Web Content Management Systems Archivability

Vangelis Banos, Yannis Manolopoulos

Department of Informatics, Aristotle University, 54124 Thessaloniki, Greece

Abstract. Web archiving is the process of collecting and preserving web content in an archive for current and future generations. One of the key issues in web archiving is that not all websites can be archived correctly due to various issues that arise from the use of different technologies, standards and implementation practices. Nevertheless, one of the common denominators of current websites is that they are implemented using a Web Content Management System (WCMS). We evaluate the Website Archivability (WA) of the most prevalent WCMS. We investigate the extent to which each WCMS meets the conditions for a safe transfer of their content to a web archive for preservation purposes, and thus identify their strengths and weaknesses. More importantly, we deduce specific recommendations to improve the WA of each WCMS, aiming to advance the general practice of web data extraction and archiving.

1 Introduction

The web has moved from small informal websites to large and complex systems, which require strong software systems to be managed effectively [4]. The increasing needs of organisations and individuals in this area have led to the rise of a new type of software, i.e. Web Content Management Systems (WCMS) [15]. WCMS are created in various different programming languages, using many new web technologies [9]. There are millions of websites using WCMS; for instance, Wordpress is used by 74.6 million websites[1], whereas Drupal is used by more than one million websites[2]. WCMS have a large contribution in the development of the web. However, we must not overlook the fact that the web is an ephemeral communication medium. The average lifetime of a web page is below 100 days [14], making it necessary to archive the web to preserve information for current and future generations. Web archiving is the process of retrieving material from the web and preserving them in an archive, making them perpetually available for access and research [16].

One of the open issues in web archiving is that not all websites are amenable to being archived with correctness and accuracy. To define and measure this website behavior we have previously introduced the metric of *Website Achivability (WA)*. WA is defined as the extent to which a website meets the conditions for

[1] https://managewp.com/14-surprising-statistics-about-wordpress-usage

[2] https://www.drupal.org

the safe transfer of its content to a web archive for preservation purposes [1, 2]. The open and continuously evolving nature of the web makes it difficult to predict the WA of a website. There is a very large number of different combinations of technologies, standards and development approaches used in web development which affect WA.

We believe that the wide adoption of WCMS has benefits for web archiving and needs to be taken into consideration. WCMS constitute a *common technical framework* which may facilitate or hinder web archiving for a large number of websites. If a web archive is compatible with a certain WCMS, it is highly probable that it will be able to archive all websites built with this WCMS.

In this work, we evaluate the WA of 12 prominent WCMS to identify their strengths and weaknesses and propose improvements to improve web content extraction and archiving. We conduct an experimental evaluation using a non-trivial dataset of websites based on these WCMS and make observations regarding their WA characteristics. We also come up with specific suggestions for each WCMS based on our experimental data.

Our aim is to improve the web archiving practice by indicating potential issues to the WCMS development community. If our findings result in advances in WCMS source code upstream, all web archiving initiatives will benefit as the websites based on these WCMS will become more archivable. The main contributions of this work are:

– specific observations regarding the WA Facets of 12 prominent WCMS,
– recommendations to improve each WCMS source code upstream to improve their WA.

This paper is organised as follows: Section 2 presents work related to WCMS archiving. Section 3 presents the CLEAR+ method to evaluate WA. Section 4 presents the ArchiveReady WA evaluation system, our experimental method and the results which are discussed further in Section 5.

