The Hydraulics Project

Empowering Communities to Build a Digital Library Utilizing Fedora and an Event-Driven Service-Oriented Messaging Framework

Andrew Curley
University of Virginia Library
Open Repositories 2011
What is Hydraulics?

End-to-end digitization workflow management tool that integrates:

• Request module
• Management system for digital production
• Archive management workflows
• Patron delivery workflows
• Fedora ingestion workflows
Hydraulics as Connection

Hydraulics serves as the canonical source by facilitating communication with Blacklight, Fedora and an archival filesystem.
Hydraulics vs. Tracksys

Hydraulics
- Generalized, open-sourced, derived from Tracksys

Tracksys
- UVA implementation, tied to local practices and needs

Hydraulics : Tracksys :: Blacklight : VIRGO
Hydraulics Stack

Ruby on Rails

- ActiveMessaging gem

Fedora

- 3.4.2 REST API Compatible

Solr

ActiveMQ
The Hydraulics Model

Customer places request → Request Vetted by Staff → Records Created in Hydraulics

Physical Item Scanned → Metadata Creation → Manual Quality Assurance

Production Workflow Outside Hydraulics → Automated Workflows Within Hydraulics

Patron Delivery Workflow → Intended Use Not Digital Collection Building

Digital Library Delivery Workflow

All items → Archiving Workflow

Include in DL is set → Automated Quality Assurance
Request Module

Public facing request form integrates with underlying database

Distinguish internal and external patrons

Facilitates engagement between Special Collections and Preservation/Conservation staff and patron

Allows librarians to link digital objects with canonical metadata

Fee and billing management
Request Form: User Information

Integration with local LDAP

Populate with existing information

Request can be made on behalf of another person

Billing address information
Request Form: Bibliographic Info

Multiple free text fields

Flexible, non-mapped fields

Intended to guide staff in locating material

Based on years of experience handling patron request of Special Collections materials
Request Form: Intended Use

Intended uses govern type of deliverable given to patron

Guide user in choosing their deliverable type rather than explicitly asking for technical specifications

Two kinds of deliverables:
- 300dpi watermarked JPEG
- Highest possible dpi TIF
Request Approval: Bibliographic Records

Integration with Blacklight

Association accomplished by catalog key and/or barcode

Barcode is only unique value!

Bibliographic records must be “item” records; local practice is to make “manifestation” records, with “item” information in MARC 999 field
Request Approval: Routing Slip

Printed by staff after order vetting is complete

Provides workflow template and order metadata to production staff

Allows for notes or problems to be passed along throughout production

Allows production coordinators to manage priorities at a glance
Production Workflow

Workflows local to institutions, based on available equipment, computing and software

UVA Digitization Services uses:
- Phase One Capture One DB
- MS Expression Media (soon-to-be Phase One Media Pro, also formerly Iview)

Expression Media used for quality assurance and metadata entry
Production Workflow: Metadata Entry
Production Workflow: Metadata Export

<MediaItem>
  <AssetProperties>
    <Filename>000001483_0134.tif</Filename>
    <Filepath>000001483:000001483_0134.tif</Filepath>
    <UniqueID>427</UniqueID>
    <Label>0</Label>
    <Rating>0</Rating>
    <MediaType>TIFF</MediaType>
    <FileSize unit="Bytes">48510315</FileSize>
    <Created>2008:08:27 15:08:47</Created>
    <Modified>2008:11:13 15:29:34</Modified>
    <Added>2008:08:27 15:18:45</Added>
    <ThumbnailSource>000001483_0134.jpg</ThumbnailSource>
  </AssetProperties>

  <MediaProperties>
    <Width unit="Pixels">3320</Width>
    <Height unit="Pixels">4862</Height>
    <Resolution unit="DPI">600</Resolution>
    <Depth unit="Bits">24</Depth>
    <ViewRotation>1</ViewRotation>
    <SampleColor>R:D0 G:D0 B:C0</SampleColor>
    <Pages>1</Pages>
    <ColorSpace>RGB</ColorSpace>
    <Compression>65537</Compression>
    <PrimaryEncoding>TIFF (Uncompressed)</PrimaryEncoding>
    <ColorProfile>Adobe RGB (1998)</ColorProfile>
  </MediaProperties>

  <AnnotationFields>
    <Headline>119</Headline>
    <Author>Phillips, David Graham</Author>
    <Credit>Taylor 1917 .P55 S8</Credit>
    <Source>Susan Lenox, her fall and rise, with a portrait of the author</Source>
    <Location>SC-STKS</Location>
  </AnnotationFields>

  <MetaDataFields>
    <Maker>Phase One</Maker>
    <Model>P 45+</Model>
    <Software>Adobe Photoshop CS2 Macintosh</Software>
    <SourceURL></SourceURL>
    <ExifVersion>2.2</ExifVersion>
    <CaptureDate>2008:08:22 13:29:23</CaptureDate>
    <ISOSpeedRating>50</ISOSpeedRating>
    <ExposureBias>+0.0</ExposureBias>
    <ExposureTime>1/15</ExposureTime>
    <Aperture>111.0</Aperture>
    <FocalLength>120.0</FocalLength>
  </MetaDataFields>
</MediaItem>
Hydraulics Core Database Layout
Units As Coffee Beans

http://www.flickr.com/photos/niallkennedy/1780421312/
Hydraulics Core Database Layout
Hydraulics Core Database Layout 
(Using Coffee Analogy)
Finalization Workflows

Three distinct workflows:

1. Archiving
2. Patron Delivery
3. Digital Library Delivery

Decisions that govern what workflows each Unit undergoes are data-driven
Messaging in Ruby on Rails

