

Preservation is not a Place

16 February 2010

John Kunze, Stephen Abrams,
Patricia Cruse, Perry Willett

*University of California Curation Center (UC3)
California Digital Library*

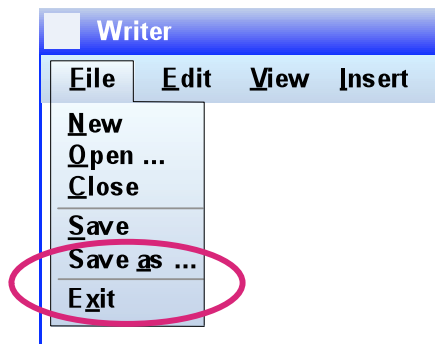
Curation design principles

Preservation is not a repository

- Preservation is an outcome
- Preservation is a relay
- All collections need some aspects of preservation
- Permanent objects live in disposable systems

The micro-services approach

Want low barrier, low commitment tools
Avoid monolithic, single-culture systems
Compose repositories from small,
independent, interoperable micro-
services



Ideally, micro-services should disappear into the surrounding infrastructure – office software, Linux distributions, etc.

The wisdom of files

After 30 years, we're *good* at modern filesystems

Files and directories (folders) are fast, plentiful, stable, and highly interoperable across platforms

Native OS tools will create, list, change, and backup

File-based micro-services will be easier...

to develop, maintain, and to *escape* from

to recombine in flexible ways

to move upstream into use by content producers

Collection storage: Pairtree

Use pairs of an object's identifier characters to create the object's file system path.

```
pairtree/  
  0=pairtree_0.01  
  pairtree-info.txt  
  pairtree_root/  
    id/  
      en/  
        ti/  
          fi/  
            er/  
              dflat...
```

Early success story

- *Pairtree* (storage service) creates paths from object id/en/ti/fi/er/s, and the resulting directory collection holds objects of any type
 - We invited Univ. of Michigan to co-author the Pairtree specification, and Hathi Trust uses our software to store Google books



cyocum

Import a pairtree and you can

- Enumerate all objects and their ids
- Produce any object by requested id
- Maintain and back up the tree with ordinary OS tools
- Rebuild a broken catalog simply by walking the filesystem

What's in this directory?

If files are important, so are file groups (directories)

- What if a file listing greeted visitors with metadata?

```
$ ls 12/34/5
0=dflat_0.01      admin/
1=Twain,_Mark    v001/
2=Huckleberry..  v002/
3=1898           v003/
4=12345
```

Directory-level metadata revealed *via* filenames

- These are Name-As-Text (*Namaste*) Tags

Object storage: Dflat

A “digital flat”: a residence for object data and metadata.

```
dflat/  
  0=dflat_0.01  
  dflat-info.txt  
  v001/  
    d-manifest.txt  
    delta/  
      redd...  
  v002/  
    manifest.txt  
    full/  
      data/  
      metadata/  
      enrichment/  
      annotation/
```

Reverse Delta Directory (ReDD)

File-level reverse delta compression.

```
redd/  
  0=redd_0.01  
  add/  
  delete.txt
```

Making (meta)data simpler

Metadata can be human and machine-manipulable

- “Email header protocols” drove the early Internet
- Why not metadata?
- Formally, ANVL – A Name Value Language

Internet-Draft: J. Kunze, B. Kahle, J. Masanes, G. Mohr

Instead of this...

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF PUBLIC "-//DUBLIN CORE//DCMES DTD 2002/07/31//EN"
    "http://dublincore.org/documents/2002/07/31/dcmes-xml/dcmes-xml-dtd.dtd">
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://www.nap.edu/books/0309064996/html/">
    <dc:title>The Digital Dilemma</dc:title>
    <dc:creator>National Research Council</dc:creator>
    <dc:date>2000-06-22</dc:date>
  </rdf:Description>
</rdf:RDF>
```

... this

erc:

who: National Research Council

what: The Digital Dilemma

when: 2000

where: <http://books.nap.edu/html/digital%5Fdilemma>

Simpler (meta)data semantics

Ask not what metadata you would *like*

- Ask instead how little metadata you need *not* to screw up *bit-level preservation*

Dublin Core is both too hard and too soft

- Dublin Kernel requires who/what/when/where, or explicit reasons for absences
- Comes with author name and date conventions
- Accommodates very rich and very poor metadata

Simpler tabular data

Excel add-in project with MS Research

- Scientific data commonly recorded in Excel, which is not good at versioning, schemas, date formats, etc.
- What add-ins could be defined to make research data more shareable, publishable, archiveable?

Early success story 2

- *BagIt* is a file package (“bag”) suitable for disk-based or fast network-based transfer of generic content
 - We wrote the BagIt specification with the Library of Congress, who now uses BagIt to receive most of its grant-funded partner content



Speaking of recycling, we are building on lots of ongoing success stories:

- JHOVE/2 (characterization service)
- ARK/NOID (identity service)
- XTF (index service)

Our micro-service specifications, and some software, are summarized at

<http://www.cdlib.org/services/uc3/curation/>

Two conference questions

- What has already failed? Can we generalize about approaches that are likely to fail over time?
- What might be the risks of building archiving systems that are “too good” or overly applied?

Other questions?

<http://www.cdlib.org/services/uc3/curation/>

John.Kunze@ucop.edu

Stephen.Abrams@ucop.edu

Patricia.Cruse@ucop.edu

Perry.Willett@ucop.edu