2 Related work

WCMS have been already studied in the context of web archiving due to their wide scale usage on the web. According to W3Techs, 38% of the top 1 million websites area are created using a WCMS. Gomez et al., the creators of the Portuguese web archive, report that there are millions of websites which are supported by a small number of publishing platforms. During the development of their web crawling process, they have come up with specific rules to harvest specific WCMS, such as Joomla, because they did not allow crawlers to harvest all their files [11]. Faheem et al. have also presented an approach to create Application Aware Helpers (AAH) that fit into the archiving crawl process chain to perform intelligent and adaptive crawling of web applications. In their work they are focusing on specific WCMS [8]. Pinsent et al. have dedicated a chapter in the Preservation of Web Resources Handbook 2008 regarding Content Management Systems Archiving. They mention that some WCMSs may present problems to a web crawler. The content gathering may be incomplete or the web crawler may get stuck in a ‘loop’ as it constantly requests pages. This behaviour depends on the specific implementation and the WCMS used [18]. Rumianek proposes a procedure to overcome the problems faced by archivists of database-driven websites such as WCMS [19]. Although interesting, this approach is not practical as it requires the implementation of new systems on top of existing web archiving platforms.

There has also been some work to try archiving content from blogs, which are a special kind of WCMSs. Pennock et al. have created ArchivePress, a specialised Wordpress CMS archiving tool, which creates plugins for WordPress to make it operate as an archiving tool [17]. Kelly et al. have also investigated multiple alternatives on archiving blogs [13]. The BlogForever project has created a new approach to harvest, preserve, manage and reuse blog content [12]. Blanvillain at al. have presented their BlogForever crawler which concentrated on techniques to automatically extract content such as articles and comments from blog posts using a simple and robust algorithm to generate extraction rules based on string matching using the blog’s web feed in conjunction with blog hypertext [3].

There is interest in finding methods to archive WCMS-based websites but according to our knowledge, there has been no attempt to evaluate the feasibility of archiving different WCMS and highlight their strengths and weaknesses regarding web archiving. In our work, we try to conduct such an analysis using a substantial experimental dataset and a novel method.

3 CLEAR+: A Credible Live Evaluation of Archive Readiness Plus

The Credible Live Evaluation of Archive Readiness Plus method (CLEAR+) is an approach to produce on-the-fly measurement of WA, which is defined as the extent to which a website meets the conditions for the safe transfer of its content to a web archive for preservation purposes [1, 2]. We use the latest iteration of the CLEAR+ method as of 02/2015.

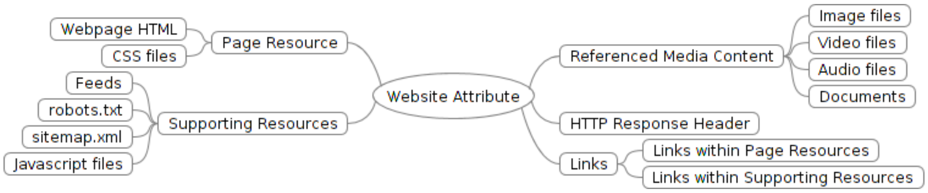

Fig. 1. Website attributes used for WA evaluation [1]

In short, the basic operating principles of our method is that we communicate through standard HTTP requests and responses with the target website in a similar manner as regular web archiving systems and retrieve information that we evaluate using recognized practices in digital curation (e.g. using adopted standards, validating formats, and assigning metadata) to generate a credible score representing the target WA. We measure WA from several different perspectives, which we call *WA Facets*, by conducting specific *Evaluations* on *Website Attributes* (Figure 1). For instance: What is the percentage of valid versus invalid hyperlinks? Do the CSS files used in a website comply with W3C standards? What is the percentage of corrupt image files, if any? The score for each WA Facet is computed as the weighted average of the scores of the evaluations associated with this Facet. The significance of each evaluation defines its weight. The WA Facets can be summarised as follows:

- F_A: *Accessibility* indicates the facilitation of web archiving systems to access and retrieve website content via standard web communication methods.
- F_C: *Cohesion* is the robustness of a website against the failure of different web services. This Facet is concerning websites which are dispersed across different services (e.g. different servers for images, javascript widgets, and other resources) in different domains.
- F_M: *Metadata Usage* indicates the adequate provision of metadata [6]).
- F_S: *Standards Compliance* indicates the encoding of digital resources using known and transparent standards.

