Archiving dynamic websites: a considerations framework and a tool comparison

Allard Oelen
Student number: 2560801

Supervisor: Victor de Boer
30 June 2017
Vrije Universiteit Amsterdam
a.b.oelen@student.vu.nl

Abstract. Websites need to be archived to prevent loss of knowledge. Sites are becoming more dynamic, and traditional archiving techniques for static website are not sufficiently able to capture a website with all of its functionalities. In this research, considerations of archiving dynamic websites are identified and put into the Web Archiving Considerations Framework. The features of the framework have been determined using semi-structured interviews that were conducted at a national audiovisual archive. The framework can be used for setting up new web archiving projects, as well as evaluating existing archiving approaches. Additionally, three web archiving tools (Brozzler, Heritrix and Webrecorder.io) are empirically compared with each other, to find out which tool is best able to capture dynamic web content. The comparison has shown that Webrecorder.io performs best in preserving the functionalities of dynamic websites.

Keywords: Web Archiving, Digital Preservation, Dynamic Web Archiving, Web Archiving Considerations Framework

1 Introduction

Archiving always had an important role in understanding the past and constructing the future [12]. This also applies to archiving the World Wide Web. Websites should be archived to prevent loss of knowledge [5] and to pass on this knowledge to future generations. In 1996 web archiving was initiated [3], but due to the development of web technologies, new archiving approaches are necessary. Sites nowadays do not consist solely of static text and images, but also of dynamic generated elements, for example using JavaScript. Archiving these sites using traditional methods can result in captures that are not functioning correctly, as these methods are often not able to capture dynamic content. For an archived website, it should be possible to reproduce the look and feel of the original website. If the dynamic functionalities are left out, this is no longer
possible. Without the ability to reproduce this way of presenting the content, its expressive power is lost.

In this paper, the considerations of archiving dynamic sites are going to be identified. This is done by constructing a framework that lists those considerations. This framework is constructed by interviewing researchers who are involved with archiving in general. The interviewees have different roles in the archiving field, and will therefore have different views on what is important for web archiving. To ensure the completeness of the framework, all these views are taken into account. The framework is used to answer the first research question: *What are considerations of archiving dynamic websites?* Afterwards, the framework is applied to describe the implications of its features on three existing archiving tools. In addition to this tool evaluation, the tools are going to be compared with each other. This comparison will focus on how well the tools are able to capture dynamic websites. With this comparison, the second research questions will be answered: *How do existing web preservation tools differ from each other?* The findings of this research can be used to ensure that the most important considerations are taken into account while examining web archiving approaches. Additionally, the tool comparison can be used as a guidance for selecting an archiving tool.

### 2 Related work

Websites are becoming more dynamic due to the adoption of several technologies. JavaScript is making sites more dynamic by enabling client-side interaction. Because of this, JavaScript has an impact on the archivability of websites. In a study of Brunelle et al. [1] this impact is measured. According to this research, client-side rendering was not part of web archiving in the past. But because the web has changed from online documents to applications, archiving JavaScript (and other related technologies) is becoming more important. One of the findings of this research, is that JavaScript is responsible for 52.7% of the missing content in their tests. This implies that capturing JavaScript is an important factor in archiving dynamic websites. The findings of Brunelle et al. are relevant for the tool comparison in this paper. Since JavaScript is largely responsible for missing content, and thus for incomplete archives, the archiving tools can be compared on how well they are able to capture JavaScript.

Mannerheim [2] describes a web preservation approach where the publisher of the website is responsible for the achievability of the site. If this approach would be adapted, archiving websites would become easier since achievability is something that concerns the web developer as well. A drawback of this so called “institutional approach”, is that each publisher or web developer will use his own solutions for archiving. This will result in a diversity of archiving standards, which does not benefit the achievability of websites. A standard is needed for making a site suitable for archiving. Web developers could use the method of Banos et al. [6] to measure the achievability their website. The method is called CLEAR and it presents an online tool which can be used to estimate the likelihood that a website could be archived correctly. The web developers subsequently can improve the achievability of their site based on the results of this method.
According to Brunelle et al. [4] the amount of missing resources is not a good indicator to determine the damage of a memento (an archived version of a website). It has turned out that the perception of the web user is more important than the actual damage. This finding can be used for the framework that is going to be constructed in this paper. Since perception is an essential indicator for damaged mementos, one missing CSS file (responsible for the styling of a website) can damage the memento more than multiple missing JavaScript files (which mainly provide functionality of a website).

