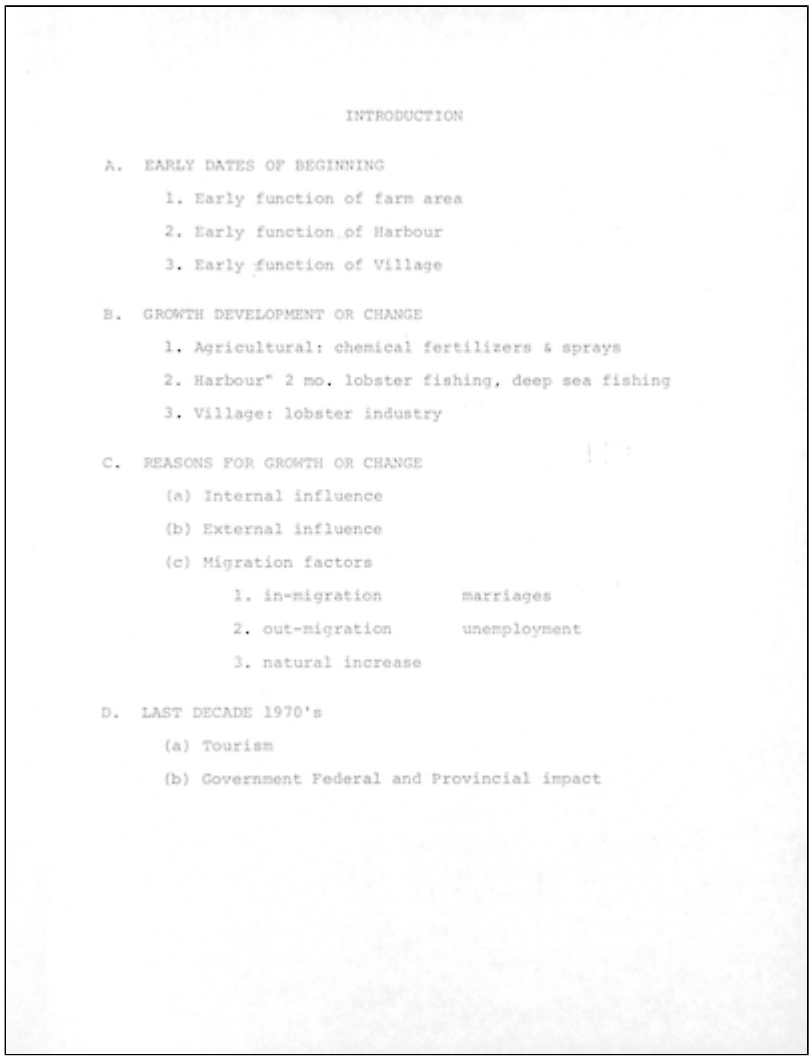
Case Study - Thinking Through a Data Problem

Developing a Content Model for a collection of typewritten letters

Our starting point is a scanned version of a typewritten letter, which has been saved as a TIFF.



INTRODUCTION

A. EARLY DATES OF BEGINNING

1. Early function of farm area
2. Early function of Harbour
3. Early function of Village

B. GROWTH DEVELOPMENT OR CHANGE

1. Agricultural: chemical fertilizers & sprays
2. Harbour" 2 mo. lobster fishing, deep sea fishing
3. Village: lobster industry

C. REASONS FOR GROWTH OR CHANGE

(a) Internal influence
(b) External influence
(c) Migration factors

1. in-migration marriages
2. out-migration unemployment
3. natural increase

D. LAST DECADE 1970's

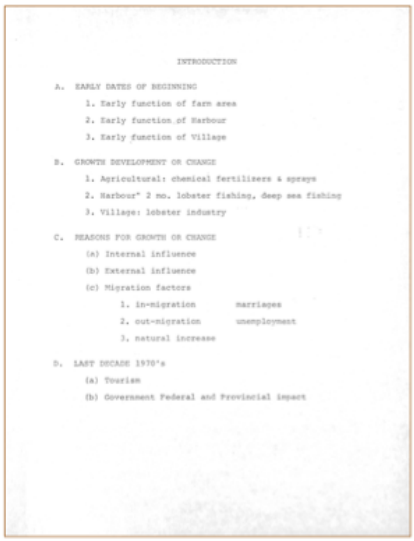
(a) Tourism
(b) Government Federal and Provincial impact

