Software Stories for landmark legacy code

Morane Gruenpeter, Roberto Di Cosmo, Katherine Thornton, Kenneth Seals-Nutt, Carlo Montangero, Guido Scatena

Software Stories
for landmark legacy code

Final report - November 30th, 2021

Morane Gruenpeter (Software Heritage, Inria)
Roberto Di Cosmo (Software Heritage, Inria and University of Paris)
Katherine Thornton (sciencestories.io)
Kenneth Seals-Nutt (sciencestories.io)
Carlo Montangero (University of Pisa)
Guido Scatena (University of Pisa)

Abstract: Software Heritage in collaboration with the sciencestories.io team and the University of Pisa are introducing a new way to showcase software, that can be accessible to a wide range of software enthusiasts without any technical background, Software Stories. The project is supported by UNESCO as part of the shared mission to *collect, preserve and share source code as precious asset of humankind* [1]. The Software Stories interface is designed to highlight materials about a software title in a visual manner, similar to a digital software museum. The engine provides a semi-automatic tool for curators to create the presentation layer of the Software Heritage Acquisition Process (SWHAP) [2] allowing curators to generate a multimedia overview of a landmark legacy software title.

For the prototype, the University of Pisa has provided three software titles that were curated in 2019 during the creation of the SWHAP. These software titles were collected and curated in GitHub and then archived in Software Heritage. For the presentation layer the collected materials and metadata were deposited in Wikidata and Wikimedia Commons. You can visit `https://stories.softwareheritage.org` to check the Pisa collection and watch the demonstration video online on `https://youtu.be/s6uVdRDh5Xk`.

With the support of unesco

Contents

List of Figures

1 Introduction

The Software Heritage Acquisition Process (SWHAP) is a protocol to collect, preserve, curate and present legacy software of historical or scientific relevance [2]. It was established in 2019 as a result of a collaboration between Software Heritage (SWH), the University of Pisa and UNESCO. Recovering legacy source code, where the human readable knowledge is kept, is a delicate task. It requires a careful methodology to collect and reconstruct the innovative creation of its authors which might reveal a major scientific discovery.

In the SWHAP protocol, different actors have different tasks to collect the legacy software and all the physical and digital artifacts that are related to the software itself or its creators. One important aspect of preserving the software is also preserving the contextual information and artifacts in which it was built. The context of the software is a window to the past, which is essential to understand the software and its story. The story of the software is the one that will be told to the next generations. Therefore, curating the story around the software is as important as curating the software itself.

Software Heritage in collaboration with the `https:\sciencestories.io` team and the Pisa university are introducing a new way to showcase software, which is accessible for a wide range of software enthusiasts without any technical background, the software story. Each software story is a collection of pages, called moments, where different elements about a software are visualised, like in a virtual software museum.

In this report we present the steps taken by SWH, the sciencestories.io team and the Pisa university curators to create the software stories prototype. First, we describe the complete timeline of the project from June 2021 to the end of 2021. Then, in 3, we present the software titles from the Pisa collection, which were chosen for the prototype. Additionally, we detail the global structure of the Software Stories interface. In 4 We illustrate the storyboards of the prototype showcasing a single software story with the TAUmus example. In 5 We specify the workflow of a curator when interacting with the software stories interface. Finally, we conclude the report with the next steps of the collaboration toward the promotion and adoption of the complete SWHAP specifications.

2 Project summary

The Software Stories project is coordinated by Software Heritage. The implementation of the Software Stories interface is mandated to the sciencestories.io team. The following tasks have been completed during 2021:

- June: initialize project with the identified partners

- June: plan project schedule
- June: first meeting with all the partners (Software Heritage, sciencestories.io team and Pisa University)
- July: choose software examples for initial design
- August: cost estimation and formal invoice received from sciencestories.io team
- August: design of data model
- August: draft engine's workflow
- September: design of storyboards (user interface and curator interface)
- September: implement and deploy iframe for source code moment - SWH team
- September: prepare first progress report
- September: organise partners meetings to review first iteration
- October: get feedback on the first iteration from the Pisa University curators
- October: identify three software items from the Pisa collection for the prototype and complete their curation with the following tasks:
  - collect and curate metadata of the software items on Wikidata and Wikimedia Commons using an inventory file
  - verify or create Wikidata entities
  - collect links to media on the internet (e.g YouTube, Vimeo, Open Access repositories, etc.)
  - add to Wikimedia Commons the images that were collected with the source code during the collection phase of the SWHAP
  - identify SWHID of source code for each software title
- November: finalize demo of software Stories
  - develop the engine and storage to create and save a software story
  - deploy Software Stories application
  - redirect from `https:\stories.softwareheritage.org` to Software Stories application
  - create collection page with collaborators logos
  - choose image for the collection page
  - add on the collection page image for each story

  - verify that the rendering engine renders the three stories
  - verify that metadata for the software stories is shown (few iterations on Wikidata)
  - add a landing page for each story
  - check the source code moment and verify that the iframe is properly accessing SWH
  - modify SWHID to access source code branch in the SWHAP repositories

- November: specify how to collect and curate materials for presentation layer
- November: create video of the demo accessible on `https://www.youtube.com/watch?v=s6uVdRDh5Xk`
- November: prepare first curator guide to create a software story
- November: prepare final report

In this final report we present the status of the project in November 2021 with the description of the prototype, the design of the data model and the curator's interactions with the software stories interface.

In parallel, we started developing a curator guide to provide instructions to curators while collecting software and metadata and how to use the software stories interface. The next steps of this collaboration to complete the Software Heritage Acquisition Process (SWHAP) specifications and to promote both the Software Stories and the SWHAP are ordered with the timeline below:

- December: review the moments order and community feedback
- December: coordinate promotion software Stories after prototype validation through Software Heritage website, newsletter and social media, with explicit mention of UNESCO support
- January 2022: create a checklist for the curator to be used during the SWHAP (as the first step towards a full guide for the curator)
- January 2022: iterate on curator guide on how to create a Software story
- March-June 2022: promote completed SWHAP specifications

3 Software Stories: from design to implementation

The Software Stories application is a collection of story instances showcasing different software titles. For the prototype, three landmark scientific software titles from the University of Pisa were chosen and are detailed in [3]:

- **TAUmus (1972-1977) - Q107316563**: TAUmus is computer music software developed during the 70s in Pisa, first at IEI and then at CNUCE, by a team led by Maestro Pietro Grossi. TAUmus was basically a command line interpreter running on an IBM 370. It enabled the user to create and modify early pieces of computer music and play them. The sound was produced, under the control of the 370, by the TAU2 audio terminal, a device then developed by the same team at IEI.

