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

Stephen Abrams*, Sheila Morrissey, Tom Cramer*****

*California Digital Library
University of California
415 20th Street
Oakland, CA 94612, US
Stephen.Abrams@ucop.edu

**Portico
100 Campus Drive
Princeton, NJ 08450, US
Sheila.Morrissey@portico.org

***Stanford University
314 Meyer Library
Stanford, CA 94305, US
tcramer@stanford.edu

Abstract

The JHOVE characterization framework is widely used by international digital library programs and preservation repositories. However, its extensive use over the past four years has revealed a number of limitations imposed by idiosyncrasies of design and implementation. With funding from the Library of Congress under its National Digital Information Infrastructure Preservation Program (NDIIPP), the California Digital Library, Portico, and Stanford University are collaborating on a two year project to develop and deploy a next-generation architecture providing enhanced performance, streamlined APIs, and significant new features. The JHOVE2 project generalizes the concept of format characterization to include identification, validation, feature extraction, and policy-based assessment. The target of this characterization is not a simple digital file, but a (potentially) complex digital object that may be instantiated in multiple files.

Introduction

Digital preservation is the set of intentions, strategies, and activities aimed at ensuring the continuing usability of digital objects over time. However, since digital objects rely on explicit technological mediation in order to be useful, they are inherently fragile with respect to technological change. Over any significant time period, a gap inevitably arises in the ability of a digital object to function in contemporaneous technological contexts. Put most simply, digital preservation is concerned with effectively managing the consequences of this gap, which is achievable only to the extent to which the gap is quantifiable. The necessary quantification comes, in part, from characterization.

Characterization exposes the significant properties of a digital object and provides a stable starting point for iterative preservation planning and action, as shown in Figure 1 (Brown 2007). Characterization is particularly pertinent to any significant transformative process. The comparison of an object's pre- and post-transformation properties is a valuable mechanism for quantifying potential transformative loss. In this scenario, the characterization data functions as a canonical representation or surrogate for the object itself (Lynch 1999).

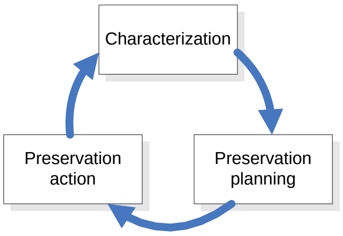

Figure 1. Iterative preservation cycle,
adapted from (Brown 2007).

While manual characterization is possible, it is tedious and error prone and requires highly trained staff. Preservation characterization can only be effective at scale through automated efforts (Green and Awre 2007). The original JHOVE framework was developed to provide comprehensive characterization functionality for use in automated systems and workflows (Abrams 2003).

JHOVE was a collaborative project between the Harvard University Library and the JSTOR Electronic-Archiving Initiative (now called Portico) with funding from the Andrew W. Mellon Foundation. (More information is available at http://hul.harvard.edu/jhove/.) It has found wide acceptance by the international digital library and preservation communities. However, its extensive use over the past four years has revealed a number of limitations imposed by idiosyncrasies of design and implementation. With funding from the Library of Congress under its National Digital Information Infrastructure Preservation Program (NDIIPP), the California Digital Library, Portico, and Stanford University are collaborating on a two year project to develop and deploy JHOVE2, a next-generation architecture providing enhanced performance, streamlined APIs, and significant new features.

Characterization

The description of the original JHOVE framework used the terms *identification*, *validation*, and *characterization* to denote independent concepts. In the context of the JHOVE2 project there has been a shift in terminology under which *characterization* is now defined generically as the totality of description about a formatted digital object, encompassing four specific aspects:

- *Identification.* Identification is the process of determining the presumptive format of a digital object on the basis of suggestive extrinsic hints (for example, an HTTP Content-type header) and intrinsic signatures, both internal (a magic number) and external (a file extension). Ideally, format identification should be reported in terms of a level of confidence.

- *Validation.* Validation is the process of determining a digital object's level of conformance to the requirements of its presumptive format. These requirements are expressed by the normative syntactic and semantic rules of that format's authoritative specification.