Questions to Ask

1. What kind of metadata do you want to gather? Metadata that describes the object, the administrative data related to the object, the technical metadata of the object you digitized, and/or metadata that relates to the longterm preservation of the object.
2. What kind of **metadata schema** will I use to describe each letter?
    a. If you are concerned with descriptive metadata is Dublin Core sufficient or would MODS be more appropriate (or EAD, etc.)?
        i. You'll need to review your content and select a schema that best matches your needs. Avoid creating your own schema.
    b. You need to use the FormBuilder to create your metadata form.
3. If the letters are **more than a single page**, how will you deal with that?
    a. There a few options here:
        i. Each letter is its own digital object and is related (using RELS-EXT or embedded in the metadata) to a 'collection object' that gathers the pages of the letter together.
        ii. A single letter object could have several several page datastreams.
        iii. Our preference would be to take an 'atomistic' approach: each page of a letter would be created as a digital object.
4. How will your users **view/search your collection** of letters?
    a. Will you have a grid display of your letter images? Or a list view? Or both?
    b. Will you need a **thumbnail** for each of your letter images?
        i. If so you'll need to create a thumbnail datastream that is part of your letter object. What happens if you have many pages in the letter? Just a thumbnail for the first page? What if in a search a user gets a list of letters/pages?
5. What will the **view a single letter** look like?
    a. Will you display the metadata of the letter and **web based image** of the letter (you may want to use some wireframing tools to sketch out your views, eg. try the Pencil Project, a plugin for Firefox)?
6. What **derivatives** will you need to provide the various views to your users?
    a. Thumbnail.
    b. Web based image of the letter.
    c. You could add tremendous value to your collection by extracting the text from the page image using an OCR program and including the resulting text in your index for search/discovery.

Based on the outline above we can start to determine the datastreams that will make up a typical letter digital object, which will then help us define the content model for this type of digital object.

Letter Digital Object



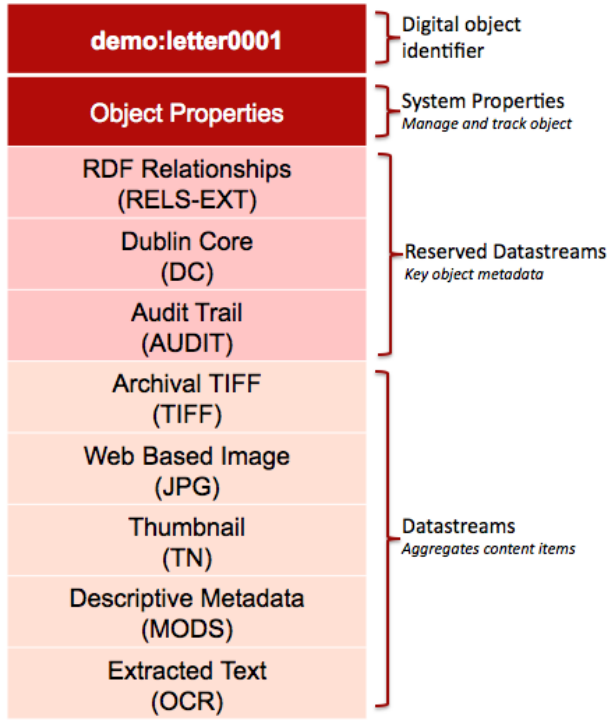
INTRODUCTION

A. EARLY DATES OF BEGINNING
1. Early function of farm area
2. Early function of Harbour
3. Early function of Village

B. GROWTH DEVELOPMENT OR CHANGE
1. Agricultural: chemical fertilizers & sprays
2. Harbour* 2 no. lobster fishing, deep sea fishing
3. Village: lobster industry