- **Softi (1968) - Q108929297**: The purpose of this code is described as Softening of a curve and performs a simple smoothing of the values of a function. It is an exercise in using the CEP, and implements a general purpose numerical algorithm. The code is interesting for its structure, representative of the typical usage of the CEP at the time. It uses three programming languages:
  - FORTRAN CEP, a dialect - developed inhouse at CSCE - of the IBM FORTRAN II, for the main algorithm;
  - the CEP assembly language (LSDC, Symbolic Decimal CEP Language), for the input/output routines to/from paper tape;
  - the command language of the execution control system of the CEP, to launch the execution.

  The author, Tonina Starita (CSCE), was an innovator in the use of digital computing machines to study biological data, like EEG and similar measurements.

- **CMM (1994-1997) - Q109375562**: The Customizable Memory Management (CMM) is a memory management facility (aka garbage collector) supporting complex memory-intensive applications in C++. The CMM can manage several heaps, each one implementing a different storage discipline. The CMM has been developed to support the implementation of the Buchberger algorithm in the context of the European research project PoSSo (Polynomial System Solving) active from 1992 to 1995. In 1994 it was used by Sun Microsystems in the development of the Oak programming language, later known as Java.

Each software title is represented by a single story and is identified with the Wikidata identifier, starting with Q (Qxxx). Each story is composed by a set of moments, each moment is a page in the stories interface. The moments are divided into two main categories, Wikidata moments which are generated automatically from Wikidata, and custom moments which can be created through the Publisher Workspace. The Wikidata moments, with a collection of Wikidata properties, can be used directly when generating a software story. The custom moments are stored in the software stories storage after they are created and saved by a curator. In this report we will describe the Wikidata moments with the TAUmus example.

4 The storyboards

The Pisa collection is accessed through a collection page presented in figure 1, where a visitor can click on each item and explore the story and its different moments.

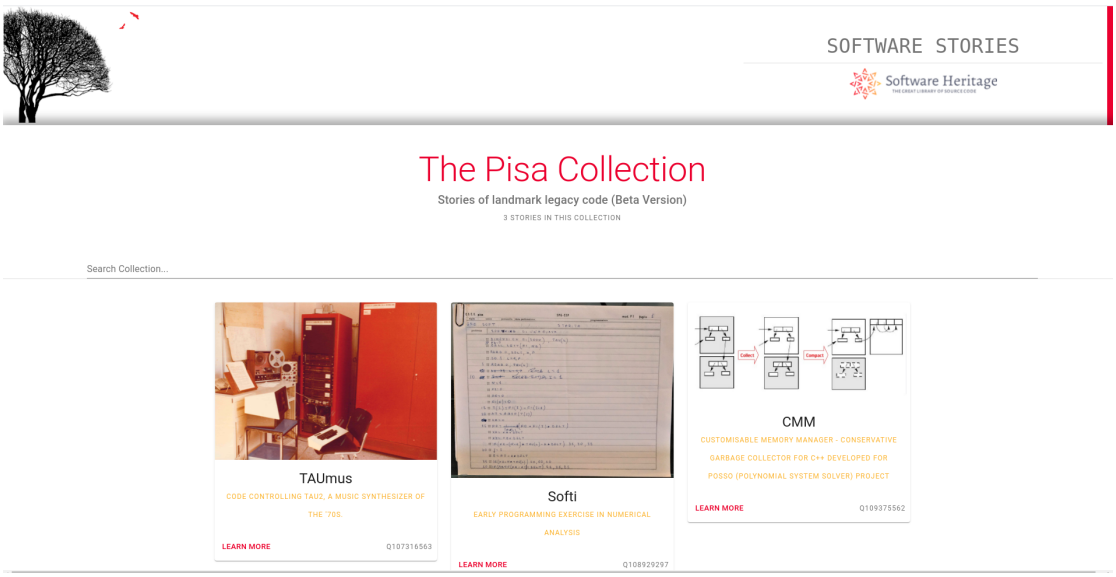

Figure 1: The Pisa collection page on the Software Stories interface

In this section we have used the first software title in the Pisa collection, TAUmus (Q107316563). The storyboards portray the way the prototype's interface presents the software entity using the moments structure. Most moments presented are Wikidata moment types. The last storyboard is the realisation of the Software Heritage source code moment.

The left bar starts with the software label, description and logo image of the entity. Below the main properties, the moment menu lists all available moments in a story. When accessing the story, the first moment is visible and is identified with a parameter in the url (moment=0). In a moment view there is a title on the top left of the moment panel and a subtitle that is usually used to contain the source of the moment's content. In the main frame are loaded the elements composing the moment, depending on the moment type.

4.0.1 Formal language describing a story

A curator can use a formal language to create a software story with the set of moments. This formal language is represented in a *json* format and starts with the Qxxx identifier in Wikidata:

```
{
  "id": "Q107316563",
  "label": "TAUmus",
  "description": "Code controlling TAU2, a music synthesizer of the
     '70s.",
  "alt_labels": null,
  "conformance": null,
  "moments": [ <list-of-moments> ]
 }
```

We have selected a few moments to show how a software story is composed visually and what is the formal language used to create it (using json input).

4.1 The landing page

The landing page is also a moment, the curator can choose the first moment on the Publisher Workspace. As for the SWH story template generated from Wikidata, the first moment will depend on what information is available on Wikidata. The first moment in this version's design is the media gallery moment.