  Ideally, the determination of conformance should be based on commonly accepted objective criteria. However, many format specifications – particularly those not created as part of explicit standardization efforts – suffer from ambiguous language requiring subjective interpretation. The incorporation of such interpretative decisions into automated systems should be highly configurable to support local variation of preservation policy and practice.

- *Feature extraction.* Feature extraction is the process of reporting the intrinsic properties of a digital object significant to preservation planning and action. These features can function in many contexts as a surrogate for the object itself for purposes of evaluation and decision making.

  Note that since digital preservation is concerned with planning for future activities, potentially in response to unforeseeable circumstances, predicting which properties will one day be significant can be problematic. Prudence therefore suggests reporting the most inclusive set of properties possible, while providing sufficiently fine granularity of control to allow for appropriate localized configuration.

- *Assessment.* Assessment is the process of determining the level of acceptability of a digital object for a specific use on the basis of locally-defined policies. Assessments can be used to select appropriate processing actions. In a repository ingest workflow, for example, the range of possible actions could include rejection, normalization, or acceptance in original form.

Reduced to simpler terms, characterization answers the following questions relevant to the preservation of a digital object:

- What is it?
- What is it really?
- What are its salient characteristics?
- What should be done with it?

Or even more reductively, What? and So what?

The JHOVE2 Project

The high-level goals of the JHOVE2 project are three-fold:

- To *refactor* the existing JHOVE architecture and APIs to increase performance, simplify integration, and encourage third-party maintenance and development.

- To provide significant *enhancements* to existing JHOVE functionality to increase its utility to preservation practitioners and workflows.

- To develop JHOVE2 *modules* supporting characterization of a variety of digital formats commonly used to represent audio, geospatial, image, and textual content.

Redesign and Implementation

While JHOVE was implemented in Java 1.4, it used the older stream-style I/O of the standard *java.io* package. JHOVE2 will use the buffer-based NIO package, which has the potential for significantly higher performance through the use of memory mapped I/O (Hitchens 2002).

Although all JHOVE modules implement the same Module interface, and thus share a common method signature, their internal coding is not always similar. Understanding the construction details of one module is not necessarily helpful in understanding the internals of any other module. In order to provide a greater level of conceptual and practical uniformity of implementation, the JHOVE2 design process will establish common design patterns to which all modules will adhere (Fowler 2006). These patterns will also facilitate the integration of individual modules into other systems independent of the core JHOVE2 framework.

The intention of the JHOVE2 project is to continue to provide all existing JHOVE functionality – although implemented in the context of the new framework and APIs – while adding a number of significant new features. The new JHOVE2 code base will be released under the BSD open source license.

More Sophisticated Data Model

JHOVE was designed and implemented with the implicit assumption that a single digital object was equivalent to a single digital file in a single format:

1 object = 1 file = 1 format

(While not strictly true of all modules, the few exceptions to this assumption were dealt with idiosyncratically.) There are, of course, many important examples for which this assumption is not true. For example, a TIFF file encapsulating an ICC color profile and XMP metadata. While still a single object and file, there are essentially three formats (TIFF, ICC, and XML/RDF):

1 object = 1 file = 3 formats

The JPEG 2000 JPX profile defines a fragmentation feature in which an encoded image can be manifest in an arbitrary number of individual files:

1 object = *n* files = 1 format

The ESRI Shapefile constitutes a single object that is always manifested by three files, each with its own format:

1object = 3 files = 3 formats

JHOVE2 data modeling will support the general case of an object manifested by an arbitrary number of component files and formats:

1 object = *n* files = *m* formats

From another perspective, however, these kinds of multi-file aggregates can be considered to constitute high-level formats in their own right. For purposes of the JHOVE2 project *format* is defined expansively as a class of objects sharing a common set of syntactic and semantic rules for mapping from abstract information content to serialized bit streams (Abrams 2007). Thus, a page-turning format could be defined consisting of METS descriptive and structural metadata, TIFF master and JPEG delivery page images, and OCR text files:

1 object = 1 + 4*n* files = 1 format

Conceptually, there is no meaningful difference between the traversal of a nested container file – for example, the TIFF with embedded profile and metadata described previously – and a multi-file, multi- directory file system hierarchy. A JHOVE2 module could be developed that would start its recursive parsing at the root “page-turning format” level. As the traversal encounters each lower-level component (image files, OCR files, etc.), JHOVE2 would automatically invoke the appropriate format-specific parser.

