

Sitemap Generators

Paul Ganster, Peter Grassberger, Magdalena Mayerhofer, Ana Lopez Camarero

706.041 Information Architecture and Web Usability WS 2019/2020
Graz University of Technology

2 Dec 2019

Abstract

A common task for information architects, web designers and many more is to get an overview of the hierarchy of a website. Creating sitemaps is a usual approach to this problem. Since the manual task of creating them is time consuming, one may often consider to use sitemap generators. Visual sitemaps are graphical representations of sitemaps and present the more difficult task, which can be seen since there is no professional tool that totally works as expected. The task of creating a content inventory on the other hand can be automated more easily. Not only will the tools offered reduce the manual work required for their users, but there even exists a free and open source tool that produces comparable results to paid tools. This survey will focus on sitemap generator tools and compare their produced results and performance.

© Copyright 2019 by the author(s), except as otherwise noted.

This work is placed under a Creative Commons Attribution 4.0 International (CC BY 4.0) licence.

Contents

Contents	i
List of Figures	ii
List of Tables	iii
List of Listings	iv
1 Introduction	1
1.1 Sitemaps	1
1.2 Short History	1
1.3 Generators	3
1.4 General Information about tools	3
1.5 Tested Websites	3
2 Visual Sitemaps	5
2.1 PowerMapper	5
2.1.1 Results	5
2.1.2 Improvements	7
2.2 VisualSitemaps	9
2.2.1 Results	10
2.2.2 Improvements	11
2.3 DynoMapper	12
2.3.1 Results	12
2.3.2 Conclusion	13
2.4 Visual Site Mapper	13
2.4.1 Results	14
2.4.2 Conclusion	16
3 Content Audit	17
3.1 Screaming Frog	17
3.2 URL Profiler	17
3.3 Content Analysis Tool (CAT)	21
3.4 Scrapy	21
3.4.1 Proof of Concept	22
4 Conclusion	26
Bibliography	27

List of Figures

1.1	Tested Websites: oebb.at, audi.at, developer.db.com	4
2.1	The sitemap progress screen in PowerMapper.	6
2.2	PowerMapper Overview of most Site Map Styles	6
2.3	The start page of PowerMapper	7
2.4	The result of the ÖBB website with PowerMapper	8
2.5	The result of db.developer.com	8
2.6	The result of db.developer.com after trying to resolve issue	8
2.7	The sitemap creation form only requires URL as input.	9
2.8	On the dashboard, an orange icon in the status column indicates that a crawl is in progress.	10
2.9	The result screen offers 1.) a shareable web link, 2.) a PDF export, 3.) toggle between top-to-bottom and left-to-right hierarchy and 4.) the visual sitemap itself, whereas the visibility of child nodes can be toggled per node.	10
2.10	Fully zoomed out PDF Export	11
2.11	Three screenshots of the PDF Export zoomed in at 5-times the zoom level	11
2.12	Cropped PDF export at 50-times the the zoom level as in Figure 2.10.	12
2.13	A large number of tag pages on the Audi website with connectors lead to a blur in the export.	12
2.14	DynoMapper ÖBB PDF export pages 1 - 10 of 17.	13
2.15	DynoMapper Audi subtree in the online viewer.	14
2.16	DynoMapper: audi.com tree view.	14
2.17	The unique style that the tool offers.	15
2.18	The resulting visual sitemap of the ÖBB website	15
3.1	Solution of the crawl of the ÖBB website	18
3.2	Visualization Tree by Screaming Frog of the ÖBB website.	18
3.3	Initial status of the program	19
3.4	Spreadsheet output of the URL Profiler with the ÖBB website as an example.	20
3.5	Screenshot folder of the ÖBB website as an example.	20
3.6	Spreadsheet result with some of the most interesting columns to take into account of the ÖBB website as an example.	21
3.7	Screenshot of a URL with lack of content of the ÖBB website.	21
3.8	Content Analysis Tool online	22
3.9	Content Analysis Tool export	23
3.10	The resulting content inventory after running the oebb_scraper Scrapy project.	25

List of Tables

2.1	Prices and feature restrictions per license with PowerMapper.	7
2.2	Prices and feature restrictions per license with VisualSitemaps.	9
2.3	Prices and feature restrictions per license with Dyno Mapper.	13
3.1	Prices and feature restrictions per license with URL Profiler.	18
3.2	Prices and feature restrictions per license for Content Analysis Tool.	22

List of Listings

1.1	TXT sitemap example oebb.at	2
1.2	XML sitemap example oebb.at	2
1.3	CSV sitemap example oebb.at.	3
3.1	Item class for the ÖBB content audit	23
3.2	Spider class for the ÖBB content audit.	24

Chapter 1

Introduction

1.1 Sitemaps

A sitemap is a list of pages of a website. In some instances, the sitemap also represents the hierarchy of pages, how subpages relate to their parents' pages.

There are different purposes for sitemaps. They are used to plan the information architecture of website before creation or to analyze existing websites, this process is also called content auditing. Sitemaps can also be included in websites to give users an overview of all the pages that exist. Lastly sitemaps in an XML can be used to provide a list of pages to web crawlers and search engines.

The hierarchy of pages can be defined in two ways. Based on hyperlinks a page is a subpage of another page if there is a link from one to the other. Based on the URL path component, that can be understood as a file system directory structure, a page is a subpage if it comes after another page's name in the path separated by a slash character. Therefore `https://www.oebb.at/en/fahrplan.html` is a subpage of `https://www.oebb.at/en/`.

The simplest representation of a sitemap is a list, these lists can be in the format of a TXT file with one link per line, as can be seen in Listing 1.1. The CSV format can contain additional data for every page, for example: HTTP status code, HTML title tag, HTML first h1 tag, an example can be viewed in Listing 1.3. Another structured approach is the XML Sitemaps format introduced by Google, mostly used to inform crawlers about all pages, an example can be seen in Listing 1.2. Sitemaps included in websites for the user to view are in the format of HTML, as a list or hierarchy.

Visual representations of sitemaps can be in the format of images (PNG, SVG, etc.) or in the PDF format. HTML is also a common format for visual sitemaps. HTML allows for a more interactive approach where nodes in the hierarchy can be links to the original pages or they can open a detailed view of a sub hierarchy or sub tree.

In the process of auditing a content inventory is most often represented as a CSV file with metadata about every page, shown in Listing 1.3. There are also specific formats for some tools, like the Screaming Frog SEO Spider File. The columns of these files are used to identify pages that are missing or need improving.

1.2 Short History

In 2005 Google introduced a Google Sitemaps 0.84 Standard under Creative Commons license, which was also supported by Yahoo! and Microsoft in 2006. It had the purpose to inform search engines about new pages and increasing the coverage of sites already in the search index. The latest version of the XML Sitemaps standard is 0.9

```
1 https://www.oebb.at/en/  
2 https://www.oebb.at/de/  
3 https://www.oebb.at/en/tickets-kundenkarten.html  
4 https://www.oebb.at/en/fahrplan.html  
5 https://www.oebb.at/en/reiseplanung-services.html  
6 https://www.oebb.at/en/regionale-angebote.html  
7 https://www.oebb.at/en/regionale-angebote/burgenland/zum-neusiedler-see.html  
8 https://www.oebb.at/en/regionale-angebote/kaernten/autoschleuse-tauernbahn.html  
9 https://www.oebb.at/en/regionale-angebote/niederoesterreich/waldviertel-express.html
```

Listing 1.1: TXT sitemap example oebb.at

```
1 <?xml version="1.0" encoding="UTF-8"?>  
2 <urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">  
3   <url>  
4     <loc>https://www.oebb.at/en/</loc>  
5     <lastmod>2019-11-05</lastmod>  
6     <changefreq>monthly</changefreq>  
7     <priority>0.5</priority>  
8   </url>  
9   <url>  
10    <loc>https://www.oebb.at/de/</loc>  
11    <lastmod>2019-11-05</lastmod>  
12    <changefreq>monthly</changefreq>  
13    <priority>0.5</priority>  
14  </url>  
15  <url>  
16    <loc>https://www.oebb.at/en/tickets-kundenkarten.html</loc>  
17    <lastmod>2019-11-05</lastmod>  
18    <changefreq>monthly</changefreq>  
19    <priority>0.3</priority>  
20  </url>  
21  <url>  
22    <loc>https://www.oebb.at/en/fahrplan.html</loc>  
23    <lastmod>2019-11-05</lastmod>  
24    <changefreq>monthly</changefreq>  
25    <priority>0.3</priority>  
26  </url>  
27 </urlset>
```

Listing 1.2: XML sitemap example oebb.at

```
1 url , content_type , status , title , h1
2 https://www.oebb.at/ , text/html ; charset=UTF-8,200,ÖBB - Startseite , Topaktuelle
  Informationen
3 https://www.oebb.at/de/reiseplanung-services/vor-ihrer-reise/reisereservierung.html ,
  text/html ; charset=UTF-8,200,ÖBB - Reisereservierung , Reisereservierung
4 https://www.oebb.at/de/reiseplanung-services/vor-ihrer-reise/mit-haustieren-
  verreisen.html , text/html ; charset=UTF-8,200,ÖBB - Mit Haustieren verreisen , Mit
  Haustieren verreisen
5 https://www.oebb.at/de/reiseplanung-services/vor-ihrer-reise/reiseversicherung.html ,
  text/html ; charset=UTF-8,200,ÖBB - Reiseversicherung , Stornoversicherung
```

Listing 1.3: CSV sitemap example oebb.at

1.3 Generators

Sitemap generators are programs that create sitemaps or content inventories. They contain a web crawler which is a program that tries to find every page, starting with one page and looking for links to follow to discover new pages again and again. Some visual sitemap generators can also start from an existing sitemap as a list or XML Sitemaps file, skipping the step of crawling pages which can be done by another tool.

Web crawling can be limited by websites either through requiring user authentication with a login form or with the robots exclusion standard by specifying a robots.txt file. Some generators have features to manage both limitations, one can provide login details or choose to ignore the robots exclusion standard.

1.4 General Information about tools

The tools are categorized into the two main categories of visual sitemaps and content inventory. However, this does not mean that the tools only belong into that category, but that the main purpose of this tool is in that category. Some of the tools offer some features for visual sitemaps and for content inventory at the same time.

1.5 Tested Websites

The main websites which were used for the purpose of testing the tools are:

- <https://www.oebb.at>
- <https://www.audi.at>
- <https://developer.db.com>

oebb.at is the website of the Austrian Federal Railways, one of its main utilities is the ability to check train schedules and book tickets. audi.at is the Austrian website of Audi AG which is a German automobile manufacturer. developer.db.com is the developer portal website of Deutsche Bank AG which is a German bank. The DB developer portal is single-page application website that loads content dynamically instead of doing full page reloads, it is created with JavaScript and Angular. Figure 1.1 shows the home pages of the three tested pages.