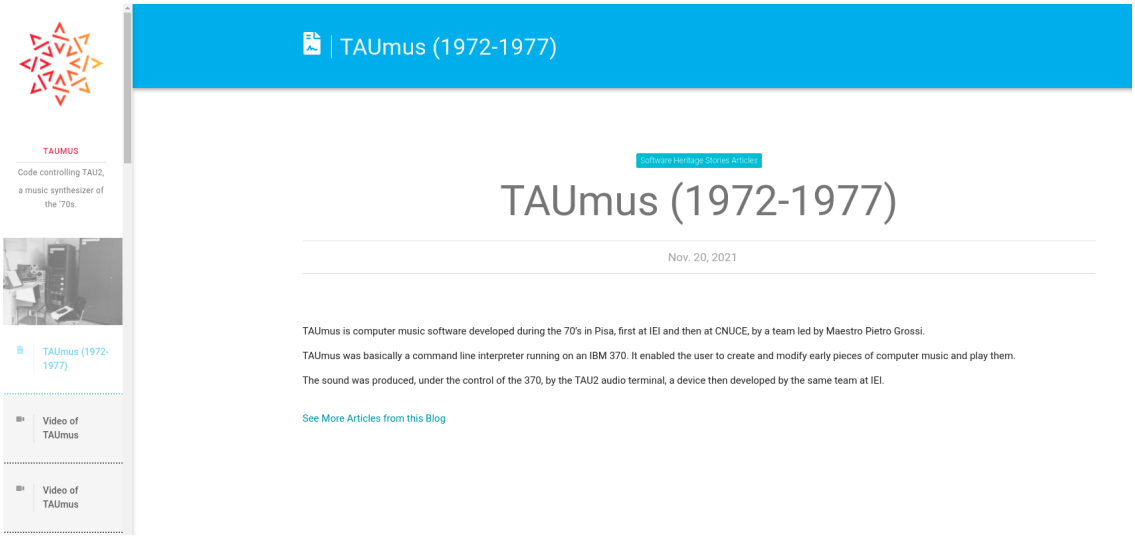

Figure 2: The landing page in the TAUmus story from the Pisa collection's Software Stories interface

4.2 Media gallery moment

The Media gallery moment has a set of 7 images retrieved from Wikimedia Commons. A IIIF manifest is generated for the resources and they are displayed in an instance of Universal Viewer [4]. This moment has an internal left menu with the collection of images, a top bar to control the gallery and a right tab to view the additional metadata on each item in the gallery. The moment is visible in figure 3.

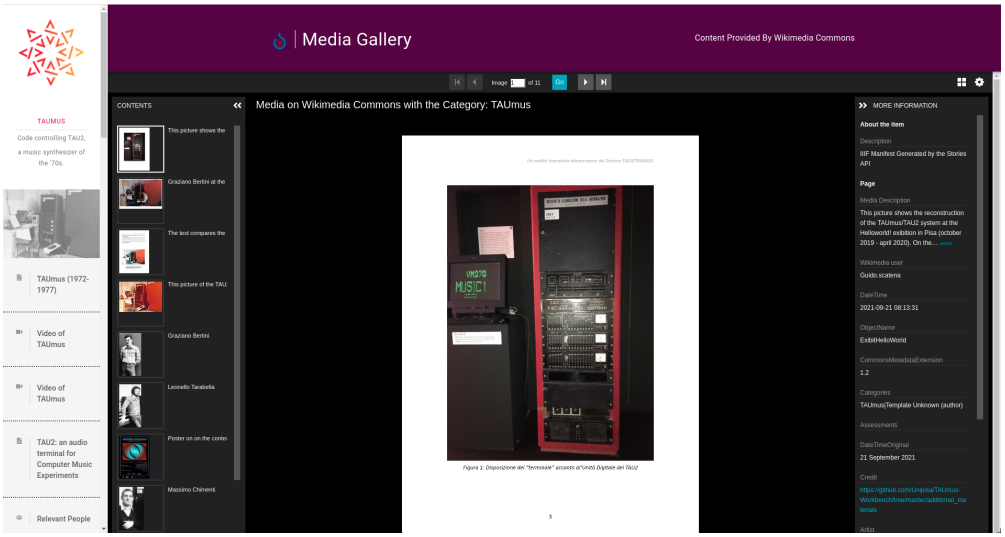

Figure 3: The media gallery moment in the TAUmus story

4.2.1 Formal language describing the media gallery moment

The formal language for a moment in *json* format includes a collection of properties used for the moment. The *index* property is used for the moment location in the story. Here is the example for a the media gallery moment in json:

```
{
  ...
    "moments": [
            {
          "index": 15,
          "type": "iframe",
          "title": "Media Gallery",
          "subtitle": "Content Provided By Wikimedia Commons",
          "icon": {
            "name": "Wikimedia Commons logo",
            "source": "img",
```

```
            "url": "https://upload.wikimedia.org/wikipedia/commons/4/4a/Commons-logo.svg"
          },
          "color": {
            "type": "hex",
            "background": "#530244",
            "text": "#fff"
          },
          "label": "Wikimedia Gallery",
          "reference": {
            "title": "Wikimedia Commons",
            "url": "https://commons.wikimedia.org/wiki/Category:TAUmus",
            "description": "Online repository of free-use images, sound and other media files, part of the Wikimedia ecosystem",
            "logo": "https://upload.wikimedia.org/wikipedia/commons/4/4a/Commons-logo.svg"
          },
          "url": "https://stories-api-stage.herokuapp.com/api/iiif/viewer/universal_viewer?manifest_uri=https://stories-api-stage.herokuapp.com/api/iiif/wikimedia/commons/category/TAUmus"
        }
    ]
}
```

4.3 Timeline moment

The timeline moment is a representation of dates available on Wikidata. Timeline items may contain images related to the event itself. The moment is visible in figure 4.