Archiving a website once is not sufficient since websites continue to change over time. Web pages will be added or changed, and these changes should be archived as well to ensure the completeness of the archive. Brewington et al. [7] did research on how dynamic web pages are. Dynamic in this context means how often the pages are changed. This research focused on search engines, and how often a search engine should re-index a site. The findings are relevant for archiving as well, since archiving tools also need to make a decision in how often to crawl and store a website. Denev et al. [8] introduce the term blur to indicate the quality of an archived website. Blur arises if a website is changing after the initial capture. To repair a blurred archived version, the site should be re-captured, to make the capture sharp again.

Different approaches of website archiving are proposed by researchers in the web preservation field. In the end, those approaches and considerations are important to include in the framework. The whole breadth of archiving should be taken into account to ensure a complete framework. This research complements the existing literature by providing a high-level overview instead of a detailed analysis of specific components.

3 Research method

The research is divided into three parts: 1) conducting semi-structured interviews to discover features, 2) clustering the features into a framework, 3) making a tool evaluation and comparison.

3.1 Discovering of features using interviews

To find out which features should be included into the framework, interviews have been conducted. Interviews were the most appropriate method for this research. With a questionnaire, a larger amount of people could have been reached and for the participants it would have been less time consuming. However, for this research most of the features were not known on beforehand. Different participants have different views on archiving, and a questionnaire is not suitable to further explore those different views. On the other hand, a focus group could have led to general consensus between the participants. Since the goal of the interviews was exploring different points of views, this was not desirable.

The purpose of the interviews was to make sure that different views of archiving were considered, therefore semi-structured interviews were conducted. An unstructured interview would be more appropriate if the topics were completely unknown on beforehand. On the other hand, a structured interview would be more appropriate if the exact
topics were already known. In the end, a semi-structured interview offered the right balance between open and closed interviews. The topics were determined on beforehand likewise most of the questions. To ensure the full breadth of archiving, all interviews had the following structure: 1) what is archiving, 2) what is digital archiving, 3) what is web archiving. This structure was chosen so the interviewees would not only provide their view on web archiving, but also on archiving in general. A question like: “when do you call something an archive?”, is relevant for web archiving because the actual goal of archiving becomes clear. Most of the questions have been asked to every interviewee, so the provided answers could be validated against the answers of the other interviewees. Some questions were focused to the background of the interviewee. For example, during the interview with the software engineer, more technical questions about capturing specific web technologies were asked. The interview script can be found online [13]. The semi-structured interview setting allowed it to further explore considerations that were not determined on beforehand.

The interviews were conducted at The Netherlands Institute for Sound and Vision1. This location has been chosen because all interviewees are employees of this institute. In total four people have been interviewed. The interviews took on average 45 minutes per interviewee. All participants agreed with being recorded. Therefore, no hand-written notes have been taken. The interviewees also gave consent to publishing the transcripts of the interviews online. The full recordings are available online [13].

3.2 Construction of the framework

To construct the framework, potential features were extracted from the interviews. For each interview, the proposed features of archiving were listed. Per feature, quotes of the interviewee were added to substantiate the validity of the feature. Afterwards the features per interview were merged together. The result of this was a list with features, complemented with: the quotes, which interviewee proposed the feature and a description of how to interpret this feature. Afterwards, features that were solely relevant to the Sound and Vision institute were removed from the feature list. The more interviewees proposed a certain feature, the most important this feature was considered. The 16 most important features were included in the framework. After the creation of the feature list, four main topics were determined. These topics form the basis of the framework, each topic contains three to five features. The features were shortened to a description of one sentence that was used to build the final framework.

