Robustifying Links To Combat Reference Rot

Jones, Shawn Morgan
Klein, Martin
Van De Sompel, Herbert

Robustifying Links To Combat Reference Rot

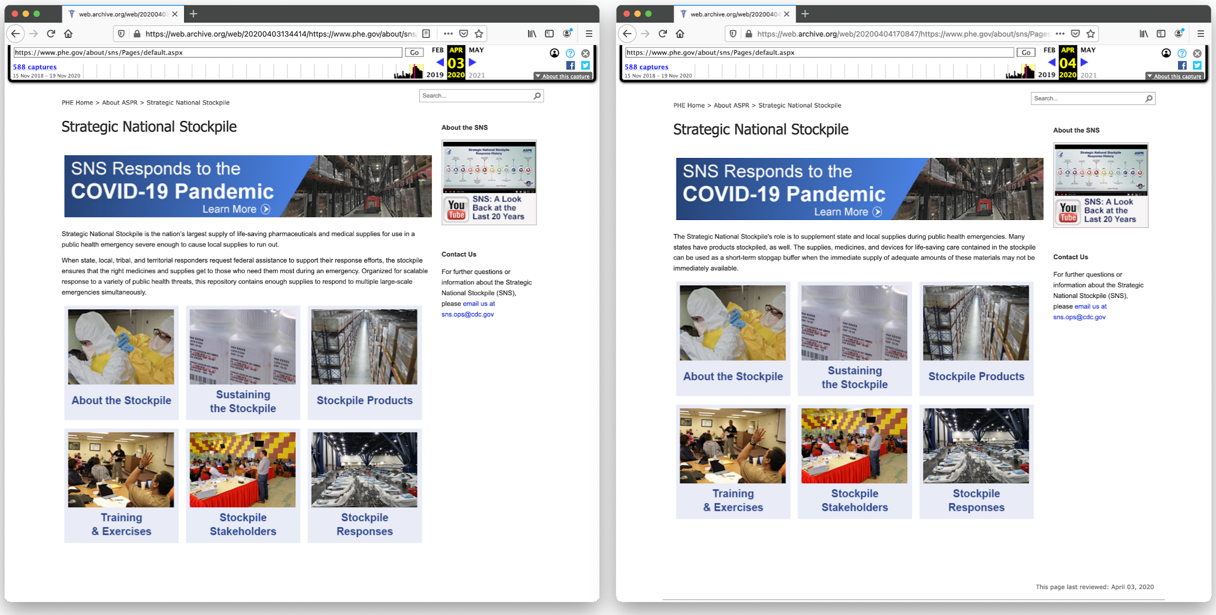
Shawn M. Jones (0000-0002-4372-870X), Los Alamos National Laboratory
Martin Klein (0000-0003-0130-2097), Los Alamos National Laboratory
Herbert Van de Sompel (0000-0002-0715-6126), Data Archiving and Networked Services

Abstract

Links to web resources frequently break, and linked content can change at unpredictable rates. These dynamics of the Web are detrimental when references to web resources provide evidence or supporting information. In this paper, we highlight the significance of reference rot, provide an overview of existing techniques and their characteristics to address it, and introduce our Robust Links approach, including its web service and underlying API. Robustifying links offers a proactive, uniform, and machine-actionable way to combat reference rot. In addition, we discuss our reasoning and approach aimed at keeping the approach functional for the long term. To showcase our approach, we have robustified all links in this article.

Information Integrity Is Impeded by Reference Rot

Two characteristics of the dynamic nature of the Web are **link rot** and **content drift**. Link rot occurs when links break over time -- a detriment we have all encountered numerous times, and that has been studied and quantified abundantly. Content drift happens when content at a given web location changes over time, something we expect when gathering information on the current state of affairs -- e.g., weather, news, stock prices. Because we often assume that the latest web content is the most desirable, the effects of content drift are not immediately apparent. In past work, we quantified the extent of content drift for links to web pages referenced in scholarly papers. We argued that it significantly undermines the integrity of the web-based scholarly record because it prevents readers from revisiting the exact web content that authors referenced. But the detrimental effect of content drift is not limited to scholarship on the Web.



https://web.archive.org/web/20200403134414/https://www.phe.gov/about/sns/

https://www.phe.gov/about/sns/Pages/default.aspx

588 captures
15 Nov 2018 – 19 Nov 2020

FEB APR MAY
03
2019 2020 2021

PHE Home > About ASPR > Strategic National Stockpile

Strategic National Stockpile

SNS Responds to the COVID-19 Pandemic
Learn More

Strategic National Stockpile is the nation's largest supply of life-saving pharmaceuticals and medical supplies for use in a public health emergency severe enough to cause local supplies to run out.

When state, local, tribal, and territorial responders request federal assistance to support their response efforts, the stockpile ensures that the right medicines and supplies get to those who need them most during an emergency. Organized for scalable response to a variety of public health threats, this repository contains enough supplies to respond to multiple large-scale emergencies simultaneously.

About the SNS

SNS: A Look Back at the Last 20 Years

Contact Us

For further questions or information about the Strategic National Stockpile (SNS), please email us at sns.ops@cdc.gov

About the Stockpile | Sustaining the Stockpile | Stockpile Products | Training & Exercises | Stockpile Stakeholders | Stockpile Responses

https://web.archive.org/web/20200404170847/https://www.phe.gov/about/sns/Pages

https://www.phe.gov/about/sns/Pages/default.aspx

588 captures
15 Nov 2018 – 19 Nov 2020

FEB APR MAY
04
2019 2020 2021

PHE Home > About ASPR > Strategic National Stockpile

Strategic National Stockpile

SNS Responds to the COVID-19 Pandemic
Learn More

The Strategic National Stockpile's role is to supplement state and local supplies during public health emergencies. Many states have products stockpiled, as well. The supplies, medicines, and devices for life-saving care contained in the stockpile can be used as a short-term stopgap buffer when the immediate supply of adequate amounts of these materials may not be immediately available.

