The SWHAP@DIUNIPI.IT

Here we present the SoftwareHeritageAcquisitionProcess (SHAPv0.1) we elaborated to acquire software at the Department of Computer Science of University of Pisa. It should be considered as v0.1 (its first version): revisions and extension will surely come as the topics is waste and somehow sprawling and tentacular.

What follows is organized with the intent to distinguish the design of the logical process from the instantiation of such a process. In fact, acquiring some case studies software we aim at defining an operational model and a possible implementation.

The Process

The Software Heritage Acquisition Process (SWHAP) is the process of acquisition of software source code from a request (*push* acquisition) or from a

To acquire software, we consider *source code* - rather than executables, for its value in terms of readability and understanding of the operation of the system - possibly with its *version history* - as an analogous of the strata of n archeological site, providing insights on the development process of the software.

The process is build of the following steps:

- **Collect**: the process in which the material is recognized, digitally acquired when needed and cataloged. This process may be particularly hard and costly when the original software is not already accessible in its digital form: it may involve the discovery of the coding scheme of the stored information and the recovery, may be even the building anew of the reading device, when the source code is stored on obsolete data supports.
- **Curate**: the process of restore, refinement, correction, integration and reshape into standard forms of materials, in such a way that they can be easily accessed and consulted. In particular, a work of identifying the different versions of a software, with their authors is required.
- **Present**: when collected and refined data are organized to be divulgated. In particular, apart from the submission to the Software Heritage Archive (that is the primary goal of the project, that is, to preserve the software base), many different supports may be needed for different purposes. For instance, Wikidata/Wikipedia, to preserve the related historical information in a machine-accessible way, for further generic dissemination of the knowledge embedded in the software. Open Access repositories can be used for articles, books, technical reports. Presentation tools/mechanisms, to provide the relevant information available in the previous archives in a way tailored to specific situations, like public events, lectures, etc. A good example is the work that has been done for Science Stories.

Each part of the process can be done by different people with different technical and cultural knowledge. The artifacts resulting at the end of each step are somehow milestones and represents a goal. When someone reaches a milestone, the administrator of the project and other users should be able to know that.

To realize the process go through three phases; each phase correspond to a virtual place and it is equipped with:

- a **Catalogue**, similar to a library catalogue, is a complete list of items, typically one in alphabetical or other systematic order;
- a **Journal**, that is a registry where all the done operations are written. For the depository and the warehouse, the journal contains records for acquisitions, their date, notes about the origin of the artifacts, information about where they are archived. For the workbench the journal is more detailed: is a sort of lab notebook where every activity is tracked.

In particular, the stages of the process are the following:

- **Depository**: it is a collection of raw (digital) material: they may be cases where it is needed to recover the digital representation from printouts of a software system, blueprints of a system architecture, punched cards, floppy disks, etc. If the software acquisition concerns physical items they pass by a **warehouse**, that is a physical place where items are stored and maintained since the pull or the push acquisition request.

- **Workbench**:
- **Vault**:

The final product of the process is the **curated source code (CuratedSC)** that will be stored into the vault, that is the Software Heritage Archive (HAL). Both the depository and the workbench are introduced to being able to tracing the origin and the evolution of the artifacts leading to the CuratedSC. In the end, we must be able to answer the questions *"What we have ?"*, *"Where and when we found it ?"*, *"In what way has been archived and transformed?"*.
After being either pulled or pushed, the software is Collected from an origin, and stored in its digital copy into the depository. The origin, in the case of physical repositories, can be a physical place, we speak of a warehouse. From the warehouse, a digital "as is" copy is made into the depository.

The Instantiation of the process: the di.unipi case

To acquire legacy software of Department of Computer Science at the University of Pisa, we instantiated the model of SWHAP using GitHub as support.

In particular, we choose GitHub as the material implementation of the depository and of the workbench as :

- it is a well-established platform for storing open source project and to collaborate with others. It offers an extensive and reachable disk space at a convenient price - it is free for open source projects;
- at the moment, the Software Heritage has already a crawler that feeds the vault from GitHub - we have an instantaneous realization of redundancy for persistence;
- it offers facilities to compile the journal (in the form of commit history, where each change is tracked) and the catalogue;
- it offers collaboration tools (via team, issues, etc) and it is integrated with presentation tools.

The main GitHub repository involved are

1. **DIUNIPI4SWH** This is the front-page of the project. It presents the project, the software acquisition process, collects the documentation, contains the catalogue of acquired softwares and the journal of acquisitions. It also links the template repository that has to be used to start each acquisition process. It is curated by one or more maintainers in charge of accepting records for the catalogue.

2. **DIUNIPI SWH TEMPLATE** This repository defines the skeleton of directories and files that has to be used for each software acquisition. In particular it is structured as

   - DEPOSITORY TMP:
     - CATALOGUE.md Contains references on where the physical source material is stored, possibly with some instructional references to contact. When the depository is done, the file content will be copied inside the DIUNIPI4SWH CATALOGUE.md .
     - JOURNAL.md Contains a log of what has ben done, by whom and in what way is been done.
     - README.md Contains a brief presentation of the SW_NAME and on their authors. Contains a link to other files and section of the repository itself.
     - METADATA.json Contains a list of pair value=data, using CodeMeta anthology, of all the raw - subject of further correction and integration - collected metadata. These metadata should be possibly filled by the author of the software himself, or the person in charge of the software in the moment of acquisition. If the software is already digitally acquired on other platforms (eg HAL) it

can be omitted (the link to the other platform should be annotated into DIUNIPI4SWH catalogue).