The outcomes of the WA evaluation are different scores in the range of 0-100 for F_A, F_C, F_M and F_S. The final WA score is the average of the WA Facets scores.

4 Evaluation

We present the ArchiveReady system to evaluate WA and the method we follow to define and evaluate a significant corpus of websites. Finally, we present detailed results and we identify specific characteristics of different WCMSs.

4.1 ArchiveReady WA evaluation system

ArchiveReady [1, 2] is a real-time WA evaluation system, which is the reference implementation of the CLEAR+ method. It is available at `http://archiveready.com`. ArchiveReady is based on standard open source software and uses specialised tools to evaluate websites such as JHOVE for media file validation [7], W3C HTML[3], CSS[4] and RSS[5] validation services, as well as the PhantomJS headless WebKit browser to access and process websites[6]. The WA evaluation process can be summarised as follows:

[3] `http://validator.w3.org/`

[4] `http://jigsaw.w3.org/css-validator/`

[5] `http://validator.w3.org/feed/`

[6] `http://phantomjs.org/`

1. ArchiveReady receives a target URL and performs an HTTP request to retrieve the webpage hypertext.
2. After analysing it, multiple HTTP connections are initiated in parallel to retrieve all web resources referenced in the target webpage, imitating a web spider.
3. In stage 3, Website Attributes (Figure 1) are evaluated. In more detail: a) HTML and CSS analysis and validation, b) HTTP response headers analysis and validation, c) Media files (images, other objects) retrieval, analysis, and validation. d) Sitemap.xml and Robots.txt retrieval, analysis and validation, e) RSS feeds detection, retrieval, analysis and validation, f) Network transfer performance evaluation.
4. The metrics for the WA Facets are calculated according to the CLEAR+ method and the final WA rating is produced.

ArchiveReady provides a simple REST API to enable WA evaluation from 3rd party applications.

4.2 Website corpus evaluation method

We use 5.821 random WCMS samples from the Alexa top 1 million websites[7] as our experimental dataset. We use this dataset because it contains high quality websites from multiple domains and disciplines. This dataset is also used in other related research [20] [11]. We select our corpus with the following process:

1. We implement a simple python script to visit each homepage and look for the <meta name="generator" content="software name" /> tag.
2. For each website having the required meta tag, we evaluate if it belongs to one of the WCMSs listed in wikipedia[8]. If yes, we record it in our database.
3. We continue this process until we have a significant number of instances for 12 WCMSs (Blogger, DataLife Engine, DotNetNuke, Drupal, Joomla, Mediawiki, MovableType, Plone, PrestaShop, Typo3, vBulletin, Wordpress).
4. We evaluate each website using the ArchiveReady REST API and record the outcomes in our database.
5. We analyse the results using SQL to calculate various metrics.

The generator meta tag is not used universally on the web due to a variety of reasons such as security. Thus, we have skipped a large number of websites, which did not indicate the system they use. Also, we do not take into consideration the version number of each WCMS as it would be impractical. There would be too many different variables in our experiment to conduct useful research. Also, it is highly improbable that the top internet websites would use legacy versions of their WCMS. The Git repository for this paper[9] contains all the captured data and the necessary scripts to reproduce all the evaluation experiments.

[7] http://s3.amazonaws.com/alexa-static/top-1m.csv.zip

[8] http://en.wikipedia.org/wiki/List_of_content_management_systems

[9] https://github.com/vbanos/wcms-archivability-paper-data

4.3 Evaluation results and observations

For each WCMS, we present the average and standard deviation for each WA Facet, as well as their cumulative WA (Figure 2). First of all, our results are consistent. While the WA Facet range is 0-100%, the standard deviation of all WA Facet values for each WCMS ranges from 4.2% (Blogger, F_A) to 13.2% (Mediawiki, F_S). There are considerable differences between different WCMS regarding their WA. The top WCMS is DataLife Engine with a WA score of 83.52% with Plone and Drupal scoring also very high (83.06% and 82.08%). The rest of the WCMS score between 80.3% and 77.2%, whereas the lowest score belongs to Blogger (65.91%). In many cases, even though two or more WCMS may have similar WA score, their WA Facet scores are significantly different and each WCMS has different strengths and weaknesses. Thus, it is beneficial to look into each WA Facet differences.