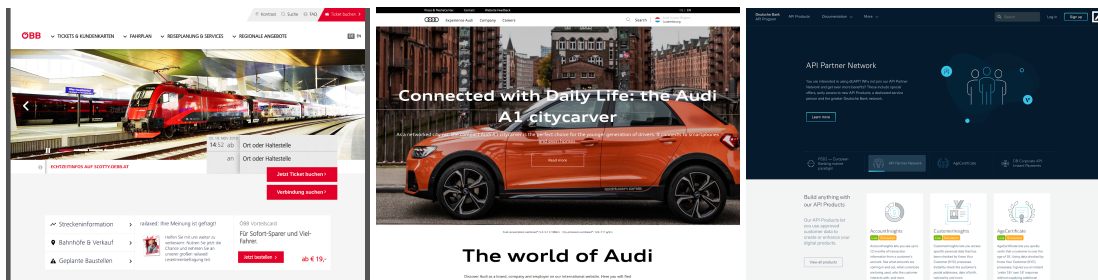


Figure 1.1: Tested Websites: oebb.at, audi.at, developer.db.com

Chapter 2

Visual Sitemaps

2.1 PowerMapper

PowerMapper PM 2019 is a tool for creating visual sitemaps via either an online tool or an application for MacOS or Windows. The company behind it is based in Edinburgh (UK) and PowerMapper was first launched in 1997.

The online tool is free but very limited compared to the MacOS or Windows application. It covers only one level of the hierarchy, which is not very helpful in most of the cases. The desktop version is more powerful, but this comes at a price: There are two types of licenses available, namely the Standard and the Professional edition. The main features and restrictions can be seen in Table 2.1.

The Sales department of PowerMapper provided us with a complimentary Professional license, because the online version and the trial version did not meet our needs to test the tool to the full extent of possibilities.

The input required by PowerMapper is the URL of the page which should be crawled. For the tested websites the crawls took over an hour, even though, according to the PowerMapper website, a crawl should build a sitemap in under 5 minutes. During the crawl, PowerMapper shows the number of nodes that have been found, the time still left, how many nodes are already mapped and the percentage, as it can be seen in Figure 2.1.

The result of the page can be shown in 12 different styles – even if the website says 13. From those 12 available styles, only a few of them are visual sitemaps and some are just site maps. Also, some of the visual styles are very strange, and in our opinion, would not be considered a visual sitemap style. All the different styles can be seen in Figure 2.2. The start page of the desktop application can be seen in Figure 2.3

2.1.1 Results

After entering the URL into the input field, first you have to search for a while how to actually start the crawl. The button for that is not very intuitive, because it is not next to where you entered the URL and due to it looking more like some information than an actual button.

When starting the crawl with the ÖBB website, after a short moment of waiting you could see the approximated time of about 10 minutes. However, in the end it took over an hour to crawl the actual page. During the crawl, it also shows you the percentage of how far the process is along, but most of the times it got stuck around 30% until it is finished. So, it is not really helpful to show the percentage. You can also see the number of found and already mapped nodes during the process, but after the application is done with the search there is no such summary available anymore. Approximately there were about 2000 pages for ÖBB.

After the crawl is finished, you immediately see the map in the standard style inside the application. You now can save or export the map and also change the style and manually manipulate the map by hiding

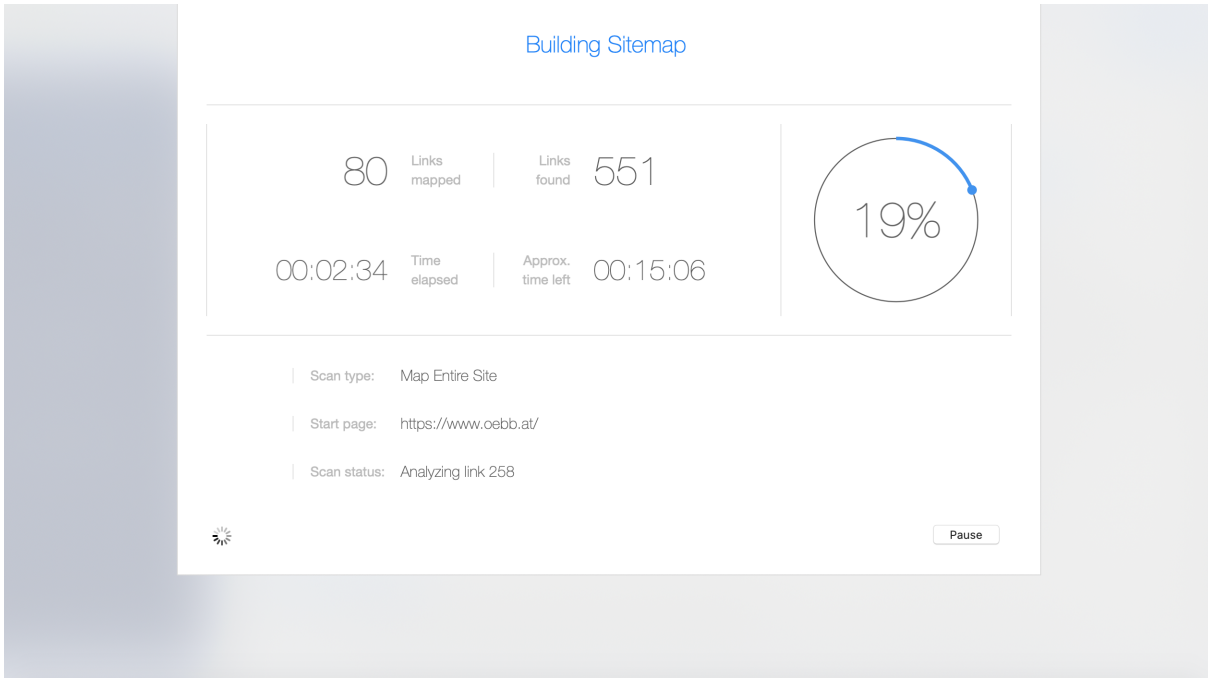


Figure 2.1: The sitemap progress screen in PowerMapper. [Screenshot taken by the authors of this paper.]

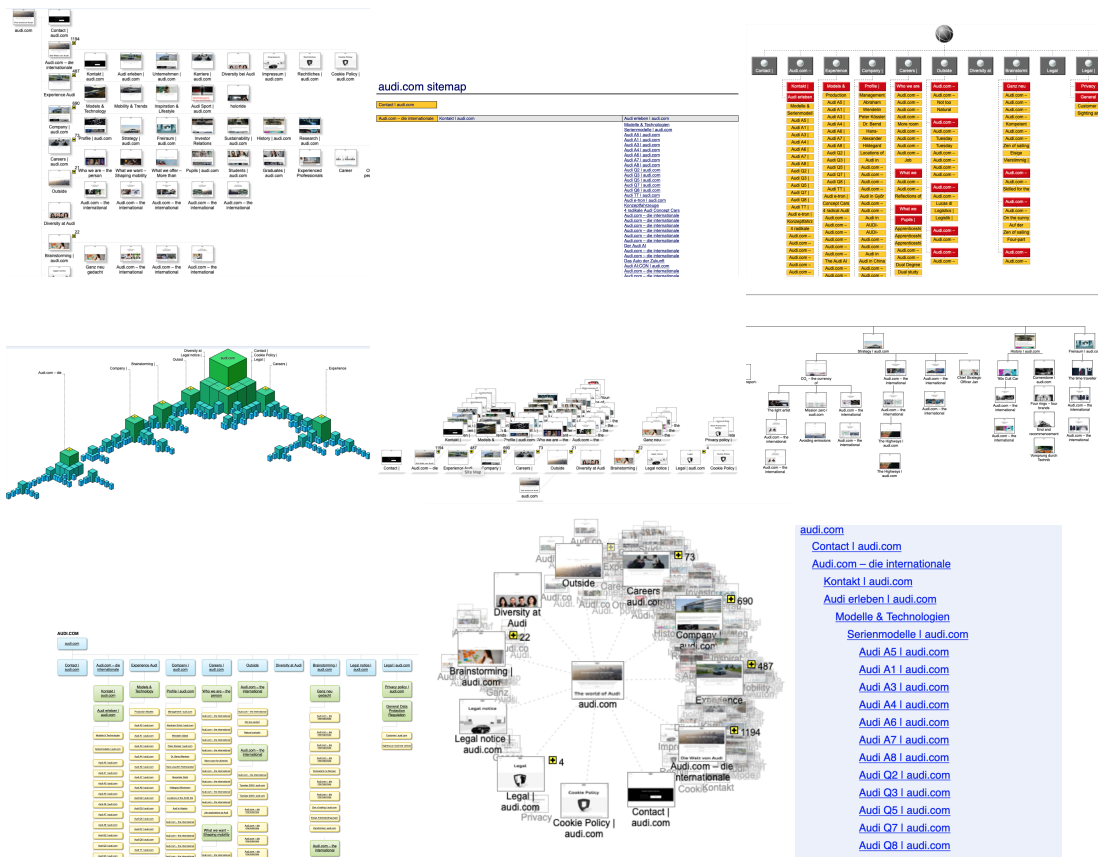


Figure 2.2: PowerMapper Overview of most Site Map Styles [Screenshot taken by the authors of this paper.]

License	Standard	Professional
Price/License	€99	€239
Pages/Crawl	22000	22000
Sitemap styles	7	13
Import and visualize	no	yes

Table 2.1: Prices and feature restrictions per license with PowerMapper.