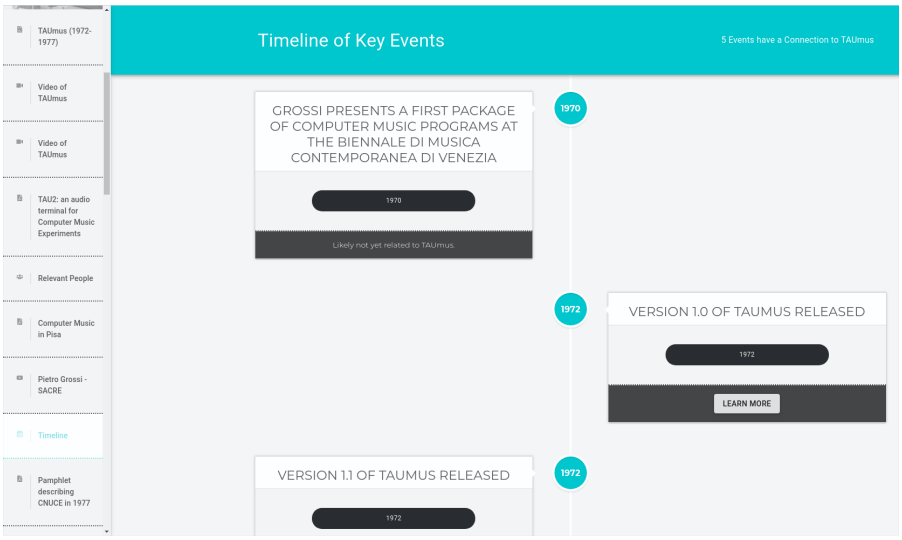

Figure 4: The timeline moment in the TAUmus story

4.3.1 Formal language describing the timeline moment

Here is the example of the formal language used for the timeline moment in json format:

```
{
  ...
    "moments": [
          {
           "index": 6,
           "type": "timeline",
           "title": "Timeline of Key Events",
           "subtitle": "3 Events are Currently Recorded in Wikidata
               with a Connection to TAUmus",
           "icon": {
             "name": "FaRegCalendarAlt",
             "source": "fa"
           },
           "color": {
             "type": "hex",
             "background": "#1dc7ce",
             "text": "#fff"
           },
           "label": "Timeline",
           "reference": {
```

```
            "title": "Wikidata",
            "url": "https://www.wikidata.org/entity/Q107316563",
            "description": "Free knowledge database project hosted by
                Wikimedia and edited by volunteers",
            "logo": "https://upload.wikimedia.org/wikipedia/commons
                /6/66/Wikidata-logo-en.svg"
        }
    ]
}
```

4.4 Map moment

The Map moment contains a full earth map, where specific locations are tagged with location markers. These locations correspond to geocoordinates available in statements from Wikidata items. The moment is visible in figure 5.

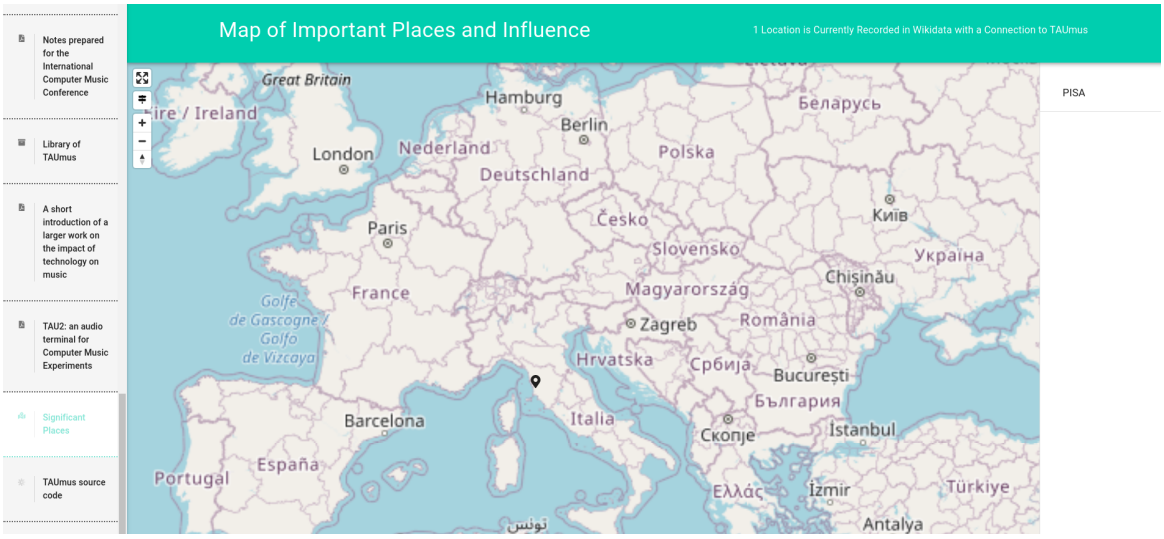

Figure 5: The map moment in the TAUmus story

4.4.1 Formal language describing the map moment

Here is the example of the formal language used for the map moment in json format:

```
{
  ...
    "moments": [
            {
```

```
        "index": 23,
        "type": "map",
        "title": "Map of Important Places and Influence",
        "subtitle": "1 Location is Currently Recorded in Wikidata
            with a Connection to TAUmus",
        "icon": {
          "name": "FaMapMarkedAlt",
          "source": "fa"
        },
        "color": {
          "type": "hex",
          "background": "#1dceae",
          "text": "#fff"
        },
        "label": "Significant Places",
        "reference": {
          "title": "Wikidata",
          "url": "https://www.wikidata.org/entity/Q107316563",
          "description": "Free knowledge database project hosted by
              Wikimedia and edited by volunteers",
          "logo": "https://upload.wikimedia.org/wikipedia/commons
              /6/66/Wikidata-logo-en.svg"
        }

    ]
}
```

4.5 People moment

The People moment contains a list of people related to the software with their image and metadata. This moment requires that the relevant people have also a Wikidata entity. The moment is visible in figure 6.

