I T H A K A

JSTOR | PORTICO | ITHAKA S+R

Portico

ConPrep System Overview

17th December 2010

Our Approach to E-Journal Archiving

I T H A K A

- Source file archiving
  - Preserve the components not the rendition
  - Include high-resolution files (PDF and figures) if available
  - All e-only components (data, media, etc.)
  - SGML / XML structured text by preference
    - HTML as last resort
- Preserve intellectual content not "look and feel" of HTML
  - HTML renditions are an artifact of current technology
    - Often dynamically generated
    - Fragile technology, overdue for change
- Preserve only essential features of the user interface
  - Reference linking, other content-based features
  - Not generic navigation or search or e-commerce features
- Why this approach?
  - Based on Mellon-funded study by Harvard University Library
  - Based on practical realities of works with multiple manifestations
  - Based on assessment as to instability of current web technologies

PORTICO

Our Approach to Long-Term Preservation

ITHAKA

- Format-based migration strategy
  - Driven by Portico Format Registry
- Preservation policies:
  - Fully supported
  - Reasonable effort
  - Byte-preserve only
- Preservation policies based on
  - Format validity
  - File format action plans and archive capabilities
  - Business rules such as publisher preferences
- Archive must also preserve supporting information
  - Required files such as DTDs and entity files
  - Documentation
  - Contracts
  - Archive policy documents
  - Archival actions documents

Key Challenges

ITHAKA

- Diversity of incoming data streams
  - Lack of a packaging standard
- Automating identification, classification, validation of formats
  - Metadata harvesting
- Normalization of proprietary data formats to Archival DTD
  - No industry standard article DTD
- Large number of very small files
- Building a system that can manage non-trivial intervention in the content prior to archiving and preserve the record of the source data, the normalized data, and everything that happened during the normalization
- A big step toward managing future migrations!

Key Architectural Goals

ITHAKA

- Pluggable tools to facilitate new providers and replacement tools
- Configurable workflows to add new business flows and content types Elsevier checksums.
- Clear and clean separation of process view of content model from structural view
- Scalable to very high content volumes

PORTICO

System Components

ITHAKA

- Workflow
  - Per content type (E-Journals, Business artifacts, Technical artifacts)
  - New and updated content
- Profiles (per provider)
  - Provider-specific rules and policies
  - Packaging rules
  - File name extract rules
- Format registry
  - List of formats known to the archive
  - Links to policy documents, technical documentation, and "required files"
- Tools registry & Tools service
  - What tools for which formats?
  - Where are they located?
  - How are they invoked?

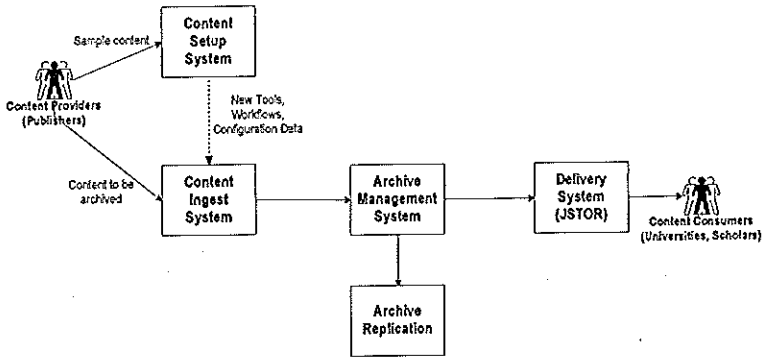
DTD to filter. (local copy?

