

Server-side Preservation of Dynamic Websites

White Paper

July 2018

Prepared by Rasa Bočytė

University Supervisor Annet Dekker (UvA)

Internship Supervisors Erwin Verbruggen, Jesse de Vos (Sound and Vision)



Cite as: Bočytė, Rasa, Server-side Preservation of Dynamic Websites
(Hilversum, NL: Netherlands Institute for Sound and Vision, 2018).

This paper is published under the following license:



Table of Contents

| | |
|--|-----------|
| INTRODUCTION | 03 |
| • The Preservation Challenge | 04 |
| • Research Approach | 05 |
| DYNAMIC WEBSITES | 06 |
| CASE STUDIES | 08 |
| • Refugee Republic | 09 |
| • Taxodus | 10 |
| • Modular Body | 11 |
| CURRENT WEB ARCHIVING OPTIONS | 12 |
| WEBSITE SOURCE FILES | 16 |
| REVIEW OF BEST PRACTICES | 19 |
| • Collaboration with Creators | 19 |
| • Standards and Community Support | 20 |
| • Reduce Surface Area | 20 |
| • Disk Imaging | 21 |
| • Package | 21 |
| • Borrowing Practices from Software Developers | 21 |
| • Finding Useful Metadata | 22 |
| • Video documentation | 22 |
| RECOMMENDATIONS FOR THE ARCHIVAL INFORMATION PACKAGE | 23 |
| • Collaboration with Website Creators | 23 |
| • High-Level Approach | 25 |
| • Descriptive Documentation and Metadata for Digital Sustainability | 28 |
| • Documenting the Performance of a Live Website | 29 |
| CONCLUSION | 32 |
| LIMITATIONS AND FURTHER RESEARCH | 33 |
| APPENDIX 1: REPROZIP IMPLEMENTATION | 34 |
| APPENDIX 2: QUESTIONNAIRE TEMPLATE | 36 |
| WORKS CITED | 37 |

INTRODUCTION

The aim of this report is to offer new insights into the possibilities for website preservation. Server-side preservation complements large-scale web crawling solutions with an in-depth conservation approach for individual online productions. The method relies on the belief that digital preservation needs to start with website creators at the pre-ingest stage before the umbilical cord connecting the website to its server-side environment is cut off. Websites are complex ecosystems, most of which lie hidden under the surface of a webpage. A server-side preservation approach aims to include individual components of that ecosystem and dependencies between them in order to preserve a fully functioning and contextualised dynamic website. Since there are few publicly available examples of this strategy, the Netherlands Institute for Sound and Vision has recently initiated research around the subject. This report is the result of a five-month internship exploring the topic. It provides advice for the institute on the server-side preservation of dynamic websites, highlights further steps to be taken and offers a reflection on the challenges that might be encountered along the way.

While the institute already has a web archiving strategy that relies on crawling, it is not fit for every purpose. Crawling spiders are very good at ingesting large numbers of websites and capturing their numerous versions through time, but they can only scratch a website's surface as it is displayed through the browser. The institute has identified the need to preserve highly interactive web projects that use the latest online media technologies. They are therefore often "uncrawable" and functioning on the verge of obsolescence. A dynamic website that relies on complex server-side processes and user interaction falls through the crawling cracks and risks disappearing before any alternative preservation action can be taken. A server-side preservation strategy aims to fill in this gap and provide a solution that would allow the institute to safeguard such projects from extinction. To do that, however, it is necessary to understand what materials need to be acquired from website creators and what information is needed to support the digital object so that its full dynamic functionality could be preserved in the archive. During this internship project, I conducted research using three case studies in order to define what the archival information package for a dynamic website would have to include and how it could be created.

Server-side preservation promises to be a rather demanding and complex path, where the needs, availability and skills of all parties involved (creators, developers, production and funding bodies, archivists, IT specialists) need to be carefully juggled. Various trials and tribulations encountered while conducting this research only proved this point. However, the results promise to be unarguably rewarding. They will help Sound and Vision implement strategies to safeguard complex dynamic web productions. Results of this project expose some of the challenges that would have to be tackled by the institute and provide solutions that could be taken into consideration.

THE PRESERVATION CHALLENGE

In recent years, Sound and Vision has taken on an active role as a Dutch media landscape curator, expanding its collection policy far beyond linear audiovisual sources. In line with this, the institute has taken a prominent interest in preserving new media forms, ranging from web to 360-degree videos and interactives like games and online documentaries. Each medium comes with its own challenges, which have been explored in several research projects - de Vos (2013) and Verbruggen (2018) looked into the preservation possibilities for online documentaries, Cranmer (2017) discussed VR productions and 360-degree videos and Glas et al. (2017) delved into the world of games. Questions about server-side website preservation have largely emerged from two projects – the ongoing resurrection of *De Digitale Stad* ('The Digital City' or DDS) website and the collaboration with the International Documentary Film festival Amsterdam (IDFA) for the preservation of its DocLab programme nominees.

Efforts to resurrect *De Digitale Stad*, an online platform for the citizens of Amsterdam that was active from 1994 to 2001, presented an almost archaeological challenge - digging through the source materials that rely on now outdated systems in order to reconstruct the platform as it would have been experienced by its original users. The project culminated in the construction of two versions of the website – original source code run on an emulated environment and a replica of the code adapted to work on a contemporary system (Alberts et al. 147). While these two methods took very different perspectives, they both highlighted fundamental questions about digital objects that are composed of millions of files, scattered in various places and highly dependent on specific software or hardware processes - how to take control of all this? Which parts of a website, apart from its source files, need to be taken care of and which ones can be replaced? What preservation perspective should be taken – stay as authentic as possible and keep changes to a minimum or keep the digital bits up-to-date and refashion them to fit within the contemporary technological environments?

Whereas the *De Digitale Stad* project takes on the challenge of restoring a no longer working website, the institute's involvement with IDFA DocLab highlighted the urgency to look after contemporary online projects that could disappear any day now. An "Interactive Canon"¹ of 95 works was curated from the festival's programme to showcase the most innovative uses of new online media. Some of the projects, even though only a couple of years old, have already become inaccessible and no preservation actions have been executed yet. To avoid the digital archaeological excavation approach which, as in the case of *De Digitale Stad*, can last for years, the archiving community along with the creators call for actions that need to be taken today. In line with this, Sound and Vision needs to be ready to face the challenge and come up with solutions within this time frame while the works are still alive and well.

1 <https://www.doclab.org/tag/interactive-canon/>

RESEARCH APPROACH

Server-side website preservation as described in this report is a response to the above-mentioned challenges. Since the institute currently has no guidelines for how to deal with digital objects that fall under the broad category of dynamic and interactive websites, the goal of this project was to formulate advisory requirements for constructing the archival information package for dynamic websites. Within the OAIS (Open Archival Information System) model, which delineates a framework for developing digital archives, the archival information package plays a central role. It conceptually links “the object that is the primary focus of preservation together with all of the additional types of information (or metadata) that are necessary to support its continued use over time” (Day 190). In other words, the package maintains semantic links between the digital object and the information that is needed to understand and reconstruct it in the long term. This report investigates what exactly that digital object and the additional information should be in the case of dynamic websites and what form they should take.

Building on the challenges encountered in the aforementioned projects, the following questions also run through the report:

- What should be the boundaries of a website that is being preserved?
- What preservation strategy could be used for dynamic websites?
- What kind of technical, descriptive and preservation metadata should be generated and acquired from the creators to ensure long-term sustainability?
- How could the dynamic and performative elements of a website be preserved or documented?
- What could be gained from server-side website preservation?

Recommendations formulated during this research resulted from the review of best practices in consultation with other archivists and conservators working on similar projects and hands-on work with three case studies. I entered this project from a rather theoretical background, with little practical experience in the field, and had to acquire various technical skills along the way. This report documents the steps taken during this project and the challenges that were encountered in order to shed light on the possible problems and solutions that the institute might have to confront.

DYNAMIC WEBSITES

As a starting point of the research, I began by looking into some of the dynamic websites that Sound and Vision would be interested in preserving. The primary suspects for this selection were websites that could not be successfully archived with the *Archiefweb*² crawler, a tool Sound and Vision currently uses. Previous tests revealed that crawling errors occur mainly due to the use of JavaScript, Flash and videos embedded on web pages (Baltussen et al. 73-74).³ However, that is not to say that every website that uses JavaScript or Flash would be considered for server-side preservation. On the contrary, since in comparison to client-side web archiving or remote crawling, server-side preservation is a much more resource-intensive task, only a relatively small number of websites would be chosen for it.

This selection would be largely determined by the significant properties - elements of an object that are identified as central to its understanding and use - that the institute would choose to capture in each case. These properties need to remain persistent in the archived version of the website and here the archival information package should ensure their long-term preservation. The collection policy and the aims of the institute largely dictate what significant properties will be selected - in some cases, the primary interest might be in preserving the content that is published on a web page - audiovisual materials, text or links to other sources, whereas in others the whole look and feel will be important. In the latter case, if the dynamic elements like complex JavaScript animate that look and feel, the main concern should be in finding a preservation approach that would be capable of retaining it.

It all comes down to the purpose of each web archiving strategy and the significant properties that it can preserve. While crawling is aimed at collecting content that is published on the web and tracking how that content changes over time, it offers a very broad overview of a website as it is displayed through a browser. Since Sound and Vision is highlighting its role as a media curator, it needs to dig through this representational surface of the web and gain a better understanding of the way online media functions and the kind of possibilities it offers. The questions that one would ask are no longer only about what was published on the web, but how it was published and represented for users, how the stories were told, and what kind of creative possibilities were open to developers with the technology available at the time. As a curator, the institute wants to showcase the innovative examples of various online media expressions.

It is these innovative websites that are the focus of server-side preservation. In most cases, they will be stand-alone projects commissioned by various creative agencies. For the purposes of this research, I looked at websites funded by either the *Mediafonds*⁴ or by Dutch public broadcasters whose websites Sound and Vision has agreed to archive. I assembled a list of websites that had

² <https://www.archiefweb.eu/>

³ These tests were performed with an earlier Heritrix crawlers version (1.0), but the same issues, albeit to a lesser degree, persist with the Archiefweb tool which uses Heritrix 3.0.

⁴ Mediafonds was a fund dedicated to stimulating cutting-edge projects by the public broadcasters in the Netherlands. It promoted the development of high-quality productions across diverse media and genres, including TV and radio programmes, online documentaries, games and web videos.

already been identified by previous researchers as difficult to preserve with crawlers (fig. 1). It should be noted that I only considered productions that were still fully accessible online. That way live websites would serve as a point of reference in comparison to their archived versions and help identify errors that occurred because of the chosen preservation strategy rather than any issues associated with obsolete technology or corrupted files. After inspecting each website individually, I identified their dynamic characteristics that pose a challenge to current web archiving tools and that a server-side strategy would have to overcome.