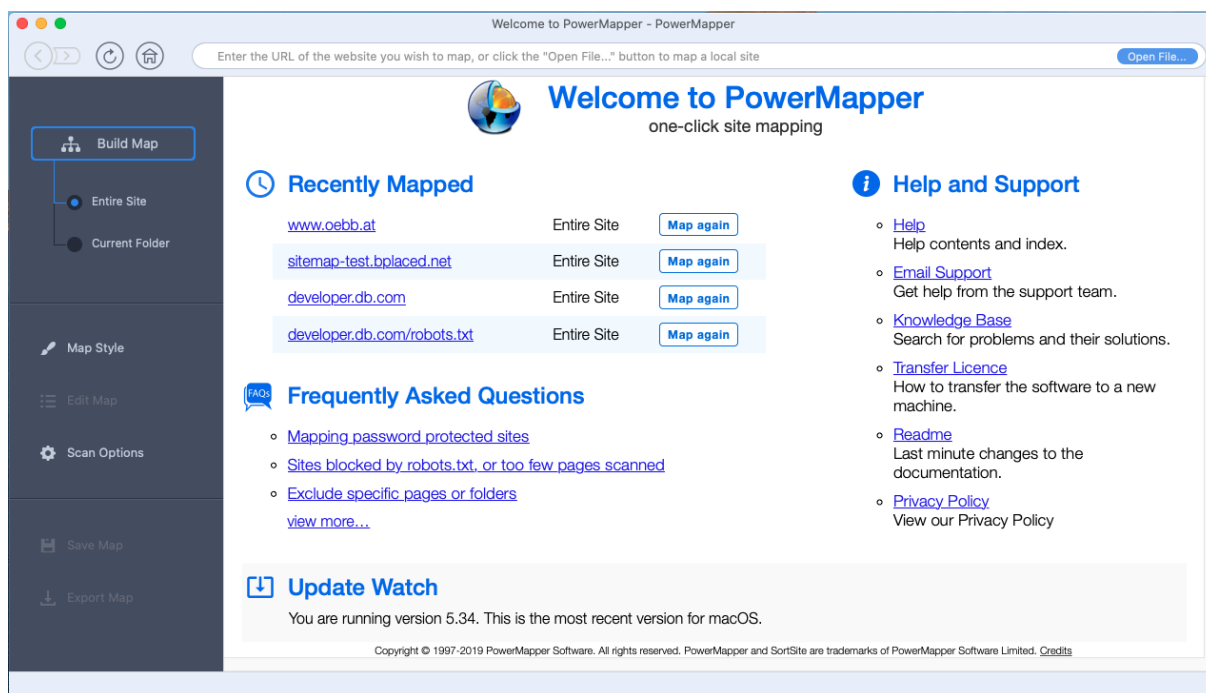


Figure 2.3: The start page of PowerMapper [Screenshot taken by the authors of this paper.]

single nodes. This is a very nice feature which gives you the possibility to clean up the result to your needs. The result can be seen in Figure 2.4

The way of navigating through the levels inside the application works in the same way then when you export it as HTML. The HTML export is the only way to export the map as visual sitemap, however you can also export it as XML or CSV, but than it is not visual anymore. The nice thing about the HTML Export is, that not all the nodes are totally visible at once, but you can expand subtrees or hide them again. In this way you can actually inspect the results in a reasonable way.

After performing a crawl on db.developer.com, you can see that the PowerMapper is not without flaws. In Figure 2.5 you see the result of this crawl. After trying to resolve the issue by following the suggested steps, PowerMapper gave the following result seen in Figure 2.6. This is still not really the expected result, due there being just one node.

2.1.2 Improvements

Recommended improvements would be some usability enhancements, for example so it is clearer how to start the crawl. Also, some summary information after the crawl would be very nice, like the time it needed and the actual number of mapped nodes. Additionally, issues it seems to have with some of the websites, for example with db.developer.com, would be great if they could be removed. Especially because the website worked perfectly in other tools. Last but not least, the functionality to test a website behind a login is missing, because you cannot provide login data to PowerMapper.

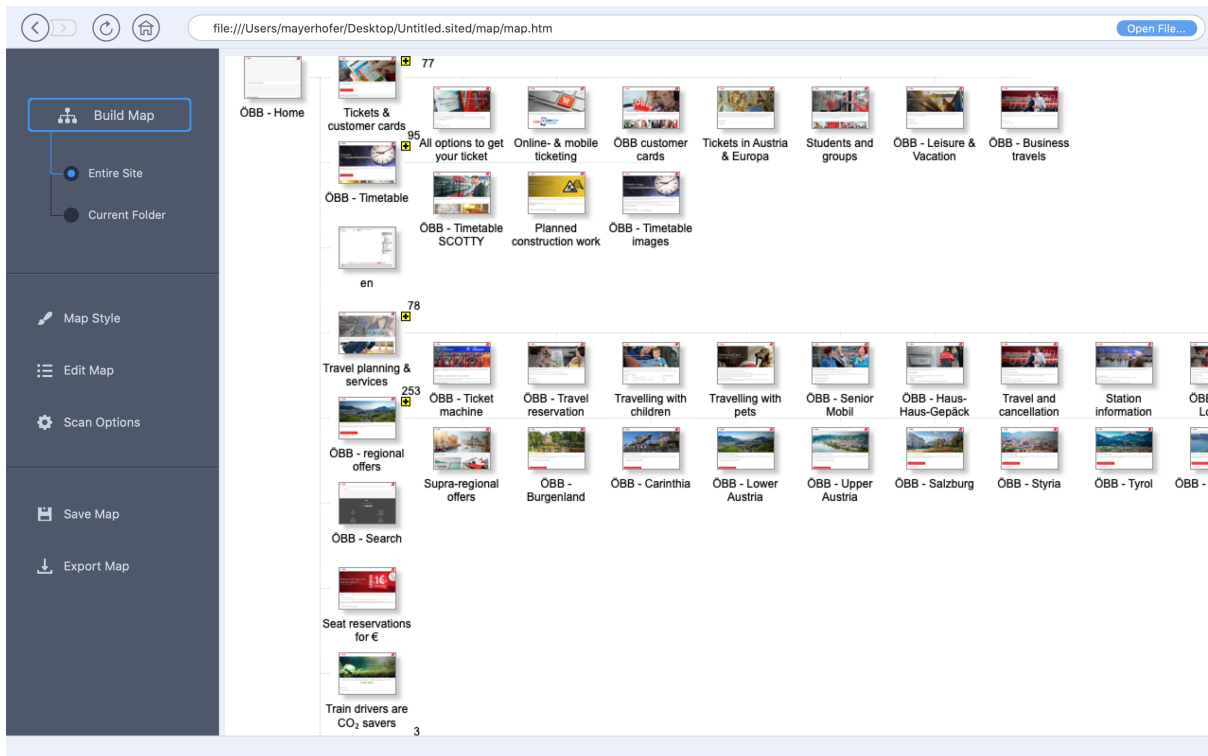


Figure 2.4: The result of the ÖBB website with PowerMapper [Screenshot taken by the authors of this paper.]

This site cannot be scanned

The site's start page cannot be opened. Possible causes are:

- The site's administrator has excluded Web crawlers using the Robot Exclusion Standard (robots.txt)
- The start page is blocked by a Robot Exclusion Standard meta tag: <meta name='robots' content='nofollow' >
- You've blocked the page using the Blocks tab of the Options window.
- The start page isn't HTML.

You can override these behaviours using the Blocks tab on the Scan Options window.

Blocked by Robots.txt or Blocked Links

<https://developer.db.com/robots.txt>

Figure 2.5: The result of db.developer.com [Screenshot taken by the authors of this paper.]



Figure 2.6: The result of db.developer.com after trying to resolve issue [Screenshot taken by the authors of this paper.]

License	Free	Mini	Freelancer	Team
Price/month	\$0	\$19	\$39	\$159
Pages/month	50	1000	3000	10000
Pages/sitemap	50	500	1500	3000
Crawlable depth	2	4	Unlimited	Unlimited

Table 2.2: Prices and feature restrictions per license with VisualSitemaps.

Figure 2.7: The sitemap creation form only requires URL as input. [Screenshot taken by the authors of this paper.]

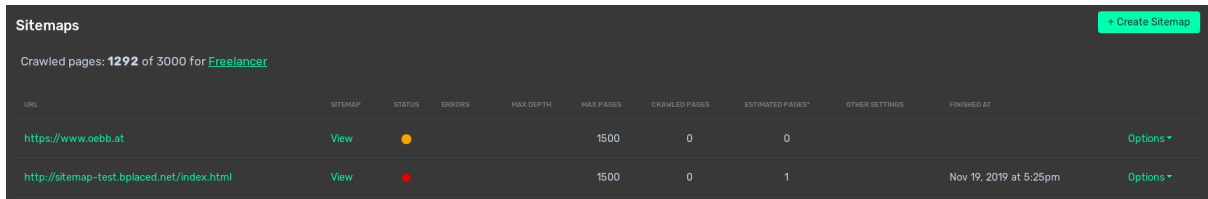
2.2 VisualSitemaps

VisualSitemaps VS 2019 is a tool for creating visual sitemaps via a web application. It has been in the making for 2 years since 2017 and was the result of the founders' frustration of needing to manually create sitemap with screenshots for every new web project.

This online tool is available for any device supporting a browser. This implies the usage of VisualSitemaps with smartphones is possible, but the web application is rather optimized for desktop usage and larger tables. Several license types are available for the proprietary software, namely the Free, Mini, Freelancer and Team license. These licenses dictate not only the price, but also restrict important features, such as crawlable depth and pages per month, as can be seen in Table 2.2. The support of VisualSitemaps was so kind as to provide a Freelancer license for this survey, as the limitations of the Free license were too harsh to make any kind of statement about this product.

The input required by VisualSitemaps is the URL of the page one wants to start the crawl from, as can be seen in Figure 2.7. The crawl takes several minutes, indicated by an orange status icon in the dashboard, as in Figure 2.8. Upon crawl completion, an e-mail will be sent. Within the dashboard, one can click "View" and view the created visual sitemap. In Figure 2.9, a completed crawl on ÖBB can be seen. The result can be toggled from a top-to-down to a left-to-right hierarchy. Each node offers a possibility to hide its children. An export function is also provided, which either creates a shareable web link or a PDF, whereas the PDF creation may additionally take up to a minute.

Further settings can also be applied before starting a crawl, such as login data, URL restrictions, maximum pages and maximum depth. The former consists of a login URL, where the login form is located, a username and a password. In addition, CSS selectors can be defined, which help VisualSitemaps



URL	SITEMAP	STATUS	ERRORS	MAX DEPTH	MAX PAGES	CRAWLED PAGES	ESTIMATED PAGES*	OTHER SETTINGS	FINISHED AT
https://www.oebb.at	View	●			1500	0	0		
http://sitemap-test.bplaced.net/index.html	View	●			1500	0	1		Nov 19, 2019 at 5:25pm

Figure 2.8: On the dashboard, an orange icon in the status column indicates that a crawl is in progress. [Screenshot taken by the authors of this paper.]

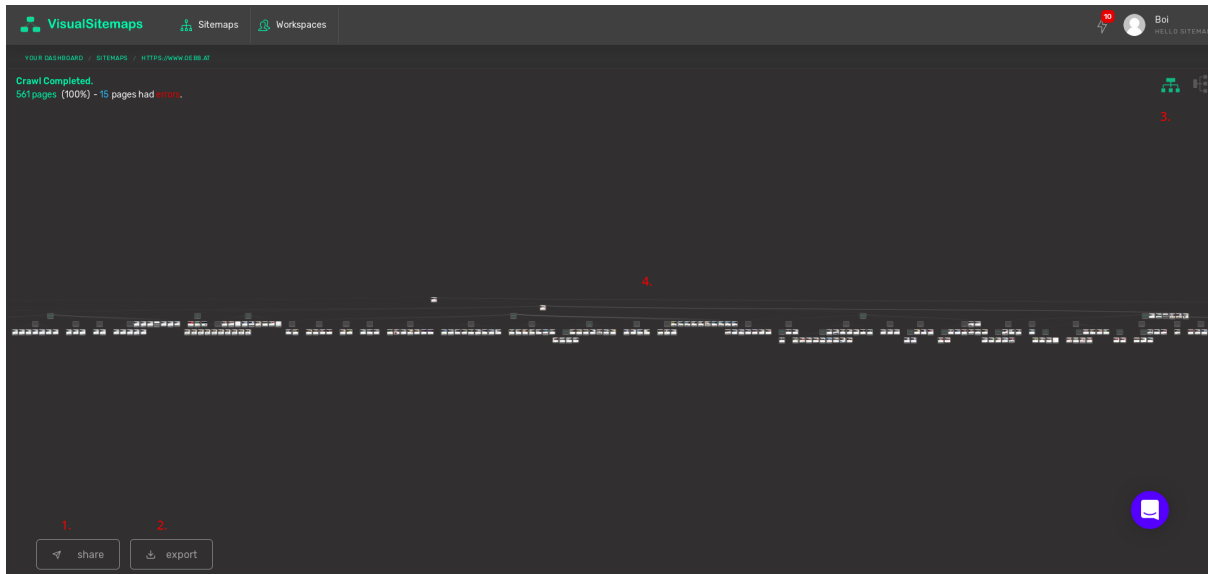


Figure 2.9: The result screen offers 1.) a shareable web link, 2.) a PDF export, 3.) toggle between top-to-bottom and left-to-right hierarchy and 4.) the visual sitemap itself, whereas the visibility of child nodes can be toggled per node. [Screenshot taken by the authors of this paper.]

to locate the login form. This enables VisualSitemaps to also work with pages protected by a login. The URL restriction settings consist of both a keyword, directory blacklist and whitelist, which limits the pages that are visited depending on its URL.