Joe

PORTICO

Portico Systems Overview

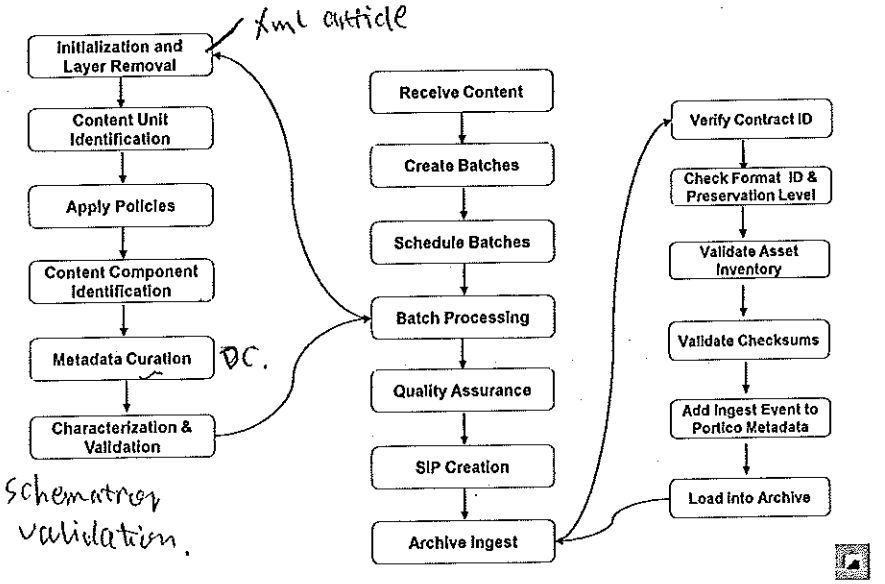
ITHAKA



PORTICO

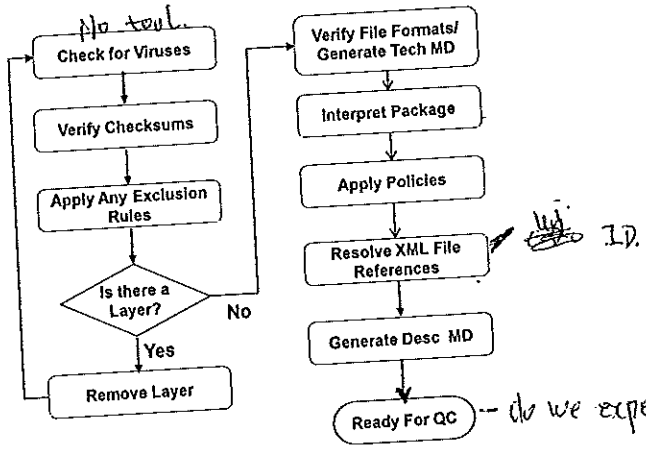
General Portico Workflow

ITHAKA



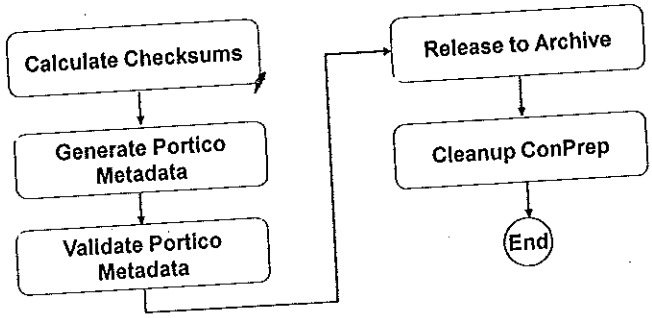
PORTICO

Batch Processing Detail



SIP CreationDetail

reserve Publisher Source data as it is.



Provider Submission Profile

- Captures provider-specific file naming conventions, directory hierarchies, and processing characteristics including rules
- Defines pattern rules based on regular expressions
- Key mechanism to externalize provider-specific behavior from software
- Maintained on a per-provider basis
- Assigned to batches at the time of submission; however can be changed later during QC

Format Registry

- File formats for assets
  - Page Images (e.g., PDF)
  - Graphics
  - SGML & XML
- Each DTD or schema version is a separate format
- File formats for metadata ("metadata is data")
  - XML schemas
- Unique verbose names for all formats recognized by Portico
  - "Recognized" not "supported" or "preserved"
- Current design based on preliminary designs for GDFR
  - By Stephen Abrams of Harvard Library

Format Registry Implementation

- An XML-based registry consulted to identify, validate, characterize, and render various format instances. An instance of the *Portico Format Registry schema*.
- One registry shared by all systems and applications – ConPrep, Archive Management, and Distribution
- Flexible design to "accommodate" future GDFR initiative
- Information on tools/services used to manipulate formats, and policies enforced to preserve formats over time, are in separate registries

