Flexible Design for Simple Digital Library Tools and Services

Lighton Phiri    Hussein Suleman

Digital Libraries Laboratory
Department of Computer Science
University of Cape Town

October 8, 2013
Contextual overview

- Problem and motivation
  - Preservation costs
  - Technical skills and education
  - Computing resources

- Proposed solution
  - Simplicity and minimalism
    - Successes of minimalism — Project Gutenburg
  - Principled DL design

- Prior work
  - Derivation of design principles
  - Repository implementation
    - Real-world case studies
Repository prototype architectural design

- File-based
  - Digital objects stored on native operating system
  - Hierarchical collection structure

- Metadata objects
  - Plain text files
  - Encoded using Dublin Core
  - Relationships modelled using metadata elements

- Object organisation
  - Metadata records stored alongside objects
  - Content objects and container objects nested within other container objects
User study experiment

- **Objective**
  - Developer-oriented
  - Simplicity and flexibility of file-based store

- **Target population**
  - 34 Computer Science honours students
  - 12 groups of twos and threes

- **Skillset**
  - Technologies relevant to study —DBMS, XML, Web apps
  - Storage solutions
  - Digital Libraries concepts

- **Approach**
  - Subjects tasked to build layered services using file-based store
  - Marks awarded for innovation —among other facets
  - Subjects answered post-experiment survey
## User study experiment - results

- **Survey participants**
  - 76% response rate — representation from all 12 groups

<table>
<thead>
<tr>
<th>Group</th>
<th>Web service</th>
<th>Candidates</th>
<th>Respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>Transcription</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Group 2</td>
<td>Downloader</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Group 3</td>
<td>Commenting</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Group 4</td>
<td>Visualisation</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Group 5</td>
<td>Transcription</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Group 6</td>
<td>Annotation</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Group 7</td>
<td>Visualisation</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Group 8</td>
<td>Browsing</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Group 9</td>
<td>Annotation</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Group 10</td>
<td>Rating</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Group 11</td>
<td>Gestures</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Group 12</td>
<td>Visualisation</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
User study experiment - results (1)

Programming languages

<table>
<thead>
<tr>
<th>Programming languages</th>
<th>Number of subjects</th>
</tr>
</thead>
<tbody>
<tr>
<td>JavaScript</td>
<td>16</td>
</tr>
<tr>
<td>PHP</td>
<td>15</td>
</tr>
<tr>
<td>HTML5</td>
<td>11</td>
</tr>
<tr>
<td>Python</td>
<td>8</td>
</tr>
<tr>
<td>Java</td>
<td>4</td>
</tr>
<tr>
<td>C#</td>
<td>3</td>
</tr>
</tbody>
</table>
User study experiment - results (2)

Repository aspects

Simplicity

Structure

Metadata

Understandability

Structure

Metadata

Number of subjects

0 5 10 15 20 25

Strongly agree Agree Neutral Disagree Strongly disagree

10 of 26
User study experiment - results (3)

- Users asked to rank storage solutions in order of preference
- What aspects of your most preferred solution [database] above do you find particularly valuable?
  - “I understand databases better.”
  - “Simple to set up and sheer control”
  - “Easy setup and connection to MySQL database”
  - “Ease of data manipulation and relations”
  - “Centralised management, ease of design, availability of support/literature”
  - “The existing infrastructure for storing and retrieving data”
Do you have any general comments about the data structure or format?

- “Had some difficulty working the metadata, despite looking at how to process DC metadata online, it slowed us down considerably.”
- “Good structure although confusing that each page has no metadata of its own (only the story).”
- “The hierarchy was not intuitive therefore took a while to understand however having crossed that hurdle was fairly easy to process.”
- “I guess it was OK but took some getting used to”
User study experiment - findings

- Simplicity resulted in more understandable structure
  - 69% agreed that XML-files were simple
  - 61% found XML format easy to work with
  - 62% found hierarchical structure simple to work with
  - 46% found hierarchical structure easily understandable

- Simplicity does not affect flexibility of interaction with file-store
  - No influence on choice of language
  - Only 15% of subjects thought it did
Performance experiment

- **Objective**
  - Assess performance relative to collection size

- **Test Environment**
  - Pentium(R) Dual-Core CPU E5200@ 2.50GHz; 4GB RAM
  - 32 bit Ubuntu 12.01 LTS
  - Siege and ApacheBench for benchmarking