2.2.1 Results

After applying the necessary settings, a crawl can be started. For the ÖBB website, it took about 30 minutes to yield a result of 561 crawled pages. The shareable link can be provided right away to clients with a browser, but the PDF takes another minute to be generated. This can become annoying as the browser needs to be open during this time. The resulting PDF can be seen in Figure 2.10. As one can already infer, at full width, the export with a larger number of nodes is unreadable. Figure 2.11 shows the PDF at five times the zoom level. Even now, no usable information can be read from the export. Only after setting the zoom level to fiftyfold, the content of the nodes can be read and analyzed. A cropped export at that zoom level is displayed in Figure 2.12. The main problem beside the necessary zoom level is the performance drops in PDF readers opening the export. The size of the PDF is about 35 megabytes. After applying the essential zoom-level, the PDF reader takes a few seconds to fully render the content currently within the viewport. This may become a nuisance, as to navigate in such a wide PDF, one needs to zoom out, scroll to the desired node and zoom in again. Every zoom level change takes a few seconds caused by the large size. Furthermore, the navigation in the PDF is rather tedious as horizontal scrolling may take forever if one wants to look at a specific node and its children. These problems (zoom-level and navigation) also exist with the shareable link as an export. But at least scrolling through the visual



Figure 2.10: Fully zoomed out PDF Export [Screenshot taken by the authors of this paper.]



Figure 2.11: Three screenshots of the PDF Export zoomed in at 5-times the zoom level as in Figure 2.10. [Screenshot taken by the authors of this paper.]

sitemap in the web application is far more smoother than through the PDF.

If one takes a look back at Figure 2.12, one can analyze the results more thoroughly. The leaf nodes in the tree represent web pages, whereas such a node is visualized by adding the HTML title and a screenshot within the node. Nodes with children represent the URL hierarchy of its leaves. This initial assumption that VisualSitemaps is creating the hierarchy based on the URL /directory/ structure, was confirmed by the VisualSitemaps support. This means there is no actual parent > child linkage one would expect from a visual sitemap, but rather a URL directory tree. For example, if the third level in a navbar contains this example URL "oebb.at/ticket.html", the ticket.html node would not be in the third level of a category, but rather directly under root. Additionally, external link redirections will not work, which was confirmed by the support team. Under the condition that the crawled website has well maintained internal URLs, the hierarchy created by VisualSitemaps is definitely usable. The number of crawled pages, 561, is still by far larger than the number of nodes in a manually created sitemap, 139. Since ÖBB sets the language via URL (e.g. oebb.at/en/home.html vs oebb.at/de/home.html), 561 can be halved, as VisualSitemaps duplicated nodes of pages which are available for both German and English.

Similar results regarding zoom-level, navigation and URL hierarchy can be observed when using Audi as input website. One striking problem that was occurred with Audi's website were the tag pages. Several pages on Audi can be tagged, e.g. "Design", "Innovation" etc. Clicking on these tags leads to a page that shows multiple pages with the same tag. Since these tag pages were in no URL directory, they were all inserted directly under root, which can be seen in Figure 2.13. The large number of such pages lead to a huge number of connectors, which seem more like a blur on the page than children of a node of a tree.

After performing a crawl on db.developer.com, one can see that dynamic web applications can be crawled without any problem. Additionally, due to the small size of this website the resulting sitemap is actually usable, and one can easily navigate through the nodes.

2.2.2 Improvements

Recommended improvements could be providing a better navigation throughout the hierarchy. For example, by clicking on a node, only its children could be shown and therefore reducing the clutter in the graph. Hiding single nodes (instead of only the children) would also be a possibility. Such dynamic interactions could be exported as multiple static HTML files, so the navigation features would be available offline too. This could even replace the PDF export which not only takes long, but also has rather degrading user experience for larger number of nodes.

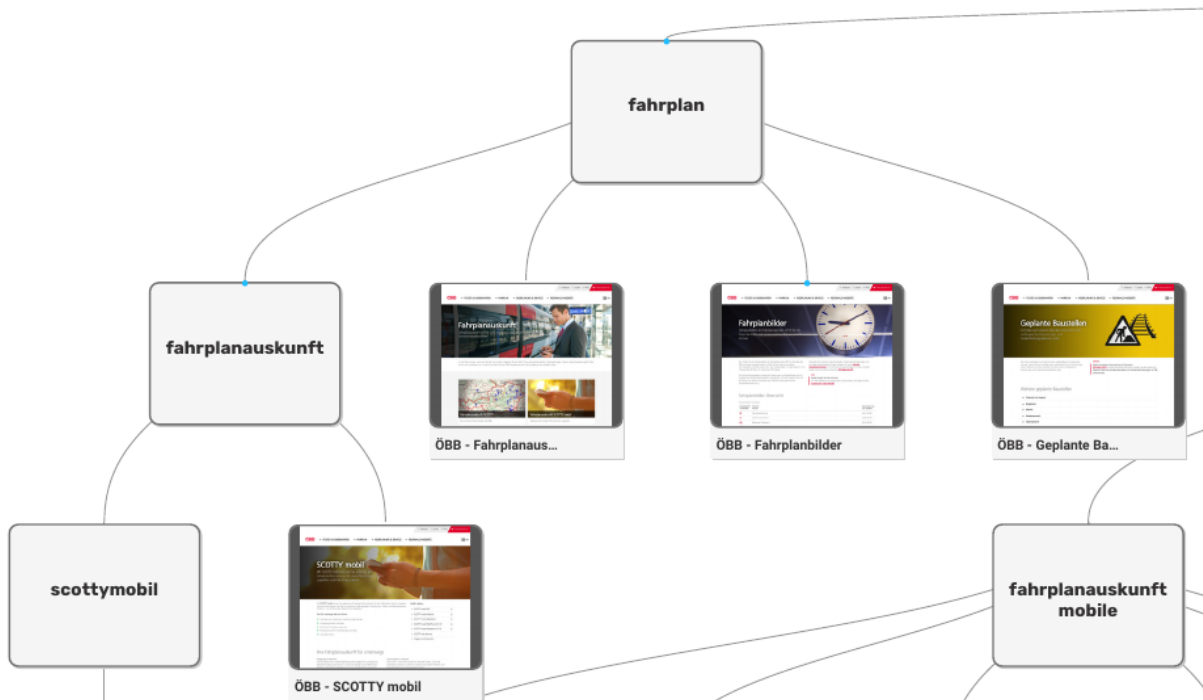


Figure 2.12: Cropped PDF export at 50-times the zoom level as in Figure 2.10. [Screenshot taken by the authors of this paper.]

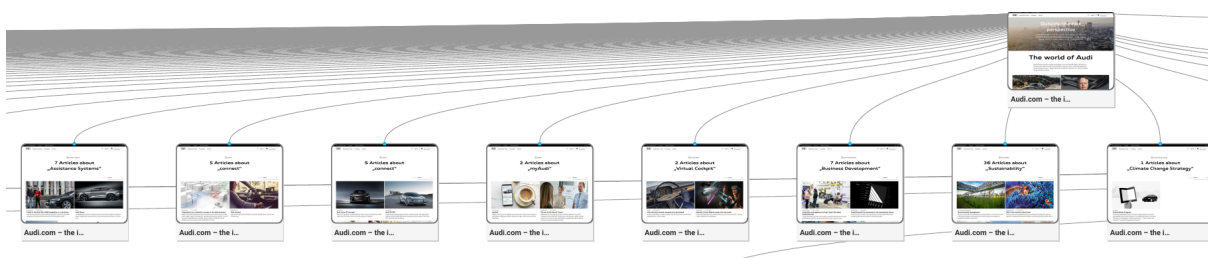


Figure 2.13: A large number of tag pages on the Audi website with connectors lead to a blur in the export. [Screenshot taken by the authors of this paper.]

2.3 DynoMapper

DynoMapper DM 2019 is an online tool to create visual sitemaps, it works in every major browser and therefore does not depend on any specific operating system. DynoMapper was evaluated on the basis of a trial version with limited functionality as seen in Table 2.3. Even after contacting its support team and insisting on a more capable license for the purpose of research and education it was not provided. Trial licenses only allow to create one project analyzing one website, crawling a maximum of 100 pages and only one trial license can be obtained per IP address.

2.3.1 Results

Sitemaps can be created by providing an URL to the site or an XML sitemap file containing several page URLs that was already created by parsing the site in another tool. Creation of sitemaps took under 5 minutes for 100 sitemap nodes representing pages. Visual sitemaps can be created in several styles, they are available under the names: Default, Tree, Circle, Folder, Thumbnail.

As DynoMapper is already an online tool with an online viewer of sitemaps it provides the option to create shareable web links where the sitemap can be viewed without needing to log into the site. Another

	License	Trial	Standard	Organization	Enterprise
Price/month		\$0	\$40	\$159	\$399
Saved Sitemaps		0	25	50	100
Pages/sitemap		100	5000	25000	200000

Table 2.3: Prices and feature restrictions per license with Dyno Mapper.