One common thread that characterises all of these websites is that they are all suspended in a limbo state, waiting to be activated and engaged with to reveal their full potential. Their content is never static but is always in the process of becoming; media theorist Wolfgang Ernst refers to this as the “latency” of the digital where content reveals itself through dynamic processes (161-162). While static web pages reveal everything they have at the first glance – all content is visible as soon as a specific URL loads - a lot of content and its functionalities on dynamic websites are hidden until a user performs certain actions. Those might be as simple as the movement of YouTube videos around the web page in response to the mouse movement initiated by a user in the interactive documentary *Modular Body* or as complex as the creation of PDF files based on user actions and input in the online game *Taxodus*. It is this state of constant becoming that is central to dynamic websites and thus needs to be taken into consideration when thinking about what the archived website should look like and how it should function.

| WEBSITE | MAIN DYNAMIC ELEMENTS |
|--------------------------------|--|
| Andere Achterhuizen | Interactive images – a map with hovering elements and hyperlinks, zooming in option Embedded videos |
| Collapsus | Interactive video – additional content appears at specific points during the video Navigation between different content without interrupting videos or jumping to a new web page Google maps |
| The Last Hijack | Interactive timeline – various content attached to specific point on it Embedded videos Google maps |
| Taxodus | Interactive image – map with hovering elements and data attached to specific points on it Embedded audio Content generated through user interaction (database with PDF reports) |
| Refugee Republic | Interactive images – a map with hovering elements and hyperlinks, zooming in option Embedded videos and audio Content arranged in overlapping layers that unveil through user interaction (scrolling horizontally) |
| Modular Body | Embedded YouTube videos Content moving around the web page in response to mouse movement |
| The Sochi Project | Embedded videos Content arranged in layers that unfold when scrolling down the page Google maps |
| Waarom Srebrenica moest vallen | Content unveiling through user’s navigation (scrolling down the page) Embedded videos |
| Propaganda by the People | Content created and submitted by users (a database of drawings) Embedded YouTube videos |

Fig. 1 List of dynamic websites funded by *Mediafonds* and Dutch public broadcasters.

CASE STUDIES

Following the initial overview of dynamic websites, I decided to narrow down the focus of this research to specific case studies. Here the list of dynamic elements described above (fig. 1) suggested a possible approach. While it is not comprehensive or representative of all dynamic websites (and with newly created websites, more characteristics are bound to emerge), three categories of dynamic elements emerged:

- elements rendered with JavaScript or Flash (maps, timelines)
- external resources (YouTube videos, Google maps)
- content that is activated or created via user interaction (gameplay)

I decided that it would be best to focus on three case studies that would represent these three categories – *Taxodus* for user interaction, *Modular Body* for external resources and *Refugee Republic* for extensive use of JavaScript.⁵ My intention behind choosing three very different productions was to, first, examine how server-side preservation could be used to deal with a wide spectrum of dynamic properties and second, to determine whether some common guidelines could be formulated for very different dynamic websites. Following are brief descriptions of the case studies and some initial observations about their dynamic characteristics.

⁵ During this stage of the project, I got in touch with the creators of these websites and all of them agreed to participate.

CASE STUDY 1

REFUGEE REPUBLIC

Refugee Republic is an interactive online documentary that gives viewers a glimpse into the daily life of Camp Domiz, a refugee camp in northern Iraq. The main camp map (fig. 2) serves as a gateway into four virtual walking tours. Each walk is filled with overlapping illustrations, videos and audio recordings made at the camp. Horizontally scrolling through the web page, users are able to visit different areas of this territory and meet protagonists who share narratives from their daily lives.

The website is composed of many complex elements, each animated by different JavaScript functions. Their successful execution is a challenge to constantly updating software - some more recent browsers versions already have trouble successfully recreating certain functionalities. Linking the dynamic components of this website to software and hardware processes and maintaining these dependencies in the long term will be a major sustainability risk.

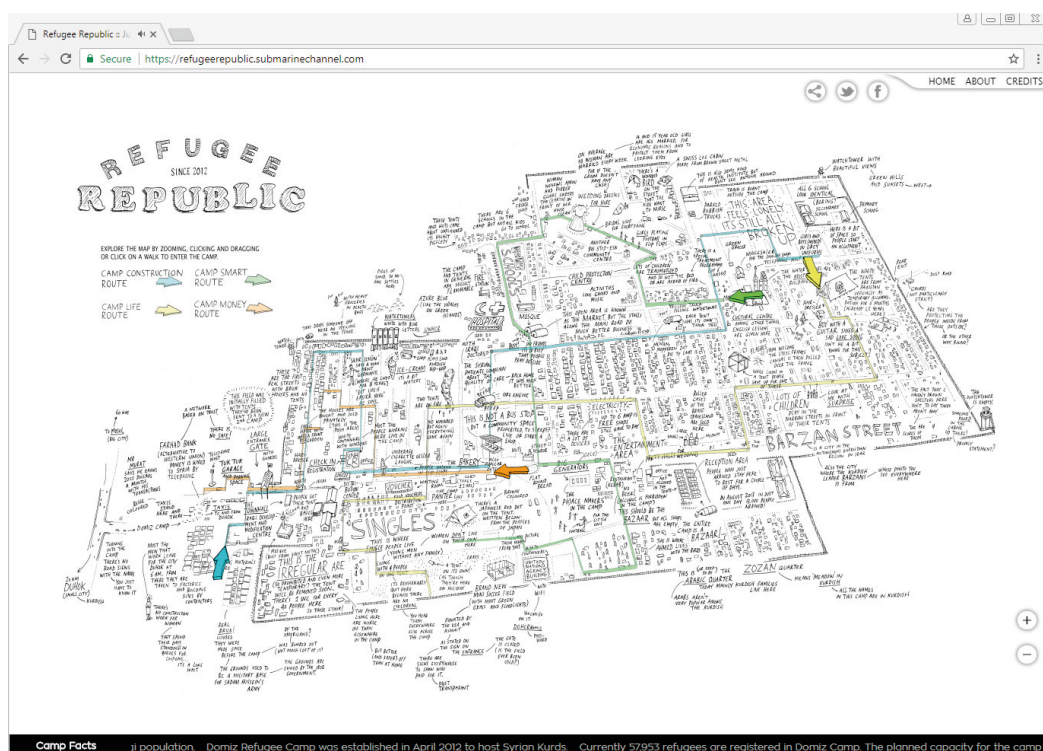


Fig. 2 Screenshot of the *Refugee Republic* website.

CASE STUDY 2

TAXODUS



Fig. 3 Screenshot of *Taxodus* website.

Taxodus (fig. 3) is a browser-based online game that puts its players into the position of large company CEOs who can manipulate their investments to escape from paying taxes. Each player can choose from a list of companies, most of which are known to have taken advantage of tax havens. Navigating through a world map, users can transfer their assets offshore, make treaties and establish private foundations in order to pay as little tax as possible. At the end of the game, a report is generated showing how the user performed.

Games as a genre put a lot of established preservation practices to a test. The user interaction, an essential part of gameplay that activates the website, needs to happen here and now, not in the past. For current web archiving practices aimed at automation rather than subjective and personal user experiences, games like *Taxodus* are unreachable. They demand practices that are not concerned with preserving the “original” but rather that document and give insight into what this experience would have looked and felt like.

CASE STUDY 3

MODULAR BODY



| Fig. 4 Screenshot of the *Modular Body* website.

Modular Body (fig. 4) is an online science fiction documentary that narrates the story of a rather controversial experiment - the creation of OSCAR, a living organism grown entirely from human cells. The story is pieced together from separate YouTube videos embedded on the website. Users can select the order in which they can view these videos and in this way, construct their own personal take on the narrative.

Of the three case studies, this website may seem to be the least technologically complex, yet its reliance on an external platform (YouTube) poses a great challenge for preservation – how to take care of something that exists externally, outside of the digital object? Is it possible gather all the materials and dependencies dispersed across multiple platforms and put them all into one package? A website like this encourages archivists to think about hybrid approaches where several strategies need to be combined like pieces of a puzzle.

CURRENT WEB ARCHIVING OPTIONS

The next step in the project was to investigate the extent to which current tools succeed or fail to capture dynamic websites. At the moment, crawling is the main web archiving strategy at Sound and Vision. Since the Autumn of 2017, the institute uses *Archiefwab's* tool WAD (Web Archiving Dashboard) based on the latest Heritrix 3 crawler version. It provides an easy, hands-off, large scale-strategy, something that is very appealing to archives and libraries that are looking for relatively undemanding workflows and immediate results. A crawling script collects all the materials that it can find published or linked on a web page and saves them to a standard WARC file format which can then be replayed on the web archiving dashboard. While it is admittedly a very attractive strategy, it should be clearly stated that it serves a specific purpose – it captures timestamped impressions of a website's content. Essentially, it freezes a version of a web page as it is represented through a browser at a particular point in time.

The Webrecorder⁶ tool is another possible web archiving strategy that Sound and Vision is interested in adopting in order to complement the current crawling workflow. Arts organization Rhizome developed the tool with a particular focus on dynamic websites to ensure that their “performance is preserved and replicable in the future” (Webrecorder). At the centre of Webrecorder's functionality is user interaction – the tool records all the transactions and content that a user engages with. While crawling is very much an automated strategy, Webrecorder is much more subjective and personal; it only captures the things that one browses through and clicks on.

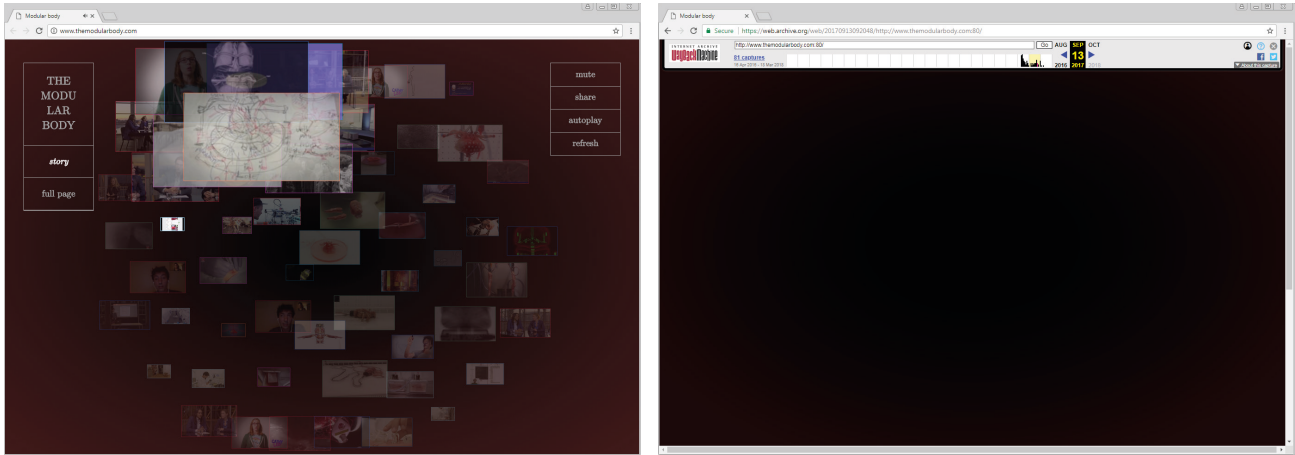
To see how “archivable” the selected dynamic websites were, I used the Internet Archive's Wayback Machine⁷, *Archiefwab's* harvesting tool that Sound and Vision recently started using and Webrecorder. I was not so much interested in comparing the results of these tools but much more in the different aspects of remote web archiving that each one of them highlighted - the highly automated perspective of the Wayback Machine, the more personal approach of Webrecorder and the collector's view from the *Archiefwab* tool.