3.3 Implications of framework and tool comparison

The implications that the framework has are evaluated with respect to three existing web archiving tools. Most of the implications are the same for the tools, and therefore the results of the evaluation are put together into one table. Per feature the implications are described by performing tests on the tools, consulting the tool documentation or by

1 http://beeldengeluid.nl
using literature. For one feature a more detailed evaluation has been performed by empirically comparing the tool with each other. Based on literature, the following tools turned out to be the most prominent ones and were therefore selected for the evaluation:

1. Brozzler\(^2\) – A web crawler with the capability of capturing media. Uses warcprox\(^3\) to make the tool suitable for web archiving.
2. Heritrix\(^4\) – A web crawler specially designed for web archiving. Written and used by The Internet Archive. The tool is able to capture large amounts of websites (called broadcrawling) [11].
3. Webrecorder.io\(^5\) – An integrated platform for archiving websites together with a graphical user interface (GUI) to browse and share the archived sites.

These tools have in common that they are able to archive a website and that they output the archived site as a WARC file. The tool comparison has shown the differences between how well each tool is able to capture dynamic web content. Four quantities were determined to compare the tools on. The quantities can be measured on a scale from one to five. Each tool has been tested against the same set of five websites. These sites can be found in section 4.4. This set is provided by Sound and Vision and is considered as hard to archive due to the dynamic functionalities of the sites. To scope the research, five websites are selected out of the complete set of 39 sites. This set of five is considered to be representative to the full set of websites. Each site has been archived by the three tools. Afterwards the captured sites were empirically evaluated and scored based on the four quantities. The results of this process are put into tabular form, and a total score per tool has been calculated.

4 Results

4.1 Features extracted from the interviews

The total duration of the four interviews was 180 minutes. Based on the interviews a list of proposed features is composed and four topics are determined. Each feature is categorized into one of these topics. The topics are listed below, together with an explanation of what considerations are related to this topic.

- **Scope** – What should be archived?
- **Accessibility** – How should the archives be accessed?
- **Dynamic archiving** – How to capture dynamic content?
- **Miscellaneous** – Features that do not fit into the above-mentioned categories

In Table 1 proposed features and properties of archiving are listed. These features are classified by the previously determined topics. The interviewees are referred to as:

\(^2\) https://github.com/internetarchive/brozzler
\(^3\) https://github.com/internetarchive/warcprox
\(^4\) https://webarchive.jira.com/wiki/display/Heritrix
\(^5\) https://webrecorder.io/
• I1 – Researcher Interactive and New Media
• I2 – Software Engineer
• I3 – Coordinator Ingest
• I4 – Media Manager Ingest

Table 1. Features proposed during the interviews

<table>
<thead>
<tr>
<th>Topic</th>
<th>#</th>
<th>Description</th>
<th>Interviewees</th>
</tr>
</thead>
</table>
| Scope   | F1.1| Capture should function as close to the original site as possible. This has consequences for the archiving techniques (e.g. archiving sites using screenshots is not considered sufficient).   
  – “An archived website is a website as I would visit it today” I2  
  – “You want to reproduce the look and feel of a site” I3 | I1; I2; I3; I4 |
|         | F1.2| Most decisions about the scope of archiving (e.g. the depth of crawling, what to archive, the frequency of archiving) are depending on the purpose of the archive (do you want to capture a large number of sites in a lower quality, or a small number of sites in a higher quality).   
  – “Depending on the case of what you want to archive, you go on to a certain level of depth” I2 – “[Questions about the scope can be answered by asking:] why would we archive that specific website?” I2 | I1; I2 |
|         | F1.3| Versioning should be used to capture updates of sites. Therefore, archiving a site once is not sufficient. As described in F1.2 the frequency of archiving is determined by considering the scope of the archive.   
  – “You want to be able to show the development of the web [by capturing updates of sites as well]” I1 | I1; I2; I3; I4 |
|         | F1.4| Decisions concerning the scope of archiving should be determined by an archivist. Preferably this archivist has knowledge of web archiving to support the decision making.   
  – “The level of depth would be decided by the person [the archivist] who would evaluate how important the content of the site is” I2 – “These questions [questions regarding what to archive] must be answered by the person that selects the sites in the first place” I2 | I1; I2; I4 |
| Accessibility | F2.1| An archive must be accessible in order to be called an archive. Preferably the archive is easily accessible. This means that browsing through the archive is possible with, for example, a Graphical User Interface (GUI). The accessibility includes that | I1; I2; I3; I4 |
the captures within the archive are findable.
– “An archive without access is already lost” 11