About the SNS

SNS: A Look Back at the Last 20 Years

Contact Us

For further questions or information about the Strategic National Stockpile (SNS), please email us at sns.ops@cdc.gov

About the Stockpile | Sustaining the Stockpile | Stockpile Products | Training & Exercises | Stockpile Stakeholders | Stockpile Responses

This page last reviewed: April 03, 2020

Figure 1: The Internet Archive holds archived versions (mementos) of the U.S. National Stockpile website from April 3, 2020 (left) and April 4, 2020 (right). Significant changes in the textual description of the stockpile can be seen when comparing both mementos.

Content drift can mask changes in government policy. For example, the United States government has maintained a national stockpile of medical equipment since the 1950s and consolidated it at the Strategic National Stockpile in 2003. As shown in the left archived web page ("**memento**", defined in the next section) of Figure 1, the Stockpile existed to support “state, local, tribal, and territorial responders” in the case of an emergency. While managing the COVID-19 pandemic, senior White House official Jared Kushner directed that the stockpile was not for use by these municipalities, in contrast to the policy stated on its website. The right memento in Figure 1 reflects this change in policy one day after Kushner’s statement. Journalists from Ars Technica, Politico, The Hill, The Washington Post, and others used mementos, like those in Figure 1, to prove that the agency policy existed on the website up until Kushner’s comments. If journalists had only been able to rely upon the most recent version of this resource, content drift would have prevented them from proving that the agency changed its policy after Kushner’s comments.

Figure 2 demonstrates this problem generically. At time t_0 an author wrote Article A, which we formally refer to as a **linking page**. The article links to eight web pages, referred to as **linked resources**. At a later time t_m, a reader visits the linking page and finds link rot for two and content drift for four of the linked resources. Only two linked resources are the same as when the author originally linked to them. Thus, a reader visiting the linked resources at time t_m will not see the same content to which the author linked at time t_0, and, in the case of content drift, may not even be aware of it.

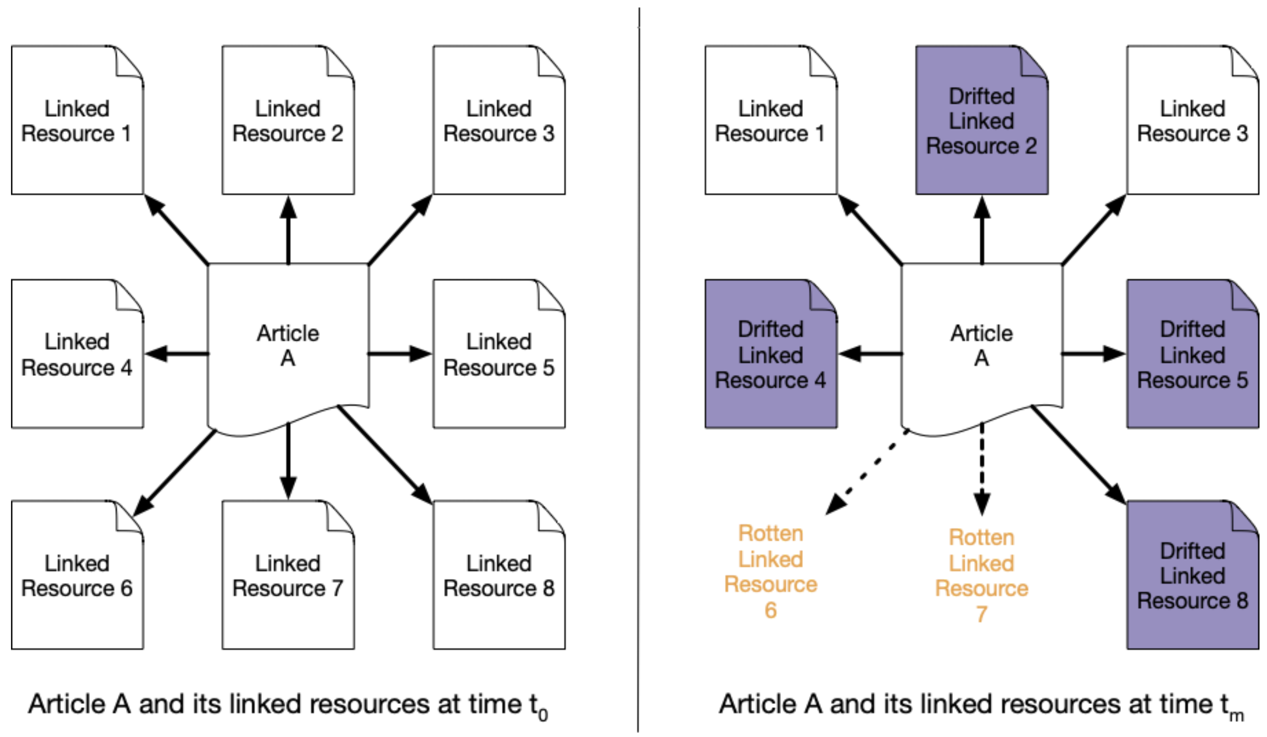


Figure 2: An abstract example of reference rot over time. The linking page (Article A) contains eight linked resources. At time t_0, all linked resources match what the author viewed. At time t_m, two links are rotten, four linked resources have content drift, and only two links remain that accurately reflect what the author saw at t_0.

All linked resources are subject to link rot and content drift, a Web phenomenon which we collectively refer to as **reference rot**. Investigative pursuits in journalism, as illustrated above, the integrity of the scholarly record, and evidence provided to support United States Supreme Court opinions, are only some of the examples that highlight the essential need to be able to revisit referenced content as it was at the time the authors of linking pages created links. We are therefore motivated to address the reference rot problem and propose Robust Links as a solution that we devised with an eye on long term functionality in an ever-changing Web environment.