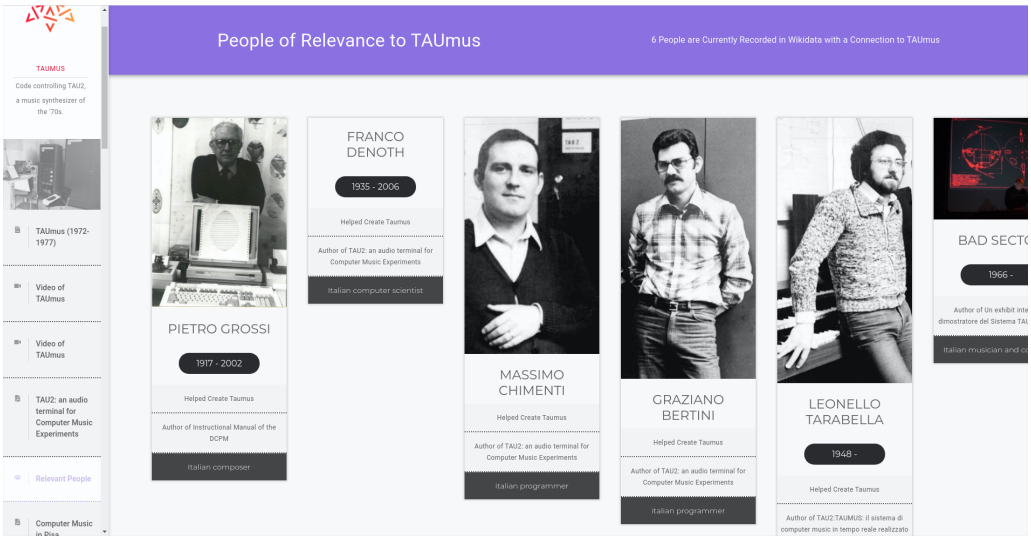

Figure 6: The relevant people moment in the TAUmus story

4.5.1 Formal language describing the people moment

Here is the example of the formal language used for the people moment in json format:

```
{
  ...
    "moments": [
          {
           "index": 3,
           "type": "people",
           "title": "People of Relevance to TAUmus",
           "subtitle": "5 People are Currently Recorded in Wikidata
               with a Connection to TAUmus",
           "icon": {
             "name": "FaUsers",
             "source": "fa"
           },
           "color": {
             "type": "hex",
             "background": "#8d6fe6",
             "text": "#fff"
           },
           "label": "Relevant People",
           "reference": {
```

```
            "title": "Wikidata",
            "url": "https://www.wikidata.org/entity/Q107316563",
            "description": "Free knowledge database project hosted by
                Wikimedia and edited by volunteers",
            "logo": "https://upload.wikimedia.org/wikipedia/commons
                /6/66/Wikidata-logo-en.svg"
          },
          "data": [
            {
              "label": "Pietro Grossi",
              "id": "Q3388139",
              "instance": {
                "label": "human",
                "id": "Q5"
              },
              "website": "http://www.pietrogrossi.org",
              "image": "http://commons.wikimedia.org/wiki/Special:
                  FilePath/Pietro%20Grossi%20with%20computer%20Acorn%20
                  Archimedes%20.jpg",
              "description": "Italian composer",
              "birth": {
                "label": "1917",
                "year": 1917
              },
              "death": {
                "label": "2002",
                "year": 2002
              },
              "references": {
                "value/P813": {
                  "property": {
                    "label": "http://www.wikidata.org/prop/reference/
                        value/P813",
                    "id": "value/P813"
                  },
                  "value": {
                    "label": "http://www.wikidata.org/value/
                        bec00327ac698eecf84b6bd3d0a91a40",
                    "id": ""
                  }
                },
```

```
                "P813": {
                  "property": {
                    "label": "http://www.wikidata.org/prop/reference/
                        P813",
                    "id": "P813"
                  },
                  "value": {
                    "label": "2021-06-22T00:00:00Z",
                    "id": ""
                  }
                },
                "P854": {
                  "property": {
                    "label": "http://www.wikidata.org/prop/reference/
                        P854",
                    "id": "P854"
                  },
                  "value": {
                    "label": "https://github.com/Unipisa/TAUmus-
                        Workbench/blob/master/metadata/codemeta.json",
                    "id": ""
                  }
                }
              },
              "years": "1917 - 2002",
              "relations": [
                "Helped Create Taumus"
              ]
            },
          ]
        }
    ]
}
```


4.6 Wikidata statements moment

Similar to the Wikipedia article moment, the Wikidata moment is a complete copy of the current page of the entity on Wikidata. The moment is visible in figure 7.

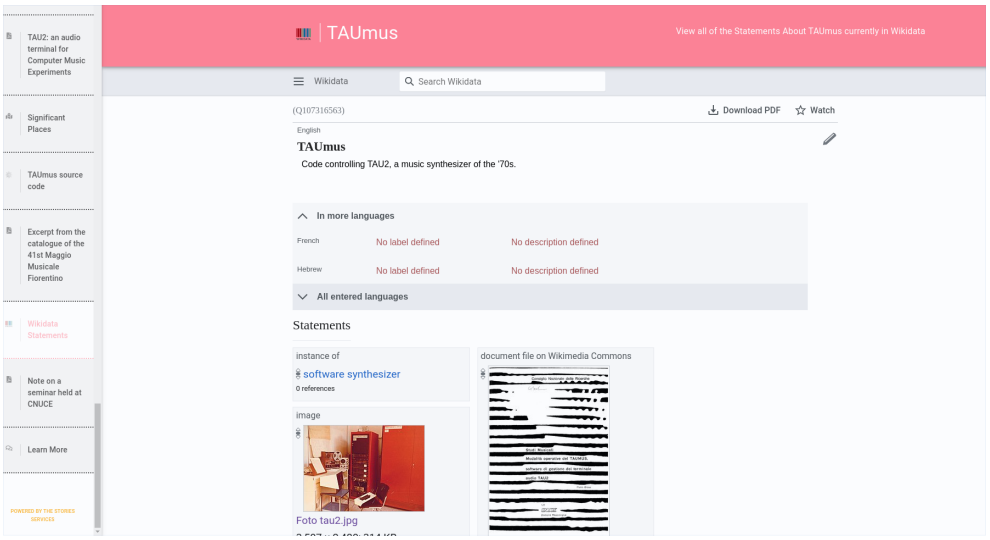

Figure 7: The Wikidata statements moment in the TAUmus story

4.6.1 Formal language describing the Wikidata statements moment

Here is the example of the formal language used for the Wikidata statements moment in json format:

```
{
  ...
    "moments": [
            {
          "index": 26,
          "type": "wikidata",
          "title": "TAUmus",
          "subtitle": "View all of the Statements About TAUmus
              currently in Wikidata",
          "icon": {
            "name": "Wikidata Logo",
            "source": "img",
            "url": "https://upload.wikimedia.org/wikipedia/commons
                /6/66/Wikidata-logo-en.svg"
          },
          "color": {
            "type": "hex",
            "background": "#ff8394",
            "text": "#fff"
          },
```

```
            "label": "Wikidata Statements",
            "reference": {
              "title": "Wikidata",
              "url": "https://www.wikidata.org/entity/Q107316563",
              "description": "Free knowledge database project hosted by
                  Wikimedia and edited by volunteers",
              "logo": "https://upload.wikimedia.org/wikipedia/commons
                  /6/66/Wikidata-logo-en.svg"
            },
            "url": "https://m.wikidata.org/wiki/Q107316563"
          },
          {
    ]
}
```

4.7 Learn more moment

The learn more moment contains a library of links to other locations where the software title is identified using external identifier properties of the entity and all references used in support of statements on the software item in Wikidata. Each item has a "learn more" button that redirects the visitor to external resources. The moment is visible in figure 8.