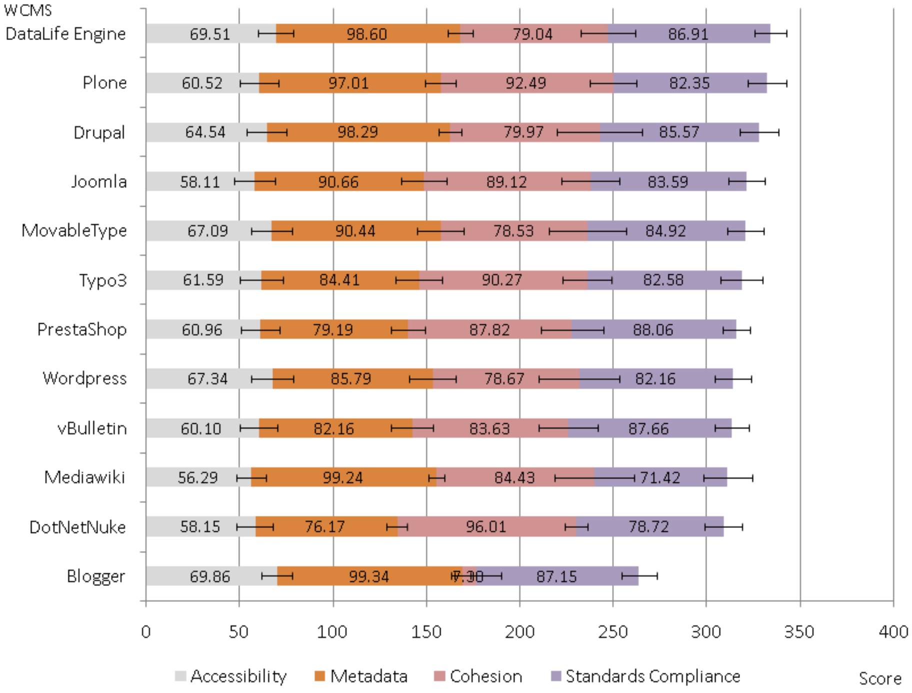

Fig. 2. WA Facets average values and standard deviation for each WCMS

F_A: Top value is around 69.85% for Blogger and 69.51% for DataLife Engine, whereas the minimum value is below 60, at 56.29% for Mediawiki and 58.15% for DotNetNuke.

F_M: Top value is 99.24% for Mediawiki, whereas the minimum value is 76.17% for DotNetNuke. The difference between the minimum and the maximum value is around 23 points, which almost twice the difference between F_A range (13).

F_C: Appears to have the greatest differentiation between WCMS. The minimum value is only 7.38% for Blogger and the maximum value is 96.01% for DotNetNuke. At first sight, there seems to be an issue with the way Blogger is using multiple online services to host its web resources. Other WCMSs also vary from 78.5% (MovableType) to 92% (Plone), which is a considerable variation.

F_S: Range is between 71.42% for Mediawiki and 88.06% for PrestaShop. Again these differences should be considered significant.

F_A has the smallest differentiation and F_C has the greatest one among all WA Facets.

We continue our research with more detailed observations regarding specific evaluations. Due to the large number of WA evaluations and the space restrictions imposed, we cannot present everything. We choose to discuss only highly significant rules. Similar research is easy to be contacted by anyone interested using the full dataset and source code available on github. We present our observations grouped by the four different WA Facets.