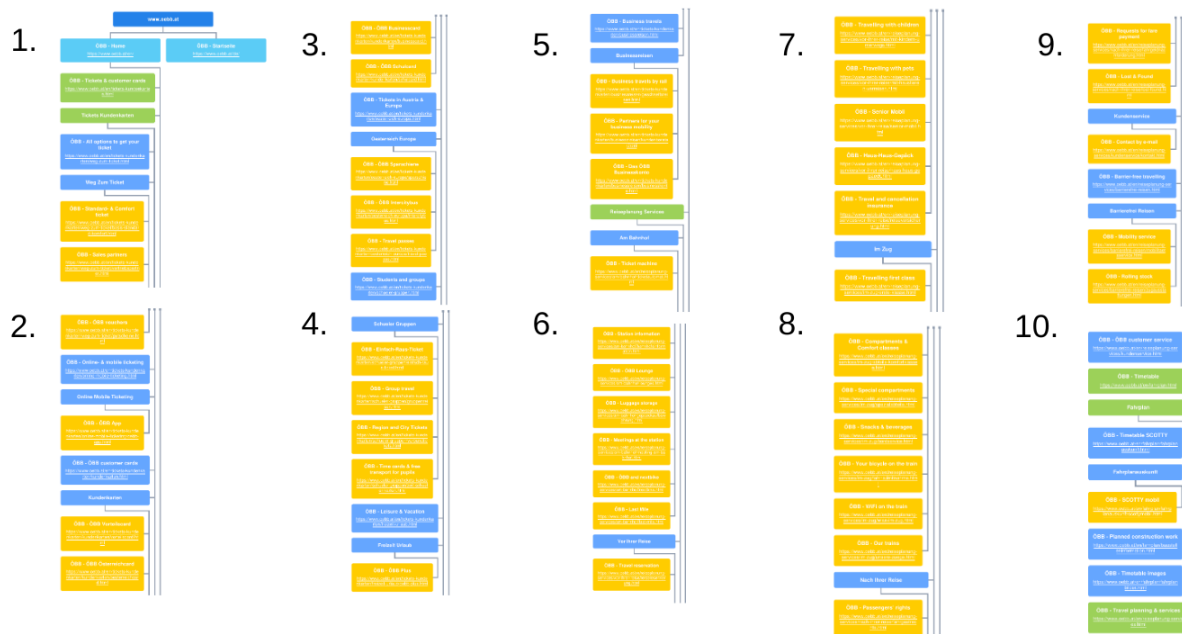


Figure 2.14: DynoMapper ÖBB PDF export pages 1 - 10 of 17. [Screenshot taken by the authors of this paper.]

important export format is PDF. Besides that, there are export format for just the sitemap like XML and TXT.

2.3.2 Conclusion

The default style works well when exploring the visual sitemap in the online viewer, one can click on nodes to get an overview of a subtree as seen in Figure 2.15. Exporting this default view as PDF in turn does not give a good overview, nodes are listed under each other, PDF page for page, and the connections cannot be traced, as shown in Figure 2.14. The best visualization style to export as PDF is the Tree view, it provides a good overview over the whole tree on one page of the PDF, as can be seen in Figure 2.16.

2.4 Visual Site Mapper

This free online service Visual Site Mapper AS 2019 was released by the software allentum in 2014. As its definition says, it has only an online application, therefore, no commercial version is available. This service consists on a very basic process that starts with the URL of a website as an input. The whole crawl process takes about two minutes and the result is a conceptual map of about 100 to 200 nodes. It is not possible to export the results, nor to share them. The service offers a unique style of site mapping which can be seen in Figure 2.17.

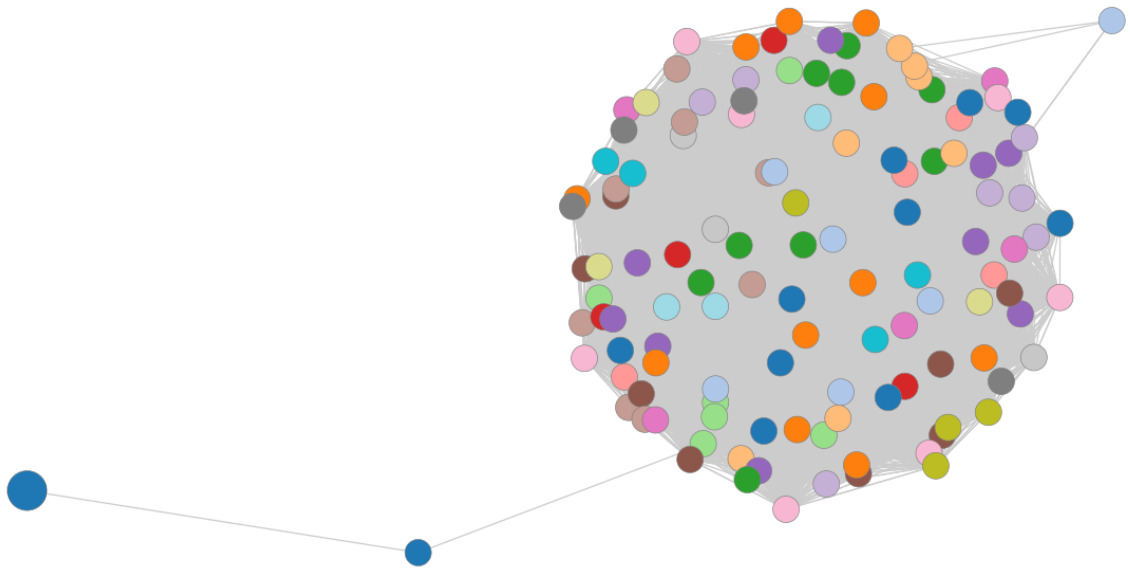


Figure 2.17: The unique style that the tool offers. [Screenshot taken by the authors of this paper.]

www.oebb.at

Top 165 pages are shown

Highlight links: All Outgoing Incoming

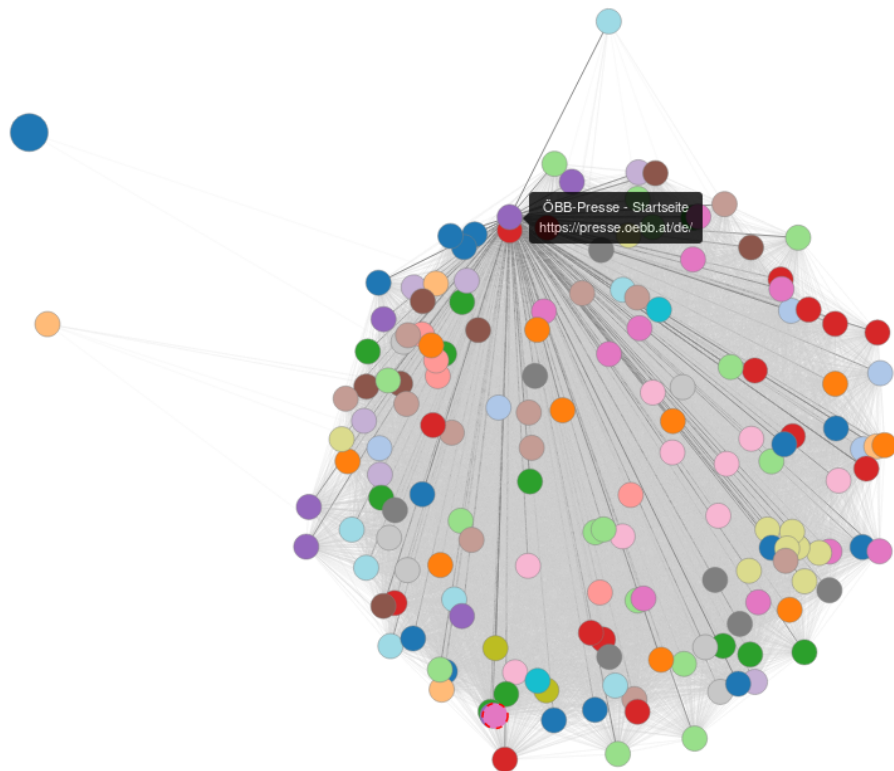


Figure 2.18: The resulting visual sitemap of the ÖBB website [Screenshot taken by the authors of this paper.]

2.4.2 Conclusion

As a conclusion, this basic tool does not allow to choose or define any preference for the resulting visual sitemap. Therefore, the result of the crawl is neither well structured nor explained and does not clear anything about the information architecture of the website. For this reason, this visual sitemap generator does not fulfill the main objective of its purpose, which is to make a clear vision of the hierarchy and information architecture of a website. In addition, if the input is `developer.db.com`, which is a single page application website, we got an error due to the `robots.txt` settings. Therefore, it could be said that this tool is not prepared for all kinds of websites.

Chapter 3

Content Audit

Content audit is the process of inventorying all the indexable content on a domain to analyze it. This part of the survey will analyze the programs that assist with this process. Thus, we will analyze 4 different tools (Screaming Frog, URL profiler, Scrapy and Content Analysis Tool) taking into account that we are looking for a list of every URL on the site with its characteristics, trusting that the designer has designed and kept the site with reasonable URLs. In addition to the main utility of these content inventory programs, some of them have the possibility of creating visualization trees or XML sitemaps.

3.1 Screaming Frog

The Screaming Frog SF 2019 site crawler, compatible with MacOS, Windows and Ubuntu, is one of the most commonly used tools for the content audit process. A professional license (£149.00 per year) was kindly provided by the Screaming Frog support. Using said license, an unlimited number of URLs could be crawled (only 500 URLs for the free version) with customizable analysis and the possibility of having XML sitemaps and visualization trees of the results. The crawling process lasts for about 5 minutes and only the URL of the website is needed as an input for the analysis. As a result there will be a deep study of each URL, as can be seen in Figure 3.1, of the website with the possibility of exporting the result as a CSV or Excel sheet. Nevertheless, as this tool is the first step of a two step progress, one only need to work with the exports of all the internal links that one will analyze again in the next tool (URL Profiler).

For this tool the input example keeps being the ÖBB website. The process lasted for about 4 minutes and analyzed 2359 nodes. As it can be seen each URL has its own analysis taking into account different elements that can be taken off as the customize analysis option. Overall, it is a very simple and efficient tool that gives a wide analysis of all the URLs and, as an extra, it creates visualization trees and XML sitemaps (see in Figure 3.2) .

3.2 URL Profiler

URL Profiler 3M 2019 provided us with a one-week trial version, which works as the professional version. Regarding the price of the tool, different options can be find due to getting adjusted to the customer preferences (more details in Table 3.1) but, in general terms, the prices range from €19,95 - €89,95 per month.

Regarding compatibility, it is suitable for Windows and MacOS. As input, there are several option such as XML sitemap, Screaming Frog SEO Spider File or a CSV document. On the contrary, as output options, there is only the XML export option.