13

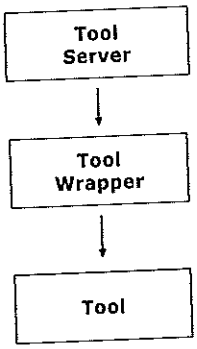
Portico Tools Services

- Format-neutral services:
  - Virus check (ClamAV)
  - Checksum (various)
  - Identification (JHOVE, BSD file; returns a format ID and/or MIME type)
- Format- or MIME type-specific services:
  - Validation (JHOVE)
  - Characterization (JHOVE)
  - Layer removal (e.g., unzip)
  - Transformation (XSLT; per source format and destination format)
- DTD-Specific XML services:
  - Descriptive metadata extraction (XSLT)
  - HTML rendition (XSLT)
  - Descriptive metadata curation (java & XSLT)
  - File reference extraction (XSLT)
  - File reference replacement (XSLT)
  - QC errors & warnings (Schematron)
- And more to come

14 Portico / Ontario Scholars Portal Meeting

Tool Registry & Services Implementation

- Registry provides information about tools utilized to process content
- Registry does not know whereabouts of tools or itself offer services
- Loose coupling of tool and format registries to facilitate independent evolution



- Dispatcher that listens for requests; upon arrival, spawns a worker thread to process
- Adapter that hides tool-specific behavior and converts tool-specific interface to tool-neutral interface
  - e.g., maps specific return values to standard values
- A COTS product, open-source, or custom software that provides a specific service
  - e.g., JHOVE, ClamAV, gzip

A Major Issue: Varying Degrees of Badness

- What format is a defective file?
  - The purported format? The actual format?
  - Format "Re-identified" (a business concern as well as technical)
- Can a file be damaged but still usable?
  - XML: No, we have to have valid XML file to extract metadata!
  - PDF: Yes, Acrobat reader can read some WFNV or NWF PDF?
- On what do you base the preservation policy for a bad file?
  - The actual format?
  - Best-effort on purported format?
  - What about well-formed but not valid?
- Some use cases:
  - Defective file (varying degrees)
  - Purported format is in error (e.g. wrong extension)
  - Both of the above

Verification / Identification Sequence

To distinguish between bad files and mislabeled files:

- Verify purported format (MIME type)
- If verification succeeds
  - Record format
  - Capture technical metadata
- If verification fails, do identification
- If identified format is same as purported format
  - File is bad
- If identified format is not same as purported format
  - Might be mislabeled
- Verify identified format
  - If fails again, file is bad

17

Portico Content Model

- Based loosely on MPEG-21 Concepts
  - Not MPEG-21
  - Not METS
  - Developed as "Ithaka Configurable Repository" R&D project
- Key requirements
  - Support for complex objects such as E-Journal articles
  - Flexible enough for future content
  - Treat metadata as first class asset
    - "Metadata is data"
  - Classify assets by function as well as format
    - To support function-based migration strategies
    - To support complex objects

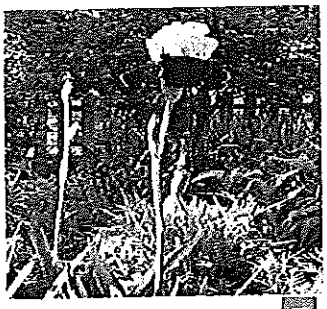
18 Portico / Ontario Scholars Portal Meeting

Portico Content Model Key Concepts

- Content Type
  - Broad division into business lines or genres
- Content Set
  - Divisions within a Content Type
- Content Unit
  - Single intellectual unit of archived content
- Functional Unit
  - One or more files with same intellectual identity and functional type:
    - Renditions: Page, Web
    - Text: Full, Header
    - Components: Graphics, Media, Other
    - Metadata Records (Portico Metadata)
- Storage Unit
  - The preserved file

Content Model Example: Imaginary Photo Album Archive

- Content Type: Digital Photo Albums
- Content Set: Evan's Chicago Garden Album
- Content Unit: Iris Germanica (variety unknown)
- Functional Unit: Image
  - Storage Unit: Pic1.jpg
  - Storage Unit: Pic1.tiff
  - Storage Unit: Pic1.gif
- Functional Unit: Metadata Record
  - Storage Unit: Pic1.pmets



Content Model Example: E-Journals

ITHAKA

- Content Unit: One E-Journal article
- Functional Unit: Full Text
  - Storage Unit: SGML file (normalized & inactive)
  - Storage Unit: XML
- Functional Unit: Print Rendition
  - Storage Unit: PDF file
- Functional Unit: Component: Image
  - Storage Unit: Figure1.jpg
  - Storage Unit: Figure1.gif
- Functional Unit: Metadata record
  - Storage Unit: Portico Metadata file