WCMS	Valid URLs	Invalid URLs	Correct (%)
Blogger	45425	1148	97%
Mediawiki	39178	1763	96%
Drupal	52501	2185	96%
MovableType	22442	1009	96%
vBulletin	104492	5841	95%
PrestaShop	57238	3287	94%
DataLife Engine	31981	2342	93%
Plone	25719	1856	93%
Wordpress	47717	3515	93%
DotNetNuke	38144	2791	93%
Typo3	30945	3747	89%
Joomla	37956	4886	88%

Table 1. A_1 The percentage of valid URLs. Higher is better.

***F_A*: Accessibility** refers to the web archiving systems' ability to traverse all website content via standard HTTP protocol requests [10].

A_1: The percentage of valid versus invalid hyperlink and CSS URLs (Table 1). These are critical for web archives to retrieve all WCMS published content. Hyperlinks are created not only by users but also by WCMS subsystems. In any case, some WCMS check if they are valid whereas others don't. In addition, some WCMS may be incurred with invalid hyperlinks due to bugs. The results show that not all WCMSs have the same frequency of invalid hyperlinks. Joomla and Typo3 have a high percentage (88% and 89%), whereas Blogger, Mediawiki, Drupal and MovableType have the highest percentage of invalid hyperlinks (97% and 96%).

A_2: The number of inline JavaScript scripts per WCMS instance (Table 2). The excessive use of inline scripts in modern web development results in web archiving problems. Plone, MovableType and Typo3 have the lowest number of inline scripts per instance (4.82, 6.82 and 6.89). The highest usage by far comes from Blogger (27.11), while Drupal (15.09) and vBulletin (12.38) follow.

WCMS	Inst.	Inline scripts	scripts/inst.
Plone	431	2076	4.82
MovableType	295	2011	6.82
Typo3	624	4298	6.89
Mediawiki	408	3753	9.20
DataLife Engine	321	3159	9.84
Wordpress	863	8646	10.02
DotNetNuke	598	6028	10.08
Joomla	501	5163	10.31
PrestaShop	466	5130	11.01
vBulletin	462	5721	12.38
Drupal	528	7969	15.09
Blogger	324	8783	27.11

Table 2. A_2 The number of inline scripts per WCMS instance. Lower is better.

WCMS	Instances	Issues	Correct
DataLife Engine	321	46	86%
Wordpress	863	272	68%
Drupal	528	189	64%
PrestaShop	466	237	49%
MovableType	295	152	48%
Typo3	624	322	48%
Plone	431	249	42%
vBulletin	462	329	29%
Joomla	501	359	28%
Blogger	324	240	26%
DotNetNuke	598	461	23%
Mediawiki	408	335	18%

Table 3. A_3 Sitemap.xml is present. Higher is better.

The sitemap.xml[10] protocol is meant to create files which include references to all the webpages of the website. Sitemap.xml files are generated automatically by WCMS when their content is updated. The results of the A_3 evalution

[10] `http://www.sitemap.org/`

(Table 3) indicate that most WCMS lack proper support for this feature. Only DataLife Engine has a very high score (86%). Also Wordpress and Drupal score over 60%. All other WCMSs perform very poorly, which is surprising.

***F_C*: Cohesion** is relevant to the level of dispersion of files comprising a single website to multiple servers in different domains. The lower the dispersion of a website's files, the lower the susceptibility to errors because of a failed third-party system. We evaluate the performance for two F_C related evaluations.

C_1: The percentage of local versus remote images is presented in Table 4). Blogger is suffering from the highest dispersion of images. On the contrary, Plone, DotNetNuke, PrestaShop, Typo3 and Joomla have the higher F_C, over 90%.

C_2: The percentage of local versus remote CSS (Table 5). Again, Blogger has a very low score (2%), whereas every one WCMS is performing very well.