Other Approaches to Address Reference Rot

Reference rot, and especially the link rot component of the problem, has attracted significant attention. As such, it is not surprising that a variety of approaches exist that attempt to address it. Here we cover persistent identifiers, approaches to cite web resources, and techniques that leverage web archives. We show that each has shortcomings when it comes to combating reference rot.

To discuss these approaches, we borrow terms from the Memento Protocol. An **original resource** is the current, most recent version of a web resource. An original resource is identified by a **URI-R**. **Mementos** are captures of original resources at specific points in time. A memento of an original resource is identified by a **URI-M** and has a specific **archival datetime**.

Persistent Identifiers

At first thought, we might conclude that persistent identifiers such as Digital Object Identifiers (DOIs), PURLs, and W3IDs are viable solutions to the reference rot problem. However, a quick analysis of the way they function reveals they can successfully be used in certain settings but not generally. All types of persistent identifiers use a similar approach to avoid link rot for links to a web resource. They consider the actual URL of the resource to be temporary, subject to change. They introduce a permanent URL for the resource that authors should use for linking purposes. The persistent identifier registrar manages a central lookup table to express the correspondences between the permanent URLs and the respective actual, current URLs. When a reader follows the link to the permanent identifier of a web resource, the registrar applies the lookup table to generate an HTTP redirect from that permanent URL to the current URL of the resource. As time goes by and the resource changes web location, its custodian must update the lookup table and associate the permanent URL with the new web location. This approach works successfully when the custodian fundamentally cares about the long term accessibility of its web resources and hence is willing to invest time and energy to keep the lookup table up-to-date. While this may be the case for scholarly publishers, cultural heritage institutions, and authors of web ontologies, for example, we cannot expect such commitment from the middle-of-the-road webmaster. The basic premise of the value proposition of persistent identifiers, namely a custodian's commitment regarding long term access to its web resources, does not hold for the majority of web resources. For those resources, the author of a link rather than the linked resource's custodian has to invest in making the link durable.

Citing Web Resources

Approaches for citing web resources have evolved. Initially, web resources were infrequently cited, leading citation standard bodies to ignore them. As web resources became more commonly used as references, these same bodies recognized the need to update their standards to include web resources and also to support addressing the reference rot problem. Figure 3 shows an example reference to a web page in the APA format. It includes a URI-R so that the reader can visit the current version of the cited resource, assuming it still exists. The reference format also specifies a “retrieved” date, an explicit recognition of the existence of reference rot. For consistency between reference standards, we refer to this date as the **author view date**. The reader can use this date along with the URI-R to manually locate a memento of the linked resource in a web archive. However, there are no guarantees that such a memento exists or that it was archived on or close to the date specified in the reference. As such, the reference leaves it to the reader to manually solve link rot and provides no solution to content drift.

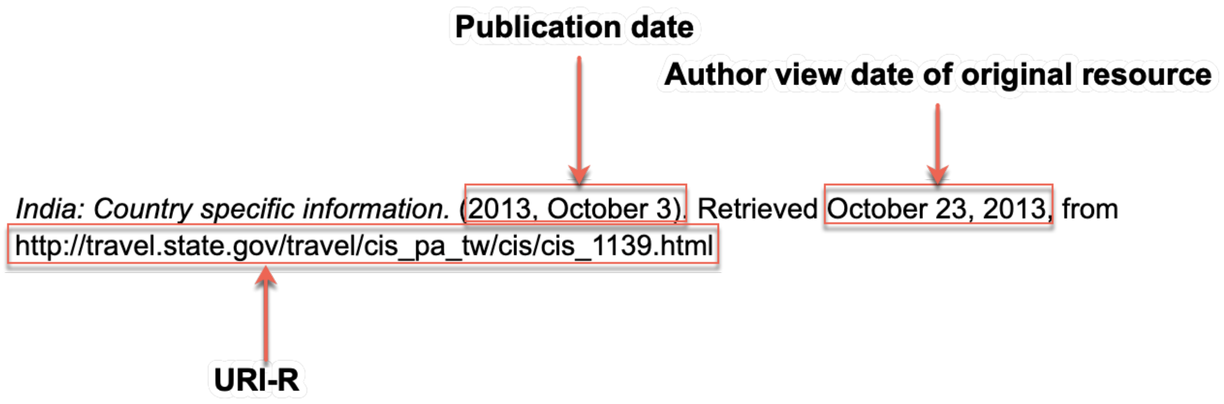

India: Country specific information. (2013, October 3). Retrieved October 23, 2013, from http://travel.state.gov/travel/cis_pa_tw/cis/cis_1139.html

Figure 3: An annotated example reference to a web page in APA format.

