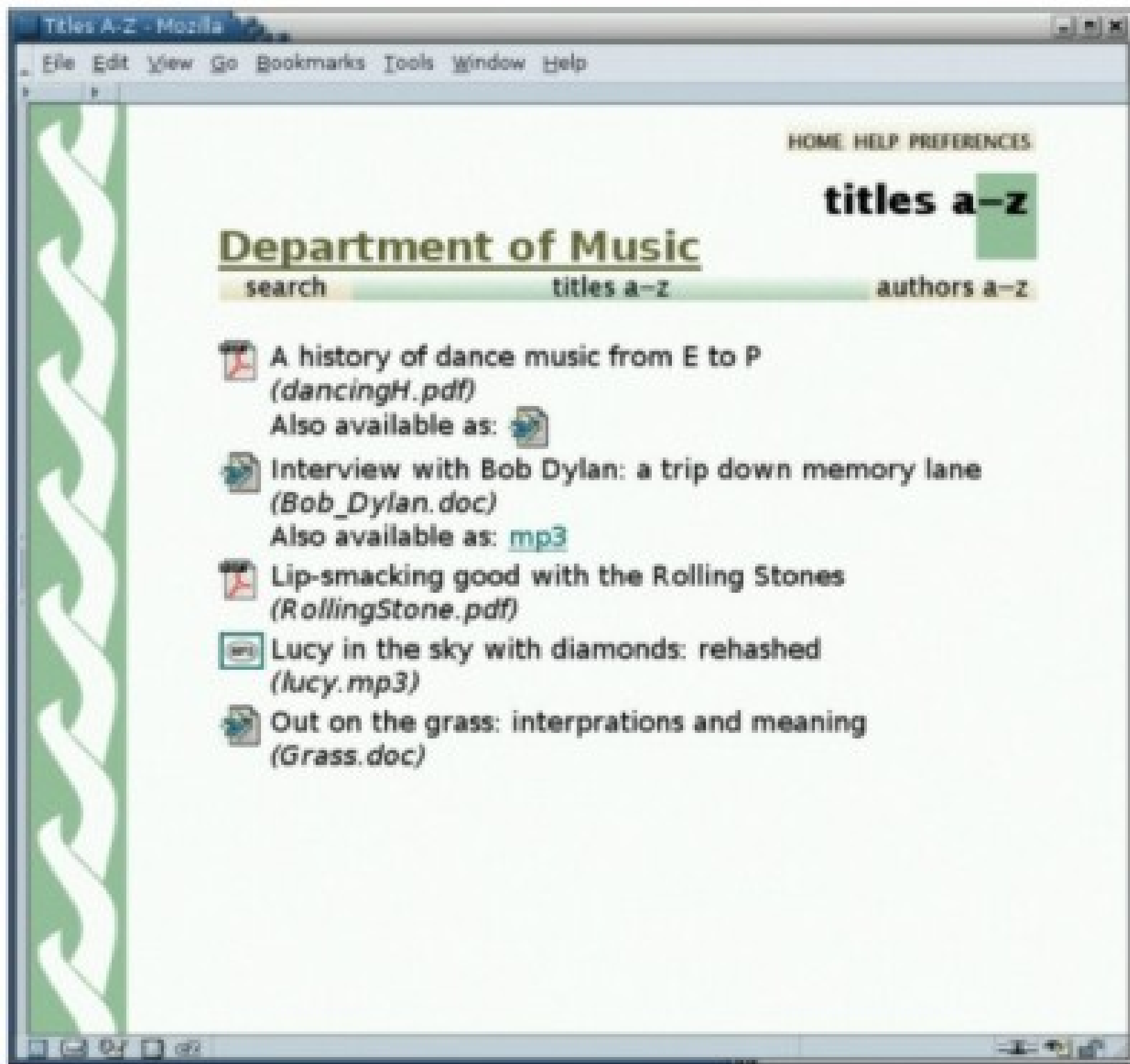


# CALJAX

in-Browser Digital Repository using AJAX

# Digital Repositories

- Manage collection of digital objects
  - Browsing
  - Searching
  - Updating
  - ... and more
- Several solutions available
  - DSpace
  - EPrints
  - Fedora
  - Greenstone
- Only Greenstone can export a collection



# AJAX

- Asynchronous Javascript and XML
- Adds a programmable layer to otherwise static HTML
- XMLHttpRequest
- Typically used to make websites more responsive
  - Partial updates of pages
- Security is a big concern
  - Cross-site scripting
- Javascript is limited
  - Network access
  - Disk access
- Motivation?
  - Bundled with modern Web browsers
  - Limitations can be overcome