As expected, a lot of content was lost during the collection process with all three tools. In the case of *Modular Body*, none of the YouTube videos that required connection to an external website were collected; for Refugee Republic, images and videos that were layered on top of the web page's background disappeared or were rendered incorrectly; any content that could only be accessed via user interaction on *Taxodus* remained out of scope (fig. 5-7). All of these vanished elements are not statically placed on a web page but rather are generated via certain performative processes - whether that is a connection to the live web or a database, an enactment of instructions in the source code, or user engagement. Both crawlers and Webrecorder remain indifferent to this dynamic execution of a website. It is essentially a semantic problem where the performance is lost in translation for web archiving tools that only speak the language of easily identifiable hyperlinks and changing URLs. Hyperlinks and URLs enable these tools to unlock content and follow the labyrinthic paths of a website. But paths on a dynamic website are not necessarily connected to the changes in its address line on a browser; indeed, one can go through the entire *Taxodus* game without ever leaving the homepage. These strategies attempt to tie the

6 <https://webrecorder.io/>

7 <https://archive.org/web/>

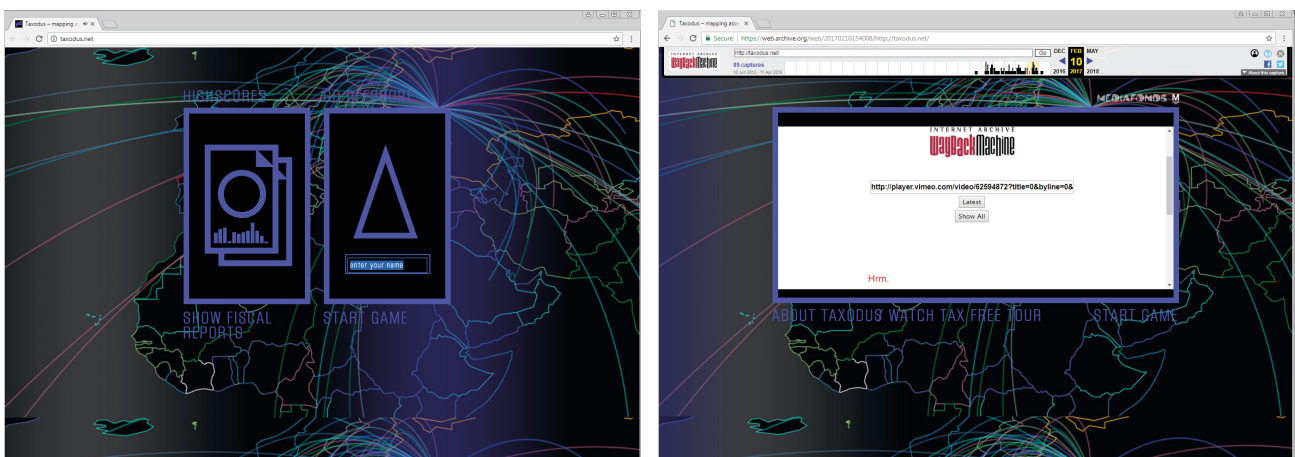
content of a website to specific timestamped URLs rather than the processes that execute them. If we are to maintain all of the functionality of these websites, it will be necessary to shift focus from timestamped content to these interaction-based processes.



| Fig. 5 Screenshots from the live *Modular Body* website and its archived version in the Wayback Machine.



| Fig. 6 Screenshots from the live *Refugee Republic* website and its archived version in the Wayback Machine.



| Fig. 7 Screenshots from the live *Taxodus* website and its archived version in the Wayback Machine.

Refugee Republic was the only website where Webrecorder significantly outperformed the crawlers. All of the videos, music and overlapping images were rendered correctly, which is largely due to the fact that all this content is placed along with other source materials on the web server, making it much more easily reachable. A lot of the material from *Taxodus*, on the other hand, is attached to databases and the *Modular Body* videos are stored externally on YouTube. This remote content remains out of reach for crawlers. However, a new problem emerged with Webrecorder. Once you start moving around the recorder website, going back and forth between the web pages, not following the same order that was used during the recording, some elements start disappearing or appearing in the wrong places (fig. 8). Again, this is related not so much to the content itself but the processes that gather that content onto a webpage. The way Webrecorder saves and then replays a website is completely different from the way it is executed via a web server. On a server, each element of a web page is stored as separate files that are then compiled anew each time via interaction between requests from the user side and responses from the server-side software. It is software that puts all these layers of the digital object together enabling its dynamic performance. Each time Webrecorder attempts to replay a specific URL, it looks for content that is tied to that URL, but miscommunication happens because it tries to assemble content using a different logic than the one originally used via server-side processes. Software and its connection to source files thus will be key components when it comes to the preservation of dynamic websites.

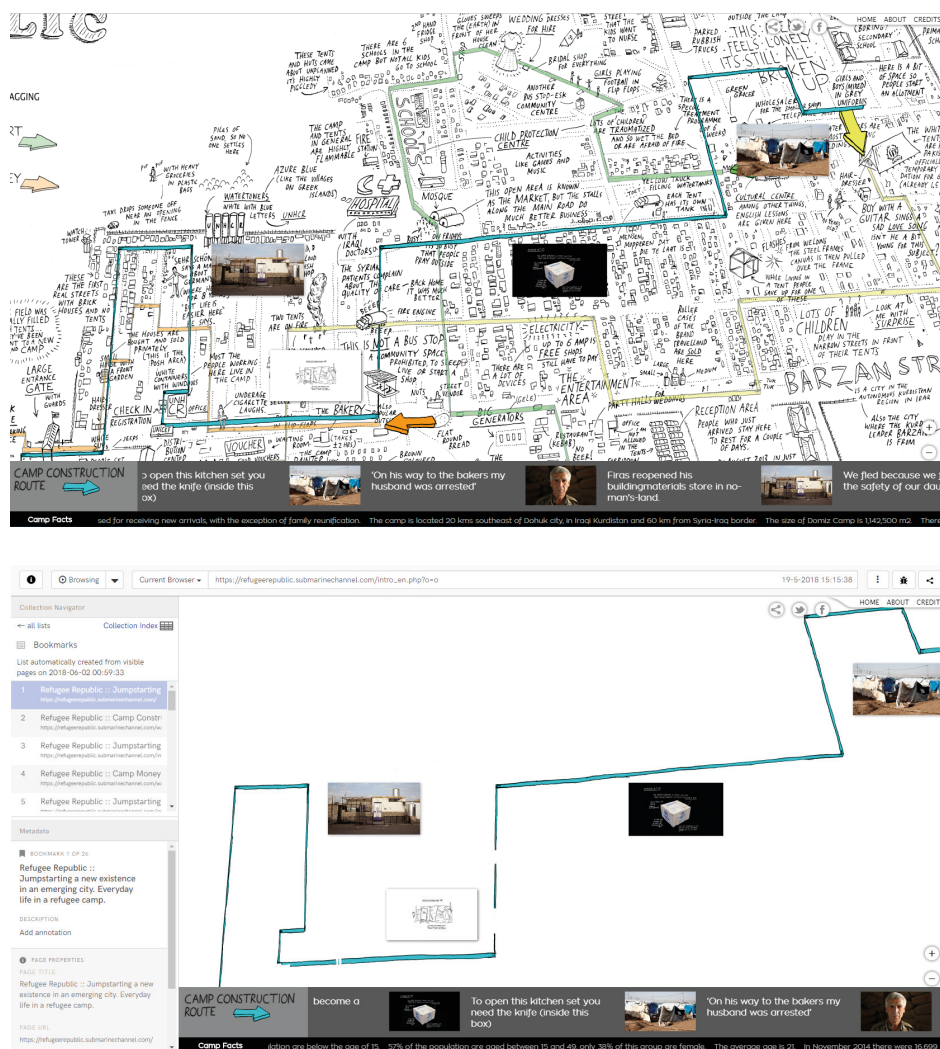


Fig. 8 Screenshots from the live *Refugee Republic* website and its archived version from Webrecorder.

Even if all the web pages could be perfectly rendered, the question that needs to be asked is what kind of contextual information is captured? The answer is that, unfortunately, only very basic metadata is harvested and it is primarily focused on the crawling process itself rather than the website that is being collected. The *Archiefw* tool displays the extent of the crawl (URLs that have been captured, whether external web pages and audiovisual media were collected, etc.), how much data was captured, when and how often crawling was performed, etc. This is very much a perspective that focuses on the collector rather than the website and its creator. Indeed, creators themselves might not always be aware that their content is being crawled.⁸ It is because of this hands-off approach that a lot of the contextual information never even enters into the picture. Literature often mentions the “deep web” (Pennock 11) or the “dark web” (Masanes 13) to describe the content that is locked under interactive elements like JavaScript or fill-in forms through which crawlers cannot penetrate. However, hidden in even more remote black boxes are website’s components that are not visible through a browser but that give invaluable insights into the creative and technical process as well as sociocultural contexts. The Web Archiving Metadata Working Group at OCLC have similarly concluded that crawling might not be the most appropriate time in the web archiving workflow to gather sufficient metadata, and extra steps need to be taken to properly contextualise websites (Samouelian and Dooley 9). That step is removing the barrier between the archival agency and the website creator and allowing the latter to take a more prominent role in the preservation decisions.

⁸ Creators can include a specific script that prevents crawler access; Sound and Vision’s policy is to send opt-out forms for website owners, and while permission is barely ever denied, at the moment of writing no active contact with creators was established.

WEBSITE SOURCE FILES

So how could server-side website preservation help to address the above-mentioned issues? As it is not a commonly used approach, there is little consensus on how it should be performed. Most authors describe it as a strategy where “files are copied directly from the server” (Pennock 7), i.e. migration to a new environment, and simply mention that some usability issues might result from the transfer (Masanes 27) but do not go into more detail. The focus in these descriptions is on a website as a collection of files rather than a uniform performative digital object.⁹ It is easy to copy-paste source files but reconstructing a dynamic website from them is another matter. I found this migration strategy a useful way to examine what it would take to recreate the execution of a dynamic website and, in the process, discover what needs to be included in the archival information package apart from the source files.

