# Process of developing

1. Learning about input files.

I sorted all input files according to their size and chose ten files. Two of them are over 20 KB, two of them are between 10 KB and 20 KB, four of them are between 1 KB and 10 KB and the last two are below 1 KB.

I found these elements are included in input files and I need to consider how to deal with them.

<record>

<metadata>

<oai\_datacite>

<payload>

<resource>

<identifier>

<creators> (have several <creator> inside)

<title>

<publisher>

<publicationYear>

<subjects> (have several <subject> inside)

<date>

<resourceType>

<alternativeidentifier>

<relatedIdentifiers> (have several <relatedIdentifier> inside)

<size>

<rights>

<descriptions>

1. Learning about Datacite schema and Dublin Core schema and figuring out differences.

Form below shows elements for Simple Dublin Core Schema

Dublin Core Metadata Element Set, Version 1.1

|  |  |
| --- | --- |
| Term Name | Definition |
| Contributor | An entity responsible for making contributions to the resource. |
| Coverage | The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant. |
| Creator | An entity primarily responsible for making the resource. |
| Date | A point or period of time associated with an event in the lifecycle of the resource. |
| Description | An account of the resource. |
| Format | The file format, physical medium, or dimensions of the resource. |
| Identifier | An unambiguous reference to the resource within a given context. |
| Language | A language of the resource. |
| Publisher | An entity responsible for making the resource available. |
| Relation | A related resource. |
| Rights | Information about rights held in and over the resource. |
| Source | A related resource from which the described resource is derived. |
| Subject | The topic of the resource. |
| Title | A name given to the resource. |
| Type | The nature or genre of the resource. |

What highlighted with yellow are those shared elements for both Datacite and Dublin Core. It seems that they can be put into new xml files directly.

What highlighted with blue are those there are similar elements in Dublin Core but need some modification.

What are not highlighted are those there are no same or similar elements in Simple Dublin Core element set.

I tried to extract information from input files and put them into Dublin Core generator (references 6) to check whether the output is the same with what I assumed.

1. After reading and comparing xml examples (oai-sample-1.0-datasite.xml and oai-sample-db.xml, reference 8 and 9) from OAI, I found that those within <metadata> are in fact the information part of input files.

The <record> part, as my personal perspective, is an element that cannot be changed no matter what metadata schema the xml file is under. In another word, suppose we have two xml files, one is under Datacite schema and the other is under Dublin Core schema, but they described the same data file. Though they are different, since they are under different metadata schema, they share same content and “talked” about the same thing. The <record> part is the key to indicate that they are with same thing.

Because of this understanding, I choose to extract the whole part of <record> (outside of <metadata>) and put it directly into output files.

1. Within <metadata> part, according to examples from OAI, attributes of <oai\_datacite> and attributes of <resource> are namespaces related to datacite. For output file, I choose to put all namespaces as attributes of <oai\_dc>
2. For those highlighted with yellow, I extract element text from input files and put them into new tree directly.
3. For <publicationYear>

I choose <date> tag.

1. For <date>

For <date> within <dates> I figured out that for those input files I randomly chose (about 30 in total, I randomly chose files when writing and debugging ) they all have attribute dateType=”Accepted”. Thus, I choose to <dateAccepted> as tag for Dublin Core. Since dateAccepted is not element term for Simple Dublin Core, obviously, I have to use qualified Dublin core and /terms/ namespace need to be added.

1. For <resourceType>

I found from randomly chosen input files that they all have attribute resourceTypeGeneral="Dataset" Dataset is a DCMI type. Therefore, I choose to use two <type> tags, one for element text of <resourceType> and one for attribute value of resourceTypeGeneral. And within the <type> for resourceTypeGeneral, I put attribute type with value “DCMIType”.

1. For <alternativeIdentifier>

Not all input files has tag <alternativeIdentifier>, but for all <alternativeIdentifier>, they all have attribute alternativeIdentifierType with value “citation”. Therefore, I use < bibliographicCitation> tag for output xmls under Dublin Core.

1. For <relatedIdentifier>

All <relatedIdentifier> have attribute relationType. Value of relationType attribute could be “HasPart”, “IsReferenceBy” and so on. All these attributes are properties in the /terms/ namespace.

1. For <size>

Use tag <extent>, the definition of which is “the size or duration of the resource”. It is also a term under DC terms.

1. Dealing with output

# What I learned

* Learning about what is xml, what is metadata schema and how to read them
* The experience of dealing with real problem

At the beginning, I totally had no idea what was Datacite schema, what was Dublin Core schema, and even not understood what those tons of xml files were and what should I do. However, I searched from internet to know about them, found out what I need to do and finally solved it. It is a very valuable experience since I will, with no doubt, have to deal with problems that I do not know what they are. This experience really gives me confidence to face those challenges in my future career.

* Python coding and debugging experience

I chose to run my code every time I did modification to it, even tiny modification. Right now I think it is really a good idea. It avoid the situation that I wrote a long paragraph and found out it did not work and did not know where problem lied. Debugging is the part that really costs time, much more than writing codes.

# References

1. Guidelines for implementing Dublin Core in XML

<http://www.dublincore.org/documents/dc-xml-guidelines/>

1. DATACITE DUBLIN CORE APPLICATION PROFILE DRAFT 1.8

Download: <https://groups.google.com/a/datacite.org/forum/#!topic/dc2map/-RLKw75WjEM>

1. OAI, Dublin Core part

<https://www.openarchives.org/OAI/openarchivesprotocol.html#dublincore>

1. Expressing Dublin Core Description Sets using XML (DC-DS-XML)

<http://www.dublincore.org/documents/dc-ds-xml/>

1. DCMI Metadata Terms

<http://dublincore.org/documents/dcmi-terms/>

1. Dublin Core generator

<http://www.dublincoregenerator.com/generator.html>

1. DataCite Metadata Schema 2.2

<https://schema.datacite.org/meta/kernel-2.2/>

1. OAI Dublin Core xml example

Download: <http://www.openarchives.org/OAI/2.0/oai_dc.xsd>

1. DataCite OAI Schema v 1.0

<http://schema.datacite.org/oai/oai-1.0/>

1. The lxml.etree tutorial

<http://lxml.de/tutorial.html>

1. 20.5. xml.etree.ElementTree — The ElementTree XML API

<https://docs.python.org/3.5/library/xml.etree.elementtree.html>

# Drawbacks and restrictions

* This python script deals specifically with input files in “dryad” folders.
* Output files are all in the same folder with dryad.py, rather than in one separate folder. It would bring some troubles when the amount of input files is huge.
* For transformation of <publicationYear>, I use <date> directly. Therefore, if users read output files only, there is no indication to tell them that this date is publication year.
* For all identifiers, including related identifiers, I ignored identifiertype attribute and its value, usually is DOI. Fortunately, it does not encounter any problem with input files I got, but if there is some files with identifiertype other than DOI, the output, to some extent, is not as easy and clear as the original input files to understand.