F2.2 Preferably, an archive is publicly available (via a website). Legal issues can arise when making an archive public.
– “You want to preserve and present the archive” 12

F2.3 To make an archive accessible and to make the captures searchable, metadata should be included during the archival process. This metadata can be generated automatically and can be complemented with manually created metadata.
– “Try to add metadata about the original environment... that is the metadata that is needed to ensure the preservation” 11

F2.4 To fulfill the findability property of F2.1, captures should be full text searchable. This means that end-users of the archive are able to find sites without explicitly entering the site URL. This feature is comparable with the functionality of a search engine.
– “Preferably the web archive is full text searchable” 11
– “The Internet Archive is not full text searchable, which is unfortunate” 13

F3.1 Dynamic functionalities on websites should be preserved.
– “The archive should capture the interaction with the site... you should be able to experience the site” 11

F3.2 Archiving approaches like textual descriptions, screenshots or screen captures should only be used to complement the archived site. They should never be a desired end-product of the archival process. Some sites are so complex that these approaches are the only working methods to capture a part of the functionality of the original sites.
– “It [the above described approaches] does not sufficiently represent the functionality of which the original resource was made for” 12
– “A screen capture can be used as an addition to a full capture if archiving a certain site is not fully possible” 13

F3.3 The goal of archiving makes different web technologies (like JavaScript, Flash or videos) subordinate to this goal: preserve websites as close to the original as possible. For example, archiving web videos or Flash elements should not be considered a goal as such, but as means to achieve the goal of archiving.
– “The issue is: how concrete, how well presented and how close to reality it [the capture] is” 12
In addition to archiving a specific site, the process of archiving should be described as well. This information can be helpful in the future for multiple reasons: describe why something is archived, describe the context of a site, provide information about what is not archived correctly, how to view the archived site (e.g. which system/browser requirements) and what software/set-up is used to archive the site. A consequence of this feature is that in-house knowledge of archiving is desirable, since the knowledge is then available to describe such processes.

- "By providing proper documentation you enable people to access this resource in the same way in the future" [12] – "You should describe why a certain capture is not complete" [13]

Quality assurance is an important process of archiving. This process ensures that all pages of a site are captured and that the pages are (mostly) error free. This is a time-consuming process, a tool with quality check functionalities could speed-up this process.

- "I try to visit all pages within a website, to find out if they are archived correctly" [14] – "...it is a trade-off between quality and quantity" [14]

An open and standard file format should be used to store the captures. The WARC file type is expected to be the standard type for archiving [9] and is therefore recommended.

- "Try to limit the amount to file types, to make migration easier" [13]

Web developers can be involved in the archival process. Firstly, they can help with repairing broken functionalities by providing source codes. Secondly, they can keep the archivability in mind while developing a website. For example, they could leave out Flash functionalities since Flash is known to cause errors while archiving. This has also a drawback:

- "You do not want to limit the creative mind of a developer [they might leave out some functionalities in order to make that site easier to archive, which is not desirable]" [13]

Although many decisions should be made while archiving sites, automatizing is still preferable.

- "An automated solution is the golden solution for everything" [11]
4.2 Constructing the framework

Based on the extracted features in section 4.1, the Web Archiving Considerations Framework has been constructed, see Fig. 1. The framework provides a high-level overview of what is considered important for archiving dynamic websites.

Fig. 1. Web Archiving Considerations Framework

4.3 Using the framework to evaluate archiving tools

The previously created framework is used to evaluate three archiving tools. These archiving tools are: Brozzler, Heritrix and Webrecorder.io. Per feature is described what the implications of this feature are, as listed in Table 2. For most features, the implications are the same for each tool. For framework feature F3.1, the tools differ from each other, and they require a more extensive explanation. Therefore, the evaluation of this feature is discussed in more detail in section 4.4.