Early on during this research project, I received source files of the *Refugee Republic* website - over 30,000 files in various formats.¹⁰ Since I had little experience working with this type of digital objects, it was difficult to find a starting point – there was no executable file that one could easily open to inspect the website or any instructions how to set it up – just a collection of files distributed in many folders. To help me find the best approach for reconstructing the website, I consulted a couple of IT specialists at the institute. After the initial inspection, we discovered that the website was developed using PHP¹¹ and that all of the source materials were simply placed in folders rather than attached to a database or any other software package. This meant that I could perform some tests and set up the website using a generic open-source web server software.¹²

9 In fact, the only software tool I found in the literature recommended specifically for server-side preservation was DeepArc (<http://deeparc.sourceforge.net/>) developed by the National Library of France. It is not actually aimed at server-side preservation per se. As the creator of the tool Sara Aubry confirmed, it is meant to deal with a website's database contents such as repositories of ebooks or images and extract metadata about them. It offers a mix between server- and client-side archiving, where the crawled version of a website is combined with documents from a database that a website owner is asked to deposit. Aubry pointed out that since its development in 2005, the tool had only been used for some experiments but had not been picked up on a larger scale at the National Library of France or elsewhere, as the focus had mainly shifted towards web crawling tools (personal communication, March 21, 2018). It is not suitable for the purposes of this research either as database contents are not the primary concern here.

10 These files were received during a previous pilot project but nothing was done with them at the time.

11 PHP is a scripting language often used for the development of dynamic web pages. Well-known examples include WordPress, which relies heavily on PHP's server-side scripting language.

12 I performed all tests using my personal laptop. I installed Linux operating system (Ubuntu 16.04) since it is most commonly used for web servers and it was compatible with one of the software packages that is described in the later stages of this report. In the long run, however, the institute should set up a workstation dedicated to web archiving.

I used a PHP built-in web server to host the website locally on my computer without exposing it to an open network.¹³ While setting it up required minimal effort - just a few instructions executed via the command line - the result was not immediately satisfactory. Some of the content from the web pages disappeared or was not rendered correctly. The upside of this exercise was that a lot of these processes and errors were transparent – I had a clear indication of where in the source files, with the accuracy of an exact line number, the problem occurred (fig. 9); I could also observe this via the command line terminal where every executed process was indicated (fig. 10). Mostly problems emerged from certain libraries and extensions (e.g. SimpleXML) that were missing on my machine but were necessary to render certain types of audio-visual files; where it was obvious from error messages which libraries were missing, they could be easily installed. Another challenge was incorrect file locations in the source files; in a few places, PHP files pointed to either live resources or locations that would have changed when files were transferred to a new machine. With a help of one of the IT specialists at Sound and Vision, I was able to find the files where these locations had to be updated. However, we were not able to trace all the problems. This would have required a very intimate understanding of the construction of the website and its configuration in a specific environment, something that we did not have without consulting its developers.

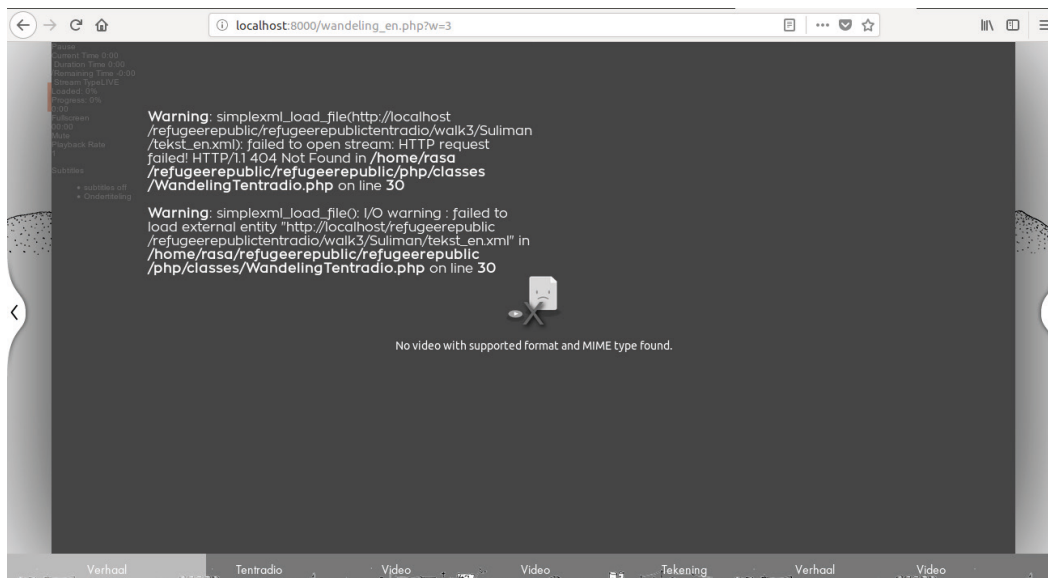


Fig. 9 *Refugee Republic* website test errors with PHP server.

¹³ The built-in server automatically comes with the installation of PHP, and is often used for testing websites during development stages.

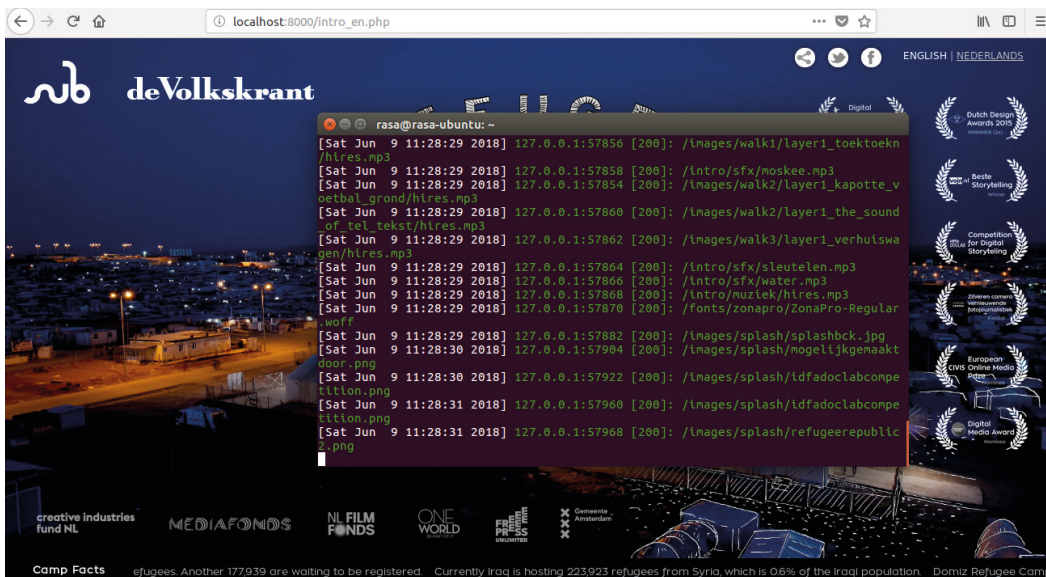


Fig. 10 *Refugee Republic* website test with PHP server with command line terminal showing the executed actions.

I tested a couple of other web server environments (XAMPP, LAMP) but the results were similar – again it required chasing errors and libraries that were needed to support the execution of all the dynamic elements on the website. It was a rather time-consuming process, one that demanded learning how to use new software tools each time and become intimately familiar with the whole website both from the client and the server side. Often, I did not feel competent enough to judge where the problems were coming from, let alone make any changes directly in the code. This migration approach would be difficult to implement on a larger scale in an archival institution. Nor could it be a sustainable solution; constant maintenance and further migration would be required to keep up with technological changes.

A higher-level approach is needed for website preservation. It needs to take into consideration all of the technical environment that perform a website – all of the software, specific libraries and settings that after the migration of files to a new environment I had to chase. Source files are tightly attached to software processes and environments - they are like life support mechanisms for a website. But they are important not only because of their ability to perfectly execute it; these environments themselves have significant archival value. After all, digital objects derive their meaning “from the sociotechnical framework in which they are created, rendered and experiences” (Konstantelos 235). In other words, source materials from the server-side environment are only one part of a much larger ecosystem. All of this ecosystem together is the primary digital object that needs to be preserved.

Of course, the challenge is that while source files usually come in one clearly defined folder, the whole environment is not contained in a ready-made package; it does not have clear-cut boundaries but is distributed all over the place and tightly attached to the systems that it is functioning in. Extracting it might result not only in its disconnection from the contextual information but also in the complete loss of functionality. It will be necessary to find a way to identify and document any dependencies that could prove to be a potential risk for the future sustainability of a website.

REVIEW OF BEST PRACTICES

During the next stages of the research, I explored how other institutes are handling complex website preservation and how their experiences could be adapted and implemented at Sound and Vision. Here I referred to the previous investigation of best practices¹⁴ and also consulted researchers and professionals working on similar projects. Since current efforts mostly concentrate on web crawling, the examples are few, especially amongst archival organisations. Examples I was able to find mostly dealt with the retrospective reconstruction of obsolete websites. Art museums and galleries, on the other hand, have explored this area more extensively. This comes as no surprise since server-side preservation requires in-depth work with each individual production, an approach that art institutions are accustomed to.

COLLABORATION WITH CREATORS

Creators of complex websites are the starting point of preservation. They have the most intimate understanding of the logic behind their website and the risks that should be considered during the preservation process. Art institutions have started several initiatives that try to simplify the collaboration between creators and conservators. Tools like the *Variable Art Questionnaire*¹⁵ and the *Artwork Documentation Tool*¹⁶ are aimed at encouraging artists to start considering preservation early on in their creative activities. However, while in art institutes the focus is on uncovering artistic intent, for archival purposes attention should be paid to the context and processes of production and creation. This is not something that can be captured or extracted from digital documents automatically; it needs to come directly from website creators.

As tests with crawling tools demonstrated, metadata extracted from websites pertains to a collector rather than a creator. One way that the above-mentioned tools highlight provenance and creative processes is through interviews with artists and submissions of sketches, working notes or any other materials that can contextualise their artworks. In the case of the University of Bologna website Unibo.it, which was excluded from the Wayback Machine, interviews with its creators who had an intimate understanding of the website's architecture and materials from various media outlets became the primary source for at least imagining its reconstruction and studying its technical and sociocultural contexts (Nanni par. 28-31). Even when a website can be fully preserved, these details would give invaluable insights to future researchers.

