What? So What?
The Next-Generation JHOVE2 Architecture
for Format-Aware Characterization

Stephen Abrams
California Digital Library
Stephen.Abrams@ucop.edu

Sheila Morrissey
Portico
Sheila.Morrissey@portico.org

Tom Cramer
Stanford University
tcramer@stanford.edu
Digital preservation

- Managing the gap between what you were given and what you need
- That gap is only manageable if it is quantifiable
- Characterization tells you what you have, as the starting point for iterative preservation planning and action

What? So what?

• What is it?
  – Identification Determining presumptive format through signature matching

• What is it, really?
  – Validation Determining conformance to commonly-accepted normative requirements

• What about it?
  – Feature extraction Reporting intrinsic properties significant to preservation planning and action

• What should you do with it?
  – Assessment Determining acceptability for a given purpose on the basis of locally-defined policies
Characterization in ingest workflows

Diagram showing the processes involved in characterizing content in ingest workflows, including identification, validation, feature extraction, SIP, Unpackage, consistency assessment, and ingestion. The workflow involves producer content, metadata, policy rules, and archive.
Characterization in migration workflows
JHVE2 project

• A next-generation architecture for format-aware object characterization
  – Three-fold goals:
    • Re-factor the existing architecture to achieve higher performance, simplify system integration, and encourage third-party enhancement
    • Provide significant new function
    • Implement modules

• Collaborative project of CDL, Portico, and Stanford University
  – Funded by Library of Congress/NDIIPP
  – Open source BSD license
What is a format, anyway?

- A set of syntactic and semantic rules for mapping between abstract information content and bit streams

- If we are interested in preserving *content*, and not merely bit streams, managing format is fundamentally important

```
ffd8ffe000104a46
4946000102010083
00830000ffed0fb0
50686f746f73686f
7020332e30003842
494d03e90a507269
6e7420496e666f00
0000007800000000
0048004800000000
02f40240fffefffe
0306025203470...
```
What is a format, anyway?

- A set of syntactic and semantic rules for mapping between abstract information content and bit streams
- If we are interested in preserving *content*, and not merely bit streams, managing format is fundamentally important

```
ffd8ffe000104a46  SOI
4946000102010083  APP0  JFIF 1.2
00830000ffed0fb0  APP13  IPTC
50686f746f73686f  APP2  ICC
7020332e30003842  DQT
494d03e90a507269  SOF0  183x512
6e7420496e666f70  DRI
0000007800000000  DHT
0048004800000000  SOS
02f40240ffeeffee  ECS0
0306025203470...  ...
```
What is a format, anyway?

• A set of syntactic and semantic rules for mapping between abstract information content and bit streams

• If we are interested in preserving content, and not merely bit streams, managing format is fundamentally important

  ffd8ffe000104a46  SOI
  4946000102010083  APP0  JFIF 1.2
  00830000ffed0fb0  APP13  IPTC
  50686f746f73686f  APP2  ICC
  7020332e30003842  DQT
  494d03e90a507269  SOF0  183x512
  6e7420496e7420332e300038  DRI
  4946000102010083  DHT
  0000007800000000  SOS
  0048004800000000  ECS0
  02f40240ffeefee  ...
Objects, not files

• JHOVE assumed 1 object = 1 file = 1 format

• But what about…
  – TIFF with embedded ICC profile and XMP metadata
    1 object = 1 file = 3 formats
  – JPEG 2000 JPX fragmentation
    1 object = n files = 1 format
  – ESRI Shapefile
    1 object = 3 files = 3 formats

• JHOVE2 will support 1 object = n files = m formats
Objects, not files

Source units

Reportable units
Other enhancements

• Generic plug-in interface
• Configurable set of modules iteratively invoked against each object
• Common data structure passed between modules to enable stateful processing
• Identification de-coupled from validation
• Standardized handling of format profiles and error reporting
• Symbolic display of binary formats
• API-level support for editing
Data abstraction

• Based on the “natural” conceptual structures of a format and their component attributes
  – Each such structure maps to a class with methods for parsing, validating, reporting, and serializing
  – Each such attribute maps to a field with accessor and mutator methods

  • UTF-8 ⇒ Character
  • TIFF ⇒ Image File Header and Image File Directory
  • JPEG 2000 ⇒ Box
  • PDF ⇒ boolean, number, string, name, array, dictionary, and stream
Format support

• Based on project partner requirements and budgetary constraints
  – *Image:* JPEG 2000, TIFF
  – *Audio:* WAVE
  – *Text:* SGML, UTF-8, XML
  – *Document:* PDF
  – *GIS:* Shapefile
  – *Color:* ICC
  – And their well-known variants, e.g. TIFF/IT, TIFF/EP, GeoTIFF, EXIF, DNG, …

• Unfortunately precluding some JHOVE-supported formats
  – AIFF, GIF, HTML, JPEG
Technical components

• Java 1.5
  – java.nio package

• OSGi/Spring frameworks
  – Component versioning and dependency management
  – Fine-grained control of component invocation
  – Inversion of control

• SourceForge
  – Distribution platform
  – Issue tracking
Schedule

- Months 1-6  Outreach, design, and prototyping
- Months 7-9  Core APIs and framework
- Months 10-24 Module implementation
Advisory board

- Deutsche Nationalbibliothek
- Ex Libris
- Fedora Commons
- Florida Center for Library Automation
- Harvard University
- Koninklijke Bibliotheek
- MIT/DSpace
- National Archives (UK)
- National Library of Australia
- National Library of New Zealand
- Planets project
Questions?

Wiki  confluence.ucop.edu/display/JHOVE2Info/Home
Mailing lists  JHOVE2-Announce-L
             JHOVE2-Techtalk-L

(Subscribe via the wiki)