Table 2. Implications of framework features on three archiving tools

<table>
<thead>
<tr>
<th>Feature</th>
<th>Implications</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1.1</td>
<td>By the use of one of the selected archiving tools, this requirement is fulfilled. The tools are built to capture a site as close to the original as possible. This in contrast to other archiving techniques, like screenshots or screen captures, that only capture the appearance and not the functionalities. WARC files are created so a capture can be viewed in the same way as the original site.</td>
</tr>
</tbody>
</table>
F1.2 There is no direct implication for the tools to comply to this feature. An indirect implication is the ability to configure a tool so it does fulfill to the purpose of the archive.

F1.3 By default, the tools do not have the ability to capture only changed pages of a site. An incremental crawling strategy, as proposed by Sigurðsson [10], would make it possible to only capture the changes of a website instead of the full site.

F1.4 The archivist should have sufficient resources to be able to define the scope of the capture. Like the implication for F1.2, the possibility to configure a tool to fit to the scope ensures that a tool complies to this feature.

F2.1 By default, Brozzler and Webrecorder.io do have a GUI installed. This tool is called PyWb, and can be used to playback WARC files by entering the URL of an archived site. PyWb is not specifically made for these tools, and can therefore be used with other archiving tools as well. Heritrix does not offer a tool to browse through the WARC files.

F2.2 The tools are not built to make the archive publicly available. However, it is possible to make the access to PyWb public. This has limitations since the interface is PyWb is not user-friendly: browsing through the archive is only possible by entering a URL (so the user has to know in advance what sites are available in the archive) and the user has to select a WARC file from a list of files (so the user has to find out which file is the right one). Although public access is possible, changes are needed to make the interface user-friendly.

F2.3 Some metadata is automatically included while capturing a site. The header data is stored for every request, based on this it is possible to find out on what server the content is hosted. Besides that, the date of the crawls is stored. Brozzler and Heritrix do also include which version of the tool is used for archiving. Metadata that is not captured but relevant, includes the browser name (and version) that is used to make the capture.

F2.4 By default, the tools do not provide an interface to make the archive full text searchable. This is beyond the scope of the tools, since they focus on the archiving process itself, instead of accessibility. However, by storing the archived sites as WARC files, it is possible to develop a tool that supports full text search. If, for example, screenshots were used to archive a site, it would not be possible to perform a text search on the sites.

F3.1 In an ideal situation, all dynamic functionality is preserved. But due to the variety and complexity of websites, it is not possible to preserve all functionalities. A more detailed analysis of this feature can be found in section 4.4.

F3.2 There are no implications for the tools to comply to this feature.

F3.3 See implication F3.1.

F4.1 All the tools are open-source, therefore describing the archiving process is possible. If a tool would not be open-source, describing the process would depend on the openness of the creator of the software.

F4.2 There are no implications for the tools to comply to this feature.

---

6 https://github.com/ikreymer/pywb
F4.3 The output of all tools is a WARC file. This is an open file format, and is a commonly used file type for archiving sites [9].

F4.4 There are no implications for the tools to comply to this feature.

F4.5 Brozzler and Heritrix are using automated crawlers to capture a website. This means that by entering a single URL (for example the homepage of a site) the full site is automatically being captured. Such a crawler helps to automate the archival process. Webrecorder.io demands that every page is manually visited before it is being captured. It is time-consuming compared to the automated crawler.

4.4 Detailed analysis of feature 3.1

Feature 3.1 requires that the full dynamic functionalities of sites should be preserved. Because websites can be complex, capturing the full functionality of a site is not always possible. An empirical comparison has been made to find out which tool performs best in archiving this dynamic content. Each tool is tested against the same set of five websites, as listed in Table 3. Those websites are selected by Sound and Vision and are considered as hard to archive. The full tool comparison is shown in Table 4.