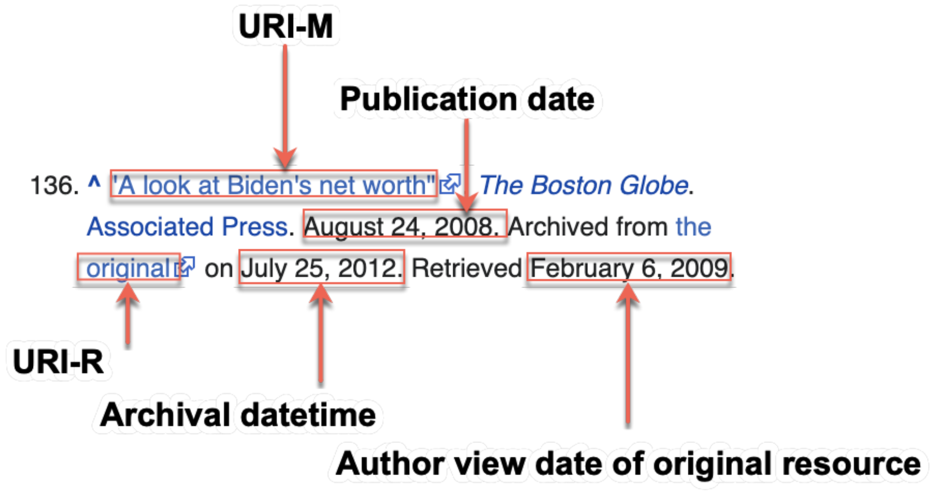
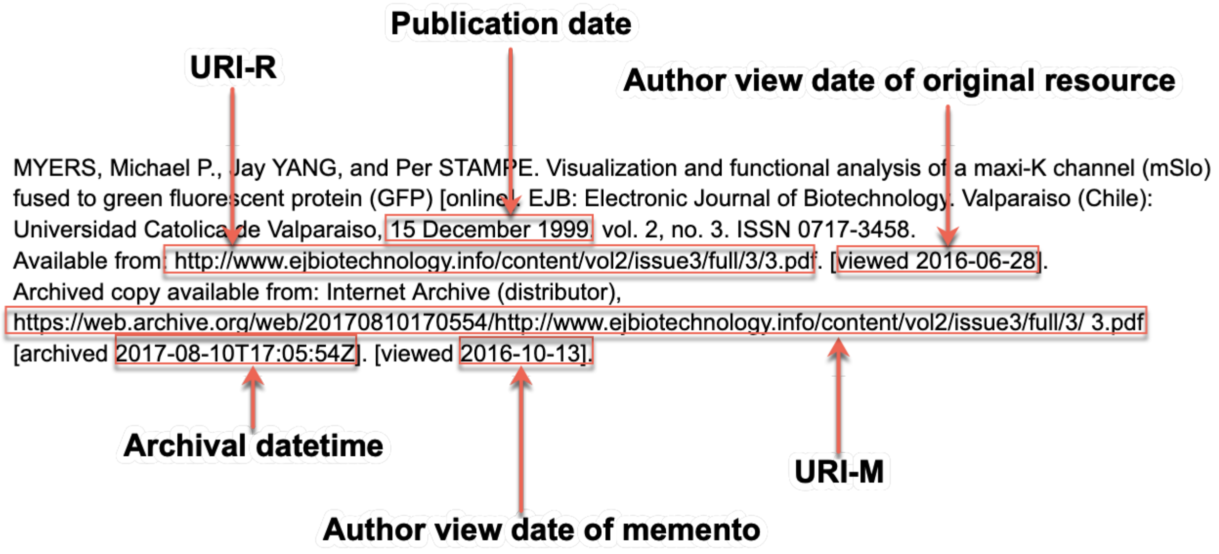

136. ^ "A look at Biden's net worth". *The Boston Globe*. Associated Press. August 24, 2008. Archived from the original on July 25, 2012. Retrieved February 6, 2009.

Figure 4: An annotated example reference from Wikipedia.


MYERS, Michael P., Jay YANG, and Per STAMPE. Visualization and functional analysis of a maxi-K channel (mSlo) fused to green fluorescent protein (GFP) [online]. EJB: Electronic Journal of Biotechnology. Valparaiso (Chile): Universidad Catolica de Valparaiso, 15 December 1999. vol. 2, no. 3. ISSN 0717-3458.
Available from: http://www.ejbiotechnology.info/content/vol2/issue3/full/3/3.pdf. [viewed 2016-06-28].
Archived copy available from: Internet Archive (distributor),
https://web.archive.org/web/20170810170554/http://www.ejbiotechnology.info/content/vol2/issue3/full/3/ 3.pdf
[archived 2017-08-10T17:05:54Z]. [viewed 2016-10-13].

Publication date
URI-R
Author view date of original resource
Archival datetime
URI-M
Author view date of memento

Figure 5: An annotated example reference to a web page in ISO/DIS 690 format

Where possible, Wikipedia article citations to linked resources contain information that might be used to combat reference rot. They contain a link to the original resource, the date when the linked resource was retrieved by the author of the Wikipedia article (the linking page), a link to a memento of the linked resource, and the archival datetime of that memento as shown in Figure 4. With the URI-M, the reader avoids link rot. The archival datetime of the URI-M is not always near the author view date, meaning that this approach does not reliably address content drift. While it is possible for an author to create their own memento and manually insert its URL into one of these citations, many of these URI-Ms are inserted by scripts that are run after the Wikipedia article is updated. The URI-Ms added by these scripts often reflect a best effort to find a memento near the author view date.

This approach used by Wikipedia and others has found its way into the recommendations of the recent ISO/DIS 690 reference standard, as shown in an example in Figure 5. It includes the URI-R and a URI-M for the resource. The reader can use the URI-M to visit a specific version of the linked resource. Because the author explicitly endorses this URI-M as a reference, this reference format is, in principle, able to address both link rot and content drift. If the URI-M is inaccessible due to technical problems or other issues, the reader could still use the author view date and the URI-R to manually find another memento that might match the version captured on that date. The standard mentions, however, that the date specified by the “viewed” keyword does not always correspond to the author view date of the original resource. The example in Figure 5 shows two dates with the “viewed” keyword. The first one, following the URI-R, is the author view date of the original resource. A reader can use this date in combination with the URI-R to locate a memento in a web archive. The second “viewed” date, following the URI-M, refers to the author view date of the memento. Thus, the textual position of dates in ISO/DIS 690 citations must be carefully considered when consulting referenced web resources.

Authors can adapt these citation standards into any human-readable file format, like PDF, HTML, or DOCX. Robust Links leverage HTML and thus are confined to that format.

Techniques that Leverage Web Archives

A variety of infrastructural approaches to address the reference rot problem have emerged in which web archives play an essential role because they hold prior, timestamped versions of web pages.

Criteria for Characterizing the Techniques

All techniques to combat reference rot that leverage web archives rely on the ability to deliver a memento for a linked resource as a means to counteract the fact that, some time after the link was established, the resource may have vanished from the live web or its content may have drifted. As such, these techniques rely on the existence and hence creation of mementos of linked resources. A number of factors contribute to the extent to which a given memento that is delivered addresses reference rot in an accurate and robust manner. We first discuss these factors and then describe and characterize the various techniques accordingly.