- WORKBENCH TMP
- SRC Sinthetic git of source code versions (int the style of (Spinellis 2017)).
- CATALOGUE.md
- JOURNAL.md
- README.md
- LICENCE.md
- METADATA.json Contains the CodeMeta sheet with all the structured completed and refined metadata - in the style of other platforms (eg HAL).
- Pointer to actively developed branch
- DATA Additional data

README.md and LICENCE.md as written by the origina author for historic memory.

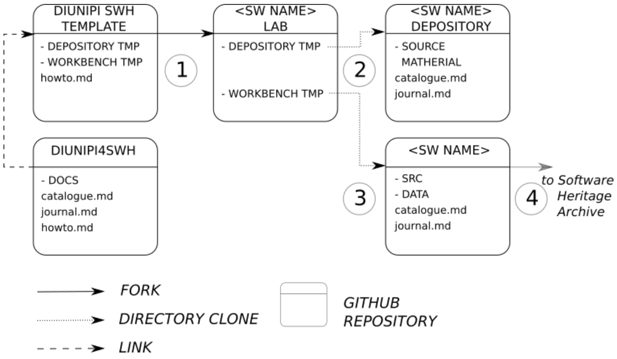

Figure 1: DIUNIPI SWHAP

The process here is as follows (see Fig. 1):

1. For every new software acquisition, the DIUNIPI SWH TEMPLATE, which contains the depository skeleton template, is forked into a LAB repository. The forked repository is named with the pattern LAB, where is in the form of Software_name+Main author surname+Year.
2. The acquisition process begins filling the DEPOSITORY TMP directory with all the digital version of original materials and the traces are written into the specific Depository Journal. Once the acquisition process is terminated the specific Depository catalog is compiled the DEPOSITORY TMP directory is cloned into a DEPOSITORY repository. This repository will be set as read-only (that is, writable only by the owner).
3. The curation process begins filling the WORKBENCH TMP directory from the material of the DEPOSITORY TMP directory. In particular, the SRC folder will contain a synthetic git build following what done by Spinellis (Spinellis 2017). The DATA directory will contain all additional information. As for depository, once the curation process is completed, the WORKBENCH TMP directory is cloned into a repository and will be set as read-only. This is the curated software, the result of the acquisition process. When (both the collect and) the curation process is terminated the LAB depository is deleted and the link to SW NAME DEPOSITORY and to SW NAME are added to the DIUNIPI4SWH CATALOGUE.
4. Once the curated software repository is done, the curator proposes a record into the catalog of DIUNIPI4SWH repository. The owner of DIUNIPI4SWH repository will accept the record and will submit the curated software to the Software Heritage Archive (HAL).

The whole process requires some administrators and moderators of DIUNIPI4SWH process, while each step can be done by different people. For this reason, HOWTO.md contains distinct guidelines for the many different roles involved.

The final picture, with curation process ended, is as in see Fig. 2.

Use cases

We focused our attention to the following legacy software:

- **CMM** - A garbage collector written by Giuseppe Attardi and Tito Flagella
- Grossi Tarabella musica elettronica
- *Martelli Montanari*

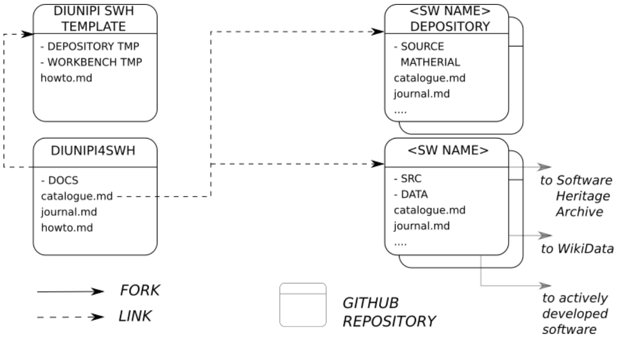

Figure 2: DIUNIPI SWHAP final result

- *Macchina ridotta*
- *Compilatore fortran cep*
- *Index Dantesco di Padre Busa*
- *Programma primo Ping di Lenzini*
- *Parser di linguistica computazionale*

Web Ref

- Latest generated pdf
- software acquisition template repository
- Directory clone
- DIUNIPI4SWH
- *google doc precedente*
- CodeMeta
- Esempio di acquisione di Silab1.1 su HAL

Code

Generare pdf con bibliografia ~ pandoc –filter pandoc-citeproc –bibliography=WorkingNotepad.bib –variable classoption=twocolumn –variable papersize=a4paper -s WorkingNotepad.md -o WorkingNotepad.pdf ~

Spinellis, Diomidis. 2017. “A Repository of Unix History and Evolution.” *Empirical Software Engineering* 22 (3). Springer US: 1372–1404. doi:10.1007/s10664-016-9445-5.