In order to support the new concept of arbitrary recursive parsing of complex object formats, three types of identification are needed:

- Identification of the format of *files* based on internal and external signatures.
- Identification of the format of *bit streams* – proper subsets of files – based on internal signatures.
- Identification of the format of *objects* instantiated in multiple files – in other words, a PREMIS representation – based on signatures defined in terms of file-level characteristics and structural relationships.

  For example, a Shapefile object can be presumptively identified whenever three sibling files – that is, existing within the same directory – share a common filename stem but have the extensions *dbf*, *shp*, and *shx*, respectively:

```
abcd/
    1234.dbf
    1234.shp
    1234.spx
```

While object- and file-level identification can occur independent of the parsing necessary for validation and feature extraction, bit stream identification will occur only during the parsing stage.

Generic Plug-in Mechanism

All JHOVE plug-in modules perform the same function – validation and feature extraction – and only a single module is invoked against each digital object. JHOVE2 will implement a more generic processing model in which a configurable sequence of modules, each capable of performing an arbitrary function, can be invoked against each object (see Figure 2). A persistent memory structure for representation information, as defined by the OASIS reference model, will be passed between modules (ISO 2003). Since a given module in the sequence will have access to the results of all subsequent modules, it will be possible to define sophisticated stateful processing flows.

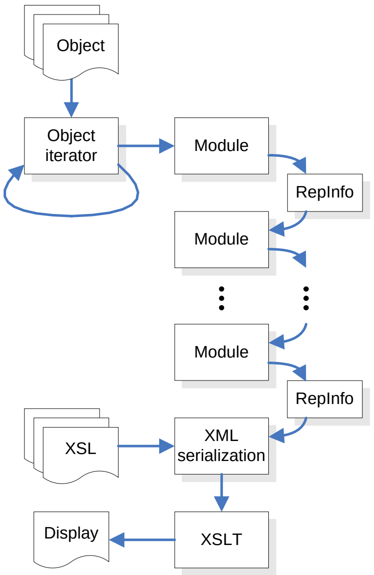

Figure 2. Processing flow.

De-Coupling Identification from Validation

JHOVE performs identification of a digital object’s format by iteratively invoking all configured modules until one reports the object to be valid. Since JHOVE validation is rigorous, this makes identification extremely reliable. However, this benefit is outweighed by the fact that *any* validation error, no matter how trivial, will cause JHOVE to iterate to the subsequent module. Thus, JHOVE will identify a damaged object as, say, a valid bytestream rather than an invalid PDF, which, while technically correct – by definition, *all* objects are valid bytestreams – is not particularly useful in most preservation contexts.

JHOVE2 will de-couple the identification and validation operations. Identification will be performed on the basis of matching file-level characteristics and internal and external signatures. The working assumption is that DROID will be used for file- and bit stream-level identification (Brown 2006).

Standardized Profile and Error Handling

JHOVE modules exist at the granularity of format *families*, but can recognize and distinguish between the many variant formats, or profiles, of the family. For example, the TIFF (Tagged Image File Format) family encompasses a number of specific profiles possessing differences significant in many preservation contexts, such as TIFF/EP, TIFF/IT, GeoTIFF, EXIF, DNG, etc. While at a functional level JHOVE modules provide equivalent handling of profiles, each module's implementation of this function is somewhat idiosyncratic. JHOVE2 will introduce standardized patterns of module design for dealing with profiles in a common and easily extended manner.

Module error handling in JHOVE is similarly idiosyncratic. Again, JHOVE2 will introduce a standardized pattern of error handling with more precise error messages using terminology and references drawn from the appropriate specification documents.