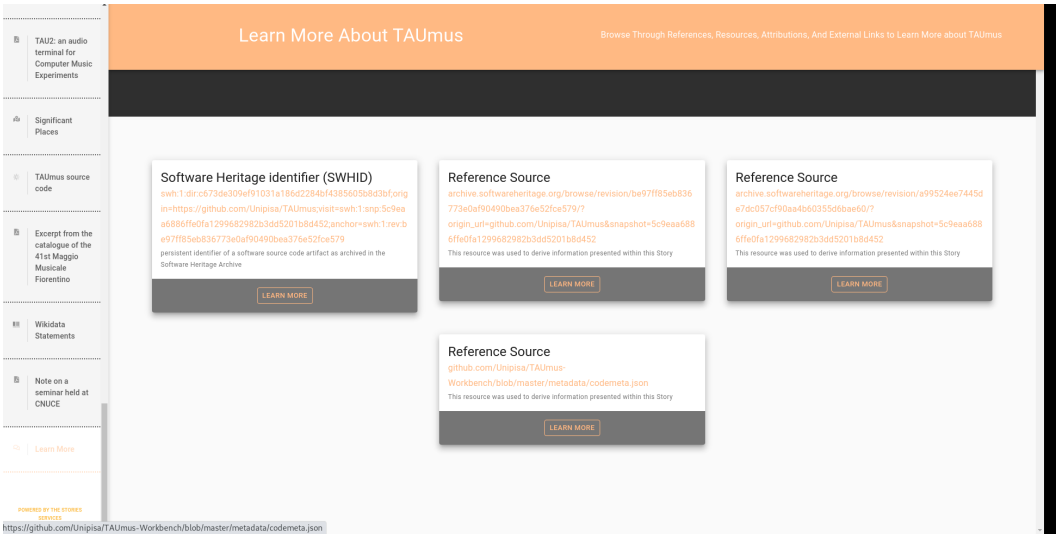

Figure 8: The learn more moment in the TAUmus story

4.8 SWH source code moment

The SWH source code moment contains an iframe of a source code directory, file or code fragment in the Software Heritage archive. The iframe structure presents the objects SWHID on top of the iframe and an easy access to the archive on the top right of the iframe. Furthermore the source code in a directory iframe can be browsed in the stories interface. The moment is visible in figure 9.

4.8.1 Formal language describing the SWH source code moment

Here is the example of the formal language used for the source code moment in json format:

```
{
  ...
    "moments": [
            {
          "index": 24,
          "type": "software_heritage",
          "title": "TAUmus source code",
          "subtitle": "Content Provided by Software Heritage",
          "icon": {
            "name": "Software Heritage Logo",
            "source": "img",
            "url": "https://upload.wikimedia.org/wikipedia/commons
                /9/94/Software-heritage-logo.1024px.png"
          },
          "color": {
            "type": "hex",
            "background": "linear-gradient(90deg, rgb(255 201 58) 00%,
                 rgba(226,0,38,1) 100%)",
            "text": "white"
          },
          "label": "TAUmus source code",
          "reference": null,
          "url": "",
          "swhid": "swh:1:dir:c673de309ef91031a186d2284bf4385605b8d3bf
              ;origin=https://github.com/Unipisa/TAUmus;visit=swh:1:snp
              :5c9eaa6886ffe0fa1299682982b3dd5201b8d452;anchor=swh:1:
              rev:be97ff85eb836773e0af90490bea376e52fce579"
        }
    ]
}
```

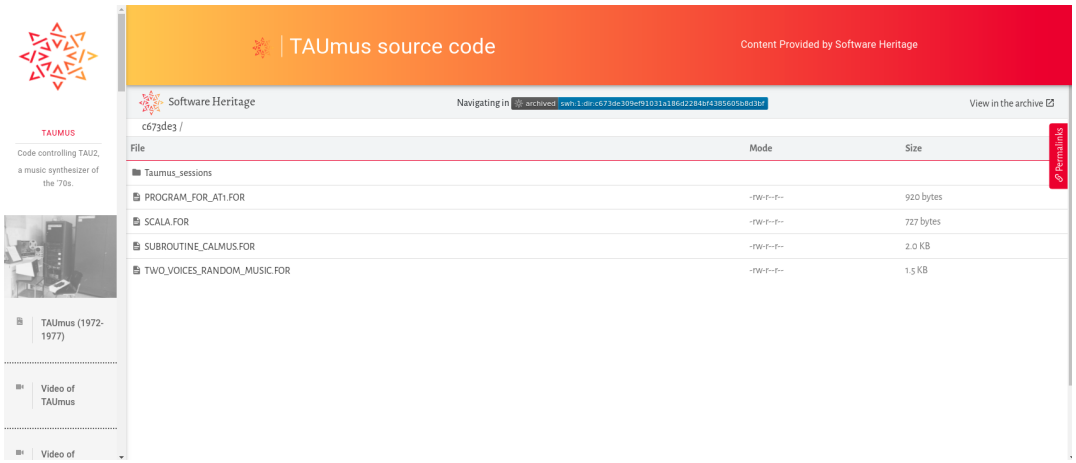

Figure 9: The SWH source code moment in the TAUmus story

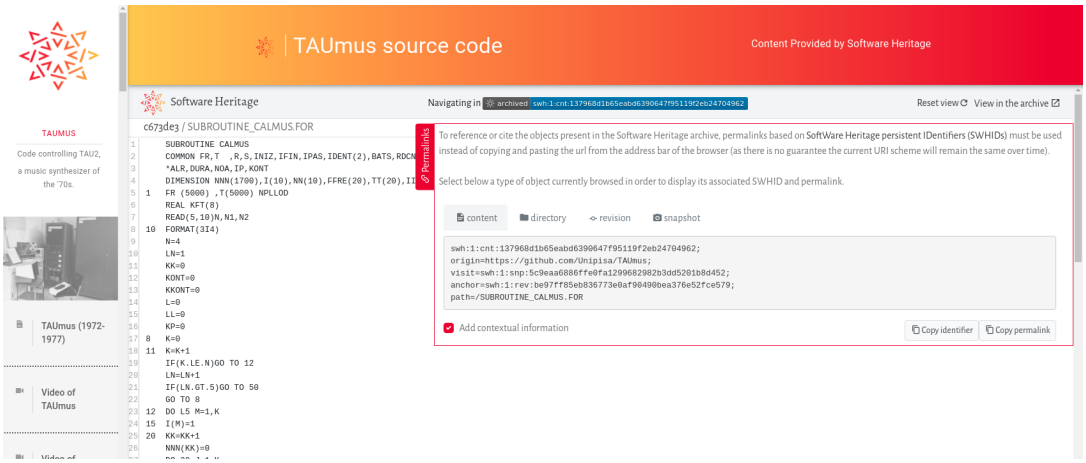

Figure 10: Browsing into the SWH source code moment in the TAUmus story

5 From curation to presentation

In the SWH stories there are two potential actors, the curator and the visitor. The former is responsible of creating the story, while the latter will benefit from the SWH story when visiting and interacting with the interface.

