Long Term Preservation of Websites

Alexander Herschung

startext GmbH, Dottendorfer Straße 86, 53129 Bonn, Germany
Alexander.Herschung@startext.de

Abstract. While websites are of great interest for digital archives, the digital long term preservation of websites is a huge problem. The reason is that websites consist of a large number of file formats and require today's hardware and software environment to work properly. PABLO is a software tool that processes websites and transforms them into a dramatically simplified form that is simple enough for digital archiving and exhaustive enough to preserve the websites content and appearance and allows users to browse the entire site like the original.

Keywords: Digital Long Term Preservation, Website Archiving

1 Introduction

The challenge of really preserving websites for an unlimited period of time has been a standard example in Startext company to discuss the difference between storing files and real archiving since it is easy to store a website offline but just keeping this set of files is not sufficient to ensure access to the website in the far future. On the repeating discussions about possible ways to preserve websites one point of view emerged: "Maybe it's not really possible to archive websites. So let's define a walkthrough, a defined sequence of user actions and record what happens on screen as a movie file. This will not preserve the website. But it will preserve an impression of the website".

At this point the idea came up not to record a movie file but instead "let's take photos of every single page and save information about where links are and where they point to. Based on this information we will always be able to reproduce the core features of the website". That idea was the starting point in the development of PABLO. And this is exactly what PABLO does: creating a documentation of a web-site that allows to recreate the most essential features of the original website.

2 About startext company

Startext is a German software company, located in the city Bonn, founded 1980. Startext developes software mostly for cultural heritage organisations as archives and museums. Startext produces its own software products, such as inventory software for archives and museums and – today most important – a full featured digital long term preservation software, compliant to the international standard model OAIS.

Startexts digital archive software covers all aspects of OAIS, is delivered with a reday-to-use configuration of ingest (including virus-scan, format recognition and -validation and -migration) and targets also small and medium sized organizations allowing them to start with real digital archiving at a reasonable price.

Startext is also partner for customer-specific software development projects. On the 3rd of march 2009 the municipal archive of cologne was completely destroyed when the whole building collapsed into an opening hole in the ground. About 40km of archival good where affected, damaged, destroyed. Within two weeks Startext produced a whole new software to support the salvage operation. This software has since then been continuously improved and today covers even the full management and documentation of all efforts spent on restoring the damaged archival goods.

3 Regarding the digital long term preservation of websites

Websites could for sure be of interest for digital archives. Be it to archive a website that is to be closed down shortly or to document the evolution of a website by taking a snapshot e.g. once a year.

But digital long term preservation ("archiving") of websites is a problem class of its own. That is because one of the most important strategies to ensure digital long term preservation is to control and limit file formats that are stored in the digital archive. The transformation of complex file formats to more simple ones is probably the most important approach to increase the "archiveability" of digital content.

For text documents PDF/A is state-of-the-art standard archive format. For images it is uncompressed tiff. But what about websites?

3.1 Significant Properties

When it comes to significant properties, the question on which aspect of a website should be preserved, the answer has to be given for each specific website independently.

There are at least the following significant properties that could be relevant:

1. textual content – all texts presented within the website
2. appearance – how the website appears in a (today's) browser
3. interactivity – the way the website reacts and interacts with an user
4. "browseability" – the core feature of websites: allowing users to follow links

3.2 File Formats

Storing a website offline is simple. There are multiple tools allowing to do so. But what is really stored this way? It's the html source code with all attached file formats: css, java-script, linked images of multiple formats etc.. Most of these file formats are not suitable for digital archiving, they work only properly if their file-/folder-structure

is preserved too and – on top of it – this set of files does only produce adequate result in a today's browser running in a today's operating system.

Most of the websites interactivity is lost too because the underlying database is not preserved.

So what one gets with this approach is a snapshot of the website that only works in today's browser and today's operation systems.

3.3 Software archiving

So if one really wants to preserve a today's website one has to do a lot more than storing away a couple of html- and other files. One would have to preserve the html code, the underlying database, the content management system, the browser and all underlying operation systems both on server and client side.

This is not only archiving one website. It's software archiving. And software archiving is – at least – extremely difficult.

3.4 A different approach – PABLO

What if there would be a way to preserve a website's appearance, content and it's essential aspects of user experience in a much more simple form?

This question was the starting point in the development of PABLO.

At first, PABLO is a kind of crawler. While accessing a Firefox browser it crawls through a whole website (or a part of it, the scope of harvesting can be configured), opening every single page and creates an image file of the full page (taking a photograph of the page as it is displayed in the browser). In addition PABLO processes the page and determines where on this page links are located and where they lead to. The information about links is stored in a METS-XML-file (along with some other information such as e.g. the full text content).

As a result for each and every singe web page PABLO produces two (and only two) files:

- • an image file that preserves the appearance of the web page in a today's browser
- • an XML-file that preserves the position and target of links

This result is so simple that is readable and understandable even without any additional context information.

But on the other hand it allows to create a reproduction of the original website that preserves it's appearance, it's content and it's most essential behavior.