14 As part of the De Digitale Stad preservation project, Erwin Verbruggen conducted a survey of best practices for server-side preservation by posting questions via online forums and contacting selected individuals. Questions can be found here: <http://qanda.digipres.org/1137/ingesting-large-%26-hybrid-digital-collection>

15 <http://variablemediaquestionnaire.net/>

16 <https://www.li-ma.nl/adt/>

STANDARDS AND COMMUNITY SUPPORT

One very basic reason why server-side preservation has not taken off yet as a viable strategy in the archival institutes is the lack of standards around it. Here crawling has a major advantage over it, with a large number of companies offering web crawling services and a widespread community with a lot of experience to offer. Precisely because of this, many website preservation projects that could have adopted a server-side approach instead chose the client-side path. When the Stanford University Library initiated their reconstruction of SLACK, the first website in the US, contents from the backup files systems were reconstructed into web pages that could then be crawled and packaged into WARC files (AISum 285-286). The compatibility with widely used crawling tools and standardised file formats is an appealing solution, one that helps to address sustainability challenges in the long run. Compiling source files into neat WARC files provides a manageable approach, whereas finding standards for dealing with millions of heterogeneous files at the file level is nearly impossible.

REDUCE SURFACE AREA

As the test with *Refugee Republic* website demonstrated, setting up a dynamic website only from source files and migrating them to a completely new environment is not a viable solution that could ensure that the look and feel of a website can be fully preserved in the long run. A more comprehensive view of a website is needed, one that includes its surrounding environment. As McHugh et al. put it:

When the relationships between scripts, users, web services, databases and rights management systems become more intricate and integral, preservation becomes less akin to photocopying and more like performing organ transplant surgery, with all of the risks that digital materials will be 'rejected' within their anticipated preservation environment. (n.p.)

To minimise these risks of rejection, the most common approach, as Rhizome's preservation director Dragan Espenschied advised, is to "reduce the surface area" of the digital object that conservators need to look after and make the software and hardware incompatibilities as generic as possible (personal communication, April 18, 2018). There is no need to know the purpose of each individual file as long as the archivist knows how to make the digital object as a whole work. From an archival point of view, while it is important to keep the surface area of a record manageable and scalable, it is also important to think about giving entry points into the object so that it does not become impenetrable. In other words, the selected boundaries and surface area of the website should be able to highlight technical processes and decisions instead of hiding them.

DISK IMAGING

Emulation of disk images has become a preferred strategy in many institutes since it leaves few dependencies and minimises the surface area around the digital object - as long as the emulating platform itself is supported by hardware, the performance of a digital object could be reconstructed. The main appeal of disk images is that they can combine numerous software packages that are necessary to run and perform a website – server-side software as well as client-side browsers.¹⁷ Since Sound and Vision already has experience with disk imaging and emulation from game preservation, it could be a viable solution. It is a technologically difficult and time-consuming strategy but its applicability is wide-ranging and worth investing in.

PACKAGES

Another possible solution that focuses on the reproducibility of digital processes came from the investigation of software tools that could be employed for server-side preservation. ReProZip¹⁸ was recommended as a tool that could capture and reproduce web server environments. It can trace all the processes, data files, libraries and environment variables that are necessary to run a web server and package it all into a compressed file (.rpz). This package could then either be opened as any other compressed file or run using a containerisation platform to virtualise the web server in any environment. In a way it is similar to an executable file one would use to install a software programme, only here it includes not only the digital object but the environment that executes it as well.

The main difference between such a package and a disk image is its scope. Disk images are less selective, they capture all the content stored on a device, even the things that might be completely irrelevant. Packages created with ReProZip, on the other hand, are much more lightweight since they focus only on the specific processes that power the performance of a digital object. This specificity, however, might come at a cost - it might be rather complex to identify all the interwoven software and hardware processes that are needed to execute dynamic websites and not to leave anything out.

BORROWING PRACTICES FROM SOFTWARE DEVELOPERS

A big challenge to digital sustainability is what Schlieder describes as “semantic aging” - not only do the data formats, software and hardware get updated and become obsolete, but the knowledge that is needed to run and interact with them disappears as well (144). This semantic knowledge needs to be passed down to the future alongside with the digital objects; while the technology can be emulated, the knowledge and skills of archivists and users cannot. Documentation thus needs to serve as a sort of instruction manual that creates a link between the knowledge, technology and the preserved website.

One way to prepare for this digital aging, as recommended by the Guggenheim Museum’s conservator Jonathan Farbowitz, is to start learning from software developers and adapt their practices for specifying technological dependencies - README files, platforms like GitHub, etc. (personal communication, April 12, 2018). This type of documentation serves as a kind of score

¹⁷ Emulation as a Service initiative in particular could be of use here as it provides a scalable solution for such emulation; it offers to automatically set up all the processes that perform the digital object. See <http://eaas.uni-freiburg.de/>

¹⁸ <https://www.reprozip.org/>

that delineates how digital objects are to be performed; that performance might have to change depending on the environments available but the dependencies documented in the score should enable users to execute it nevertheless.¹⁹

FINDING USEFUL METADATA

A lot of technical metadata can be extracted automatically from a digital object, but that does not mean that all of it is useful for preservation purposes. This is still something that the website preservation community is still trying to figure out. There are important questions that need answering: what kind of metadata is actually useful? How much of it is needed and what is the best way to collect it?²⁰

The main focus currently is on documenting optimal running conditions. Art institutes, in particular, are moving away from the idea of an “original”- their conservation efforts for new media artworks are much more concerned with enabling creation of ideal conditions or a “second original” that is as close as possible to the initial performance of the object with the means available (Guez et al. 106). The already mentioned initiatives like *Variable Art Questionnaire* and *Artwork Documentation Tool* offer guidelines for documenting such conditions; a similar approach could be employed for dynamic website preservation.

VIDEO DOCUMENTATION

A user’s perspective is most commonly adopted to provide a glimpse into the look and feel of a dynamic digital object and its live performance. Different strategies are possible, each highlighting different aspects of a website and how it is experienced:

- The conservation team at the Guggenheim Museum created a walkthrough video for the recently restored Brandon website; here a curator navigates through the website explaining what interaction is possible.²¹
- The Dullaart-Sarkowski Method focuses on a more personal engagement with internet-based art, documenting what is on the computer screen as well as the user in an environment where the interaction is taking place (Dekker and Fauconnier).
- The game preservation project at Sound and Vision has also adopted a strategy to record user engagement via “Let’s Play” videos, where a player walks through a part of a game and records his or her experiences (Glas et al. 138).

19 This related to Richard Rinehart’s argument that media art should be treated in a similar manner to a musical work - it can be performed differently each time, but an abstracted score tells the performer (whether that is a human actor or a software component) how to achieve a result that retains the work’s integrity (181-182).

20 A recently started research project at the Guggenheim museum will be investigating these questions.
See <http://www.softwarepreservationnetwork.org/fcop/#fcop-team2>

21 See <https://www.guggenheim.org/blogs/checklist/restoring-brandon-shu-lea-cheangs-early-web-artwork>

RECOMMENDATIONS FOR THE ARCHIVAL INFORMATION PACKAGE

Following is a list of recommendations for steps that could be taken to create an archival information package for server-side website preservation. These were formulated from the above-described tests with case studies, literature review and the overview of the best practices. I used the three case study websites to exemplify how these recommendations could be implemented practically.

COLLABORATION WITH WEBSITE CREATORS

Server-side preservation needs to start with a collaboration with creators. The recommended approach is to conduct an interview with the creators to establish responsibilities on both sides, discuss preservation solutions and identify materials that creators would be able to provide. To simplify the process, Sound and Vision needs some basic guidelines that it could provide to the website creators, specifying what the institute needs. These should not be prescriptive and adjustments would have to be made and discussed in each individual case.

Interviews. To start off this collaborative approach, I contacted the creators of the three case studies to arrange meetings. Although all of them initially agreed to participate, unfortunately, Floris Kaay who developed *Modular Body* was not able to do so anymore. Maintaining contact with website creators will definitely be one of the main challenges of server-side preservation. While collaboration brings fruitful results, it is also a demanding process that not all creators will have enough time to commit to. This encouraged me to think about two things:

- How could the institute make this collaboration easier and more appealing to website creators? Having a clear idea of what exactly is expected of them is one of the key factors here. While conducting this research I was not able to promise any practical outcomes and was only exploring possible options. I could not, therefore, expect the creators to be fully committed. Contacting them with a defined list of documents and metadata that the institute requires should help establish more sustainable collaborations.
- How would the institute approach website preservation in the absence of a creator? A scenario where creators will simply transfer files but will not be available for further consultation is quite likely. Or, as with the case at hand, not even the source files will be received. Despite this, the institute could still consider alternative ways to document such websites (as discussed in the last part of this chapter).

Meetings that I was able to arrange with the creators of *Taxodus* (artist Femke Herregraven) and *Refugee Republic* (developer Aart Jan van der Linden and producer at Submarine Channel Remco Vlaanderen) proved to be particularly insightful. Their insider's perspective brought up important topics and concerns – the day-to-day maintenance of their websites, issues related to constantly updating technology, presentation of websites in exhibiting environments, etc. These conversations with creators give an opportunity to:

- Define responsibilities on both sides and agree on what each party can commit to. For example, in the case of *Taxodus*, it was interesting to see whether the artists wanted to preserve the database with reports generated by website users and whether new reports would be continuously updated and included in the archived version of the website.
- Find an optimum solution between what creators expect and what the collecting institution is capable of offering. Let us say, if a website relies on external materials, instead of the institute making that decision on their own, conversations with creators would help to find an appropriate solution. During the initial email exchange, the creator of *Modular Body* explained that he chose to embed YouTube videos on the website rather than use videos from source files in order to reach wider audiences and connect to the YouTube platform. If that was an important part of the website experience, it could be discussed whether, to complement the website, crawls of YouTube pages should be added to the archival information package as well.
- Identify technical aspects that need to be considered. I would recommend meeting with developers who maintain the website as they would be able to point to possible issues and technical challenges. For instance, for *Refugee Republic*, browser updates were identified as a constant problem, thus choosing a suitable browser version and incorporating it into the archival information package became an important aspect in the preservation.

Contextual materials. Creators of both websites seemed particularly keen to preserve information about the creative processes and the development of their websites. They wanted to share and document stories that are not visible on the website itself but nevertheless highlighted important social, political, technical and personal contexts, and provided insights into their creative decisions. During these conversations, we came up with a list of additional materials that the institute could acquire from the creators for the archival information package:

- Website user analytics that show how the users interacted with a website.
- Trailers, press materials, etc.
- Sketches, drafts or working notes showing the development of the website.
- If applicable, documents from website's presentation in exhibitions (photographic documentation, contracts, etc.).
- Interview videos with website creators (*Refugee Republic* creators suggested making a director's cut style video where they could share stories about the creation of their website, provide commentary, etc.)

Of course, this is not a comprehensive list. Interviews with each new creator would reveal how to best contextualise their productions and what materials are available in each case. The above-suggested documents could serve as guidelines of what the institute should ask for when approaching website creators. Some of the materials would definitely require more effort than others; as in the case of making interview videos about the creation of a website, it is not likely that many will have enough time to do this. But having these requirements available for creators to consult beforehand would hopefully encourage them to consider preservation from early on and incorporate it throughout their practice, which would, in turn, make the collaborative preservation efforts easier.