WCMS	Local imgs	Remote imgs	Percent.
Plone	7833	290	96%
DotNetNuke	13136	680	95%
PrestaShop	19910	1187	94%
Typo3	15434	897	94%
Joomla	14684	1251	92%
MovableType	8147	1388	86%
Drupal	16636	3169	84%
vBulletin	11319	2314	83%
Wordpress	20350	4236	83%
Mediawiki	4935	1127	81%
DataLife Engine	9638	2356	80%
Blogger	1498	8121	16%

Table 4. C_1 The percentage of local versus remote image. Higher is better.

***F_S*: Standards Compliance** is a necessary precondition in digital curation practices [5]. We evaluate S_1: Validate if the HTML source code complies with the W3C standards using the W3C HTML validator and present the results in Table 6.

Plone has the lower number of errors (28.32), followed by Mediawiki (34.39) and Typo3 (34.41). On the contrary, Blogger has the most errors per instance (220.01), followed by far by DataLife Engine (108.31) and MovableType(101.67).

S_3: The usage of Quicktime and Flash formats is considered problematic for web archiving because web crawlers cannot process their contents to extract information, including web resource references. Results show that their use is very low in all WCMS (Table 7).

S_4: Check if the RSS feed format complies with W3C standards. The results (Table 8) indicate that Blogger has mostly correct feeds (91%), whereas every

WCMS	Local CSS	Remote CSS	Percent.
DotNetNuke	5243	101	98%
Typo3	3365	154	96%
Plone	1475	72	95%
Joomla	4539	222	95%
DataLife Engine	919	56	94%
PrestaShop	5221	400	93%
MovableType	578	42	93%
vBulletin	1459	104	93%
Mediawiki	1120	84	93%
Drupal	2320	354	87%
Wordpress	5658	1019	85%
Blogger	18	954	2%

Table 5. C_1 The percentage of local versus remote CSS. Higher is better.

WCMS	Instances	Errors	Errors/Instance
Plone	431	12205	28.32
Mediawiki	408	14032	34.39
Typo3	624	23965	38.41
Wordpress	863	35805	41.49
Joomla	501	26609	53.11
PrestaShop	466	30066	64.52
DotNetNuke	598	43009	71.92
Drupal	528	47131	89.26
vBulletin	462	46466	100.58
MovableType	295	29994	101.67
DataLife Engine	321	34768	108.31
Blogger	324	71283	220.01

Table 6. S_1 HTML errors per instance. Lower is better.

other WCMS has various levels of correctness. The lowest scores belong to Mediawiki (2%) and DotNetNuke (13%). In general, the results show that there is a problem with RSS feed standard compliance.

F_M: Metadata usage : The lack of metadata impairs the archive's ability to manage content effectively. Web sites include a lot of metadata, which need to be communicated in a correct manner to be utilised by web archives [6].

M_1: Check if the HTTP Content-type header exists (Table 9). There is virtually no issue with HTTP Content-Type in all WCMSs. Their performance is excellent.

M_2: Check if any HTTP Caching headers (Expires, Last-modified or ETag) are set. HTTP Caching is highly relevant to accessibility and performance. Blogger, Mediawiki, Drupal, DataLife Engine and Plone have very good support of HTTP Caching headers (Table 10).

WCMS	Instances	No propr. files	Success
PrestaShop	466	460	99%
Mediawiki	408	398	98%
Blogger	324	310	96%
Plone	431	412	96%
Wordpress	863	821	95%
Typo3	624	592	95%
vBulletin	462	434	94%
Drupal	528	494	94%
DotNetNuke	598	548	92%
DataLife Engine	321	294	92%
MovableType	295	263	89%
Joomla	501	439	88%

Table 7. S_2 The lack of use of proprietary files (Flash, QuickTime). Higher is better.

WCMS	valid feeds	invalid feeds	Correct
Blogger	872	83	91%
DataLife Engine	240	57	81%
Wordpress	1283	317	80%
Joomla	556	141	80%
vBulletin	299	96	76%
MovableType	271	120	69%
Drupal	133	74	64%
PrestaShop	82	112	42%
Typo3	124	191	39%
Plone	116	184	39%
DotNetNuke	2	14	13%
Mediawiki	10	521	2%

Table 8. A_5: Valid Feeds. Higher is better.