# Motivation

- Having an offline Digital Repository is desirable
- Only Greenstone can export a collection
  - Still requires a Web server
  - Updating collection can take up to 2 days
  - 50 000 copies a year (2001)
- Bleek and Lloyd Collection
  - "an archive of narratives, drawings and documents of and by the !xam and !kun people of southern Africa."
  - Completely static
  - Specific to the data
- Nothing like CALJAX exists
  - Offline
  - Lightweight
  - General solution

# Research Question

- Lots of Digital Repositories
- Heavyweight
  - Require a Web server
  - Can be difficult to setup
  - Usually no offline support

**Can we make a Digital Repository system, which can function without a Web server, relying only on an AJAX-compliant Web browser?**

# Project Description

- Proof of concept
  - Used to show that the approach is viable
- Lightweight
  - Web browser with AJAX
  - Work offline and online
- Digital Repository
  - Basic functionality

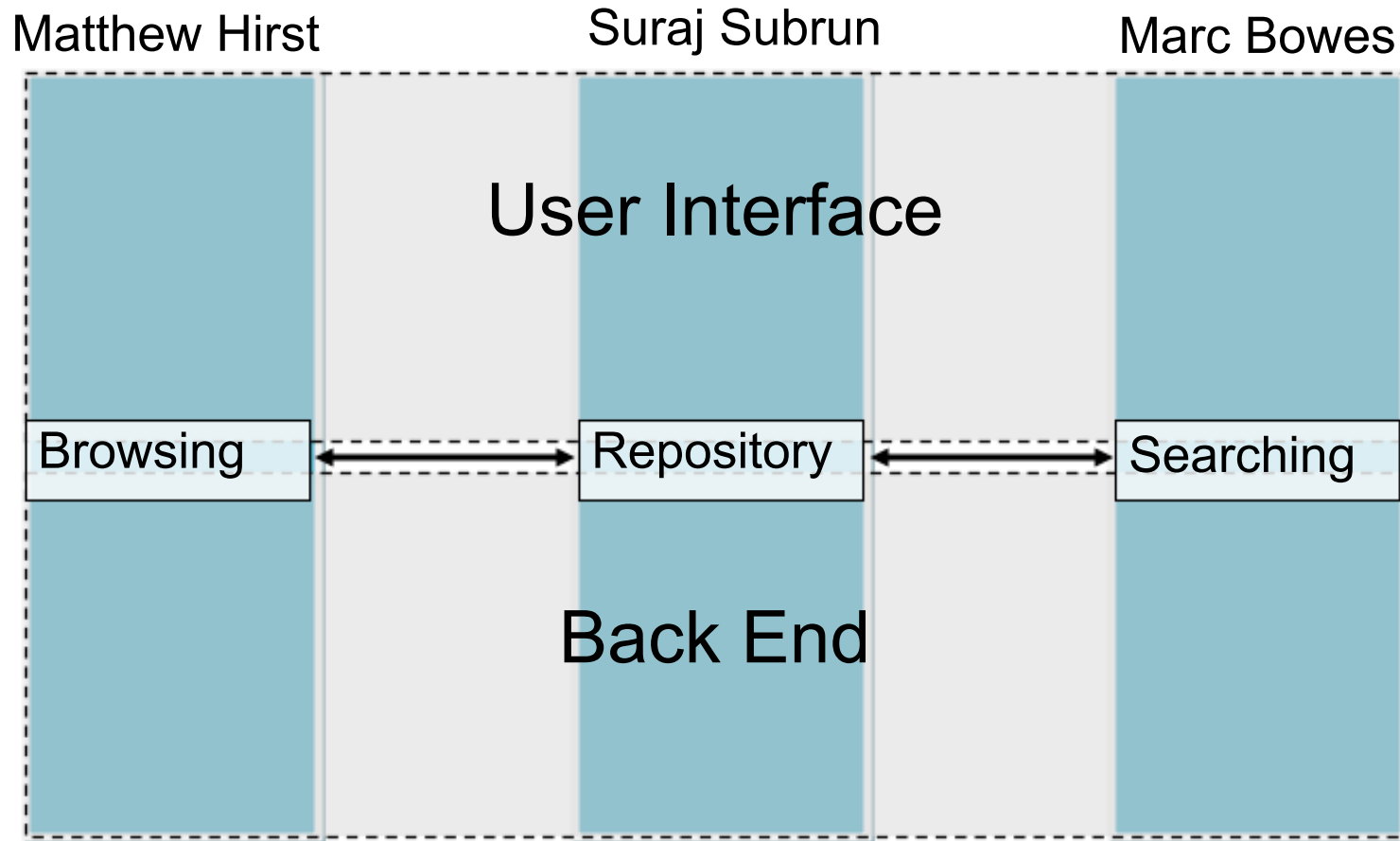
# Project Description



Top level overview



# Work Allocation (1 of 2)



# Work Allocation (2 of 2)

- Mitigates risk of people leaving or not completing work
- Stubs
  - Fabricated data for testing
- Integration
  - Common API
  - Common Data Format

