Project Context

2004-2006  VanRIMS Classification Project

2008-2009  VanDOCS ERDMS Project

2009-2010  Olympic Legacy Project
Project Phases

I - Proof of Concept (2008-2009)
- Public records
- Controlled creation environment

II - Prototype (2009-2010)
- Private records
- Uncontrolled creation environment
Initial Assumptions

• Use OAIS (Open Archival Information System Reference Model) as a starting point

• Progressively add to requirements, drawing from:
  - General Preservation Standards
    • InterPARES
    • RLG/OCLC Trusted Digital Repositories (TDR)
  - Task specific
    • E.g., PREMIS metadata
  - Institution specific requirements
Initial Assumptions: Do we Need a DA at all?

Q: Why can’t the ERDMS product serve as Vancouver’s Digital Archives?

A: - Diverse record sources
    - Public access requirement
    - Missing functional components
CoV Digital Archives: Producers and Consumers

ERDMS

City Staff Desktops (email, docs, files)

capture

City Business Systems (structured data)
- Legacy Systems & Data
- City Website(s)
- Scanning / Imaging
- Private Donations

Digital Archives

store

active documents

ERDMS Transfer Requirements

inactive documents

transfer

store

organize

destroy

organize

preserve

City Staff

Public Researchers

access

access

access
Alternatives - What’s out there already?

Already many free/open source tools available:

<table>
<thead>
<tr>
<th>Ingest Tools</th>
<th>Repository</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>JHOVE</td>
<td>DSpace</td>
<td>Archivist’s Toolkit</td>
</tr>
<tr>
<td>DROID</td>
<td>FEDORA</td>
<td>ICA AtoM</td>
</tr>
<tr>
<td>XENA</td>
<td>Greenstone</td>
<td></td>
</tr>
</tbody>
</table>

Each only does a small part in the preservation chain, no start-to-finish single solution
So, what can we do with the existing tools?

Can we piece all of the various components together to come up with a complete Digital Preservation system?

Constraints:
- Use open source tools wherever possible
- Lightweight system architecture
- Architecturally independent components
What is OAIS?

OAIS (=Open Archival Information System)

- ISO 14721:2003
- Is a high level reference model
- Defacto standard for discussing digital preservation concepts at this level
- Important concepts include
  - Information Model
  - Functional Entities
  - Mandatory Responsibilities
OAIS Information Model

Information Packages contain:
- Content (records)
- PDI = Preservation Description Information (metadata)
- Packaging Information

Three types of Information Packages:
SIP = Submission Information Package (what we get)
AIP = Archival Information Package (what we preserve)
DIP = Dissemination Information Package (what we provide)
OAIS Responsibilities

- Accept submissions from Producer
- Establish control over material
- Implement long-term preservation policies
- Determine who the users are (“designated Community”)
- Ensure preserved information is understandable to users
- Provide access
OAIS Functional Entities

- Establishes the main functional components of the system
- Defines the relationships of the components to each other in terms of the information that passes between them
Source: CCSDS (2002, pp.1-4)
Developing System Requirements from the OAIS

Problems:
- OAIS illustrates the relationships between the functional entities, and how information flows between entities
- Doesn’t show when, why, and how at low enough level to be useful for system development
- Functional entities not always equivalent to actors
To proceed, we need to develop:

- Functional requirements that specify what the system must be able to do
- Metadata requirements that stipulate what data attributes must be captured for each step
- Technical requirements that stipulate specific technical features, formats or protocols that must be implemented.
Requirements Development Methodology

- Analyze the OAIS functional model (diagram and text) and develop use cases for each instance of information being communicated between functional entities
- Translate use cases into activity diagrams (flowcharts) using UML (unified Modeling Language)
- Use iterative development to refine activity diagrams
- Develop requirements from activity diagrams
Using Requirements to Identify System Components

- Use identified requirements to test and select available software tools
- Identify deficiencies in available tools
- Identify gaps where no tool currently exists
UML Activity Diagrams

- Intuitive modeling language for flow-chart diagrams
- Useful for system development, workflow modeling, etc., because they have a defined start and end point, and the steps that take you from one to the other
Goal of Modeling

Not to say definitively how OAIS should be implemented.

Is to determine what needs to be done, and to show that the way we have chosen to do it is consistent with what the OAIS requires.
1 of 3 Ingest Diagrams

AD1 Receive SIP
AD2 Generate AIP
AD3 Prepare AIP for Storage
Iteration 3 (CoV VanDocs Specific)

AD2 Accession SIP

1. Receive notification of transfer from SIG
2. Register transfer (sub-diagram AD2.2)
3. Queue registered SIPs for audit

4. Select SIP for audit
5. Audit SIP for compliance with submission agreement (sub-diagram AD2.3.3)
   - [Fail]

6. Quarantine SIP (sub-diagram AD2.3.4)
   - [Fail]

7. Characterize SIP (sub-diagram AD2.3.5)
8. Accept SIP for processing
9. Inform SIG of action for failure

If the audit is successful, the process continues. If there is a problem detected, the process is halted.
Iteration 3 (sub-diagrams)

CoV VanDoc/ Digital Archives: Activity Diagram 2 (v1)
AD2.2 Register Transfer

1. Register transfer in workflow tracking tool
2. Select next SIP in workflow tracking tool
3. Create folder for collecting ingest report (e.g., accessions/records/[SIPs])
4. Update transfer status in workflow tracking tool

CoV VanDoc/ Digital Archives: Activity Diagram 2 (v1)
AD2.5 Audit SIP

1. Create [SIP]/audit log
2. Check SIP manifest: Confirm that each record entry in the SIP metadata [TRM/metadata] has an associated file (content object) in the SIP
3. Record [manifest] check result: for each record entry in [SIP]/audit log, and for [manifest] check as a whole [valid=Check=Pass or Fail]
4. Select next file (content object)
5. Check file integrity: Validate file integrity (MD5)
6. Record integrity check result in [SIP]/audit log
7. Check metadata: Confirm metadata record entry exists in accompanying metadata
8. Send file name: Check for invalid file name
9. Update associated metadata (TRM/metadata)
10. Record "zentralizer" activity in [SIP]/audit log
11. Finalize [SIP]/audit log to [accession/metadata][SIPs]
12. Update audit result in workflow tracking tool

Note: confirmation that transfer was complete and backup occurs in AD1.
[all SIPs registered]
End