HIGH-LEVEL APPROACH

For the archival information package, the institute should use a high-level approach to acquire a website as a digital object. That is, the digital object that is the primary preservation concern should include source files of a website as well as the software and hardware environment that is necessary to execute them. Possible solutions for this are disk images and packages that contain all of these materials and dependencies between them. These would have to be created on the web server environment that the live website is running on. Ideally, it would incorporate both server- and client-side software. The format of the digital object should be open, sustainable and not restricted by proprietary software or hardware limitations.

Due to the time restrictions for the project, I could only try out one of the strategies.²² To properly test emulation, it would have been necessary to ask website creators to make disk images of their web server environments. But, unfortunately, after the initial interviews, it was difficult to keep in touch with the creators of Refugee Republic and Taxodus and I could not rely on them to make such disk images. Instead, I decided that it would be a valuable exercise to test ReProZip tool and see what results it would bring, how it would compare to disk imaging - a much more documented and researched strategy²³ - and determine whether it is worth further investigation. I already had the Refugee Republic website set up on my computer with the PHP's built-in server thus I could try to package and reproduce it. To perform the tests, I referred to the documentation available²⁴ and examples of the already packaged websites.²⁵ I also contacted the creators of the tool, Vicky Steeves and Remi Rampin, with specific questions (see Appendix 1 for a detailed description of how the tool works).

In the end, I was not able to successfully deploy the tool, primarily due to a lack of technical experience on my side; I created a package with all the source files and dependencies but was not able to reproduce it.²⁶ However, I do believe that with some extra time and help from an IT specialist it would be possible to achieve the desired results. Nevertheless, with the examples already available and my own test, I was able to identify the main characteristics of a package that is created with ReProZip.²⁷ My focus was on evaluating how achievable and sustainable this strategy is.

²² I did not have any practical experience with either of these strategies beforehand, therefore, it would have taken too much time for me to learn how to use them both.

²³ MoMA has gathered a list of useful resources for disk images. See <https://www.mediaconservation.io/disk-imaging>

²⁴ <https://docs.reprozip.org/en/1.0.x/>

²⁵ <https://examples.reprozip.org/>

²⁶ I can only speculate why this happened but my guess is that I did not trace all the processes properly.

²⁷ These were formulated in reference to the preservation attributes for file formats identified by Kim et al 2012.

ADVANTAGES

- **Open and transparent.** The package allows random access and its content can be easily inspected after decompression (fig. 11). Files can be extracted from the package, which means that if needed, it would be possible to access only source files or migrate them to a new environment.
- **Able to trace multiple processes.** Like disk images, ReproZip packages can include multiple software packages that are needed to execute a website. Which means that both server-side and client-side environments could be preserved (unfortunately, during the tests, I was not able to figure out how to do this).
- **Detailed metadata.** A configuration file that is created when constructing a package provides very detailed human-readable information about the hardware environment, all the software dependencies and libraries that are needed to execute a website (fig. 12). With disk images, extra steps would have to be taken to generate such metadata.
- **Generic dependencies.** The surface area of a package is defined by the availability of containerisation tools - as long as they are supported, a package could be reproduced. In the long run, it might be necessary to preserve this containerisation software and emulate operating systems and hardware that support it. But these dependencies are rather generic and would not require extensive customisation in each case. Interest in emulating operating systems is already very high, therefore it should not be a major sustainability challenge.

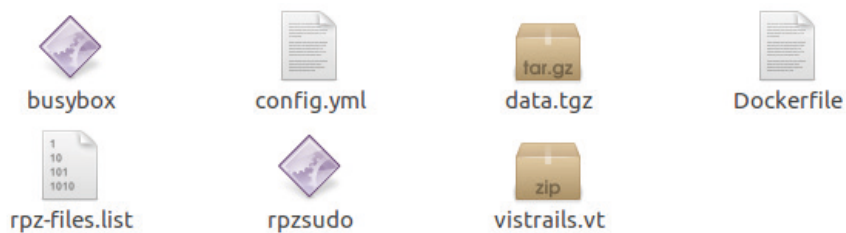


Fig. 11 Contents of the package created with ReproZip.

```
path: /home/isa/1etageel/epub/1etageel/epub/1etageel/images/splash/awad/00g.png
written_by_runs: []
read_by_runs: [0]

# Files to pack
# All the files below were used by the program; they will be included in the
# generated package

# These files come from packages; we can thus choose not to include them, as it
# will simply be possible to install that package on the destination system
# They are included anyway by default
packages:
- name: "base-files"
  version: "9.4ubuntu4.6"
  size: 319488
  packfiles: true
  files:
    # Total files used: 0.0 bytes
    # Installed package size: 312.00 KB
    - "/etc/host.conf"
- name: "libbsd0"
  version: "0.8.2-1"
  size: 162816
  packfiles: true
  files:
    # Total files used: 0.0 bytes
    # Installed package size: 159.00 KB
    - "/lib/x86_64-linux-gnu/libbsd.so.0"
    - "/lib/x86_64-linux-gnu/libbsd.so.0.8.2"
- name: "libc6"
  version: "2.23-0ubuntu10"
  size: 11215872
  packfiles: true
  files:
    # Total files used: 0.0 bytes
    # Installed package size: 10.70 MB
    - "/lib/x86_64-linux-gnu/ld-2.23.so"
    - "/lib/x86_64-linux-gnu/libc-2.23.so"
```

Fig. 12 Excerpt from the configuration file produced with ReproZip tool.

DISADVANTAGES

- **Limited flexibility.** The package could be opened with any containerisation platform on any operating system; during the test I used Docker²⁸ but an open source solution (e.g. Vagrant²⁹) could be used as well. However, currently, packages can only be created in Linux environments, which means that only a Linux-based web server could be preserved with this tool. Disk images are much more flexible in this respect.
- **Reliance on the command line.** In order to create a package with ReproZip, all the web server processes need to be executed via command line. While with the PHP's built-in server that was not an issue - it requires only a couple of very straightforward commands to start - tracing more complex environments would prove to be much more intricate, especially if specific settings need to be adjusted.³⁰ If a graphical user interface is usually employed to run a programme, it would be necessary to learn how to run it via command line.
- **Too much detail.** I mentioned detailed metadata as one of the advantages but it could also become too intricate and unnecessary. While it is important to make the processes behind the website's representational surface visible, this approach exposes an overwhelming amount of information – for example, every single library or file is listed in the configuration file. It is hard to make sense of it all today, let alone for future archivists who might not be familiar with the same technologies.
- **Limited scalability and steep learning curve.** Each time a new website is packaged, different steps would have to be taken depending on the web server software. Disk images are relatively easier to create and website developers could be asked to make them. However, it is highly unlikely they would be familiar with ReproZip, thus the institute would need an expert who would know how to apply it. The skills and knowledge needed to create these packages would put a lot of pressure on both the archivists and website creators.

My conclusion is that at this point in time, the institute should not spend time and resources on learning how to use it; while it looks promising, it needs to become a much more widely used approach to become a viable and reliable strategy with a community of users who could provide support if needed. Use of container technology for preservation purposes is still a developing practice - the standards around it are only starting to emerge³¹ and its sustainability in the long run still deserves more research.³² However, it is definitely something that the institute should keep an eye on and might be able to employ in the future. For the time being, disk imaging seems like the more promising option. As the next step after this research, the institute could reach out to several website creators to test how it would work exactly.

²⁸ <https://www.docker.com/>

²⁹ <https://www.vagrantup.com/>

³⁰ Dragan Espenschied who was consulted during this project commented that from his experience with ReproZip, the tool could not deal with "messy" web projects, where dependencies were all mixed up or not clearly articulated (personal communication, April 18, 2018); unfortunately, that might be the case with most of the dynamic website productions that Sound and Vision is interested in.

³¹ As an example <https://www.opencontainers.org/about>

³² Rechter et al. compiled a to-do list for maintenance and preservation of containers and concluded that the biggest risk is the obsolescence of the operating systems and hardware dependencies that are needed to support the containerisation software. It is not a future-proof solution and it might be necessary to emulate these environments. But this is an inevitable downside of any digital preservation solution.

DESCRIPTIVE DOCUMENTATION AND METADATA FOR DIGITAL SUSTAINABILITY

To support the sustainable preservation of dynamic websites, archival information packages for dynamic websites should include a description of the environment that is needed to run the digital object (e.g. a disk image) along with the instructions for setting it up. A questionnaire could also be developed for website creators to gather descriptive and preservation metadata about the live website and document optimal conditions that are necessary to execute its preserved version.

Environment Descriptions. If disk images were selected as the most appropriate format, description of what is needed to sustainably support them (operating system and hardware specifications) and instruction on how to set them up (emulation method) will be needed. Luckily, since the “surface area” of this file format is rather small and generic, very simple guidelines are needed and the same instructions could be applicable in many cases. Disk images of various digital objects would share the same environment settings. To simplify the process of describing these environments, a repository with such descriptions could be created so that each disk image could be linked to an appropriate description. Such a platform is already being discussed at Sound and Vision since it is needed for the preservation of other digital objects as well (games, *De Digitale Stad* project) and would thus be discussed in more detail elsewhere.

Questionnaire. From the review of best practices, it became apparent that a questionnaire is the most commonly applied approach to gather descriptive information. The purpose of such a questionnaire would be to capture metadata about the live website and assemble sufficient technical information that would clearly articulate optimal running conditions, dependencies and risks that in the future might endanger the lifespan of the preserved website. I reviewed currently available questionnaires³³ and attempted to design one that would fit the specific needs of server-side website preservation (see Appendix 2). Unfortunately, due to the already described difficulty in keeping in touch with website creators, I was not able to get feedback on this questionnaire. For now, the resulting questionnaire could be used as a reference to initiate further discussion.

The templates I was referring to were primarily designed for art galleries and museums. They focused on documenting installation conditions so that an artwork could be exhibited in various environments. They were quite detailed and offered a lot of room for open answers. Similar reproducibility issues are important from an archival point of view, however, with every metadata category I constantly found myself shifting between the two perspectives and questioning:

- Is this information really relevant and needed? For instance, is the information about the hosting services that creators chose relevant and has any value in the long term?

³³ Primarily, I referred to Artwork Documentation Tool and templates from the Matters in Media Art Project (see <https://www.tate.org.uk/about-us/projects/matters-media-art/acquiring-time-based-media-2008/templates-acquisitions0>), would document the essential information but also leave enough room for special adjustments.

- Do these questions need to be very specific or open? Clearly defined guidelines would help creators to orientate easier and better understand what is required from them but at the same time, they should be not be constricting. Open-ended questions (e.g. asking for any additional details) could be included but then there needs to be an easy way to deal with such answers and incorporate them with the rest of the metadata.
- How could it be rendered usable and easily understandable? One possible solution that could be employed to process all of this information is to create a README file describing all the optimum conditions that are necessary to run a website. An informal approach like that would document the essential information but also leave enough room for special adjustments.