After having introduced the basic information about this tool, the following paragraph will analyze the process that should be done to have the website audited. The objective of this tool is to get a deeper analysis of each URL by using the Screaming Frog results as an input and by choosing the points you

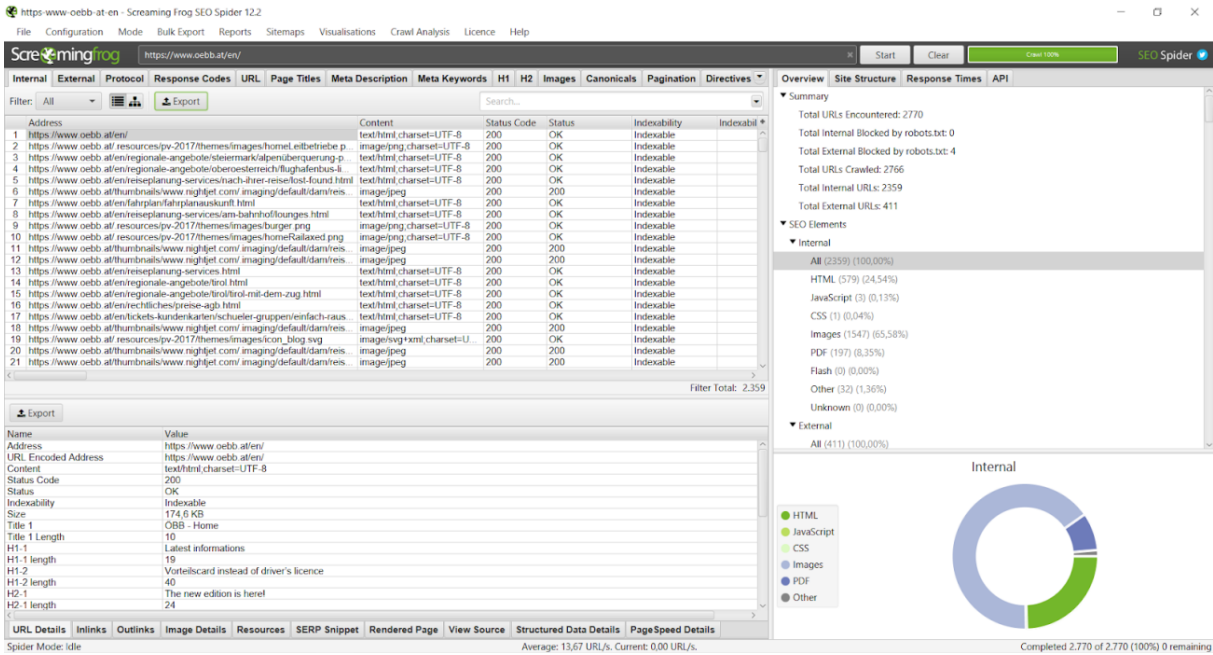


Figure 3.1: Solution of the crawl of the ÖBB website [Screenshot taken by the authors of this paper.]

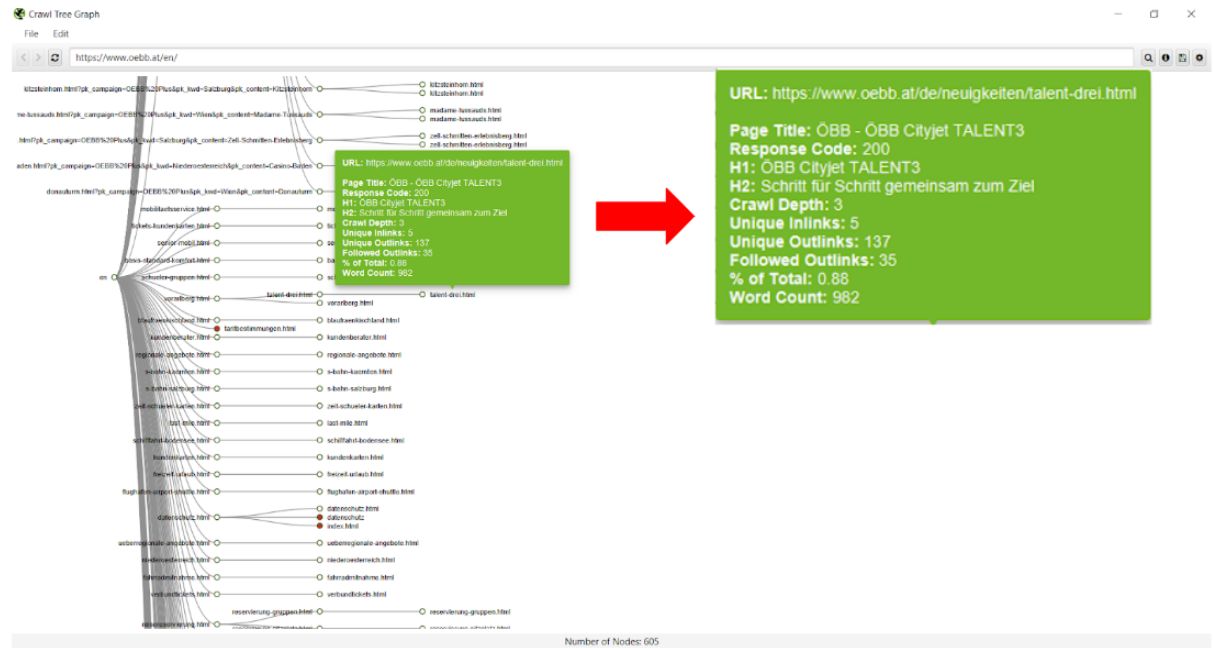


Figure 3.2: Visualization Tree by Screaming Frog of the ÖBB website. [Screenshot taken by the authors of this paper.]

License	SOLO	PRO	Agency
Price/month	\$29,95	\$39,95	\$99,95
Price/month-yearplan	\$19,95	\$25,95	\$64,95
Max URL/import	5000	100000	100000
URL/month	Unlimited	Unlimited	Unlimited
Number of Devices License	1	2	20

Table 3.1: Prices and feature restrictions per license with URL Profiler.

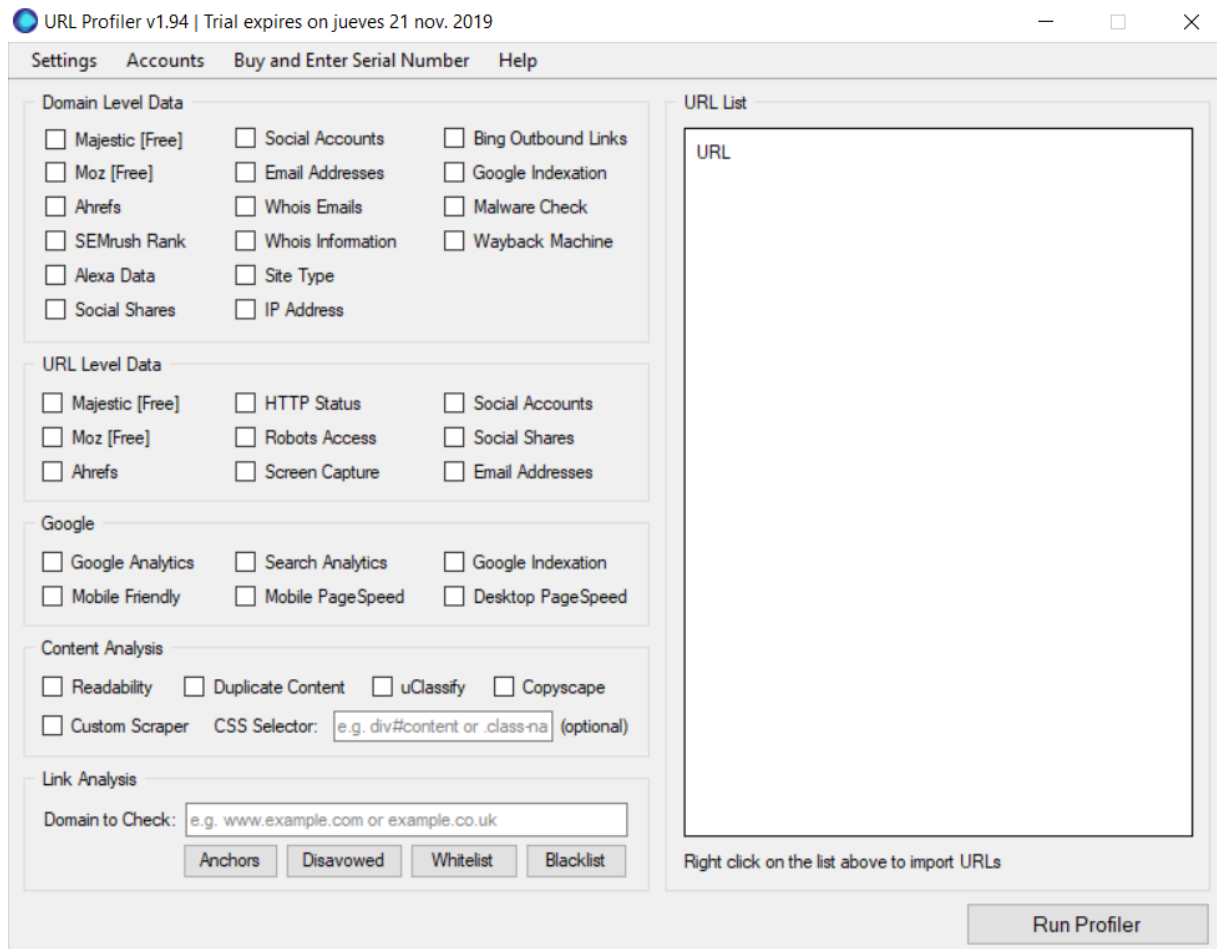


Figure 3.3: Initial status of the program [Screenshot taken by the authors of this paper.]

want to take into consideration for the analysis. As it can be seen in Figure 3.3, the analysis is completely customizable. However, for the content audit we are only interested in the URL level data and in the content analysis. Hence, the elements that are needed for this are: Majestic, HTTP status, Robot Access, Screen Capture, Social Shares, Google Analytics, Readability and Duplicate Content. Thanks to this, information about the status of each URL, about data users and about the rendering of each URL are going to be collected. Having taken all this into consideration, the tool will create two different kind of documents: A spreadsheet with the analysis of the website (seen in Figure 3.4) and a folder with an screenshot of every URL, as can be seen in Figure 3.5. The overall process will last more than 2 hours.

In the spreadsheet export a wide analysis of the website will be found. As shown in Figure 3.6, the tool does not analyze all the URLs and all the factors that have been chosen even though the crawling process lasted for more than 2 hours. Therefore, it does not work as good as it should, although it analyses has a lot of elements that can be useful to verify the URL status.

On the other hand, in relation to the images folder, it is useful to see how URLs are rendering. This can be helpful to see which URLs are lacking in content, as can be seen in Figure 3.7. In consequence, it makes the process easier and faster so you can check if the website is working as it is expected and, if it is not, you can determine it efficiently.