5.1 The curator workflow

The curator can create a JSON file with the description of the SWH story or interact with the publisher workspace as described in the scenario below and in the sequence diagram in figure 11:

1. The curator creates a workbench repository using the SWH template (part of the SWHAP workflow).
2. The curator uploads files into the raw materials directory.
3. The curator adds metadata for the software in the form of a codemeta.json file.
4. The curator searches Wikidata to see if an item for the software title exists.
5. If no item is found, the curator creates a new item.
6. The curator adds statements to the Wikidata item based on the codemeta.json file as well as any other available information using properties related to software.
7. The curator uploads images related to the software to Wikimedia Commons.
8. If there is only one image, the curator visits the Wikidata item for the software title and uses P18 to connect the image to the Wikidata item.
9. If there are multiple images, the curator creates a category for the software title in Wikimedia Commons and the returns to Wikidata and uses P373 to connect the category to the Wikidata item.
10. If there are images of files that were created with the software title, the curator creates a category in Wikimedia Commons for files created with the software title. The curator then returns to Wikidata and uses P7861 to connect the category for files created with the software to the Wikidata item.
11. The curator selects an interesting code fragment or directory and assigns anchor in Software Heritage archive.
12. The curator uses the Stories Services Publisher Workspace to add a SWH source code moment to the story and enters the URL or string of the anchor so that the moment will display the relevant fragment or directory.
13. The curator saves the story.

5.2 Curator's tools for the curation phase

In the SWHAP specifications there are four general phases, detailed in [2]: *Collect, Curate, Archive* and *Present*.

Software curation is necessary for the presentation layer and is composed with the following tasks:

- Rebuilt the development history:

  - ascertain versions
  - production dates
  - contributors for historical accuracy
- Collect descriptive metadata
  - project description
  - purpose
  - contextual information
- Procure legal authorisation
  - identify the owners
  - obtain the authorisations to publish the artifacts
- Organise metadata to be consumed for the Software Stories
  - Wikimedia Commons for images or videos
  - Wikidata for software descriptions and properties, etc.
  - open access repositories for research articles,

During the creation of the prototype, the curators from the University of Pisa used an inventory file listing for each software title the necessary materials for a story. The inventory file was created in the workbench repository. At this point, we would suggest that the file guiding the creation of the Softi story, linked above, should be refined and transformed into a template that essentially subsume the new process. Then, the issue is how to reconcile the *Catalogue* of the additional materials with this StoryInventory file, to avoid duplication, inconsistencies and useless work. A first attempt of the structure of the stories workbench is availble here.

5.3 Visitor's experience on Software Stories

After a SWH story is created, a visitor can view the story. Here is the scenario of this use case:

1. The visitor visits the SWH stories collection interface.
2. The visitor chooses a software title from the collection (using Qxxx).
3. The interface display a loading message.
4. The engine fetches the Wikidata properties for the chosen entity (data from Wikidata will always be received on the fly).
5. The engine analyses the response and fetches other content from Wikimedia Commons or other websites or repositories.

6. The engine checks for custom moments in the storage.
7. If a SWHID exists, the engine fetches the iframe content.
8. The engine builds the story with all existing moments and display the story on the interface.

6 Conclusion

The Software Stories project supported by UNESCO, is another action toward the recognition of software source code as a a key component of human creativity. In February 2019, UNESCO has published the Paris call [1] identifying a list of challenges facing software, and the importance of raising awareness among stakeholders. It is urgent and essential to collect, curate and preserve landmark legacy source code as mentioned in item 28 of the Paris call:

> *[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive*

The collaboration between Software Heritage, sciencestories.io and the curators from the University of Pisa with the support of UNESCO, has resulted in the first Software Stories prototype accessible on `https://stories.softwareheritage.org`. We will continue to complete the SWHAP specifications and promote both the Software Stories and the complete SWHAP. UNESCO's support of the SWHAP and the Software Stories is the best approach of enhancing global awareness of the importance of preserving and curating legacy source code, which in turn could determine the broad adoption by different cultural heritage stakeholders.

References

[1] Expert Group Report. Paris call: Software source code as heritage for sustainable development. Available from https://unesdoc.unesco.org/ark:/48223/pf0000366715, 2019.

[2] Laura Bussi, Roberto Di Cosmo, Carlo Montangero, and Guido Scatena. The software heritage acquisition process. Technical Report CI-2019/WS/8, UNESCO, Università di Pisa, Inria, 2019.

[3] Laura Bussi, Roberto Di Cosmo, Carlo Montangero, and Guido Scatena. Preserving landmark legacy software with the software heritage acquisition process. In *iPres2021 - 17th International Conference on Digital Preservation*, Beijing, China, 2021.

[4] Katherine Thornton and Kenneth Seals-Nutt. Science stories: Using iiif and wikidata to create a linked-data application. In *International Semantic Web Conference (P&D/Industry/BlueSky)*, 2018.

