

Digital Memory at CERN: Archiving and preserving existing knowledge

CERN Summer Student Programme - Final report

Author: Panna Georgina Lipták

Supervisors: Jean-Yves Le Meur, Antonio Vivace

June 28 - September 3 2021

Abstract

In the past decades, the value and importance of data became unparalleled. Since its foundation, a significant data set has been collected at CERN with the help of thousands of contributors. However, as a pioneer of scientific life, CERN also has the responsibility to preserve the acquired knowledge for future generations. This purpose holds many challenges: the amount of data, the deprecation of file formats, the digitization, and the diversity of both physical and computer-based data are all important aspects.

Answering the need, the Digital Memory project was born at CERN. The main goal is to provide an archiving service where contributors can initialize a request to preserve their digital assets in long-term storage. CERN hosts various services to store information, and it is important to be able to archive these records. During my Summer Student Programme, I mainly worked on creating pipelines for processing records from different upstream sources to ingest them into the archival system.

1 Introduction

The scientific world is moving in a data-oriented direction. CERN takes a large part in supplying scientists with information and not only from new researches but also by making available past data collected since its foundation. The field of physics is evolving, scientists get closer to understand how our world is built. In this process of acquiring knowledge and creating new theories, it could be powerful to be able to reflect on experiments performed in the past and view them from these recent aspects.

These resources could be in physical form (e.g. on tapes or photos) or in digital form. In the first case, the objective is to digitize them and make them available. The second case poses greater challenges because experience shows that online data is at risk of loss. This could happen due to several reasons:

hardware failure, deprecation of software and file formats, or simply because at the time the resource was not considered valuable enough to keep. Losing already existing knowledge is an avoidable error.

In 2016 CERN launched the Digital Memory Platform project [1]. The project focuses on providing long-term preservation for information available at CERN. The idea is to make the archiving process a standard for contributors. For example, if someone worked on a project for months, it is expected that there are resources created by this user and these resources could be on multiple platforms or even locally. Leaving a project, it would be standard to archive these digital assets and preserve the user’s work in the archival system for future use.

After defining the purpose of the project, implementation challenges needed to be considered. Creating a scalable and robust archiving system is not without a precedent, but integrating it in pipelines and workflows at CERN required a unique approach.

2 OAIS Reference Model

The project’s main concept is based on the widely accepted Open Archival Information System (OAIS) Reference Model [2] [3]. It defines a standard for a long-term preservation system without supplying architecture or implementation.

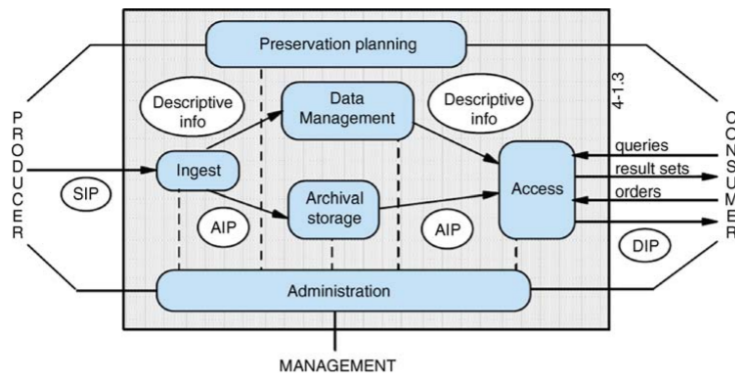


Figure 1: OAIS Functional entities [2]

Figure 1 shows the functional entities in the OAIS model. A producer means a person or a system that provides the data that needs to be archived. A consumer means a person or system who wants to access the archived information. Producers submit data using a Submission Information Package (SIP), which contains the actual payload of the artifact and Preservation Description Information (DPI). SIPs are converted into Archival Information Packages (AIP). An AIP consists of one or more Archival Information Unit (AIU) and an Archival

is a file packaging format designed to support the storage and transfer of digital contents.[7]

The OAIS model's input is a SIP that already has some prerequisites, and in the case of the Digital Platform, it is also in BagIt format to make the conversion to AIP easier. The BagIt Create tool [8] was designed to convert records harvested from CERN's digital repositories into valid SIPs. It is a Python module also available in the Python Package Index.

CERN hosts several digital repositories that contain various information created or gathered by contributors throughout the years. A record usually consists of metadata and files (optionally). The BagIt Create tool extracts necessary information from the metadata and structures the payload into the required format according to the CERN AIP specification.

3.1 Processing a record

The tool's main method is the `process` method that expects a source and a record id as a parameter. During my contribution to the project, the tool has been restructured so that the steps of processing a record could be easily separated and expanded if needed.