C. REASONS FOR GROWTH OR CHANGE
(a) Internal influence
(b) External influence
(c) Migration factors
1. in-migration marriages
2. out-migration unemployment
3. natural increase

D. LAST DECADE 1970's
(a) Tourism
(b) Government Federal and Provincial impact

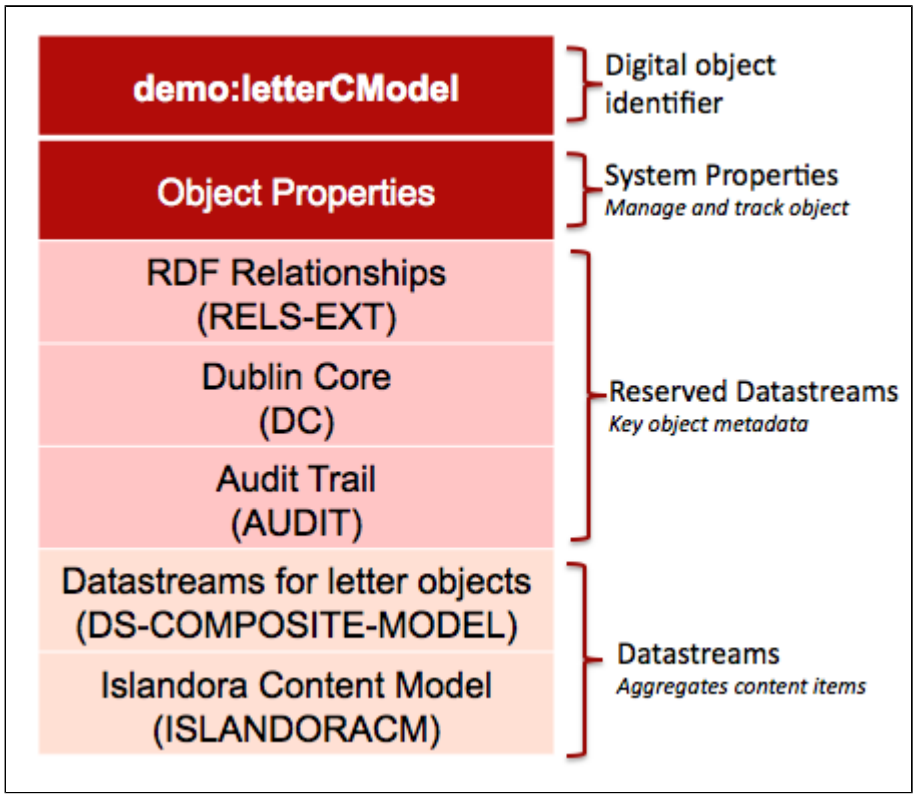


Here is a table of Datastreams, including the Datastream IDs that we've assigned, and the expected mimetype of the Datastreams.

{*}Datastream Label{*}	{*}Datastream ID{*}	**Mimetype**

Archival TIFF	TIF	image/tif, image/tiff
JPG Image	JPG	image/jpg, image /jpeg
Letter Thumbnail	TN	image/jpg
Descriptive Metadata	MODS	text/xml
Extracted Text	OCR	text/plain

Letter Content Model



When compared to the Letter Digital Object, the content model seems a bit thin. Much of the work of the content model is contained within the ISLANDORACM Datastream. Below you will find a commented FOXML version of the demo:LetterCModel content model. The bulk of the work of the ISLANDORACM is performed by a variety of functions which are contained within .inc files (PHP files) in the islandora/plugins directory.