- **Metrics**
  - Response time

- **Factors**
  - Collection hierarchical structure
  - Collection size — digital objects
## Performance experiment - test dataset

- NDLTD Union Catalog — [http://union.ndltd.org/OAI-PMH](http://union.ndltd.org/OAI-PMH)
  - Harvested 1,907,000 metadata records
  - Dublin Core-encoded plain text files
- Linearly increasing workload

<table>
<thead>
<tr>
<th>Workload</th>
<th>Objects</th>
<th>Cols</th>
<th>Size [MB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>100</td>
<td>19</td>
<td>0.54</td>
</tr>
<tr>
<td>W2</td>
<td>200</td>
<td>25</td>
<td>1.00</td>
</tr>
<tr>
<td>W3</td>
<td>400</td>
<td>42</td>
<td>2.00</td>
</tr>
<tr>
<td>W13</td>
<td>409,600</td>
<td>128</td>
<td>1945.00</td>
</tr>
<tr>
<td>W14</td>
<td>819,200</td>
<td>131</td>
<td>3788.80</td>
</tr>
<tr>
<td>W15</td>
<td>1,638,400</td>
<td>131</td>
<td>7680.00</td>
</tr>
</tbody>
</table>
Performance experiment - test dataset (2)

- Two datasets spawned from initial dataset
  - one-, two- and three-level structures

(a) Dataset #1

(b) Dataset #2

(c) Dataset #3
Performance experiment - evaluation aspects

- Transaction log analysis — http://pubs.cs.uct.ac.za
  - Ingestion
  - Full-text search
  - Indexing operations
  - OAI-PMH data provider
  - Feed generation

Welcome to UCT CS Research Document Archive

Welcome to the UCT Computer Science Research Document Archive, which archives and makes accessible documents that are products and by-products of research in the department.

Search the Titles, Authors, Abstracts and Keywords:

Browse
Browse the archive by Subject, Year, Lab or Type.

Latest Additions
View items added to the archive in the past week.
Performance experiment - experimental design

- Performance benchmarking
  - Evaluation aspects
  - Three-run averages for all scenarios
    - Datasets #1, #2 and #3
    - 15 workloads
  - Break-even points for performance degradation
  - Nielsen’s three important limits for response times

- Performance comparisons
  - Benchmark results vs DSpace 3.1
    - Ingestion
    - Full-text search
    - OAI-PMH data provider
Performance experiment - results

![Graph showing item ingestion time for different workload sizes and datasets.](image)
Performance experiment - results (2)

![Graph showing performance results for full-text search.](chart)

- **Y-axis:** Log10(Time [ms])
- **X-axis:** Workload size

Legend:
- **Blue:** Traversal time
- **Red:** Parsing time
- **Green:** XPath time

<table>
<thead>
<tr>
<th>Workload Size</th>
<th>Traversal Time</th>
<th>Parsing Time</th>
<th>XPath Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.6k</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2k</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.4k</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.8k</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25.6k</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51.2k</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102.4k</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>204.8k</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>409.6k</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>819.2k</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1638.4k</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Performance experiment - results (3)

OAI-PMH data provider

\[
\log_{10}\text{Time [ms]}
\]

Workload size

- GetRecord
- ListIdentifiers
- ListRecords
- ListSets
Performance experiment - results (4)
Performance experiment - findings

- Performance benchmarking
  - Performance within 'acceptable' limits for medium-sized collections
  - Ingestion performance NOT affected by collection scale
  - Performance generally degrades for collections > 12,800 objects
  - Performance degradation adversely affects information-discovery services —Feed generation, full-text search and OAI-PMH data provider

- Comparison with DSpace 3.1
  - Ingestion performance better than DSpace
  - Information discovery operation —search and OAI-PMH— are slower than DSpace
    - DSpace uses Apache Solr for index
    - Comparable speeds can be attained through integration with third-party search services
Conclusions and future work

Conclusions

- Feasibility of simple DL architectures
- Simplicity does not affect flexibility and potential extensibility of result tools and services
- Performance acceptable for small- and medium-sized collections
- Comparable features with well-established solutions

Reference implementation

- Packaging
- Version control
Thank You

Questions?

Additional Information

http://dl.cs.uct.ac.za