Simple Archive Architectures

Lighton Phiri and Hussein Suleman

Digital Libraries Laboratory
Department of Computer Science
University of Cape Town

IFLA '15 Workshop on Digital Libraries: research methods and tools
Contextual Overview

● Problems and challenges
  ○ Preservation costs
  ○ Technical skills and expertise
  ○ Computing resources

● Proposed solution
  ○ Explicit simplicity and minimalism
  ○ Principled design of DL tools and services

● Motivation
  ○ Successes of minimalism---Project Gutenberg
Research goals

● Is it feasible to implement DLSes based on simple architectures?
  ○ How should simplicity for DLS storage and service architectures be defined?
    ■ Derivation of design principles
    ■ Simple repository prototype + case studies
  ○ What are the implications of simplifying DLS?
    ■ Developer user study
    ■ Performance evaluation
  ○ What are some of the comparative advantages and disadvantages of simple architectures?
    ■ DSpace 3.1 comparative evaluation
Claim #1: Simplicity for DL storage and services can be defined through derivation
Design Principles (1)

- Meta-analysis of popular software applications
  - 12 candidate tools were considered---even split between DL and non-DL tools
  - Tool attributes that potentially influenced design of tools identified
  - Pair-wise comparison done to assess most appropriate attributes

- Eight guiding design principles derived [1]
  - Applicable for simple and minimalistic architectures
Design Principles (2)

- Principles mapped to potential repository architectural design decisions
  - Applicable principles derived during mapping

Table 4-1. Simple repository persistent object store design decision

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issues</td>
<td>Principles 1, 2, 6 and 8</td>
</tr>
<tr>
<td>Decision</td>
<td>Store bitstreams on the local operating system filesystem</td>
</tr>
<tr>
<td>Assumptions</td>
<td>None</td>
</tr>
<tr>
<td>Alternatives</td>
<td>Store bitstreams as blobs in a database; store bitstreams in the cloud</td>
</tr>
<tr>
<td>Rationale</td>
<td>Backup and migration tasks associate to repository objects can be potentially simplified; operating system commands can be used to perform repository management tasks</td>
</tr>
<tr>
<td>Implications</td>
<td>None—most conventional tools and services use the same approach</td>
</tr>
<tr>
<td>Notes</td>
<td>None</td>
</tr>
</tbody>
</table>
Simple Repository Prototype

- File-based
  - Digital objects stored on OS
  - Hierarchical collection structure
- Metadata objects
  - DC plain text files
- Object organisation
  - Metadata stored along content
  - Nested objects
Case studies

● Two case studies involving two different collections
  ○ The Bleek and Lloyd Collection
    ■ Honours project: “Bonolo” [5]
  ○ SARU archaeological database
    ■ Honours project: “The School of Rock Art” [6]
“The Digital Bleek and Lloyd”

- 18,924 content objects with a total size of 6.2GB
- Two-level collection structure
  - Virtual content objects representing stories
- “Bonolo” [5] DLS implemented using repository sub-layer
“SARU Archaeological database”

- 72,333 content objects with a total size of 283GB
- Four-level collection structure
- “The School of Rock Art” [6] implemented using repository sub-layer
Claim #2: There are desirable features and advantages possessed by DL tools and services implemented using simple architectures
User Study (1)

● Developer-oriented study
  ○ Assess simplicity and flexibility of simple repository architecture

● Target population
  ○ 34 computer science honours students split into 12 groups of twos and threes
  ○ Basic developer skills and DL knowledge

● Approach
  ○ Participants tasked to build layered services using simple repository
  ○ Post-experiment survey
User Study (2)

- Wide variety of layered services
- Wide variety of programming languages used
- Choice of language not influenced by repository design; only 15% indicated that it did
User Study (3)

- Dublin Core XML-encoded files perceived simple & easy to work with
  - 69% and 61% respectively
- Repository perceived simple but not easily understandable
  - 62% and 46% respectively
User Study (4)

● Simplicity resulted in more understandable repository layer
  ○ Most participants found Dublin Core XML-encoded metadata files easy and simple to work with
  ○ Most participants found hierarchical structure simple but not easily understandable

● Flexibility of interaction with repository layer unaffected by simplicity
  ○ No influence on programming languages
Performance Evaluation (1)

- Assess and benchmark performance relative to collection size
  - Typical DL service aspects evaluated. Ingestions, search, OAI-PMH data provider and feed provider
  - Log analysis of production repository informed aspects
- Comparative assessment with DSpace 3.1
- Experimental design
  - Metrics---Response time
  - Factors---Collection size and structure
Performance Evaluation (2)

- Three datasets with 15 linearly increasing workloads; data from NDLTD Union Catalog
  - One-, two- and three-level collection structures
  - Varying objects in different collection structures

(a) Dataset#1 structure
(b) Dataset#2 structure
(c) Dataset#3 structure
Performance Evaluation (3)

- Performance within acceptable limits for medium-sized collections
- Collections > 12,800 objects affected
- Information-discovery services---feed, full-text search and OAI-PMH data provider---affected
Performance Evaluation (4)

- Performance benchmarking
  - Performance within acceptable limits for medium sized collections
  - Performance degradation beyond 12,800 objects
  - Performance degradation adversely affects information discovery services; ingestion process unaffected by collection scale

- Comparison with DSpace 3.1
  - Ingestion performance outperformed DSpace 3.1
  - Information discovery services outperformed by DSpace 3.1
Conclusions

- Principled DL design approach undertaken
- Feasibility of simple DL architectures
- Minimalism does not affect flexibility and extensibility of DL tools and services
- Performance acceptable for small- and medium-sized collection
- Comparable results with well-established solutions
Bibliography


Questions?
Additional information

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