https://doi.org/10.2352/issn.2168-3204.2018.1.0.13

Long term preservation of websites

Alexander Herschung, startext GmbH, Bonn, GERMANY

Abstract

While websites are of great interest for digital archives, the digital long term preservation of websites poses a huge problem. Given that websites consist of a large number of file formats they 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. The software allows users to browse the entire site like the original.

Introduction

The challenge of effectively preserving websites for an unlimited period of time has been a standard example in startext company to discuss the difference between storing files and actual archiving. 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.

Our solution is to take photos of every single page and save the information on where links are and where they lead 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.

About startext company

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

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

Regarding the digital long term preservation of websites

Websites are certainly of interest for digital archives. Be it to archive a website that is about to be closed 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 stored in the digital archive. The transformation of complex file formats to more simple ones is probably the most important approach to increase the "archivability" of digital content.

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

Significant properties

When it comes to significant features, the question of which aspect of a website should be maintained should be answered independently for each website.

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

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

File formats

Storing a website offline is very simple. There are multiple tools allowing doing 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 only work properly if their file-/folder-structure is also preserved.

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 present-day operation systems.

Software archiving

So if one really wants to preserve a contemporary 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 and not sure to last.

A different approach – PABLO

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

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

First of all, 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), opens 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 the location of links on this page 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 (see figure 1).

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

- an image file that preserves the appearance of the web page in a present-day browser
- an XML-file that preserves the position and target of links

This result is so simple that it can be read and understood even without any additional context information. These files are particularly suitable for digital long term preservation!

On the other hand it is so complete that it allows to create a reproduction of the original website that preserves its appearance, content and most essential behavior.

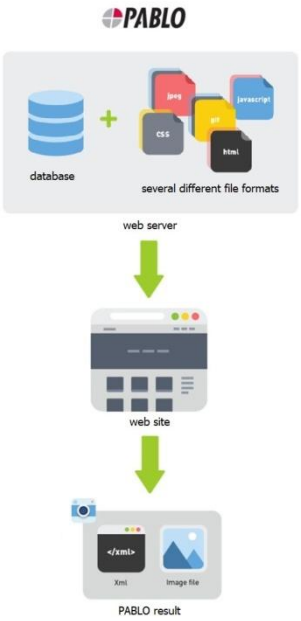

Figure 1. How PABLO works: The content of a web server is displayed on a website in a specific way → the PABLO result contains the crawling result with a XML- and an image-file corresponding to the web server's content and the look of the web site

PABLO – features and configuration options

PABLO is a stand-alone software written in JAVA. It has no prerequisites and uses its own included Firefox-browser.

The simplest way to use PABLO is to enter only the website's URL, choose the desired file format of the image files and the desired crawl-depth and then hit the start-button and watche PABLO doing its job (see figure 2).

But in reality sometimes life is more complicated.

Quite often the real size of the website is unknown, an estimated 20,000 pages can easily result in PABLO finding more than 50,000 pages and still finding more. In such a case, it may be useful 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. PABLO is even able to crawl through protected websites, as long as the password can be provided.

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 is 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 cause was found in the candidate-URLs: There was a news-calendar on the site allowing to click to the next or previous day, and the one before and so on to infinity. The solution was to exclude the calendar URL pattern from being processed.

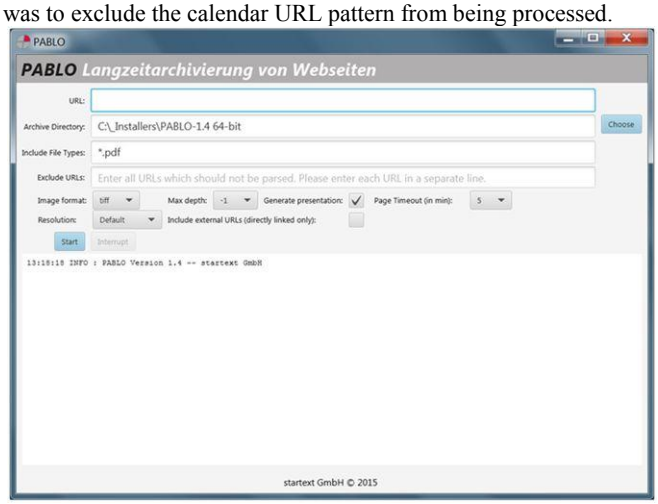

Figure 2. PABLO user interface

PABLO – status and future prospect

Today PABLO also provides a command line interface 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 the so called "archive form" of the website that was described above as well as 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 be included into harvesting.

The newest feature is the ability to harvest password-secured website areas (the biggest problem turned out to be to prevent PABLO from clicking on logout-links).

startext is currently working on archiving video content such as linked YouTube-videos as a future extension to PABLO. Another feature startext is working on, includes the scripting capability in PABLO. The purpose is to configure scripts that simulate user behavior when harvesting, e.g. entering search terms and hitting search-button. Such scripts would allow running a chosen sample of user actions and including their outcome into the harvesting results.

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

Author Biography

Alexander Herschung graduated in mathematics and psychology from the Rheinische Friedrich-Wilhelms-Universität in Bonn (1997). Afterwards he worked self-employed in the field of project development (until 2000). Since then he has worked at startext as a software engineer and as Head of Archive Software. He took over the company in 2014 and has been the managing partner since then. His work has focused on the automation in archives, libraries and museum; as of late the digitisation of the workplace as well as the private correspondence and its effects on archives has become his focal point.