Customizable Reporting

JHOVE is distributed with two output handlers: a Text handler that formats output in terms of simple mail or HTTP header-like name/value pairs, and an XML handler that produces output in terms of a JHOVE-specific container schema. JHOVE2, on the other hand, will always produce an intermediate XML output using a standard METS container schema, which can then be customized through XSL stylesheet transformations to any desired form (Cundiff 2004; Clark 1999). The METS <*StructMap*> mechanism will be particularly useful to model the arbitrary parent-child and sibling structural relationships permitted by the new JHOVE2 object modeling.

The JHOVE2 distribution will include standard stylesheets generating JHOVE-style Text and XML output so that JHOVE2 can easily replace JHOVE in existing workflows dependent upon the specific output form. As with JHOVE, JHOVE2 will report format-specific properties and other important representation information using well-known public schemas such as NISO Z39.87 for raster still images and the forthcoming AES-X098B for audio content (NISO 2006; AES 2008). In addition, the PREMIS schemas will be used for reporting event information and other general preservation metadata (Guenther and Xie 2007).

Modules

Like its predecessor, JHOVE2 will be based on an extensible plug-in framework. Since it is hoped that module development will also occur outside of the context of the JHOVE2 project it is important that JHOVE2 is based on a flexible and robust platform for module integration. The JHOVE2 project will explore the use of the OSGi (Open Services Gateway initiative) and Spring frameworks for this purpose. OSGi provides robust facilities for Java class loading and life cycle management particularly pertinent for integrating components produced in a decentralized environment (OSGi Alliance 2007). The Spring framework provides a number of functions again useful for simplifying the integration and configuration of disparate components based on the Inversion of Control (IoC) or Dependency Injection paradigm (Johnson et al. 2008).

Module function will include signature-based identification, validation, feature extraction, and assessment. JHOVE2 will also support the humanly-readable display in symbolic form of the contents of binary formatted objects. In JHOVE this functionality was provided in the form of stand-alone utility applications, *j2dump* (for JPEG 2000), *tdump* (for TIFF), etc. In JHOVE2 these functions will be incorporated into the main body of the code. Other function includes API-level support for editing and serializing formatted objects, useful for example to correct existing internal metadata or to embed additional metadata in a syntactically correct manner. It is important to note, however, that an out-of-the-box object editing capability is *not* a project deliverable. JHOVE2 will be an enabling technology for the subsequent development of a number of added-value systems and services, but the development of such products is outside the scope of currently funded JHOVE2 activities.

JHOVE2 will introduce a standard design pattern or template for plug-in modules. This will be based on the "natural" conceptual structures of a given format and their constituent attributes. Each such structure will be mapped to a Java class with methods for parsing, validating, reporting, and serializing; each such attribute will be mapped to a class instance field with appropriate accessor and mutator methods. For example, the major conceptual structures for the TIFF format are the *Image File Header* (IFH) and *Image File Directory* (IFD); for JPEG 2000, the structure is the *Box*; for PDF, the object types *boolean*, *number*, *string*, *name*, *array*, *dictionary*, and *stream*.

Compatibility

As discussed previously, JHOVE2 modules will replicate and extend existing JHOVE functionality. However, due to the nature of the newly proposed features it may not be possible to maintain backwards compatibility with existing JHOVE modules. Compatibility of output will be maintained, however, to the fullest extent possible.

JHOVE2 format identification will be possible for all formats known to the identification module. Presuming the use of DROID, this includes some 580 formats currently documented in the PRONOM database; if the signature database is extended to include the Unix magic number database (*/etc/magic*, the basis for the *file* command shell utility), the scope of identification can be extended to over 1000 formats. Detailed validation and feature extraction, on the other hand, is only available for formats for which there are explicit JHOVE2 validation/ feature extraction modules.