`process(source, recid, ...)`

1. Create a pipeline instance according to the provided source.
2. Prepare the folders of the bag and a temporary for downloading the files.
3. Create and add bagit.txt to the bag that contains the BagIt version and character encoding.
4. Prepare the AIC folder that will contain the metadata.
5. Get metadata through the pipeline. (See 3.2 The pipelines)
6. Save the metadata in the AIC folder.
7. Parse metadata and extract information to create a uniform metadata file.
8. Download the record's files to the temporary folder.
9. Copy the downloaded files to AIU folders (also created in this step).
10. Save the uniform metadata information into a JSON file inside the AIC folder.
11. Create a manifest file that contains the checksum and relative path inside the bag for every downloaded file and the uniform metadata file.
12. Verify the created bag with the bagit python package [9].
13. Delete the temporary folder.

3.2 The pipelines

Different digital repositories need to be handled differently due to their API specification. Many repositories hosted by CERN are built on the Invenio framework [10]. Currently the five supported platforms can be divided into two categories: CERN Document Server [11], ILC Document server [12] and CERN Open Data [13] are using InvenioV1, and Zenodo [14] and InvenioRDM [15] are using InvenioV3. InvenioRDM is a research data management repository platform, and for now the pipeline queries the belonging sandbox.

Every pipeline has to inherit from the `BasePipeline` class that contains common methods which are not source-specific, like creating folders. Four functions need to be implemented when someone wants to add a new pipeline to the tool:

- `__init__(source)`: Declaring initial instance variables.
- `get_metadata(record_id)`: Retrieve information corresponding to the given record id from the API of the source.
- `parse_metadata(metadata_filename)`: Create a list of files by extracting information from the metadata and add the metadata file to the list.
- `create_manifests(files, base_path, files_base_path)`: Generate the manifest file by specifying the algorithm used to create checksums for the files.
- `download_files(files, files_base_path)`: Download the files from list which were extracted from the metadata.

My initial task was to develop the InvenioV3 pipeline focusing on supporting the aforementioned two platforms. To avoid redundancy and to exploit the fact that they are built on the same foundation, they are handled by an `InvenioV3Pipeline` class. Despite the similarities, after requesting a record, the structure and syntax of the responses provided by the two sources are different, so the solution was to use a configuration file called *invenio.ini*. This configuration file contains the necessary parameters to extract the information needed from the API response. An entry's structure is the following:

```
[InvenioV3 instance name]
base_endpoint = < source API endpoint >
file_upstream = < JSON keys to the record's URI >
files = < JSON keys to the file list >
has_file_base_uri = < whether there is a separate file URI >
file_uri = < JSON keys to the file's URI >
file_name = < JSON keys to the file's name >
```

Every entry starts with the name of the source in square brackets. The *base_endpoint* parameter is the API endpoint of the server and the *file_upstream*, *files*, *file_uri*, *file_name* parameters are all for specifying JSON keys where

the corresponding information can be found in the JSON response. If the information is nested, multiple keys need to be separated by commas. The *has_file_base_uri* parameter is a boolean: in some cases, the links to the files are not explicitly included in the file list, they have to be constructed from a base URI and adding the data specified by the *file_uri* after it for every file.

3.3 Adding a new source

When a new digital repository needs to be supported by the BagIt Create tool, there are two options. If the platform was built on the InvenioV3 framework, it needs to be evaluated whether the API response could be processed by the schema specified in the configuration file. In this case, a new entry should be added to the *invenio.ini* file. On the other hand, if the platform uses a different framework or the API has a fundamentally different structure, a new class has to be implemented that inherits the *BasePipeline* class and defines the required methods (see 3.2 The pipelines). In both cases, the *main.py* file has to be updated with accepting the new source and creating the corresponding pipeline. For using the client version of the package, the new source has to be added as a valid option to the *cli.py* file.

4 OAIS platform

The Digital Memory Platform's API to trigger archiving jobs is called OAIS Platform [16], and the corresponding control interface is the OAIS web [17]. On this platform validated users can search for records in the supported digital repositories and request them to be harvested. In the current state of the project, the harvesting means that an SIP would be created locally. These requests need to be accepted by a user with the appropriate permission (called *can_approve_archive*). An approved request starts a Celery worker that handles the harvesting job by running the BagIt Create tool to process the record and a new entry in the list of archives is added with a status field. An archive's status can be: Waiting for Approval, Rejected, Pending, In Progress, Completed, and Failed. Users can list the archives for which they have permission.

My contribution to this module included adding the new pipelines to the list of available sources and improving the Search page with pagination, setting results per page, and a checkbox to only search by the unique record ID in the repositories. Figure 3 shows the OAIS Web Search page with these new features. After performing a search query, the results contain the title, the unique record ID, and the available actions. From left to right these actions are: getting more information about the record (the authors), initiating a harvesting request, and viewing the record in the original repository.

