Goal:



Current Progress:

* darus\_data\_harvester
* Example: Mainly input repo. Can be treated as a user’s repository on which they will perform our tool. Now it is mainly a collection of files with several formats.
* author\_name.xml
* book.xml
* citation.txt
* test\_mesh.vtu
* harvester: This is the main source code repository.
* specifications: A central place to define all objects or classes of metadata (suppose an ‘Author’ with attributes ‘Name’, ‘Affiliation’, ‘Contact\_information’, e.t.c.). As we are using the sdRDM library, it holds markdown texts defining specifications.
* vtu\_model: specifications for vtu file.
* crawler.py:
  + It takes the path of a repository as an input
  + Outputs a dictionary where each key is a file type, and its associated value is a list of all files with the corresponding type.
  + Comment: Can one determine filetype by reading the files not from extension.
  + Response: Python’s ‘filetype’ library is not a good choice. Used ‘python-magic’ library. It identifies a vtk or vtu file as xml. Which is expected. First get the extension of file for file type. If it is not available then read the file and collect filetype by using magic.
* file\_group.py: @Fangfang
* harvester.py:
  + It takes the path of the repository as an input.
  + Then use crawler.py to get all files of each filetype present in the repository.
  + Then it takes each file to the corresponding parser function in parser.py.
  + Gets the metadata and prints it
* parser.py: Currently there are three functions to parse.
  + Parse\_xml: To parse xml file. Not useful. Created just to check the functionality of other scripts.
  + Parse\_plain: To parse text/plain file. Not useful. Created just to check the functionality of other scripts.
    - ‘Plain’ name is inspired from magic file type. But the extension says that it is a txt file. So there is an inconsistency if the filetype is collected from extension or by magic.
    - How to handle this: Copy paste the same code twice. One for parese\_plain and again for parse\_txt.
    - Maybe not very important to consider as collecting metadata from text files is very hard.
  + Parse\_vtu: @Fangfang
* README.md
* Documentation: This document.

Next Task:

* To check how to adapt hermes workflow (@Sarbani).
* Once done with vtu example, decide other file type to parse
* Divide the work to write other parsers.