The JHOVE2 project will provide modules for new formats not supported by JHOVE, including ICC profile, SGML, and Shapefile (ICC 2004; ISO 1986; ESRI 1998). However, budgetary constraints will not permit the reimplementation of all formerly-supported formats;

in particular, modules for AIFF, GIF, HTML, and JPEG are *not* included among project deliverables. It is hoped that subsequent funded activity by project partners or other institutions will quickly remedy these omissions. The remaining JHOVE-supported formats – ASCII, JPEG 2000, PDF, TIFF, UTF-8, WAVE, and XML – will be supported in JHOVE2.

Assessment

One major new function introduced in JHOVE2 is digital object assessment based on locally-defined rules and heuristics. Risk assessment lies at the heart of the preservation decision making process: How can one determine whether a given digital object is approaching incipient obsolescence? What are the factors that make an object susceptible to loss and how can they be quantified? How can an object be evaluated for acceptability under local policy rules? JHOVE2 assessment will be performed by the evaluation of locally-defined rules in the context of prior characterization information. Assessment decisions can be used, for example, to assign appropriate repository service levels, or as factors driving business rules engines to trigger preservation events such as migration (Ferreira, Baptista, and Ramalho 2007; LeFurgy 2002; Pearson and Webb 2007).

The quantitative data necessary to perform such analyses are provided by prior JHOVE2 characterization. Assessment can therefore be seen as the next logical step in a JHOVE2 processing chain:

> Identification → Validation → Feature Extraction → Assessment → Disposition → …

The JHOVE2 project will investigate existing assessment methodologies and rules, and the means by which they can be codified into best practices and expressed in a highly-configurable, machine-actionable manner (Anderson et al. 2005; Arms and Fleischhauer 2005; Stanescu 2005; van Wijk and Rog 2007).

Schedule

The JHOVE2 project will run for two years. Broadly speaking, the schedule will proceed through three phases:

- Consultation and design (6 months)
- Core framework and APIs (6 months)
- Module development (12 months)

To facilitate communication with and review by important stakeholder communities, the JHOVE2 project will empanel an Advisory Board recruited from leading international preservation institutions, programs, and vendors. Board members will be asked to serve in three capacities: as representatives of the needs of their respective organizations; as proxies for the wider cultural and scientific memory communities; and as independent professional experts.

The capabilities of JHOVE2 described in this paper represent the intentions and plans of the project team at the time of writing. These may evolve, especially during the initial stakeholder consultation period, in order to better serve the needs of the JHOVE2 user community.

More information about the JHOVE2 pis available at the project wiki, http://confluence.ucop.edu/display/JHOVE2Info/Home.

Conclusion

An understanding of format is fundamental to the long-term preservation of digital objects. While it is possible to preserve digital objects as opaque bit streams without consideration of their format, the end result is merely well preserved bits. In order to recover the information content encoded into those bits requires knowledge of the syntactic and semantic rules governing that encoding, in other words, their format (see Figure 3).

```
ffd8ffe000104a46     SOI
4946000102010083     APP0 JFIF 1.2
00830000ffed0fb0     APP13 IPTC
50686f746f73686f     APP2  ICC
7020332e30003842     DQT
494d03e90a507269     SOF0  183x512
6e7420496e666f00     DRI
0000007800000000     DHT
0048004800000000     SOS
02f40240ffeeffee     ECS0
0306025203470...     ...
```

Syntax Semantics Content

Figure 3. Format-directed mapping from JPEG bit stream to humanly-interpretable image content (Burne-Jones 1870-1876). The example image is copyright by the President and Fellows of College Harvard.

The operations of object identification, feature validation, extraction, and assessment lie at the heart of many digital preservation activities, such as submission, ingest (see Figure 4), monitoring, and migration (Figure 5). JHOVE2 will provide a highly configurable, extensible, and functional framework for performing these important operations. Note that Figure 4 shows the deployment of characterization function on both the client *and* server sides of the ingest workflow. The use of JHOVE2 as far upstream as possible in the content lifecycle increases the overall efficiency of preservation activities by facilitating the initial creation of born-preservation amenable content.