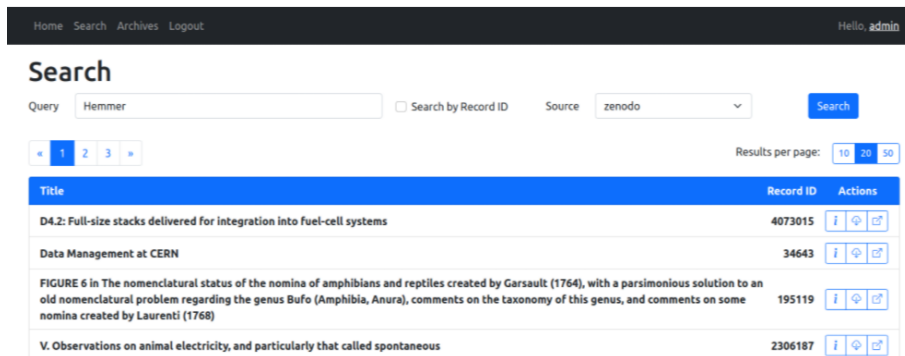


Figure 3: OAIS Web Search page with the new features

5 Future Work

Most parts of the Digital Memory Platform are still under development. In the future, new pipelines should be added to support at least all digital repositories hosted by CERN. Separating the digest and harvest methods are also an important task because this way users could not only harvest records from online sources but also provide their own SIPs to digest. Hence, it would be possible to archive local digital assets without needing to upload them to a repository just for this purpose. Relating to this, an SIP validation tool [18] needs to be implemented to check whether the SIP provided by the user is valid according to CERN's specification.

Some questions still need to be decided, and one of the most important ones is how to handle multiple versions of the same record. Accessing the evolution of a record will be a feature of the archival system, but the size of data handled by the platform is already a challenge, therefore versioning has to be implemented with the least amount of redundancy. The desirable solution is only storing the differences between versions by computing hashes for the files.

In this report, I mainly focused on the basic concept of the Digital Memory Platform and the structure of the BagIt Create tool. For further elaboration on CERN's preservation strategy, see [19].

Acknowledgments

During my ten weeks at CERN, I learned a lot about new technologies and workflows. I am deeply grateful to my supervisors for guiding me through this whole experience, making me feel part of the team, and being open to my inputs. The project I worked on was interesting and challenging, and in my opinion, it is an important but little-known part of Computer Science.

References

- [1] Digital Memory project in two words. <https://digital-memory-project.web.cern.ch/>. Accessed: 2021-08-19.
- [2] Christopher A Lee. Open archival information system (OAIS) reference model. *Encyclopedia of library and information Sciences*, 3, 2010.
- [3] CCSDS - Consultative Committee for Space Data Systems. *Reference Model for an Open Archival Information System (OAIS), Recommended Practice*, April 2019.
- [4] Jorik van Kemenade. The CERN digital memory platform: building a CERN scale OAIS compliant archival service. Master’s thesis, Vrije U., Amsterdam, 2020.
- [5] Brian Lavoie. Meeting the challenges of digital preservation: The OAIS reference model. *OCIC Newsletter*, 243:26–30, 2000.
- [6] Peter Van Garderen. Archivematica: Using Micro-Services And Open-Source Software To Deliver A Comprehensive Digital Curation Solution. In *iPRES*, 2010.
- [7] J Littman, L Madden, and B Vargas. The BagIt File Packaging Format (V0. 97) draft-kunze-bagit-07. txt. 2012.
- [8] bagit-create tool repository. <https://gitlab.cern.ch/digitalmemory/bagit-create>. Accessed: 2021-08-23.
- [9] bagit-python repository. <https://github.com/LibraryOfCongress/bagit-python>. Accessed: 2021-08-23.
- [10] Invenio. <https://inveniosoftware.org/>. Accessed: 2021-08-23.
- [11] CERN Document Server. <https://cds.cern.ch/>. Accessed: 2021-08-23.
- [12] ILC Document Server. <http://ilcdoc.linearcollider.org/>. Accessed: 2021-08-23.
- [13] CERN Open Data Portal. <https://opendata.cern.ch/>. Accessed: 2021-08-23.
- [14] Zenodo. <https://zenodo.org/>. Accessed: 2021-08-23.
- [15] InvenioRDM. <https://inveniosoftware.org/products/rdm/>. Accessed: 2021-08-23.
- [16] oais-platform repository. <https://gitlab.cern.ch/digitalmemory/oais-platform>. Accessed: 2021-08-23.
- [17] oais-web repository. <https://gitlab.cern.ch/digitalmemory/oais-web>. Accessed: 2021-08-23.

- [18] OAIS utils repository. <https://gitlab.cern.ch/digitalmemory/utils>. Accessed: 2021-08-23.
- [19] Jean-Yves Le Meur. *The Digital Memory of CERN: Status and input for a Preservation Strategic Plan*, 2018.