<table>
<thead>
<tr>
<th>#</th>
<th>URL</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td><a href="http://www.andereachterhuizen.nl/">http://www.andereachterhuizen.nl/</a></td>
<td>Site contains Flash elements and is not working in Google Chrome.</td>
</tr>
<tr>
<td>S3</td>
<td><a href="http://www.51sprints.com/">http://www.51sprints.com/</a></td>
<td>Complicated site with respect to the dynamic page loading and multiple videos.</td>
</tr>
<tr>
<td>S4</td>
<td><a href="http://www.surprisingeurope.com/">http://www.surprisingeurope.com/</a></td>
<td>Google Maps is an important functionality of this site and site has many pages.</td>
</tr>
<tr>
<td>S5</td>
<td><a href="http://www.lasthijack.com/">http://www.lasthijack.com/</a></td>
<td>Complicated navigation because of many dynamic elements on the site.</td>
</tr>
</tbody>
</table>

The tools are scored, based on four quantities:

1. Overall – The overall usability of a capture. Usability is low if important site functionalities are not correctly captured.
2. JavaScript – How well JavaScript functionalities are captured. For example: slideshows, Google Maps and dynamic page loading.
3. Flash – How well Flash content is preserved.
4. Videos – How well videos are preserved.
The following scale is used to measure the quantities:

- 1 – Nothing is working
- 2 – Most is not working
- 3 – Partially working
- 4 – Most is working
- 5 – Everything is working

### Table 4. Web archiving tool comparison

<table>
<thead>
<tr>
<th>Tool</th>
<th>Site</th>
<th>Overall</th>
<th>JavaScript</th>
<th>Flash</th>
<th>Videos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brozzler</td>
<td>S1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>1</td>
<td>3</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>S3</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>S4</td>
<td>3</td>
<td>3</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>S5</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>12</strong></td>
<td><strong>14</strong></td>
<td><strong>3</strong></td>
<td><strong>13</strong></td>
</tr>
<tr>
<td>Heritrix</td>
<td>S1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>S3</td>
<td>3</td>
<td>2</td>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>S4</td>
<td>3</td>
<td>3</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>S5</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>12</strong></td>
<td><strong>11</strong></td>
<td><strong>1</strong></td>
<td><strong>9</strong></td>
</tr>
<tr>
<td>Webrecorder.io</td>
<td>S1</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>S3</td>
<td>2</td>
<td>3</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>S4</td>
<td>4</td>
<td>5</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>S5</td>
<td>4</td>
<td>5</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>19</strong></td>
<td><strong>22</strong></td>
<td><strong>5</strong></td>
<td><strong>13</strong></td>
</tr>
</tbody>
</table>

Webrecorder.io has the highest overall score, and is scoring equally or higher on the other quantities. Therefore, it can be concluded that Webrecorder.io performs best in capture dynamic web content. This is partly due to the way Webrecorder.io is making captures. Every page should be visited manually (for example by an archivist) and requests are then stored. This means that, for example, more complicated image slideshows can be manually browsed, to ensure that all images are captured. This also holds for Flash content, which can be fully captured by ensuring that all links are clicked, and therefore stored. The downside of this approach is that automatization is not possible. For sites with a large number of pages, this approach is not preferable since archiving these sites would be time consuming. Brozzler and Heritrix work in a different way: a crawler is capturing all pages by automatically visiting all links. In contrast to Webrecorder.io, the approach is fully automated. The consequence of this approach is that more errors can occur, and that dynamic functionalities could be lost.
5 Discussion

5.1 Implications of the framework and the comparison

The framework that has been developed can be used to evaluate web archiving approaches. The framework provides a high-level overview of archiving dynamic websites, by listing the most important considerations. In the literature, such an overview did not exist. As discussed in the related literature section, most literature is focused on specific parts of web archiving, instead of providing an overview. The study of Brunelle et al. [1] has indicated that JavaScript is responsible for 52.7% of the missing content. In the tool comparison of this research, the ability of a tool to correctly capture JavaScript indeed had substantial impact on how well the sites were archived.

In practice, the framework can be useful for setting-up a new web archiving project. Besides that, is it possible to evaluate or extend an existing archiving approach. For example, the framework could be used by Sound and Vision to extend their current web archiving strategy. The framework has pointed out that multiple factors are important while considering web archiving. An archiving tool that is able to capture sites with dynamic functionalities is not the single element for a complete web archive. The scope and accessibility, together with other features that are discussed in the framework, should also be considered.