JHOVE2 will provide performance improvements and significant new features, most notably, a flexible rules-based assessment capability. The parsing of digital objects underlying JHOVE2 operations will be capable of a recursive traversal of file systems and arbitrarily nested bit streams within files. The revised core framework and APIs will facilitate third-party development and simply the integration of JHOVE2 characterization functionality into existing systems, services, and workflows. The more that JHOVE2 functionality can be dispersed into other open source products and mainstream applications, the more it will benefit from a broader community of use and support.

The JHOVE characterization system has been widely adopted by the international digital memory community.

A number of lessons have emerged from the feedback received from this community. Most significantly, it is now clear that characterization plays a fundamental role in preservation workflows. The JHOVE2 team is very excited to have the opportunity to build on the rich body of prior experience and solidify the foundations for future digital preservation efforts. Through the active input and participation of its stakeholder community, JHOVE2 will remain a central and viable component of preservation infrastructure.

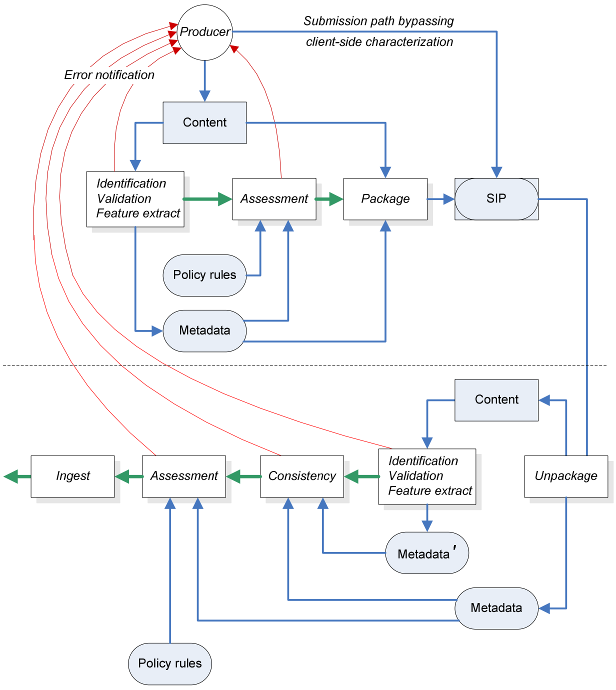

Figure 4. Generic ingest workflow incorporating characterization, adapted from (Abrams 2007).

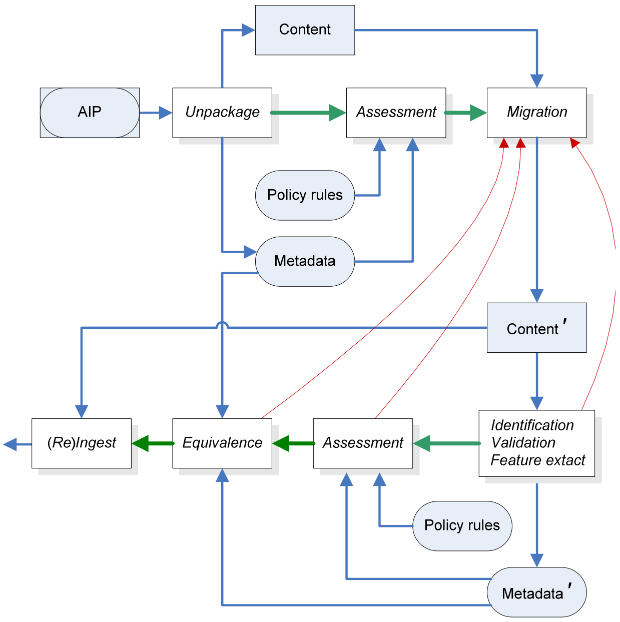

Figure 5. Generic migration workflow incorporating characterization.

Acknowledgements

The JHOVE2 project is funded by the Library of Congress as part of its National Digital Information Infrastructure Preservation Program (NDIIPP). The authors would like to acknowledge the contributions of Justin Littman, Library of Congress program officer, and the JHOVE2 project team: Patricia Cruse, John Kunze, Joan Starr, Hunter Stern, and Marisa Strong at the California Digital Library; John Meyer and Evan Owens at Portico; and Richard Anderson, Hannah Frost, Walter Henry, Nancy Hoebelheinrich, and Keith Johnson at Stanford University.