Fig. 1. How PABLO works

3.5 PABLO – features and configuration options

PABLO is a stand-alone software written in JAVA. It has no prerequisites and uses it's own included Firefox-browser. In its mot simple way to use it one enters only the websites URL, chooses the desired file format of the image files and the desired crawl-depth and then hits the start-button and watches PABLO doing it's job. But in reality sometimes life is more complicated.

Quite often the real size of the website is not known. An estimated 20,000 pages can easily result in PABLO finding more than 50,000 pages and still continuing to find more. In such case it can make sense to archive the website not as a whole but in parts. In order to do so one can configure PABLO to use only URLs of certain patterns (like "www.mywebsite/news/*") or exclude URLs of certain patterns.

One might also want to embed the archived website into its broader context. While PABLO usually restricts itself to one domain, it can be configured to follow external links too and include these linked external pages (but not their sub-pages) too.

During website harvesting it continuously writes two files holding the processed URLs and the found but still to be processed URLS (candidate URLs). These files can be very helpful in order to determine why a website I so much larger than anticipated. The above mentioned example with a website having more than 50,000 pages while about 20,000 were expected is taken from real life experience. The reason was found out by having a look at candidate-URLs: There was a news-calendar on the site allowing to click to the next or previous day, and the previous, and previous and so on to infinity. Solution was of course to exclude the calendar URL pattern from being processed.

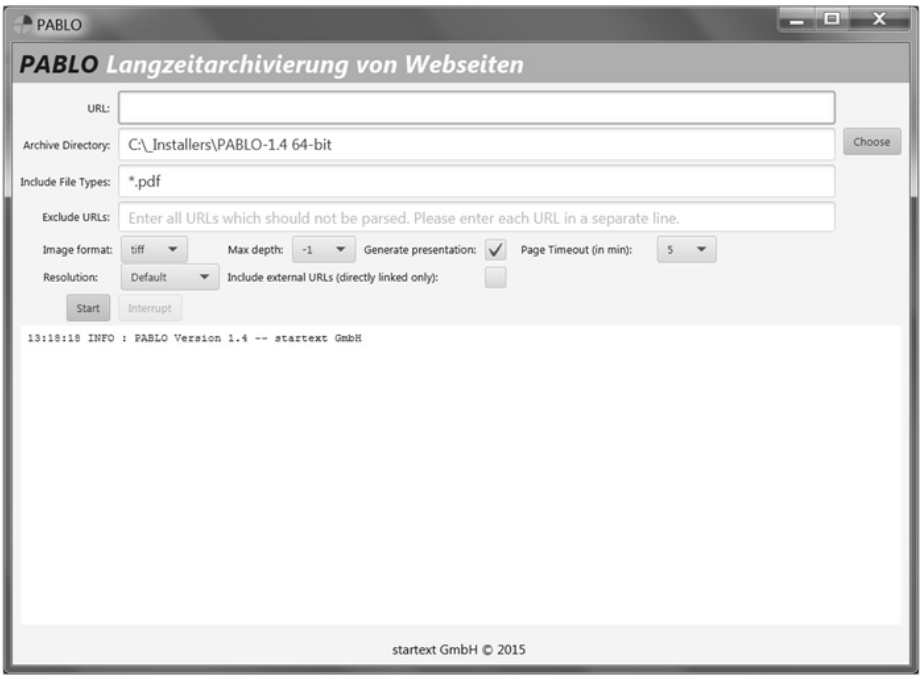

Fig. 2. PABLO user interface

3.6 PABLO – status and future prospect

Today PABLO provides a command line interface too to allow automated harvesting.

And it is able not only to find and follow simple html-links but also most of JAVA-SCRIPT-based links.

It produces not only the so called “archive form” of the website that was described above but also the so called “presentation form” which reproduces the website based on the produced image files and information stored in METS-XML-files.

PABLO also allows the user to specify file types (e.g. pdf) that should explicitly included into harvesting.

Startext currently works on allowing the harvesting of password-secured website areas too (the biggest problem turned out to be to prevent PABLO from clicking on logout-links).

The harvesting of video content such as linked youtube-videos is a future extension too.

Also startext works on including scripting capability into PABLO. Purpose is to be able to configure scripts that simulate user behavior into harvesting, e.g. entering search terms and hitting search-button. Such scripts would allow to execute a chosen sample of user actions and include their result into the harvesting.

One thing is certain: internet technology changes and evolves continuously. And so will PABLO.

References

1. http://www.staatsarchiv.sg.ch/home/auds/19/_jcr_content/Par/downloadlist_5/DownloadList-Par/download.ocFile/_Herschung,%20Alexander_%20Zur%20Langzeitarchivierung%20von%20Webseiten%20-%20ein%20L%C3%B6sungsvorschlag%20%5BPr%C3%A4sentation%5D.pdf
2. http://bibliothekartag2015.univie.ac.at/fileadmin/user_upload/k_bibliothekartag2015/pdf/BT15-6_3.pdf
3. https://opus4.kobv.de/opus4-bib-info/frontdoor/index/index/docId/2196
4. http://www.startext.de/produkte/pablo/pablo