DOCUMENTING THE PERFORMANCE OF A LIVE WEBSITE

The archival information package should include information that would showcase the performance of a live website. While it is not something that could be preserved as it relies on the live connection to the internet and user engagement, alternative documentation is possible. The recommended solution is to create a video recording (a screencast) or Webrecorder file that would showcase the look and feel of a live website and record how users would have interacted with it. This could also serve as a reference point in the future if the preserved website for some reason was no longer accessible or needed to be reconstructed.

With the three case study websites at hand, I considered what kind of documentation could be created in each case. The different dynamic characteristics and performative aspects of these websites ask for different documentation methods. I examined each website and evaluated possible options and their merits. Initially, I considered making a video for one of the websites, but it became apparent that in each case a very different approach would be necessary, and having one video documentation would not help to find solutions for all cases. Thus instead I focused on the process of how the institute could choose an appropriate type of documentation for each website. With each case study, I looked through its significant properties that need to be conveyed and identified problem areas that did not lend themselves well for documentation. Based on these, I made suggestions for the type of documentation possible and how it could be performed.

Taxodus

| | |
|----------------------------------|--|
| Significant Properties | gameplay, navigation of the map and decisions that are available for a player to make |
| Main challenges | a very large number of choices that each user can make; each player's experience would be different database with user-generated reports how much of the website's content to document? |
| Options for documentation | Screencast of a walk-through Let's Play type video with player's commentary |

Recommendations:

Playing a game like *Taxodus* is a personal and subjective experience, and so video documentation should reflect that. The goal would be not to show all of the possible actions one could take, but rather to give a general feeling of what is possible. *Taxodus* lends itself rather easily to this approach as although each user can make very different choices, the path one follows is the same each time – player chooses a company, then, using the map and the data provided, makes investment decisions, and after a specified time period a report is generated to show the results. When making a screencast, a player could follow this path and that should give a good understanding of how the whole website functions, without showing all of the content and all the possible decisions one could make.

There is an option to provide a user's commentary along with this interaction but it should be considered what information it should convey. In the above-mentioned example of the video made for *Brandon*, a curator describes the way the website works but also presents some contextual information about the images and texts used there. A similar approach could be applied to *Taxodus*, with a focus on the medium and technology used – a commentary would give a general overview of the website, even if not all of the content was shown (e.g. instead of opening every single document in the database with reports, the narrator could briefly describe it and show one report as an example), and describe how it works (connection to a database software on the server side).

Refugee Republic

| | |
|----------------------------------|--|
| Significant Properties | immersive audio-visual experience (background music, constantly changing and looping images and videos) storyline |
| Main challenges | background music videos narrative-based personal experience |
| Options for documentation | Webrecorder |

Recommendations:

Unlike *Taxodus*, *Refugee Republic* leaves very few decisions to users. The website is very flat and linear, there are four main walks on the map that one can follow. But this linearity does not make documentation any easier. Indeed, this requires intimate experience with the website, where a user needs to follow the story rather than jump between different pages. Here, documentation needs to focus more on the content rather than on the user's behaviour (which is primarily limited to scrolling). A walkthrough video like the one made for *Brandon* would not work here – it would take a long time to go through the whole website and show each video. A voice-over commentary would get in the way of engaging with the story.

In this case, it might be useful to come back to client-side web archiving strategies. While I discussed them earlier as not capable of providing adequate preservation solutions for dynamic web content, in this case, they could be considered as a type of documentation. While Webrecorder cannot recreate full performance of the website, it does give an overview of the content and the way one could engage with it, despite minor errors.

Modular Body

| | |
|----------------------------------|---|
| Significant Properties | user's ability to choose the order the videos are viewed and create a narrative from them connection to live the YouTube pages |
| Main challenges | YouTube videos number of choices available for each user |
| Options for documentation | screencast plus a Webrecorder version of YouTube pages. |

Recommendations:

As with *Refugee Republic*, playing each clip during the documentation video would be a task that would take a lot of time but the added value of it would be minimal. Perhaps a hybrid approach in this case would be more suitable; archivist could make a screencast with a commentary, showing how a user would be able to construct a personal take on the narrative from the videos. To complement this, YouTube pages with all the videos could be recorded with Webrecorder. This would allow the archivist to document both the content and the interactive experience from a user's perspective.

CONCLUSION

The archival information package for server-side website preservation will have to account for the multifaceted nature of dynamic websites. Tests performed during this research project revealed that no one-size-fits-all solution is currently possible for the server-side preservation. Different approaches will have to be chosen to support the specific characteristics of each website. Documentation of live websites will have to be adjusted in each case, different contextual information will be obtainable and, of course, the availability of website creators will prove a major factor in determining what kind of preservation solution is possible. Having said that, some common guidelines and standards for file formats and descriptions can be formulated. While I was not able to find precise answers to all of the questions that need to be answered, the process of looking for these solutions that is documented here should enable the institute to anticipate future challenges and be more prepared for them. With more experience, it will be possible to crystallise the requirements for the archival information package even further and find fitting solutions that work for both the archivists and website creators.

The recommendations outlined in this report should provide a foundation for the institute to venture into server-side website preservation. They reveal websites as complex ecosystems shrouded in complex dependencies between environments they are created, executed and experienced in. The suggestions for the archival information package made here will hopefully enable archivists to untangle these systems into something more graspable and sustainable.

LIMITATIONS AND FURTHER RESEARCH

Collaboration with creators proved to be a challenge during this research and it will remain so in the future. In terms of this project, it was unrealistic to expect a lot of involvement from the website creators since they were asked to voluntarily collaborate without any guarantees that their websites could be preserved. This encouraged me to work around these limitations and think of possible actions that could be taken in the meantime. Hopefully, this report has provided some useful guidelines that could help this collaborative process to move further. With a list of guidelines that the institute could approach creators with, it should become a less demanding process on both sides. Further interviews with website creators would also help to clarify the suggested approach.

Lack of sufficient technical knowledge also prevented me from successfully performing some of the tests. I would suggest that future research with emulation should be performed in collaboration with someone who has a more thorough understanding of how the technology works. Having said that, experimenting with unfamiliar approaches offered an extremely valuable learning experience. Finding ways to develop technical skills for digital preservation is definitely something that deserves more attention in archival institutes. As discussed, since best practices for the preservation of complex digital objects and software is currently being developed, it will be paramount to keep a close eye on new tools and practices that might offer more sustainable preservation strategies.

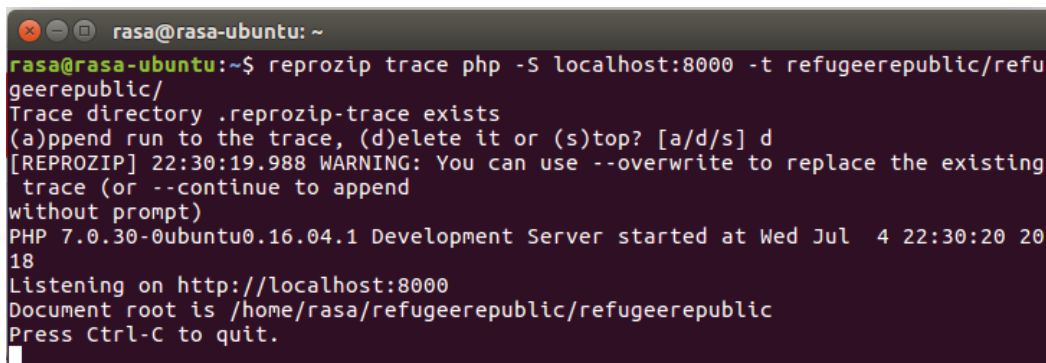
More experiments with innovative preservation approaches, even if they are not always successful, will help the institute to become more prepared for future challenges. Exploration and comparison of strategies that are available will be paramount to stay ahead of the game; instead of catching up with technology, Sound and Vision should encourage the exchange of such experiences and start a conversation about where the technology still needs to catch up with the preservation needs.

APPENDIX 1

REPROZIP IMPLEMENTATION

Packaging

To trace the processes that are needed to run the website, I started the PHP web server with an added “reprozip trace” command (\$ reprozip trace php -S localhost:8000 -t refugeerepublic/), then opened the website in a browser to ensure that it is executed and stopped the server (fig. 13).



```

rasa@rasa-ubuntu: ~
rasa@rasa-ubuntu:~$ reprozip trace php -S localhost:8000 -t refugeerepublic/refugeerepublic/
Trace directory .reprozip-trace exists
(a)ppend run to the trace, (d)elete it or (s)top? [a/d/s] d
[REPROZIP] 22:30:19.988 WARNING: You can use --overwrite to replace the existing trace (or --continue to append without prompt)
PHP 7.0.30-0ubuntu0.16.04.1 Development Server started at Wed Jul  4 22:30:20 2018
Listening on http://localhost:8000
Document root is /home/rasa/refugeerepublic/refugeerepublic
Press Ctrl-C to quit.
```

Fig. 13 Tracing the execution of the PHP server.

A configuration file was generated as a result of this containing all the information about the architecture of the system, the processes that were recorded and files that were executed (fig. 14).

```

# ReproZip configuration file
# This file was generated by reprozip 1.0.12 at 2018-07-04T22:33:50+02:00

# It was generated by the packer and you shouldn't need to edit it

# Run info
pack_id: "47fff8da-e8a5-42ea-86cc-230d62c68843"
version: "0.8"
runs:
# Run 0
- architecture: x86_64
  argv: [php, -S, 'localhost:8000', -t, refugeerepublic/refugeerepublic/]
  binary: /usr/bin/php
  distribution: [Ubuntu, '16.04']
  environ: {CLUTTER_IM_MODULE: xim, COMPIZ_BIN_PATH: /usr/bin/, COMPIZ_CONFIG_PROFILE: ubuntu,
  DBUS_SESSION_BUS_ADDRESS: 'unix:abstract=/tmp/dbus-0raaJm3XVo', DEFAULTS_PATH: /usr/share/
  DESKTOP_SESSION: ubuntu, DISPLAY: ':0', GDMSESSION: ubuntu, GDM_LANG: en_US, GNOME_DESKTOP,
  GNOME_KEYRING_CONTROL: '', GNOME_KEYRING_PID: '', GPG_AGENT_INFO: '/home/rasa/.gnupg/S.gpg
  GTK2_MODULES: overlay-scrollbar, GTK_IM_MODULE: ibus, GTK_MODULES: 'gail:atk-bridge:unity-
  HOME: /home/rasa, IM_CONFIG_PHASE: '1', INSTANCE: '', JOB: unity-settings-daemon,
  LANG: en_US.UTF-8, LANGUAGE: en_US, LC_ADDRESS: nL_NL.UTF-8, LC_IDENTIFICATION: nL_NL.UTF-
  LC_MEASUREMENT: nL_NL.UTF-8, LC_MONETARY: nL_NL.UTF-8, LC_NAME: nL_NL.UTF-8, LC_NUMERIC: n
  LC_PAPER: nL_NL.UTF-8, LC_TELEPHONE: nL_NL.UTF-8, LC_TIME: nL_NL.UTF-8, LESSCLOSE: /usr/bi
  %s %s, LESSOPEN: '| /usr/bin/lesspipe %s', LOGNAME: rasa, LS_COLORS:
  'rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:m
  MANDATORY_PATH: /usr/share/gconf/ubuntu.mandatory.path, PATH: '/home/rasa/bin:/home/rasa/.
  sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin',
  PWD: /home/rasa, QT4_IM_MODULE: xim, QT_ACCESSIBILITY: '1', QT_IM_MODULE: ibus,
  QT_LINUX_ACCESSIBILITY_ALWAYS_ON: '1', QT_QPA_PLATFORMTHEME: appmenu-qt5, SESSION: ubuntu,
  SESSIONTYPE: gnome-session, SESSION_MANAGER: 'local/rasa-ubuntu:@/tmp/.ICE-unix/1687,unix/
  SHELL: /bin/bash, SHLV: '1', SSH_AUTH_SOCK: /run/user/1000/keyring/ssh, TERM: xterm-256co
  UNITY_DEFAULT_PROFILE: unity, UNITY_HIS_3D_SUPPORT: 'false', UNITY_START_EVENTS: 'gnome3'
```

Fig. 14 Extract from the configuration file produced with ReproZip.