WCMS	Instances	Exists	Success
Blogger	324	324	100%
Drupal	528	527	100%
MovableType	295	294	100%
vBulletin	462	458	99%
Plone	431	427	99%
Typo3	624	618	99%
Joomla	501	494	99%
DotNetNuke	598	589	98%
Mediawiki	408	401	98%
DataLife Engine	321	315	98%
PrestaShop	466	456	98%
Wordpress	863	841	97%

Table 9. M_1: HTTP Content-Type header. Higher is better.

WCMS	Instances	Issues	Percentage
Blogger	324	3	99%
Mediawiki	408	12	97%
Drupal	528	23	96%
DataLife Engine	321	16	95%
Plone	431	49	89%
MovableType	295	106	64%
Joomla	501	186	63%
Wordpress	863	466	46%
Typo3	624	364	42%
vBulletin	462	326	29%
PrestaShop	466	388	17%
DotNetNuke	598	569	5%

Table 10. M_2: HTTP caching headers. Higher is better.

5 Discussion and conclusions

We evaluated the WA and presented specific statistics regarding 12 prominent WCMS. We concluded that not all WCMSs are considered equally archivable. Each one has its own strengths and weaknesses, which we highlight in the following:

1. *Blogger* has by far the worst overall WA score (65.91%, Figure 2), mainly due to the very low F_C. Blogger files are dispersed in multiple different web services, which is increasing the possibility of errors in case one of them fails. In addition, Blogger scores very low in many metrics such as the number of inline scripts per instance (Table 2) and HTML errors per instance (Table 6). On the contrary, Blogger scores very high regarding F_M and F_S.
2. *DataLife Engine* has the highest WA score (83.52%). One aspect that they should look into is HTML errors per instance (Table 6), where it has the second worst score.
3. *DotNetNuke* has the second worst WA score in our evaluation (77.2%). F_C is their strong point (96.01%) but they have issues is every other area. We suggest that they look into their RSS feeds (13% Correct) (Table 8), and lacking HTTP caching support (5%) (Table 10).
4. *Drupal* has the third highest WA score (82.08%). It has good overall performance and the only issue is the existence of too many inline scripts per instance (15.09) (Table 2).
5. *Joomla* WA score is average (80.37%). It has a large number of invalid URLs per instance (12%) (Table 1) and it has also the highest usage of proprietary files (12%) (Table 7) which is not good for accessibility and preservation.
6. *Mediawiki* WA score is low (77.81%). This can be attributed to mostly invalid feeds (only 2% are correct according to standards) and very low sitemap.xml support (18%), Table 3.
7. *MovableType* WA score is average (80.02%). It does not stand out in any evaluation either in a positive or a negative way. General improvement in all areas would be welcome.

8. *Plone* has the second highest WA score (83.06%). It must be commented for having the lowest number of HTML errors per instance (28.32) (Table 6) and very high F_C scores (96% for images, Table 4 and 95% for CSS, Table 5).
9. *PrestaShop* WA score is average (79%). It has average scores in all evaluations. but it should be commented for not using any proprietary files (top score: 99% at Table 7).
10. *Typo3* WA score is average (79%). It has the largest number of invalid URLs per instance (12%) (Table 1).
11. *vBulletin* WA score is consistenly low (78.37%). General improvement in all areas would be welcome.
12. *Wordpress* WA score is average (78.47%). We cannot highlight a specific area where it should be improved. As this is currently the most popular WCMS, Wordpress developers should look into all WA Facets and try to improve.

We recommend that the WCMS development communities investigate the presented issues and resolve them as many are easy to be fixed without causing any issues with existing users and installations. If the situation regarding the highlighted issues is improved in the next releases of the investigated WCMS, the impact would be significant. A large number of websites which could not be archived correctly would no longer have these issues once they update their software and newly created websites based on these WCMS would be more archivable. Web archiving operations around the world would see great improvement, resulting in general advancements in the state of web archiving.