URL	DNS Safe URL	Path	Title
https://www.oebb.at/en/	https://www.oebb.at/en/	/en/	ÖBB - Home
https://www.oebb.at/.resources/pv-2017/themes/images/homeLeitbetriebe.png	https://www.oebb.at/.resources/pv-2017/th	.resources/pv-2017/themes/ima	
https://www.oebb.at/en/regionale-angebote/steiermark/alpenÄ¼berquerung-panor	https://www.oebb.at/en/regionale-angebote	/en/regionale-angebote/steierm	
https://www.oebb.at/en/regionale-angebote/oberoesterreich/flughafenbus-linz.htm	https://www.oebb.at/en/regionale-angebote	/en/regionale-angebote/oberoes	ÖBB - Airport Bus Linz
https://www.oebb.at/en/reiseplanung-services/nach-ihrer-reise/lost-found.html	https://www.oebb.at/en/reiseplanung-servic	/en/reiseplanung-services/nach-i	ÖBB - Lost & Found
https://www.oebb.at/thumbnails/www.nightjet.com/.imaging/default/dam/reisepor	https://www.oebb.at/thumbnails/www.night	/thumbnails/www.nightjet.com/	
https://www.oebb.at/en/fahrplan/fahrplanauskunft.html	https://www.oebb.at/en/fahrplan/fahrplan	/en/fahrplan/fahrplanauskunft.h	ÖBB - Timetable SCOTTY
https://www.oebb.at/en/reiseplanung-services/am-bahnhof/lounges.html	https://www.oebb.at/en/reiseplanung-servic	/en/reiseplanung-services/am-be	ÖBB - ÖBB Lounge
https://www.oebb.at/.resources/pv-2017/themes/images/burger.png	https://www.oebb.at/.resources/pv-2017/th	.resources/pv-2017/themes/ima	
https://www.oebb.at/.resources/pv-2017/themes/images/homeRailaxed.png	https://www.oebb.at/.resources/pv-2017/th	.resources/pv-2017/themes/ima	

Figure 3.6: Spreadsheet result with some of the most interesting columns to take into account of the ÖBB website as an example. [Screenshot taken by the authors of this paper.]

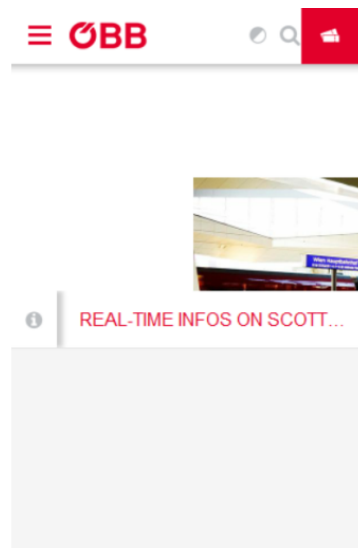


Figure 3.7: Screenshot of a URL with lack of content of the ÖBB website. [Screenshot taken by the authors of this paper.]

3.3 Content Analysis Tool (CAT)

Content Analysis Tool (CAT) CI 2019 is a crawler that works as an online tool. This evaluation is based on the trial version with a limit of 250 pages per website. It has a price structure with fixed amounts of pages and monthly amounts of pages, as shown in Table 3.2.

A web crawl can be started by providing a URL. A crawl of 265 pages took 12 minutes to complete. The web interface of CAT only shows limited meta information of the result URLs, in turn the CSV export provides many more columns with metadata, as can be seen in Figure 3.8 and Figure 3.9.

For the Deutsche Bank Developer website CAT returned a full set of URLs, but in closer inspection none of those URLs were of the subdomain provided with the single page application. It is inconclusive if this result is because of the robots.txt or the single page application.

3.4 Scrapy

Scrapy SH 2019 is an open source Python library that can be used to extract data from websites. The output of such an extraction can be JSON, JSON Lines, JL, CSV, XML, marshal or pickle file. One can interact directly with the sites structure using CSS selectors, such as tag, class name and many more. The tool also supports selections using XPath. It offers an intuitive API that simplifies the task of website scraping. As an example, Spiders and LinkExtractors can be named. A Spider is a Python class that defines how to perform a crawl and how to scrape a site. The former is defined by setting URLs as starting locations, restricting the crawl to specific domains and creating Rules. Rules define a certain behavior to

	License	Trial	Level 1	Level 2	Level 3	Basic	Professional	Enterprise
Price once	-	\$49	\$69	\$99	-	-	-	-
Price/month	-	-	-	-	\$79	\$179	\$849	
Pages Total	250	5000	10000	20000	25000	50000	250000	

Table 3.2: Prices and feature restrictions per license for Content Analysis Tool.

URL	Type	Size	Level	Title	Word Count	In Scope	View Details
https://www.deutsche-bank.de/	text/html	-1	0	(Redirect to https://www.deutsche-bank.de/pfb/content/privatkunde		true	▶
.../pfb/content/privatkunden/privatkunden.html	text/html	-1	0	(Redirect to https://www.deutsche-bank.de/pfb/content/privatkunde		true	▶
.../pfb/content/privatkunden.html	text/html	-1	0	(Redirect to https://www.deutsche-bank.de/pfb/content/privatkunde		true	▶
.../pk.html	text/html	210747	0	(Redirect to https://www.deutsche-bank.de/pfb/content/privatkunde	2703	true	▶
.../etc/designs/db-eccs-pwcc-clientlib-site.914dae82a2c4087a0106d80faed78485.css	text/css	38229	1			true	▶
.../etc/designs/db-eccs-pwcc-clientlib-site-print.5c071051bc421a402584f1df77cfaad3.css	text/css	7399	1			true	▶

Figure 3.8: Content Analysis Tool online [Screenshot taken by the authors of this paper.]

crawl the site. By defining parse methods in the Spider, one tells Scrapy how to extract data from a page it currently has crawled. LinkExtractors on the other hand offer a possibility to extract links from a web page. By setting such LinkExtractors to the Spider Rules, one can tell Scrapy to visit all links it finds on a web page recursively. This behavior can then be exploited to create a CSV containing information about the web page, such as title, HTTP status code and more.

3.4.1 Proof of Concept

As a proof of concept, a recursive link extraction example was taken and adapted with permission from Jacobs 2016. From now on, the example will be called "oebb_scraper". The oebb_scraper retrieves all links from the ÖBB website by visiting them recursively. It will create a CSV with 5 columns "url", "content_type", "status", "title" and "h1", whereas the latter is simply the content of the first h1 tag it finds. This example can be expanded to any number of fields an HTTP response returns, as this will be one of the objects the oebb_scraper uses to create the CSV.

To start a Scrapy project, one can use the CLI and call `scrapy startproject oebb_scraper`. This will create an empty Scrapy project without Spiders. In `items.py`, Scrapy items are defined, which contain the data one wants to write into the CSV export. Listing 3.1 shows the class `OebbContentAuditItem` with the fields that are needed for the oebb_scraper.

Afterwards, a Spider needs to be created to tell the scraper what and how to scrape. Calling `scrapy genspider oebb_content_audit www.oebb.at` tells Scrapy to create a Python file within the oebb_scraper Scrapy project containing a Spider called "oebb_content_audit" starting at the URL `www.oebb.at`. Using `FEED_EXPORT_FIELDS` in the `custom_settings` member of the Spider, one can set the order of the columns of the CSV export, but it is not required. By setting a LinkExtractor as a rule of the Spider with `follow=True` and `callback="parse_items"`, the Spider will now extract all links beginning at the starting URL and follow the retrieved links recursively. The visited links are then fed to the `parse_items` function of the Spider with the HTTP response as parameter. The URL and status can be directly retrieved from the HTTP response, the content type can be retrieved from the response headers and the title and h1 content can be retrieved using the `::text` CSS pseudo selector. The final spider can be viewed in Listing 3.2.

Calling `scrapy crawl oebb_content_audit -o content_inventory.csv` within the root directory of the Scrapy project will now start crawling the page and creating the CSV with the defined columns. Referring to Figure 3.10, the first 20 rows of the resulting content inventory can be seen. In the end, 802 URLs were filtered, all of them had content in the title tag and only some were missing h1

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Location	In Scope	Type	Size	Date	Level	Title	Description	Keywords	H1TagTexts	Word Count	LinksIn	LinksOut
2	https://www.deutsche-bank.de/	TRUE	unknown		2019-11-05T13:3	0	(Redirect to https://www.deutsche-bank.de/pdf/contentprivatkunden/privatkunden.html)					193	
3	https://www.deutsche-bank.de/pdf/contentprivatkunden/privatkunden.html	TRUE	unknown		2019-11-05T13:3	0	(Redirect to https://www.deutsche-bank.de/pdf/contentprivatkunden/privatkunden.html)					0	
4	https://www.deutsche-bank.de/pdf/contentprivatkunden.html	TRUE	unknown		2019-11-05T13:3	0	(Redirect to https://www.deutsche-bank.de/pdf/contentprivatkunden/privatkunden.html)					2	
5	https://www.deutsche-bank.de/ik.html	TRUE	text/html	210747	2019-11-05T13:3	0	(Redirect to https://www.deutsche-bank.de/ik.html)				2703	210	
6	https://www.deutsche-bank.de/etcdesigns/ib-ecss-pwsc/clientsite/914dae82a2c4087a010680fe	TRUE	text/css	38229	2019-10-09T22:5	1							146
7	https://www.deutsche-bank.de/etcdesigns/ib-ecss-pwsc/clientsite/print/5c071051bc421a402584	TRUE	text/css	7399	2019-10-09T22:5	1							146
8	https://www.deutsche-bank.de/pdf/content/quantierzukunft/home.html	TRUE	unknown		2019-11-05T13:4	1	(Redirect to https://www.deutsche-bank.de/quantierzukunft/home.html)					159	
9	https://www.deutsche-bank.de/pdf/content/marktinformationen/index.html	TRUE	text/html	126066		1	Deutsche Bank - ÄBERSICHT Kurse & MÄRKTE	MÄRKTE Marktberi			2173	161	
10	https://www.deutsche-bank.de/pdf/content/ik-suche.html	TRUE	text/html	27645		1	Ihre Suche bei Deutsche Bank Privatkunden	Ihre Suche			563	158	
11	https://www.deutsche-bank.de/pdf/content/ik-filialsuche.html	TRUE	text/html	51918		1	Filial- und Geldautomatenliste bei Deutsche Bank Privatkunden				954	145	
12	https://www.deutsche-bank.de/ik/international-clients.html	TRUE	text/html	256368	2019-11-05T13:3	1	International Client From account an International Client Private Clients				3612	143	
13	https://www.deutsche-bank.de/opa4/ib/advisor-appointments	TRUE	unknown		2019-07-10T16:1	1	(Redirect to https://www.deutsche-bank.de/opa4/ib/advisor-appointments)					108	
14	https://www.deutsche-bank.de/v6/index.htm	TRUE	text/html	29747	2019-01-16T15:5	1	Home AC&A: D Unser Wealth Management hilft Privatpersonen und				687	168	
15	https://www.deutsche-bank.de/ik.html	TRUE	text/html	346574	2019-11-05T13:3	1	Geschäftskund Deutsche Bank C Geschäftskunder Mehr Flexibilität				5307	210	
16	https://www.deutsche-bank.de/ik/	TRUE	text/html	37808	2019-10-28T08:4	1	Home ac: Deutsche Lösungen und Si Existenzgründun Mehr Service für				1012	192	
17	https://www.deutsche-bank.de/ik/	TRUE	text/html	38708	2019-07-04T13:3	1	Home AC&A: D Lösungen und Si Cash Managem Umfassender Lei				1019	188	
18	https://www.deutsche-bank.de/ik/investments/investments-im-ueberblick.html	TRUE	text/html	220783	2019-11-05T13:3	1	Investments im F Mit fundierter Exp Expertise Invest Finden Sie mehr				2827	143	
19	https://www.deutsche-bank.de/ik/sparen/sparen-im-ueberblick.html	TRUE	text/html	252302	2019-11-05T13:3	1	Sparen und Anle initiative Sparen Anlage Sparen und Anle				3490	143	
20	https://www.deutsche-bank.de/ik/investments/investments-im-ueberblick/private-banking/leistungsueber	TRUE	text/html	218378	2019-11-05T13:3	1	LeistungsÜber: Hier finden Sie a investments priv Jede Generation				2903	144	
21	https://www.deutsche-bank.de/ik/investments/investments-im-ueberblick/private-banking/leistungsueber	TRUE	text/html	200489	2019-11-05T13:3	1	LeistungsÜber: Hier finden Sie a investments priv Jede Generation				2411	149	