References

Abrams, S. 2003. Digital Object Format Validation. *Digital Library Federation Fall Forum*, Albuquerque, November 17-19.

Abrams, S. 2007. File Formats. *DCC Digital Curation Manual*.

AES. 2008. Report of the SC-03-06 Working Group on Digital Library and Archive Systems of the SC-03 Subcommittee on the Preservation and Restoration of Audio Recording Meeting.

Anderson, R., Frost, H., Hoebelheinrich, N., and Johnson, K. 2005. The AIHT at Stanford University: Automated Preservation Assessment of Heterogeneous Digital Collections. *D-Lib Magazine* 11(2), December.

Arms, C., and Fleischhauer, C. 2005. Digital Formats: Factors for Sustainability, Quality, and Functionality. *IS&T Archiving Conference*.

Brown, A. 2006. Automated Format Identification Using PRONOM and DROID. Technical Paper DPTP-1, Issue 2, March 7.

Brown, A. 2007. Developing Practical Approaches to Active Preservation. *International Journal of Digital Curation* 2(1): 3-11.

Burne-Jones, E. 1870-1876. The Days of Creation: The First Day. Harvard University Art Museums, 1943.454. JPEG image, http://via.lib.harvard.edu/via/deliver/deepLinkItem?recordId=HUAM303460&componentId=HUAM:51430_mddl.

Clark, J., ed. 1999. XSL Transformations (XSLT). Version 1.0, W3C Recommendation, November 16.

Cundiff, M. 2004. An Introduction to the Metadata Encoding and Transmission Standard (METS). *Library Hi Tech* 22(1): 52-64.

ESRI. 1998. ESRI Shapefile Technical Description. July.

Ferreira, N., Baptista, A., and Ramalho, J. 2007. An Intelligent Decision Support System for Digital Preservation. *International Journal on Digital Libraries* 6(4): 295-304.

Fowler, M. 2006. Writing Software Patterns. Web site, www.martinfowler.com/articles/writingPatterns.html, accessed August 9, 2008.

Green, R., and Awre, C. 2007. RepoMMan Project: Automatic Generation of Object Metadata, Technical Report D-D13, Version 1.1, October.

Guenther, R., and Xie, Z. 2007. Implementing PREMIS in Container Formats. *IS&T Archiving Conference*.

Hitchens, R. 2002. *Java NIO*. Sebastopol: O'Reilly.

ICC. 1:2004-10. 2004. Image technology colour management – Architecture, profile format, and data structure. Version 4.2.0.0.

ISO 14721. 2003. Space data and information transfer systems – Open archival information system – Reference model.

ISO 8879. 1986. Information processing – Text and office systems – Standard Generalized Markup Language (SGML).

Johnson, R., et al. 2008. The Spring Framework – Reference Documentation.

LeFurgy, W. 2002. Levels of Service for Digital Repositories. *D-Lib Magazine* 8(2), May.

Lynch, C. 1999. Canonicalization: A Fundamental Tool to Facilitate Preservation and Management of Digital Information. *D-Lib Magazine* 5(9), September.

NISO Z39.87. 2006. Data Dictionary – Technical Metadata for Digital Still Images.

OSGi Alliance. 2007. About the OSGi Service Platform. Technical Whitepaper, Revision 4.1, June 7.

Pearson, D., and Webb, C. 2007. Defining File Format Obsolescence: A Risky Journey. *3rd International Digital Curation Conference*. Washington, December 11-13.

Stanescu, A. 2005. Assessing the Durability of Formats in a Digital Preservation Environment: The INFORM Methodology. *OCLC Systems & Services* 21(1): 61-81.

van Wijk, C., and Rog, J. 2007. Evaluating File Formats for Long-term Preservation. *4th International Conference on Preservation of Digital Objects*. Beijing, October 11-12.