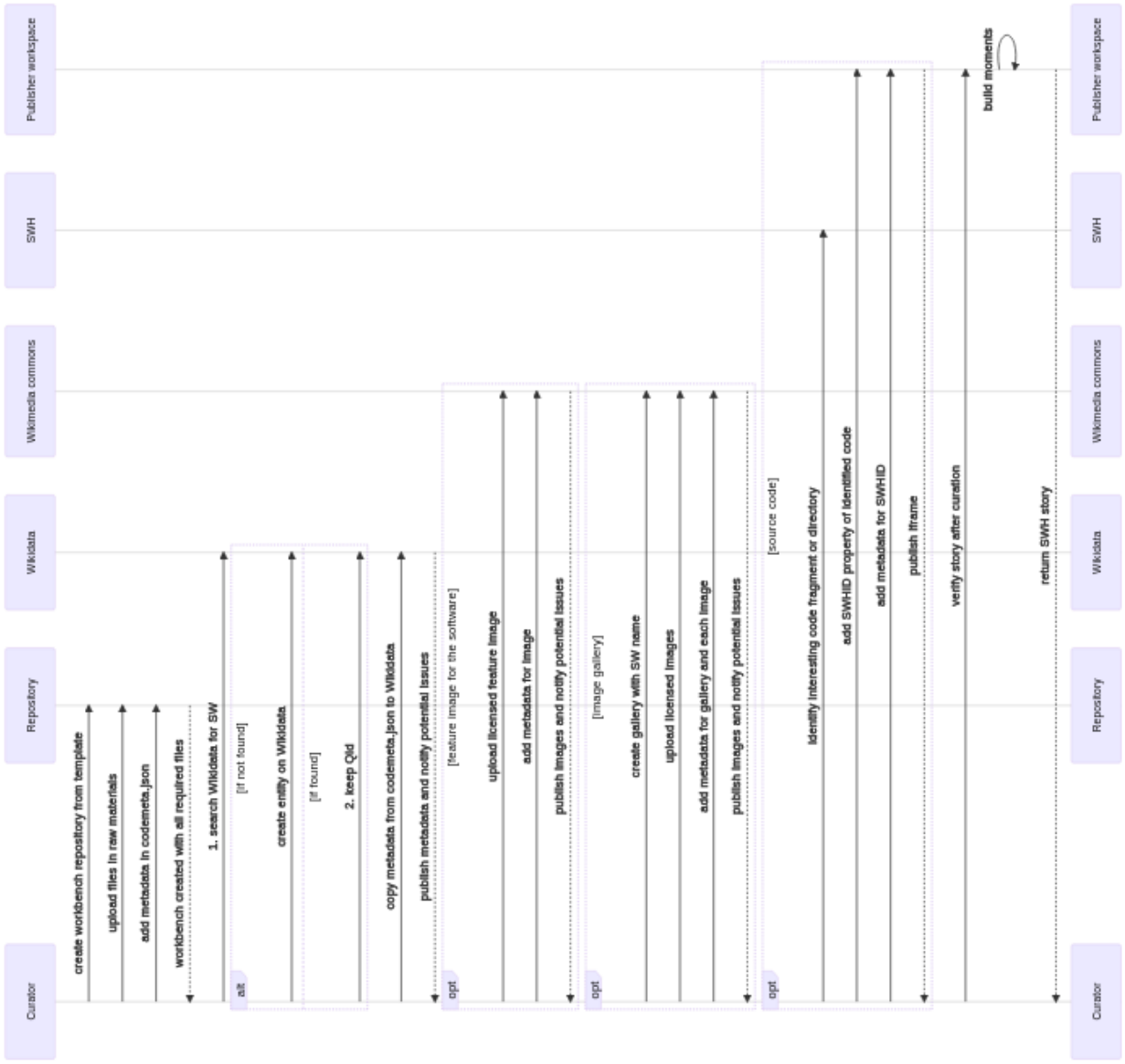

Figure 11: The workflow of the curator with the stories publishing interface

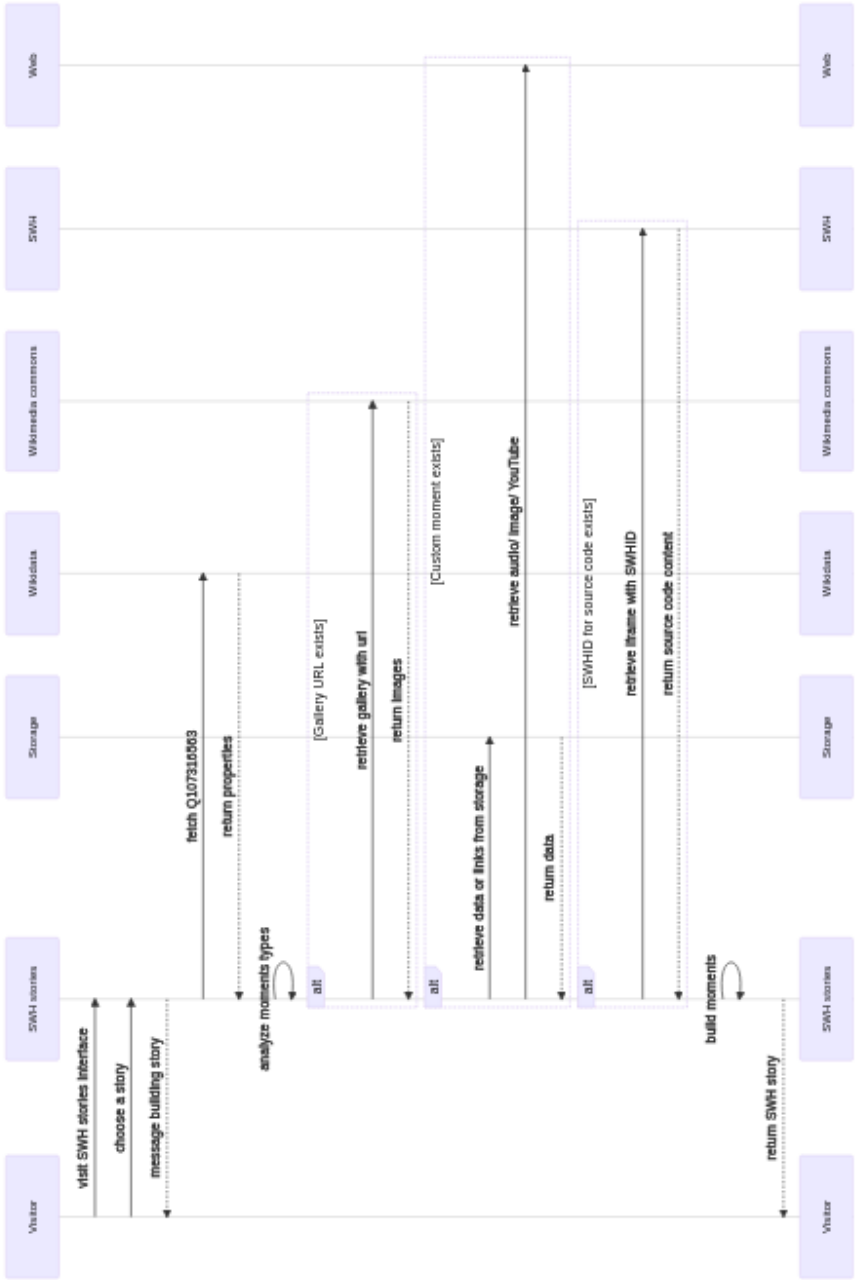

Figure 12: The visitor scenario when accessing the software stories interface