Figure 3.9: Content Analysis Tool export [Screenshot taken by the authors of this paper.]

```

1 ##### items.py #####
2 class OebbContentAuditItem(scrapy.Item):
3     # define the fields for your item here like:
4     url = scrapy.Field()
5     content_type = scrapy.Field()
6     status = scrapy.Field()
7     title = scrapy.Field()
8     h1 = scrapy.Field()

```

Listing 3.1: Item class for the ÖBB content audit

content. The results were very promising not only in regard to completeness of the content audit, but also in the extendability of Scrapy. One can further add more columns as from the HTTP response object or from the content of the page with ease. The only restriction is that the user has to have basic Python programming knowledge as no GUI is provided with this tool.

```
1 ##### spiders/oebb_content_audit.py #####
2 import scrapy
3 from scrapy.linkextractors import LinkExtractor
4 from scrapy.spiders import Rule, CrawlSpider
5 from oebb_scraper.items import OebbContentAuditItem
6
7 class OebbContentAuditSpider(CrawlSpider):
8     name = "oebb_content_audit"
9
10    allowed_domains = ["www.oebb.at"]
11
12    start_urls = ["https://www.oebb.at/"]
13    custom_settings = {
14        # specifies exported fields and order
15        'FEED_EXPORT_FIELDS': ["url", "content_type", "status", "title", "h1"],
16    }
17
18    rules = [
19        Rule(
20            LinkExtractor(
21                canonicalize=True,
22                unique=True
23            ),
24            follow=True,
25            callback="parse_items"
26        )
27    ]
28
29    def start_requests(self):
30        for url in self.start_urls:
31            yield scrapy.Request(url, callback=self.parse, dont_filter=True)
32
33    def parse_items(self, response):
34        item = OebbContentAuditItem()
35        item['url'] = response.url
36        item['content_type'] = response.headers.get('Content-Type')
37        item['status'] = response.status
38        item['title'] = self.get_tag_text(response, 'title')
39        item['h1'] = self.get_tag_text(response, 'h1')
40        return item
41
42    @staticmethod
43    def get_tag_text(response, tag):
44        # ::text css pseudo selectors
45        text = response.css(tag + '::text').get()
46        if text is None:
47            return ''
48        return text.strip()
```

Listing 3.2: Spider class for the ÖBB content audit

	A	B	C	D	E
1	url	content_type	status	title	h1
2	https://www.oebb.at/	text/html;charset=UTF-8	200	OBB - Startseite	Topaktuelle Informationen
3	https://www.oebb.at/de/reiseplanung-services/vor-ihrer-reise/reisereservierung.html	text/html;charset=UTF-8	200	OBB - Reiseservierung	Reiseservierung
4	https://www.oebb.at/de/reiseplanung-services/am-bahnhof/bahnhofsinformation.html	text/html;charset=UTF-8	200	OBB - Bahnhöfe & Verkaufsstellen	Bahnhöfe & Verkaufsstellen
5	https://www.oebb.at/de/reiseplanung-services/vor-ihrer-reise/haus-haus-gepaeck.html	text/html;charset=UTF-8	200	OBB - Haus-Haus-Gepäck	Haus-Haus-Gepäck
6	https://www.oebb.at/de/reiseplanung-services/vor-ihrer-reise/reiseversicherung.html	text/html;charset=UTF-8	200	OBB - Reiseversicherung	Stornoversicherung
7	https://www.oebb.at/de/reiseplanung-services/vor-ihrer-reise/mit-kindern-unterwegs.html	text/html;charset=UTF-8	200	OBB - Mit Kindern unterwegs	Mit Kindern unterwegs
8	https://www.oebb.at/de/reiseplanung-services/vor-ihrer-reise/senior-mobil.html	text/html;charset=UTF-8	200	OBB - Senior Mobil BeraterInnen	Senior Mobil
9	https://www.oebb.at/de/reiseplanung-services.html	text/html;charset=UTF-8	200	OBB - Reiseplanung & Services	Reiseplanung & Services
10	https://www.oebb.at/de/fahrplan/fahrplanbilder.html	text/html;charset=UTF-8	200	OBB - Fahrplanbilder	Fahrplanbilder
11	https://www.oebb.at/de/reiseplanung-services/vor-ihrer-reise/mit-haustieren-verreisen.html	text/html;charset=UTF-8	200	OBB - Mit Haustieren verreisen	Mit Haustieren verreisen
12	https://www.oebb.at/de/fahrplan/baustelleninformation.html	text/html;charset=UTF-8	200	OBB - Geplante Baustellen	Geplante Baustellen
13	https://www.oebb.at/de/fahrplan/fahrplanauskunft/scottymobil.html	text/html;charset=UTF-8	200	OBB - SCOTTY mobil	SCOTTY mobil
14	https://www.oebb.at/de/fahrplan.html	text/html;charset=UTF-8	200	OBB - Fahrplan	Fahrplan
15	https://www.oebb.at/de/fahrplan/fahrplanauskunft.html	text/html;charset=UTF-8	200	OBB - Fahrplanauskunft SCOTTY	Fahrplanauskunft
16	https://www.oebb.at/de/tickets-kundenkarten/businessreisen/businesskonto.html	text/html;charset=UTF-8	200	OBB - Das ÖBB Businesskonto	Das ÖBB Businesskonto
17	https://www.oebb.at/de/tickets-kundenkarten/businessreisen/kundenberater.html	text/html;charset=UTF-8	200	OBB - Geschäftskunden-BeraterInnen	Geschäftskunden-BeraterInnen
18	https://www.oebb.at/de/rechtliches/datenschutz-videoüberwachung.html	text/html;charset=UTF-8	200	OBB - Datenschutzinformationen zur Videoüberwachung	Informationen zur Video-Überwachung
19	https://www.oebb.at/de/rechtliches/zertifizierung.html	text/html;charset=UTF-8	200	OBB - Zertifikat für Barrierefreiheit	Unsere Website ist nun barrierefrei-zertifiziert!
20	https://www.oebb.at/static/tarife/de/index.html	text/html;charset=UTF-8	200	OBB Tarife - Tarifbestimmungen	

Figure 3.10: The resulting content inventory after running the oebb_scraper Scrapy project. [Screenshot taken by the authors of this paper.]

Chapter 4

Conclusion

A common task for information architects, web designers and many more, is to get an overview of the hierarchy of a website. Creating sitemaps is a usual approach to this problem. To get a better grasp on said sitemap, it is often represented as a tree or graph in a visual sitemap. Since the manual approach of creating them is time consuming, there exists a small variety of tools that generate visual sitemaps automatically, such as PowerMapper, DynoMapper, VisualSitemapper and VisualSitemaps. None of these tools produce a visual sitemap of the same quality as a manually created one. They all come with their downsides but an important upside can still be noted: The active time needed to generate a visual sitemap is a maximum of a few minutes (ignoring the time the tools take to generate the sitemaps). VisualSitemapper is the only free tool, but since its features are very limited it cannot really be recommended for professional use. VisualSitemaps is only available for monthly plans and can be recommended for smaller websites but the results for larger websites become quite cluttered and unreadable. DynoMapper is the most expensive tool but it offers better usability of exports for larger websites. PowerMapper is the only paid tool that can be bought with a one-time license. Not only is the hierarchy produced by the tool the best but also the export and navigation features for larger websites stand out.

Content audits on the other hand provide a possibility to get a quick overview over all pages available on a website. In addition to this overview, one can also scan the pages for their title, header, HTTP status codes and so on. The variety of content audit tools is larger than of visual sitemap tools and URL Profiler, Screaming Frog, Content Analysis Tool and Scrapy were some of them. Screaming Frog and URL Profiler are too expensive for the results they produce. The URL Profiler crawl lasts too long. After 3 hours it has not processed every URL and most of the elements that should have been analyzed were empty or its value were null. The performance of Screaming Frog is significantly better. The crawling is fast, and it offers additional features like the XML and visualization trees that work very well. The result and performance of the Content Analysis Tool (CAT) is comparable to Screaming Frog. Scrapy presents a completely different approach to this problem. It is a framework for Python and does not provide a GUI. Not only were the results comparable to the paid tools, but it even provided far better extendability than the others. Additionally, it is open source and released under the BSD license, therefore it is free to use. For users with basic programming experience, Scrapy can definitely be recommended.

Bibliography

- 3M [2019]. *Powerful Content and Back Link Auditor Software*. 301 Media. 24 Nov 2019. <https://urlprofiler.com> (cited on page 17).
- AS [2019]. *Create a visual map of your site*. Alentum Software. 24 Nov 2019. <http://visualsitemapper.com> (cited on page 13).
- CI [2019]. *Content Inventory and Analysis Made Easier*. Content Insight. 24 Nov 2019. <https://content-insight.com> (cited on page 21).
- DM [2019]. *Create Sitemaps - Sitemap Generator - Visual Sitemap Generator*. Dynamapper. 24 Nov 2019. <https://dynamapper.com> (cited on page 12).
- Jacobs, Kevin [2016]. *How to scrape a website using Python + Scrapy in 5 simple steps*. 2016. <https://www.data-blogger.com/2016/08/18/scraping-a-website-with-python-scrapy/> (cited on page 22).
- PM [2019]. *Website Testing and Site Mapping Tools*. PowerMapper. 24 Nov 2019. <https://powermapper.com> (cited on page 5).
- SF [2019]. *SEO, Search Engine Marketing & Optimisation Agency*. Screaming Frog. 24 Nov 2019. <https://screamingfrog.co.uk> (cited on page 17).
- SH [2019]. *A Fast and Powerful Scraping and Web Crawling Framework*. Scrapinghub. 24 Nov 2019. <https://scrapy.org> (cited on page 21).
- VS [2019]. *Autogenerate Beautiful Sitemaps & Screenshots*. VisualSitemaps. 24 Nov 2019. <https://visualsitemaps.com> (cited on page 9).