Sample Content Model

```
<?xml version="1.0" encoding="UTF-8"?>
<foxml:digitalObject VERSION="1.1" PID="demo:LetterCModel" xmlns:foxml="info:fedora
/fedora-system:def/foxml#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="info:fedora
/fedora-system:def/foxml#
[http://www.fedora.info/definitions/1/0/foxml1-1.xsd]">
```

```
<\!-\- Object Properties -->

<foxml:objectProperties>
<foxml:property NAME="info:fedora/fedora-system:def/model#state" VALUE="Active"/>
<foxml:property NAME="info:fedora/fedora-system:def/model#label" VALUE="Large Content
Model"/>
<foxml:property NAME="info:fedora/fedora-system:def/model#ownerId" VALUE="fedoraAdmin"/>
<foxml:property NAME="info:fedora/fedora-system:def/model#createdDate" VALUE="2011-07-
21T11:40:51.192Z"/>
<foxml:property NAME="info:fedora/fedora-system:def/view#lastModifiedDate" VALUE="2011-
07-21T13:23:53.225Z"/>
</foxml:objectProperties>

<\!-\- Datastream Composite Model -->

<foxml:datastream ID="DS-COMPOSITE-MODEL" STATE="A" CONTROL_GROUP="X" VERSIONABLE="true"
>
<foxml:datastreamVersion ID="DS-COMPOSITE-MODEL.0" LABEL="Datastreams for this object"
MIMETYPE="text/xml">

<foxml:xmlContent>
<dsCompositeModel xmlns="info:fedora/fedora-system:def/dsCompositeModel#">

<dsTypeModel ID="DC">
<form FORMAT_URI="http://www.openarchives.org/OAI/2.0/oai_dc/" MIME="text/xml"></form>
</dsTypeModel>

<dsTypeModel ID="RELS-EXT">
<form FORMAT_URI="info:fedora/fedora-system:FedoraRELSExt-1.0" MIME="application
/rdf+xml"></form>
</dsTypeModel>

<dsTypeModel ID="TIFF">
<form MIME="image/tiff"></form>
</dsTypeModel>

<dsTypeModel ID="JPG">
<form MIME="image/jpeg"></form>
</dsTypeModel>

<dsTypeModel ID="TN">
<form MIME="image/jpeg"></form>
</dsTypeModel>

<dsTypeModel ID="MODS">
<form MIME="text/xml"></form>
</dsTypeModel>
```

```
<dsTypeModel ID="OCR">
<form MIME="text/plain"></form>
</dsTypeModel>

</dsCompositeModel>
</foxml:xmlContent>
</foxml:datastreamVersion>
</foxml:datastream>

<\!-\- Dublin Core Datastream -->

<foxml:datastream ID="DC" STATE="A" CONTROL_GROUP="X" VERSIONABLE="true">
<foxml:datastreamVersion ID="DC1.0" LABEL="Dublin Core Record for this object" CREATED="
2011-07-21T11:40:51.192Z" MIMETYPE="text/xml"
FORMAT_URI="http://www.openarchives.org/OAI/2.0/oai_dc/" SIZE="393">
<foxml:xmlContent>

<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:dc="
http://purl.org/dc/elements/1.1/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.
openarchives.org/OAI/2.0/oai_dc/
[http://www.openarchives.org/OAI/2.0/oai_dc.xsd]">

<dc:title>Large Content Model</dc:title>
<dc:identifier>demo:LetterCModel</dc:identifier>

</oai_dc:dc>
</foxml:xmlContent>
</foxml:datastreamVersion>
</foxml:datastream>

<\!-\- Relationship / RDF datastream ... in this case the relationship hasModel -->

<foxml:datastream ID="RELS-EXT" STATE="A" CONTROL_GROUP="X" VERSIONABLE="true">
<foxml:datastreamVersion ID="RELS-EXT.0" LABEL="Fedora Object-to-Object Relationship
Metadata" CREATED="2011-07-21T11:40:52.105Z"
MIMETYPE="text/xml" SIZE="327">

<foxml:xmlContent>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<rdf:Description rdf:about="info:fedora/demo:LetterCModel">
<fedora-model:hasModel xmlns:fedora-model="info:fedora/fedora-system:def/model#" rdf:
resource="info:fedora/fedora-system:ContentModel-3.0">
</fedora-model:hasModel>
</rdf:Description>
</rdf:RDF>
</foxml:xmlContent>
```

```
</foxml:datastreamVersion>
</foxml:datastream>

<\!-\- Islandora Content Model Datastream for this Content Model ... it is where the
ingest rules, display methods, mimetypes, and datastream IDs related to the object to
be created -->

<foxml:datastream ID="ISLANDORACM" STATE="A" CONTROL_GROUP="X" VERSIONABLE="true">