References

1. V. Banos, Y. Kim, S. Ross, and Y. Manolopoulos. Clear: a credible method to evaluate website archivability. In *Proceedings 10th International Conference on Preservation of Digital Objects (iPRES)*, 2013.
2. V. Banos and Y. Manolopoulos. A quantitative approach to evaluate website archivability using the clear+ method. *International Journal on Digital Libraries*, pages 1–23, 2014.
3. O. Blanvillain, N. Kasioumis, and V. Banos. Blogforever crawler: Techniques and algorithms to harvest modern weblogs. In *Proceedings 4th International Conference on Web Intelligence, Mining & Semantics (WIMS)*, page 7, 2014.
4. B. Boiko. Understanding content management. *Bulletin of the American Society for Information Science & Technology*, 28(1):8–13, 2001.
5. D. P. Coalition. Institutional strategies - standards and best practice guidelines. `http://www.dpconline.org/advice/preservationhandbook/institutional-strategies/standards-and-best-practice-guidelines`, 2012. [Online; accessed 10-November-2014].
6. M. Day. Metadata, curation reference manual. `http://www.dcc.ac.uk/resources/curation-reference-\manual/completed-chapters/metadata`, 2005. [Online; accessed 10-November-2014].
7. M. Donnelly. JSTOR/Harvard Object Validation Environment (JHOVE). *Digital Curation Centre Case Studies and Interviews*, 2006.

8. M. Faheem and P. Senellart. Intelligent and adaptive crawling of web applications for web archiving. In *Proceedings 13th International Conference on Web Engineering (ICWE)*, pages 306–322, 2013.
9. N. Fernández-Garcia, L. Sánchez-Fernandez, and J. Villamor-Lugo. Next generation web technologies in content management. In *Proceedings (companion) 13th International Conference on World Wide Web (WWW)*, pages 260–261, 2004.
10. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol–http/1.1. `http://tools.ietf.org/html/rfc2616`, 1999. [Online; accessed 10-November-2014].
11. D. Gomes, M. Costa, D. Cruz, J. Miranda, and S. Fontes. Creating a billion-scale searchable web archive. In *Proceedings (companion) 22nd International Conference on World Wide Web (WWW)*, pages 1059–1066, 2013.
12. N. Kasioumis, V. Banos, and H. Kalb. Towards building a blog preservation platform. *World Wide Web*, 17(4):799–825, 2014.
13. B. Kelly and M. Guy. Approaches to archiving professional blogs hosted in the cloud. In *Proceedings 7th International Conference on Preservation of Digital Objects (iPRES)*, 2010.
14. S. Lawrence, D. M. Pennock, G. W. Flake, R. Krovetz, F. M. Coetzee, E. Glover, F. Å. Nielsen, A. Kruger, and C. L. Giles. Persistence of web references in scientific research. *IEEE Computer*, 34(2):26–31, 2001.
15. S. McKeever. Understanding web content management systems: evolution, lifecycle and market. *Industrial Management & Data Systems*, 103(9):686–692, 2003.
16. J. Niu. An overview of web archiving. *D-Lib Magazine*, 18(3/4), 2012.
17. M. Pennock and R. Davis. Archivepress: A really simple solution to archiving blog content. In *Proceedings 6th International Conference on Preservation of Digital Objects (iPRES)*, 2009.
18. E. Pinsent, R. Davis, K. Ashley, B. Kelly, M. Guy, and J. Hatcher. Powr: The preservation of web resources handbook, 2010.
19. M. Rumianek. Archiving and recovering database-driven websites. *D-Lib Magazine*, 19(1/2), 2013.
20. W3Techs. Usage of content management systems for websites. `http://w3techs.com/technologies/overview/content_management/all`, 2014. [Online; accessed 10-November-2014].