- Processor 1
  - Queue 1
    - Processor Group 1
      - Queue 2
        - Processor 2
  - Subscribes To & Publishes To
- Processor 3
  - Queue 3
    - Processor Group 2
      - Queue 4
        - Processor 4
  - Sends Messages To & Receives Messages From
- Poller
- Broker

Belongs To

Is Run By
Messaging in Ruby on Rails

- **Processor 1**
  - Subscribes To & Publishes To
  - Belongs To **Queue 1**
  - Is Run By **Poller**
  - Sends Messages To & Receives Messages From **Broker**

- **ActiveMessaging gem**
  - Creates Processor class
  - Defines queues in config/messaging.rb
  - Controls processor message handling

- **Daemons gem**
  - Runs Ruby processes in background
  - Similar to Linux ‘service’ command

- **ActiveMQ**
  - Messaging broker implementing JMS
  - Message Persistence
Logging Messages in Hydraulics: 
AutomationMessageClass

Write ‘success’, ‘failure’ and ‘error’ messages to database

Associated to an Order, Unit, Master File or Bibliographic record

Error messages are flagged; display on administrative home page; contain both stack trace and diagnostic message

Provide persistent audit trail of completed work

Tracksys has processed 2.2 million messages as of 6/10/2011
Finalization Workflows: Archiving

All Units archived to HSM for long-term preservation
  • Quantum Stornext

Checksums created

File paths systemized

One-click download of files for future redelivery

Able to ingest content into Fedora repository directly from archive
Finalization Workflows: Patron Delivery

Deliverable images created on a Unit-by-Unit basis
  • Type determined by intended use

Manifest of digital objects and invoice (PDF-format)

Zip archive of deliverables made web accessible

Email sent to customer providing URL for image pickup

Optional DVD pickup; requires staff override

Automated server cleanup
Finalization Workflows: Digital Library Delivery

Atomistic model, datastreams match Hydra guidelines

Fedora Objects will be created for:
- Bibliographic Records
- Master File

Object creation through API calls; No FOXML

Flexible access rights and ‘discoverability’ set at Unit, inherited by Master File and Bibliographic objects.
Finalization Workflows: Digital Library Delivery

Datastreams of bibliographic record object:

- MARC - External reference to Blacklight MARC XML view
- descMetadata - Transformation of MARC XML to MODS. LC stylesheet used, with local modification.
- DC - Dublin Core generated by MODS-to-DC LC stylesheet
- solrArchive - Record of solr <add> doc. Generated by custom and parameterized MODS-to-Solr XSLT
- POLICY and rightsMetadata - XACML policy
- RELS-EXT and RELS-INT

Digital Library Delivery Workflows:

- Customer places request
- Request vetted by Staff
- Automated Quality Assurance
- Include in DL is set
- Patron Delivery Workflow
- Archiving Workflow
- Intended Use Not Digital Collection Building
- All items
- Digital Library Delivery Workflow

Collection Building:

- Intended Use Not Digital
- Collection Building is set

Production Workflow:

- Outside Hydraulics
  - Physical Item
  - Scanned
  - Manual Quality Assurance
  - Automated Workflow
- Within Hydraulics
  - Request Form Front
  - End of Hydraulics
  - Metadata Creation
Finalization Workflows: Digital Library Delivery

Datastreams of Master File objects

- descMetadata - MODS created from data in Hydraulics
- DC - Dublin Core generated by MODS-to-DC LC stylesheet
- solrArchive - Record of solr <add> doc. Generated by custom and parameterized MODS-to-Solr XSLT
- POLICY and rightsMetadata - XACML policy
- RELS-EXT and RELS-INT
- content - Binary JPEG2000
- technicalMetadata - MIX generated from Hydraulics.
Demonstration of Ingestion
Empowering Communities: Production

Automated delivery and quality assurance saves staff time; more focus on digital production

More efficient access to preservation archive

Easier redelivery of already digitized content; less harm to material
Empowering Communities: Selection

Subject specialist, librarians and scholars do the selection of content by placing requests.

Demand-driven repository development.

Newly accessed collections can be added to digitization queue shortly after cataloging.

Flexibility to select content not originally requested for the digital library.
Empowering Communities: Object Creation & Maintenance

By brokering all ingestion through a messaging service:

- Metadata changes are now effectively done with a click of a button
- Policy/Access rights can be changed swiftly in response to IP violations or concerns
- Poor quality images can be replaced as soon as they are rescanned and re-archived
- New architectures can be instantiated quickly
Additional Functionality

The following topics were not covered in this presentation, but are part of Hydraulics:

1. EAD metadata: Associating Master Files with Component
2. Non-EAD MSS metadata: Associating Master Files with Box/Folder-level information provided by archivists at the time of digitization
3. Associating transcription of images with Master File records; methods of ingesting and indexing this content
4. User access and privileges
5. Extensibility of Master File class beyond TIF images
6. Statistical analysis of production metrics (for you managers out there....)
7. Hand-crafted MODS, RDF, SOLR can be provided to ingestion workflow for all object types
Next Steps

1. Community outreach and engagement
2. Possible integration with the Hydra initiative
3. Redesign code for Rails 3
4. Expand Master Files to handle A/V material
5. Develop and release app as Rails plugin
Andrew Curley
University of Virginia Library
Digital Curation Services
andrew.curley@gmail.com
@andrewcurley
#projecthydraulics

http://projecthydraulics.org/
http://demo.projecthydraulics.org/request
http://demo.projecthydraulics.org/admin
https://github.com/uvalib/hydraulics