<foxml:datastreamVersion ID="ISLANDORACM.0" LABEL="Islandora Large Content Model"
CREATED="2011-07-21T13:23:53.225Z" MIMETYPE="application/xml"
SIZE="2175">

<foxml:xmlContent>
<content_model xmlns="http://www.islandora.ca" xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance" name="Large Content Model"
xsi:schemaLocation="http://www.islandora.ca [http://localhost/islandoracm.xsd]">

<\!-\- the mimetype for uploading defined -->

<mimetypes>
<type>image/tiff</type>
<type>image/tif</type>
</mimetypes>

<\!-\- the actions (embedded in .inc files which are php files) that happen to tif
images when they are ingested -->

<ingest_rules>
<rule>
<applies_to>image/tif</applies_to>
<applies_to>image/tiff</applies_to>

<\!-\- this ingest method calls the ImageManipulation.inc file in the islandora/plugins
directory and calls the doOCR function in that .inc file.

doOCR runs the tif image through tesseract and pushes the output into the OCR
datastream of the Letter Digital Object that gets created \-->

<ingest_methods>
<ingest_method class="ImageManipulation" dsid="OCR" file="plugins/ImageManipulation.
inc" method="doOCR" modified_files_ext="txt"
module="fedora_repository"></ingest_method>
</ingest_methods>
</rule>

<rule>
<applies_to>image/tif</applies_to>
<applies_to>image/tiff</applies_to>
```

```
<\!-\- this ingest method calls the ImageManipulation.inc file in the islandora/plugins
directory and calls the createThumbnail
function and passes the width/height parameters to imagemagick which converts the tif
image into a jpg thumbnail and pushes that
result into the TN datastream -->

<ingest_methods>
<ingest_method dsid="TN" file="plugins/ImageManipulation.inc" method="createThumbnail"
modified_files_ext="jpg" module="fedora_repository">

<parameters>
<parameter name="width">120</parameter>
<parameter name="height">120</parameter>
</parameters>

</ingest_method>
</ingest_methods>
</rule>
</ingest_rules>

<\!-\- a default DC ingest form for the object ... ideally we will build a MODS based
form to hold the metadata and it will get
automatically transformed to DC on ingest/creation ... this is just a placeholder -->

<ingest_form dsid="DC" page="2">
<form_builder_method class="buildQDCForm" file="FormBuilder" handler="" method="
handleQDCForm" module="plugins/DemoFormBuilder.inc">
</form_builder_method>

<form_elements>
<element label="Title" name="dc:title" required="false" type="textfield">
<description>Title of the letter</description>
</element>
</form_elements>
</ingest_form>

<\!-\- the list of datastreams that are included in the Letter Digital Object. By
listing the datastreams here, we will see them
in the dropdown list when we administer the object -->

<datastreams>
<datastream dsid="TIFF"></datastream>
<datastream dsid="JPG">

<\!-\- Display methods are similar to ingest rules in that they call php functions that
are embedded in .inc files. In this case
we call the ShowDemoStreams.inc in the islandora/plugins/ directory. There are some
default display methods embedded in the
```

```
islandora module ... eg. for the TN datastream -->

<display_method class="ShowSlideStreamsInFieldSets" default="true" file="plugins
/ShowDemoStreams.inc" method="showJPG" module="
fedora_repository"></display_method>
</datastream>

<datastream dsid="OCR">
<display_method class="ShowTIFFStreamsInFieldSets" file="plugins/ShowTIFFStreams.inc"
method="showOCR" module="fedora_repository">
</display_method>
</datastream>

<datastream dsid="TN"></datastream>
<datastream dsid="MODS"></datastream>
</datastreams>
</content_model>
</foxml:xmlContent>
</foxml:datastreamVersion>
</foxml:datastream>
</foxml:digitalObject>
```