The tool comparison can be used to decide what archiving tool should be used for the web archive. The comparison has shown that Webrecorder.io is best in capturing dynamic web content. However, the approach that Webrecorder.io takes is not automated, in contrast to Brozzler and Heritrix. The decision on what tool is best suitable, does not solely depend on how well this tool is able to capture dynamic content, but also on if it fits into a certain archiving strategy. If many sites have to be archived, a manual crawling approach is not desirable. On the other hand, if only dozens of sites have to be archived, Webrecorder.io could be recommended because this approach is generating site captures of a higher quality.

5.2 Limitation: all interviews are conducted at Sound and Vision

The interviews are all conducted at the Sound and Vision institute. One can argue that the results of this research are therefore biased towards the way of thinking inside this institute. This is considered as a limitation of this research. During the development of the framework, the limitation is taken into account. Proposed features that were only relevant for Sound and Vison were left out of the framework. Besides using interviews, literature is used to develop the framework. Consequently, the framework, together with the other findings of this research, are relevant outside the Sound and Vision institute as well.
5.3 Hybrid approach of automated and manual crawling

Webrecorder.io performs best in capturing dynamic content due to the manual crawling approach. Because of the manual crawling, it takes more time to capture a site than with an automated crawling approach (like Brozzler or Heritrix). There is a trade-off between the completeness or quality of a capture, and the time it takes to archive a site. A hybrid approach could help by saving time and by ensuring that a capture has a high quality. For this approach, Brozzler or Heritrix can be used to crawl a site. If, during the quality assurance process, it turns out that a (dynamic) functionality is not crawled correctly, this can be fixed by using the approach of Webrecorder.io. The page can be manually crawled to ensure that the functionalities of the site are working correctly. This approach results in a scalable archiving method, where the captures have a higher quality than approaches with fully automated crawlers.

5.4 Future research: classify sites

The ability of an archivist to make decisions about the scope and the purpose of the archive, has a prominent role in the framework. On site-level, decisions have to be made about the scope. Besides that, captures should be checked to ensure that the capture does not contain errors. To make the archive scalable, websites can be classified and labelled based on some characteristics. Each site can be added to a category, and based on this category the scope can be determined. An example classification can be based on the importance of a website. The importance of a site can be determined by considering the purpose of the archive. More important sites could have a larger scope, where for example outgoing links are archived as well. This category of websites could also have a stricter quality check. Future research can try to discover which categories can be created to classify sites on.

6 Conclusion

During this research, multiple aspects of archiving dynamic websites were considered. To answer the first research question: What are considerations of archiving dynamic websites, the Web Archiving Considerations Framework has been developed. This framework can be used to identify the considerations that are related to archiving sites. The features of the framework are determined by conducting four semi-structured interviews at the Netherlands Institute for Sound and Vision. Afterwards, the framework has been used to describe the implications of its features on three existing web archiving tools. The framework provides a high-level overview of what should be taken in mind while evaluating a web archiving approach, with the focus on archiving dynamic websites. For one feature, a more in-depth analysis has been performed. This feature is used to compare three archiving tools, and to answer the second research question: How do existing web preservation tools differ from each other? These tools are empirically compared on how well they are able to capture dynamic websites.

The main findings of this research are that the scope of a capture should be determined by an archivist. Therefore, a fully automated archiving approach is not possible.
Furthermore, the accessibility of archive is considered essential. These considerations have a prominent role in the framework. Captures should function and appear like the original site, despite the technologies that the site uses. Consequently, examining these techniques is only a means to get to the goal: capturing the site with all of its functionalities. The techniques are not explicitly mentioned in the framework, which only provides a high-level overview of web archiving. The tool comparison that was made, has shown that Webrecorder.io is best able to capture dynamic web content. However, this does not imply that Webrecorder.io is therefore always the recommended tool. The decision of which tool should be used, depends on a trade-off between automatization and completeness. Webrecorder.io requires manual page crawling. Archiving a site is therefore more time consuming than using archiving tools with automated web crawlers.

The framework that has been developed, can be used as a guidance to set up a new web archive. Furthermore, the framework can be used to evaluate existing web archiving approaches. The empirical tool evaluation shows how the framework should be applied with respect to archiving tools. This evaluation includes the tool comparison, which can help in deciding which archiving tool is best suitable for a specific archiving approach.

References