It also pointed to all the files that were not captured during the tracing process but that needed to be included into the package to reproduce the website (fig. 15); the configuration file could be edited at this stage to add the necessary location paths.

```
- "/lib/x86_64-linux-gnu/libz.so.1.2.8"

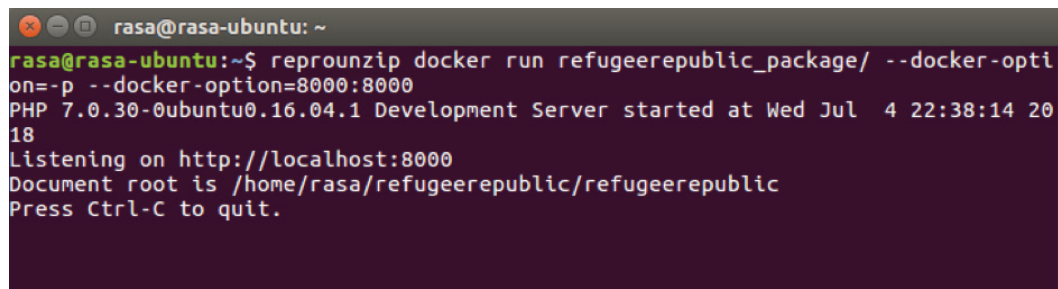
# These files do not appear to come with an installed package -- you probably
# want them packed
other_files:
- "/etc/alternatives/php"
- "/etc/hosts"
- "/etc/ld.so.cache"
- "/etc/localtime"
- "/etc/nsswitch.conf"
- "/etc/php/7.0/cli/conf.d/10-mysqlnd.ini"
- "/etc/php/7.0/cli/conf.d/10-opcache.ini"
- "/etc/php/7.0/cli/conf.d/10-pdo.ini"
- "/etc/php/7.0/cli/conf.d/15-xml.ini"
- "/etc/php/7.0/cli/conf.d/20-calendar.ini"
- "/etc/php/7.0/cli/conf.d/20-ctype.ini"
- "/etc/php/7.0/cli/conf.d/20-dom.ini"
- "/etc/php/7.0/cli/conf.d/20-exif.ini"
- "/etc/php/7.0/cli/conf.d/20-fileinfo.ini"
- "/etc/php/7.0/cli/conf.d/20-ftp.ini"
- "/etc/php/7.0/cli/conf.d/20-gettext.ini"
- "/etc/php/7.0/cli/conf.d/20-iconv.ini"
- "/etc/php/7.0/cli/conf.d/20-json.ini"
- "/etc/php/7.0/cli/conf.d/20-mcrypt.ini"
- "/etc/php/7.0/cli/conf.d/20-mysqli.ini"
- "/etc/php/7.0/cli/conf.d/20-pdo_mysql.ini"
- "/etc/php/7.0/cli/conf.d/20-phar.ini"
- "/etc/php/7.0/cli/conf.d/20-posix.ini"
- "/etc/php/7.0/cli/conf.d/20-readline.ini"
- "/etc/php/7.0/cli/conf.d/20-shmop.ini"
- "/etc/php/7.0/cli/conf.d/20-simplexml.ini"
- "/etc/php/7.0/cli/conf.d/20-sockets.ini"
- "/etc/php/7.0/cli/conf.d/20-sysvmsg.ini"
- "/etc/php/7.0/cli/conf.d/20-sysvsem.ini"
- "/etc/php/7.0/cli/conf.d/20-sysvshm.ini"
```

| Fig. 15 Extract from the configuration file produced with ReproZip.

When the editing is finished, the package can be created (`$ reprozip pack refugeerepublic`).

Unpacking

ReproZip package can be reproduced using several containerisation platforms. In this case, I used Docker but other options are available. Docker was deployed via command line to open the package (`$ reprozip docker setup refugeerepublic.rpz refugeerepublic/ | $ reprozip docker run refugeerepublic/ --docker-option=p --docker-option=8000:8000`). It then automatically set up the packaged web server (fig. 16).



```
rasa@rasa-ubuntu: ~
rasa@rasa-ubuntu:~$ reprozip docker run refugeerepublic_package/ --docker-option=p --docker-option=8000:8000
PHP 7.0.30-0ubuntu0.16.04.1 Development Server started at Wed Jul  4 22:38:14 2018
Listening on http://localhost:8000
Document root is /home/rasa/refugeerepublic/refugeerepublic
Press Ctrl-C to quit.
```

| Fig. 16 Running the ReproZip package with Docker.

APPENDIX 2

QUESTIONNAIRE TEMPLATE

| | |
|--|---|
| BASIC DESCRIPTION | Title: Creators and Developers: Producers: Dates of creation: Dates of the live website: Description: Interactive, Documentary, Online gaming, Online game, etc. |
| ENVIRONMENT | Software: Version, open-source/proprietary/custom Operating system: platform, version, etc. Hardware: processor, RAM, etc. |
| OPTIMAL RUNNING CONDITIONS | Server-side: specific software or hardware settings Client-side: particular browser version, elements that need to be supported (e. g. Flash, Javascript, specific video/audio formats, etc.) |
| CREATION & PRODUCTION PROCESSES | Programming languages used: Software: version, open-source/proprietary/custom Hosting services: |
| RISK ASSESSMENT | Risks: components that require constant maintenance and are easily susceptible to technological changes, proprietary software/hardware |
| ADDITIONAL NOTES | |

WORKS CITED

- Alberts, Gerard, et al. "Archaeology of the Amsterdam Digital City; Why Digital Data Are Dynamic and Should Be Treated Accordingly." *Internet Histories*, vol. 1, no. 1–2, Jan. 2017, pp. 146–59.
- AlSum, Ahmed. "Reconstruction of the US First Website." *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries*, Knoxville, 2015, pp. 285-286.
- Baltussen, Lotte Belice, et al. "Hard Content, Fab Front-End: Archiving Websites of Dutch Public Broadcasters." *Alexandria: The Journal of National and International Library and Information Issues*, vol. 25, no. 1–2, 2014, pp. 69–91.
- Cranmer, Candice. *Preserving the Emerging: Virtual Reality and 360-degree Video, an Internship Research Report*. Netherlands Institute for Sound and Vision, 2017.
- Day, Michael. "The Long-Term Preservation of Web Content." *Web Archiving*, edited by Julien Masanés, Springer, 2006, pp. 177-194.
- de Vos, Jesse. *Preserving Interactives. Preserving Audio-visual Materials in a Post-broadcasting Paradigm*. VU University, 2013.
- Dekker, Annet, and Sandra Fauconnier. "Documenting Internet-Based Art: The Dullaart-Sakrowski Method." AAAAN.NET, aaaan.net/documenting-internet-based-art-the-dullaart-sakrowski-method/.
- Ernst, Wolfgang. "Order by Fluctuation? Classical Archives and Their Audiovisual Counterparts; Technomathematical and Epistemological Options in Navigating Trans-Alphabetical Archive." *Archives in Liquid Time*, edited by Rienk Jonker, et al., Stichting Archiefpublicaties, 2017, pp. 160-174.
- Glas, René et al. "Playing the Archive. 'Let's Play' Videos, Game Preservation, and the Exhibition of Play". *The Interactive Past. Archaeology, Heritage and Video Games*, 2017, pp. 135–151.
- Guez, Emmanuel, et al. "The Afterlives of Network-Based Artworks." *Journal of the Institute of Conservation*, vol. 40, no. 2, May 2017, pp. 105–20.
- Kim, Yunhyong, and Seamus Ross. "Digital Forensics Formats: Seeking a Digital Preservation Storage Container Format for Web Archiving." *International Journal of Digital Curation*, vol. 7, no. 2, Dec. 2012, pp. 21–39.
- Konstantelos, Leo. "Documenting the Context of Software Artworks through Social Theory: Towards a Vocabulary for Context Classification." *The preservation of complex objects. Vol. 2, Software based art*, edited by Leo Konstantelos et al., University of Portsmouth, 2012, pp. 18-32.
- Masanés, Julien. "Web Archiving: Issues and Methods." *Web Archiving*, Springer, 2006, pp. 1–53.

McHugh, Andrew, et al. "Reflections on Preserving the State of New Media Art." *Proceedings of the Archiving Conference*, Den Haag, 2010, pp. 170-175.

Nanni, Federico. "Reconstructing a website's lost past Methodological issues concerning the history of Unibo.it." *Digital Humanities Quarterly*, vol 11, no. 2, 2017, <http://www.digitalhumanities.org/dhq/vol/11/2/000292/000292.html#d46639e627>.

Pennock, Maureen. *Web-Archiving*. DPC Technology Watch Report, 13-01, 2013.

Rechert, Klaus, et al. "Preserving Containers – Requirements and a Todo-List." *Digital Libraries: Knowledge, Information, and Data in an Open Access Society Lecture Notes in Computer Science*, 2016, pp. 225–230.

Rinehart, Richard. "The Media Art Notation System: Documenting and Preserving Digital/Media Art." *Leonardo*, vol. 40, no. 2, 2007, pp. 181–87.

Samouelian, Mary, and Jackie Dooley. *Descriptive Metadata for Web Archiving: Review of Harvesting Tools*. OCLC Research, 2018.

Schlieder, Christoph. "Digital Heritage: Semantic Challenges of Long-Term Preservation." *Semantic Web*, vol. 1, no. 1-2, Jan. 2010, pp. 143–47.

Verbruggen, Erwin. *Preserving Interactives*. Netherlands Institute for Sound and Vision, 2018. Webrecorder. Webrecorder, 2018. webrecorder.io.