Temporal accuracy of a memento - Consider a linking page with a set of linked resources. To address content drift, the archival datetime of a memento for a linked resource must be as close as possible to the date the author linked to it. If the linking page's author — or someone on their behalf — creates a memento for each linked resource around the time the author views the linked resource's content, then both link rot and content drift are addressed. Because each such memento matches the specific version of a linked resource the way the author saw it when linking, a reader can review the linked resources in the state intended by the author of the linking page. If a third party, unconnected to the development of the linking page, has created a memento for one or more of the linked resources, then this third party has incidentally addressed link rot for that specific link in the author's linking page. However, because this third party may not have created this memento at the time the author viewed the linked resource, content drift is not necessarily addressed.

Distribution of mementos - Redundancy of mementos is also important for addressing reference rot in a robust manner. Consider an approach to reference rot that creates mementos but only places them in one archive. If we link to one of these mementos, but the web archive in which it resides is - temporarily or permanently - not in service, we again have link rot. Thus, an approach that stores mementos of referenced content in multiple archives is more robust. Similarly, an approach that only retrieves mementos from one web archive is equally vulnerable and its robustness is much increased by providing access to mementos across multiple archives.

Unconstrained memento access - Consider the approach that addresses reference rot, but only for some readers or some linked resources. Such approaches apply constraints on access to mementos. Some constraints may be imposed by the tools themselves. Specific browsers or browser extensions may only address reference rot for the users that install them. A server running a special extension may only create mementos for the linked resources found in the linking pages it hosts. Other constraints regard the original resources for which mementos are accessible. For example, an approach may only provide mementos of linked resources from a specific domain, like gov.uk, or it may only address reference rot for the linked resources that are frequently navigated by the visitors of certain Cloudflare customer sites. Approaches that provide unconstrained memento access are more robust.

Datetime for memento access - Some approaches take a desired datetime into account when delivering a memento to the reader. This datetime could be the publication date of the linking page, a manually selected date, or information from the link itself such as the dates included with various citation standards shown in the previous section. The inclusion of a datetime can help a reader locate a memento near this date and thus can address content drift. Ideally, the date information should be machine-readable so that web crawlers, on-page JavaScript, and other automation can easily parse information and help readers find mementos.

Characterization of Techniques that Leverage Web Archives

Perhaps the most straightforward approach to combating reference rot is to use the URI-M as the link target when creating new links. This approach supports unconstrained memento access because it works in all browsers, is not limited to specific domains, and can be used with any web publishing platform. If the author relies upon a memento created by others, for example via web crawling by web archives, then this solution at best addresses link rot. If the author proactively creates the memento, then this approach addresses link rot and content drift. Unfortunately its effectiveness is dependent on the continued existence of a single web archive. If the web archive storing the URI-M fails for technical, financial, legal, or other reasons, then we again have link rot because the memento becomes unavailable. If the web archive is not available, we could search for the URI-R in other web archives, but how do we know what the original URI was? URI-Ms do not always convey the URI-R. For example, the URI-R https://lanl.gov is apparent at the end of the Internet Archive URI-M https://web.archive.org/web/20201027204016/lanl.gov but can not be obtained from the Perma.cc URI-M https://perma.cc/Q938-LPMG. Thus, if the URI-M fails, the reader has no way of reconstructing the reference.

The recent partnership between the Internet Archive and Cloudflare goes a step further. Every few days, Cloudflare shares its customers' popular URL paths with the Internet Archive, which then creates their mementos. If a linked resource served by Cloudflare is unavailable because its origin server is down, Cloudflare detects this failure and redirects readers to the most recent memento of the page. This solution works only for Cloudflare-supported websites and only for those customers who opt-in to this service. Similarly, the UK Government's Web Continuity Project proactively preserves UK Government web sites' pages and redirects from broken links in government web sites to these UK Government URI-Ms. While these solutions address link rot, none address content drift because they lack coordination with a linking page's author. Because they only work with specific linked resource domains, these solutions also only apply to a subset of the Web.

The Brave browser, the Wayback Machine browser extensions for multiple browsers, and the "No More 404s" browser extension all address link rot. When a browser user clicks on a broken link, each of these approaches aims to detect the broken link and seeks a replacement URI-M in the Internet Archive, ignoring mementos in other web archives. They do not provide unconstrained memento access for readers because they limit readers to only using specific browsers or extensions. The approaches do not address content drift because they rely upon the incidental preservation of linked resources by a third party, meaning if no memento of a linked resource exists, then link rot persists. They also only search for resources in a single web archive, forgoing the possibility that the linked resource might exist in other web archives.

It is worth mentioning that none of these solutions a) know the version that the linking page's author intended to reference and b) can address content drift. In contrast, the Memento Time Travel Chrome extension does allow a user to select a datetime from a calendar picker and right-click on a broken link to find a memento of the linked resource around the specified datetime and across multiple web archives. But it is up to the reader to make an informed guess for the datetime that should be used. For example, the reader could choose the creation date of the page that contains the broken link. Unless the author deliberately created a memento around this date, this approach relies upon incidental archiving and thus may not address content drift.

Harvard University's Berkman Center maintains the Amber project, which provides plugins for the content management systems Drupal and WordPress that create mementos of linked resources for linking page authors. Amber stores these mementos in multiple web archives (e.g., perma.cc and the Internet Archive) and caches them on the author's server. If a reader clicks on a linked resource and Amber detects that the link is broken, it offers to let the reader view the specific memento of the linked resource associated with the linking page. Thus, Amber satisfies multiple robustness criteria. It creates mementos near the time of linking, addressing content drift. Amber allows readers to reach these specific mementos if the link is broken, addressing link rot. However, Amber only works for Drupal and WordPress, limiting authors to those tools. It also does not provide machine-readable links, making it difficult to apply other solutions if Amber fails or its mementos no longer exist.