# Browsing - Features

- Required features obtained through
  - Research into digital repositories
  - Using other digital repository systems
  - Project Specification
- Features Include
  - Browsing via various criteria
  - Able to see information on each item
  - Able to access the original file
  - Able to access files online and offline

# Browsing - Description (1 of 3)

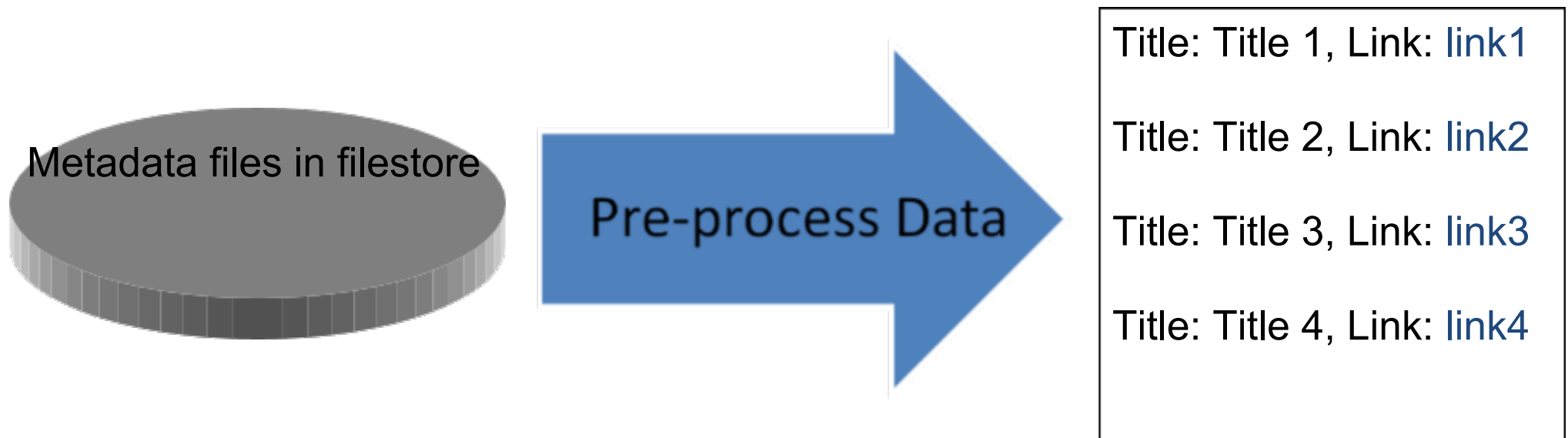
Metadata: Item:



- Title: Browsing AJAX Repositories
  - Author: Matthew Hirst
  - Subject: Digital Repositories
  - Year: 2009
  - etc.
- 
- Browse object in the repository using metadata in XML format
  - Browse by Title, Author, Subject, etc.

# Browsing - Description (2 of 3)

- Use AJAX to load metadata from the XML files.
- AJAX Security -> Cannot access filestore directly.
- Pre-process the data using external program.
- Embed title and links to the objects XML file in HTML pages.



# Browsing - Description (3 of 3)

- Use embedded AJAX to load XML through these links.
- Style the XML so it is easy to interpret by users (XSL)
- Approach previously used by Tiddlywiki
  - Lightweight wiki page.
- Adapt the approach to Digital Repositories

# Browsing - Evaluation

- Need to evaluate 3 aspects:
- Core Functionality
  - Proof of concept -> Does it work?
- Compatibility
  - On different Operating Systems
  - On different Web Browsers
- Usability
  - User tests
    - Ease of use
    - Functionality

# Searching - Description

- Expected feature of a Digital Repository system
- Adds tremendous value to the collection
  - No point having the objects if they aren't retrievable
- Browsing not a good solution if looking for a specific object

The screenshot shows the DSpace at MIT website. The header features the 'DSpace' logo and 'MIT Libraries' branding. The main content area is divided into several sections:

- Search DSpace:** Includes a search input field, a 'Go' button, and a link to 'Advanced Search'.
- DSpace at MIT >**: A section titled 'DSPACE VERSION 1.1' with a description of new features like Advanced Search and keyword searching, and a link to 'More news...'
- Search:** A section with the instruction 'Enter some text in the box below to search DSpace.' and a search input field with a 'Go' button.
- Communities in DSpace:** A section titled 'Select a community to browse its collections.' with links to 'Center for Technology, Policy, and Industrial Development (CTPID)', 'Department of Ocean Engineering', and 'Laboratory for Information and Decision Systems (LIDS)'.
- What can you find in DSpace?:** A section stating 'MIT Research in digital form, including preprints, technical reports, working papers, conference papers, images, and more.'
- Is this all of MIT's research?:** A section stating 'No. DSpace is limited to digital research products. For items in print, go to [Sutton: MIT Libraries' catalog](#). DSpace is young and growing rapidly. Check back often.'

The left sidebar contains navigation links: 'Home', 'Browse' (with sub-links for 'Communities & Collections', 'Titles', 'Authors', and 'By Date'), and 'Sign on to:' (with sub-links for 'Receive email updates', 'My DSpace authorized users', and 'Edit Profile').



# Searching - Description

- Preprocessor
  - Inverted files
- Query in Web browser
  - AJAX builds results

Doc1	apples bananas apples apples
Doc2	bananas bananas apples bananas bananas

↑  
original  
documents

inverted files →

apples	Doc1: 3 Doc2: 1	4
bananas	Doc1: 1 Doc2: 4	5

# Searching - Goals

- Results
  - Good?
  - Good enough!
- Performance
  - Javascript is slow
  - Engine performance is varied
  - Not just between browsers
- Portability
  - Ensure solution isn't proprietary
- Goal
  - Quality of results
  - Performance
  - Maximum of both

# Searching - Evaluation

- Sample data
- Find the relevant documents?
  - User testing
- Portability testing
  - Operating Systems
  - Web browsers
- Proof of concept
  - Can we search the repository?
  - Small software footprint
  - Quality of search
  - Must not detract from usability of system

# Updating and Managing

- System:
  - Portable collections meant to be used offline
  - Online central repositories
  - No major difference in software used.
- Management: Core Layer and Storage Layer
- Core Layer
  - Management and access subsystems
  - Validation and integrity of data
  - Implemented using mainly AJAX
- Storage Layer
  - Additional technologies might be considered
    - Java
  - Storage using dynamically indexed XML / databases
- Web 2.0 technologies used for interaction with central repository

# Updating and Managing - Evaluation

- Main objective
  - Operation under multiple platforms
  - Adequate feature support
- Multiple platforms
  - Conducting test runs on various systems
- Feature support
  - User evaluations

# Iterations

# Ethical, Professional & Legal Issues

- No foreseeable ethical or legal issues with implementation
- Software creation
  - Tools used carry no legal limitations (GPL licenses)
- User testing
  - Will consider all relevant ethics

# Anticipated Outcomes - System

- Basic Digital Repository
  - Browse, Search, Update
- Offline
- Small software footprint
- Cross-platform (OS & Browser)



# Anticipated Outcomes - Impacts

- Potential for profound impact
  - Huge educational value (e.g., medical, cultural, agricultural information)
    - Wide audience at a low cost
- Research in remote areas
  - Portable data
  - Can synchronise with main repository afterwards

# Key Success Factors

- Need to evaluate primary features
  - Browsing, Searching & Updating
  - Are they adequate?
- Minimal software requirements
  - Minimal or no installation
- Portability of System
  - Cross-platform

# CALJAX

- in-Browser Digital Repository using AJAX
- Three core features
  - Browse
  - Search
  - Update
- Export a collection for offline viewing
- Minimal software requirements
  - Just a Web browser!
- No such system exists
- Proof of concept

# Risk Analysis

## Risks:

- Group member leaves project group
- Group member does not complete their section of work
- Software used in approach does not work
- Loss of work due to system crash or other technical failure
- Scope of project is too large or too small

## Risk mitigation strategies:

- Split project up in such a way that members leaving or not completing work does not affect other members
- Develop a prototype early to test the technology that is used.
- Make sure proper measures are used to back up work.
- Make sure there is sufficient time left before the deadline so that if the scope is too large or too small the project can be altered.

# Milestones

- Prototype Demo - 21/08/2009
- Mainly to test functionality
- First Implementation - 2/10/2009
- First look at system which will be evaluated and developed into final (functionary -> user testing maybe)
- 
- Final Prototype - 16/10/2009