Post-Term SIParCS Documentation (PTSD) - May-July 2021 Report

The following is a document listing the completed tasks and suggestions provided by Sama Manalai and Terry Yuan. They are interns of project 7 of NCAR/UCAR's 2021 SIParCS program.

Motivation: This is solely made out of personal interest to archive and summarize the work conducted over the summer of 2021 with NCAR's SAGE team on the Metadata Search Web Application, a prototype search tool. It is intended to serve as a progress marker for anyone involved with this project.

This is also a response to the mentors' desire to receive feedback from interns on their viewpoints of the 2021 internship.

Note: This is intentionally written in an unnecessarily complicated and lengthy style. You may never read this entirely a second time because of this very reason. It is also possible that some explanations are not written properly.

Achievement Summary: After summer 2021, the Harvester service of the Metadata Search Web Application now runs automatic update checks with GitHub data repositories to provide the most recently updated and indexed XML metadata files for users to see on the search engine that was built from summer 2020.

Achieved Objectives:

Over the course of 8 weekly sprints under the agile scrum framework, the following are the accomplished tasks and observations:

- General information:
    - Running both Harvester and Search codebases now require the use of external properties files as opposed to the ones premade within the IntelliJ projects.
        - New variables are added and mentioned in the GitHub README documentation, and will require additional commands written down in the "Edit Configurations" of the "Run" menu found in IntelliJ.
    - An external YAML file (.yml) was used to house known repository information. Proper formatting is demonstrated on GitHub.
    - README files are revised early in the project, with a heavy overhaul made especially for the Harvester codebase given its status as the chief highlight of this summer's work.

- Harvester codebase:
    - At the start, the Harvester only indexes documents strictly based on what's available on the local disk folder that it is aware of. Now, it can contact GitHub to check on the status of various repositories and can make true live changes to the search service.
    - Harvester service of the Metadata Search Web Application is now a separate spring boot web application - It now has webpages that can interact with the internal program itself
    - Harvester webpages are designed for developers to conduct two jobs:
        - 1. Manual update checks, or
        - 2. Clear the Solr index and reindex everything available in the local hosting environment.
    - Solr indexing only occurs to documents and assets known to have "updated" whenever a pull request is made for the selected repository.
    - Incorporation of webhooks into the controller. This allows the program to intercept updates sent by GitHub (provided they are linked correctly through URL).
        - **Note:** The webhooks are caught under the "/webhook" path as indicated on the controller. This should also be reflected when setting up GitHub webhooks.
    - Unit tests are added for the mediator classes within the codebase to test their functionality and for any potential fail use cases.
        - **Note:** There is one test (GitFileMediatorTest.java) that checks the acceptance of "xml" and "XML" files. Thus far, only "xml" is accepted, this will need to be checked with other developers and product owners.
- Apache Solr codebase:
    - Apache Solr service has a new data field called "github_xml_url" that will have the associated GitHub URL to each indexed XML file. When sending files to index, this data field has a formula built within the Harvester codebase to facilitate proper URL construction.
- Search codebase:
    - Hidden debug mode added. To access this, type "&debug=true" at the end of your query search on the search engine URL. This displays the "github_xml_url" field attribute found on Solr.
    - Empty query search strategy implementation allows for all indexed results to display if the search input field is blank.
- Coding Techniques (on IntelliJ)
    - Command + N: Yields a small menu that can be convenient for building constructors, getters, and setters.
    - //FIXME and //TODO notations: Comment notation to highlight and inform developers what to fix/do for future reference.

Potential Improvements for the Future (Most of the notes here may have already been listed on the Jira backlog):

- General information:
    - There were some mentions throughout the internship to gradually phase out the repository information YAML file in favor of some form of database
        - **Warning:** This may mean reworking the Harvester codebase as there are code and classes dependent on the YAML file to retrieve repository information and build the GitHub URL for each indexed XML file.
    - Visual updates may be possible in the future for the web pages of both Harvester and Search
- Harvester codebase:
    - We can safely address added and updated files, but not deleted files. While no error clearly shows up, the true behavior of these "deleted" files within the indexed Solr database needs investigation.
    - Proper update messages should be implemented so as to give further information on current harvester actions (e.g. loading bar display for an "Update" or "Clean and Reharvest").
    - Implement new feature on the reharvest menu:
        - Current options are: 1. Update, 2. Clean Solr and index what's available on disk ("Clean and Reharvest").
        - New option: Clear Solr, delete local disk repositories, and reclone and reindex everything.
    - Webhook development stopped at localhost testing. The next step would be to set up webhooks for the 11 NCAR repositories to gain legitimate updates upon docker deployment.
    - It must be verified that the GitHub repository should be seen as the authoritative source, NOT the local repository. While the chances of adding files in the local directory is low, this is a preemptive measure against unknown actions made on the hosting machine's directories.
        - 4 tests are advised on the Jira backlog issue DSETSP-129.
    - Special characters are a problem right from the start. This was found on the following three documents listed in here: https://github.com/NCAR/dset-web-accessible-folder-dev/tree/main/cisl/Cloud_Collection.
    - The harvester webpages are intended ONLY for developers to use. Therefore, there should be some security measures made to prevent direct access, which is the current state as of the 2021 internship's end. The following ideas are suggested:
        - Login and/or 2FA using Duo Mobile(?)

        - Last minute suggestion/quick search: Check out Spring Security since the codebase is under the Java Spring framework to begin with.
- Search codebase:
    - Autocomplete search results or suggestions be made to misspellings made on search queries
    - Provide a landing page or preview when the cursor hovers over the main link for each result
        - Instead of the main link directly leading to the source, a separate page can be built to house and display any relevant Solr field information as a border between the search results list and the source itself (Similar to what is on the CKAN DASH search tool now and a variety of other search engines)

Mentorship Feedback:

Because each intern-mentor relationship will be different based on personalities and other factors, this is only related to the following subjects:

- Interns: Sama Manalai, Terry Yuan
- Mentors: Nathan Hook, Saquib Aziz Khan, Eric Nienhouse, Christy Grant.

The following entails intern and mentor feedback from a retrospective session hosted on 7/23/2021:

- Uncle Bob videos can be informative, so it may be possible to introduce them as some form of mandatory study material before engaging in certain tasks
- Struggle v.s. Aid balance: For the interns this summer, general behavior noted was to struggle seemingly longer than anticipated. Questions are asked only if no progress is made after an overload of information has been read, and even then this decision gauge to ask questions is different among interns. General fears among the 2021 interns on asking questions are:
    - What if I didn't look up enough information to warrant a question?
    - What am I asking about/What's the main question?
    - Are we searching in the correct direction? (Doubting over the lack of progress/understanding)
- Mentors' self-assessments on their Google Jamboard sketches suggest concerns over drawing quality and information overload. Interns suggest otherwise, citing the sketches helpful especially after "struggling" to build understanding to create the desired code to recreate the conceptual sketch
    - Interns note that words written on the sketches are incredibly vital as they provide "keyword" hints during information search "struggles". Interns also welcome the sketches in their virtual setting as they serve as visual indicators of their upcoming codework.

Thank you to Nathan Hook, Saquib Aziz Khan, Eric Nienhouse, and Christy Grant. This internship has been really informative and fulfilling. We've learned a lot during this internship, and are happy to have helped with implementing changes and features into the Harvester component of the Metadata Search Web Application. We sincerely appreciate your guidance on our work for this summer, and hope you find this long report helpful in the future.

- Sama Manalai, Terry Yuan