As mentioned in the previous section, English-language Wikipedia often links to both the URI-R and a URI-M. Additionally, it provides the date of the archiving, as seen in Figure 4. This approach focuses on combating link rot. It only addresses content drift for those linked resources for which the author explicitly created a memento. This approach does work in all modern browsers and allows linked resources from any domain, but it only addresses link rot for Wikipedia and potentially some MediaWiki installations, limiting authors' options. Even though an author can apply mementos from other web archives to their references, the Wikipedia scripts that automatically apply this solution to linked resources were built based on a specific partnership with the Internet Archive. Much like the reference standards mentioned above, a reader can use the URI-R and the archive date to find a suitable replacement URI-M in another web archive if the given URI-M is not available. Humans can read these references and understand how the pieces fit together, but machines cannot. Any client performing analysis on these linked resources would have to resort to heuristics to distinguish URI-Ms from URI-Rs. If a URI-M fails, a machine client would have no method of detecting which date on the linking page refers to which linked resource and therefore cannot collect the necessary information to find an alternative URI-M for the linked resource.

Table 1: Different web archive-based reference rot solutions and their capabilities.

	memento creation		memento access						reference rot	
	archival date is the same as the linking date	**memento stored in multiple web archives**	**linked resource from any domain**	**works in all modern browsers**	**approach applicable to any publication platform**	**searches for mementos in multiple archives**	**applies date information from linking page**	**machine -readable links**	**addresses link rot**	**addresses content drift**
Brave Browser	no	no	yes	no	yes	no	no	no	yes	no
Wayback Machine browser extensions	no	no	yes	no	yes	no	no	no	yes	no
404 No More Firefox extension	no	no	yes	no	yes	no	no	no	yes	no
Internet Archive + Cloudflare	no	no	no	yes	yes	no	no	no	yes	no
UK Government Web Continuity Project	no	no	no	yes	yes	no	no	no	yes	no
Memento Time Travel browser extensions	no	no	yes	no	yes	yes	at reader's discretion	no	yes	yes
Berkman Center's Amber Project (Drupal/WordPress extension)	yes	yes	yes	yes	no	no	no	no	yes	yes
Wikipedia + Internet Archive	sometimes	sometimes	yes	yes	no	no	yes	no	yes	sometimes
Robust Links	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes

Robustifying Links

Motivated by the severity of the problem and the lack of comprehensive and unconstrained solutions, we devised the Robust Links approach. Robust Links leverage existing web archiving infrastructure and HTML5 extension attributes. Robust Links is the first approach that actually annotates links with a date and a URI-M, making it novel compared to other infrastructure approaches. The approach is devised for links in HTML but not tied to any specific authoring platform. Robust Links' reliance upon HTML prevents the approach from being used with other document formats that contain links (eg., PDF, DOCX) in which reference formats as mentioned above can be used. In addition, Robust Links are machine-readable, inviting the creation of a variety of tools that leverage them.

Creating a Robust Link requires the following steps:

1. The linking page's author, or someone working on their behalf, archives a linked resource in one or more web archives to improve the odds of survival and ensure readers can visit that specific version in the future.
2. The author enriches the HTML of the linking page with information about the linked resources' URI-R, the URI-M of the memento created in the first step, and the datetime

of archiving, thus enabling readers and machines to find the desired version in web archives.

For example, instead of this brittle HTML link:

```
<a href=”http://lanl.gov”>Los Alamos National Laboratory</a>
```

we propose to use this Robust Link:

```
<a
href=”http://lanl.gov”
data-versionurl=”https://perma.cc/Q938-LPMG”
data-versiondate="2019-10-28">Los Alamos National Laboratory</a>
```

Readers clicking on the Robust Link by default will reach the current version of http://lanl.gov. Tools that leverage this Robust Link can additionally support visiting the URI-M specified in `data-versionurl` and can combine the URI-R found in the `href` with the date from `data-versiondate` to find additional URI-Ms.

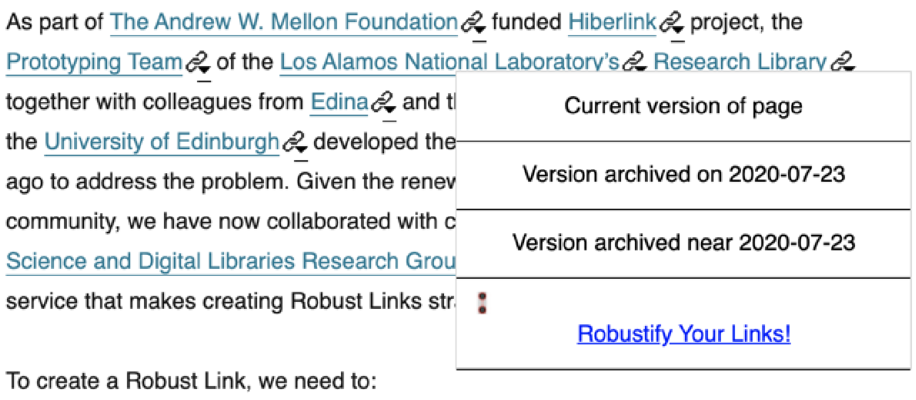
Alternatively, if the author wishes to allow readers to click on the URI-M by default, they can use this Robust Link instead:

```
<a
href=”https://perma.cc/Q938-LPMG”
data-originalurl=”http://lanl.gov”
data-versiondate="2019-10-28">Los Alamos National Laboratory</a>
```

In this case, machine clients can combine the URI-R found via the `data-originalurl` attribute and the date from the `data-versiondate` attribute to discover a replacement URI-M should the existing URI-M fail.

Actionable Robust Links

To make the machine-readable attributes provided in Robust Links actionable for human readers, we have implemented a JavaScript library (robustlinks.js) that an author can reference in their pages. The JavaScript code parses these attributes and provides readers with a drop-down menu with three choices, as shown in Figure 6.


As part of The Andrew W. Mellon Foundation funded Hiberlink project, the Prototyping Team of the Los Alamos National Laboratory's Research Library together with colleagues from Edina and t the University of Edinburgh developed the ago to address the problem. Given the renev community, we have now collaborated with c Science and Digital Libraries Research Grou service that makes creating Robust Links str

To create a Robust Link, we need to:

Current version of page
Version archived on 2020-07-23
Version archived near 2020-07-23
Robustify Your Links!

Figure 6: An example of the drop-down menu provided by robustlinks.js and robustlinks.css.

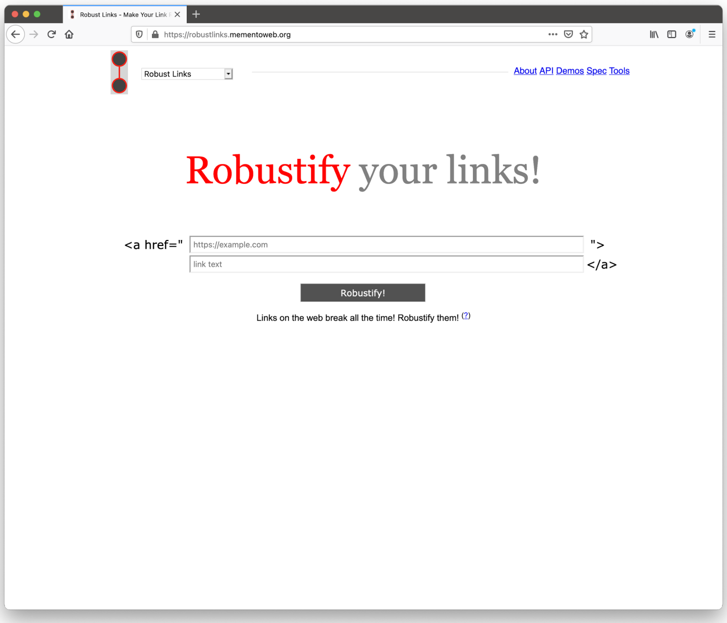
The JavaScript code builds this menu from the Robust Link's attributes and formats the menu based on a corresponding CSS file (robustlinks.css). As displayed in Figure 6, the reader has several options: they can visit the current live version of the page, which the JavaScript reads from the `href`, they can navigate to the URI-M of the "version archived on 2020-07-23", as extracted from the `data-versionurl`, or they can click "version archived near 2020-07-23" to seamlessly connect to the Memento TimeTravel service and find the memento closest to 2020-07-23 from more than 20 Memento-compliant web archives around the world. The reader does not need to install a plugin because all of this functionality is provided on the page by JavaScript linked from the HTML.

Our code is merely one example of the type of machine client that can be built upon this markup. Consider the possibility that both the URI-M and URI-R are inaccessible. A system (e.g., browser plugin, web crawler) could apply the date and the URI-R to find potential replacement linked resources from the same time period or it could analyze the spread of dates from all linked resources to find other documents on the same topic and in close temporal proximity.

When authors create Robust Links and add this JavaScript and CSS code to their articles, they allow readers to access the intended default target of their link or find a memento that matches the date the author intended. Once the author robustifies a link, they address link rot by providing a memento and combat content drift by pro-actively generating that memento, as such making sure that the memento's archival datetime matches the author's viewing date. From here, readers can obtain the content that helps them verify evidence.

The Robust Links Web Service

To assist authors in creating Robust Links, we provide a Robust Links web application. As shown in Figure 7, the minimalist entry page accepts two user inputs: the URL of the resource and the link text.

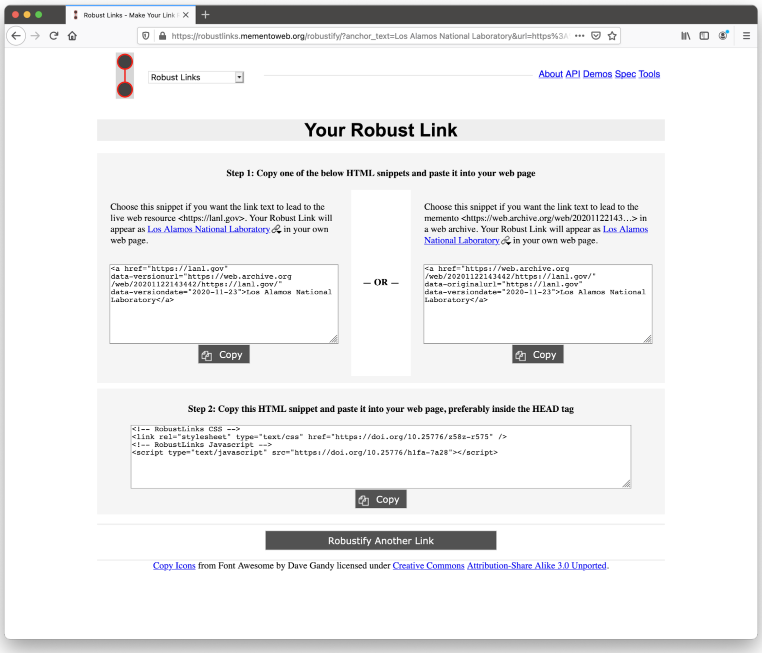

Robust Links - Make Your Link
https://robustlinks.mementoweb.org
Robust Links
About API Demos Spec Tools

Robustify your links!

```
<a href="  https://example.com  ">
           link text            </a>
```

Links on the web break all the time! Robustify them! (?)

Figure 7: The Robust Links web application entry page.


Robust Links - Make Your Link
https://robustlinks.mementoweb.org/robustify/?anchor_text=Los Alamos National Laboratory&url=https%3A
Robust Links
About API Demos Spec Tools

Your Robust Link

Step 1: Copy one of the below HTML snippets and paste it into your web page

Choose this snippet if you want the link text to lead to the live web resource <https://lanl.gov>. Your Robust Link will appear as Los Alamos National Laboratory in your own web page.

```
<a href="https://lanl.gov"
data-versionurl="https://web.archive.org
/web/20201122143442/https://lanl.gov/"
data-versiondate="2020-11-23">Los Alamos National
Laboratory</a>
```

— OR —

Choose this snippet if you want the link text to lead to the memento <https://web.archive.org/web/20201122143…> in a web archive. Your Robust Link will appear as Los Alamos National Laboratory in your own web page.

```
<a href="https://web.archive.org
/web/20201122143442/https://lanl.gov/"
data-originalurl="https://lanl.gov"
data-versiondate="2020-11-23">Los Alamos National
Laboratory</a>
```

Step 2: Copy this HTML snippet and paste it into your web page, preferably inside the HEAD tag

```
<!-- RobustLinks CSS -->
<link rel="stylesheet" type="text/css" href="https://doi.org/10.25776/z58z-r575" />
<!-- RobustLinks Javascript -->
<script type="text/javascript" src="https://doi.org/10.25776/h1fa-7a28"></script>
```

Copy Icons from Font Awesome by Dave Gandy licensed under Creative Commons Attribution-Share Alike 3.0 Unported.

Figure 8: The Robust Links web application response page.

Figure 8 shows a screenshot of the web application's response. It instructs the author to follow two steps:

1. Decide which resource they want their readers to reach with a default mouse click, the current version or the memento, and then copy the appropriate HTML.

2. From the second text area, copy the given HTML to ensure that the robustlinks.js and robustlinks.css are loaded in the linking page. This is only required once per linking page.

For users who would like to robustify many links we provide access to the Robust Links API used by our web application. With this API, a client can supply a URI-R or a URI-M and link text. The API will respond with JSON containing information about the link and how the service created it, as shown in this cURL example:

```
$ curl -G --data-urlencode "url=https://abcnews.go.com" --data-urlencode "anchor_text=ABC News for June 
15, 2020" https://robustlinks.mementoweb.org/api/
{
  "anchor_text": "ABC News for June 15, 2020",
  "api_version": "0.8.1",
  "data-originalurl": "https://abcnews.go.com",
  "data-versiondate": "2020-06-15",
  "data-versionurl": "https://archive.li/wip/hWZdd",
  "request_url": "https://abcnews.go.com",
  "request_url_resource_type": "original-resource",
  "robust_links_html": {
      "memento_url_as_href": "<a 
href=\"https://archive.li/wip/hWZdd\"\ndata-originalurl=\"https://abcnews.go.com\"\ndata-versiondate=\"202
0-06-15\">ABC News for June 15, 2020</a>",
      "original_url_as_href": "<a 
href=\"https://abcnews.go.com\"\ndata-versionurl=\"https://archive.li/wip/hWZdd\"\ndata-versiondate=\"2020
-06-15\">ABC News for June 15, 2020</a>"
  }
}
```

The JSON response from the API returns the link text and the requested URL to the user. Inside the `robust_links_html` key is the preformatted HTML for each default link target. For users' convenience to construct their own HTML, the API provides each attribute for the Robust Link as its own JSON key.

If the user supplies a URI-R, the Robust Links service will create a memento via the ArchiveNow library and use the current datetime as the value for `data-versiondate`. The service utilizes either the Internet Archive or Archive.Today to create the memento, and we are exploring the addition of Perma.cc in a future release. A client can supply an optional argument to the API to specify which web archive they desire for creating the memento. If they do not indicate a preferred web archive, the service randomly chooses one. If the user provides a URI-M to the API, the Robust Links service will dereference the URI-M and extract the values for `data-originalurl` and `data-versiondate` from the memento's HTTP response headers as specified in the Memento Protocol. Error codes, source code examples, and more are available on the API's documentation page.

Robustifying the Robust Links Browser Code

We considered advocating that authors include our JavaScript and CSS inline on every page. This would include the code **by-value**. Unfortunately, releasing JavaScript code this way has its problems. Copying and pasting code is error-prone and invites tinkering, which may result in Robust Links that do not function properly. Additionally, the hosting page will not receive bug fixes or updates that reflect changes in the environment with which it interacts, including changes to web archives that are covered and services overlaying those web archives, such as Memento Time Travel. We therefore decided that it was better to centrally host the code and recommend that web page authors link to its URLs, thus including it **by-reference**. The advantage of this decision was demonstrated during the writing of this article. We resolved an issue with conflicting CSS styles to ensure that Robust Links menus are presented as intended. All users of the Robusts Links JavaScript/CSS library will now benefit from this improvement without having to make changes to their own pages. As Web technologies evolve, providing code by-reference will allow us to devise new resolutions for future authors and improve everyone's Robust Links experience. Unfortunately, this means that our code itself becomes subject to link rot. To keep this solution operational, we need to solve two problems: First, we must select a hosting platform for the source code that is expected to remain available for the foreseeable future. Second, in case we have to move the code to another platform, we must provide continued access. We explored several options for addressing both issues.

With respect to choosing a suitable hosting platform, we realized that hosting our code at robustlinks.mementoweb.org is not a long-term solution. If robustlinks.mementoweb.org loses funding, the take-down of the site is likely. Solutions like hosting the code in a web archive creates a dependency on that specific web archive, something Robust Links is designed to avoid. The International Internet Preservation Consortium (IIPC) is an organization with members from over 45 countries, including national, university and regional libraries and archives. It focuses on providing tools, techniques, and knowledge for preserving web content and providing access to that content. The IIPC has graciously provided us with a GitHub repository in which to deposit robustlinks.js and robustlinks.css. Through GitHub Pages, we instruct GitHub to serve the code with the correct content-type so that web page authors can include it by-reference.

With the code hosted in IIPC's GitHub repository, we have improved its odds of surviving for the foreseeable future, but we have not solved the potential issue of changing URLs. This scenario is not inconceivable as, for example, GitHub Pages could change their URL paths, the IIPC could move their content to another platform such as GitLab, or even host it themselves. A change in URL would cause the Robust Links menus to fail and readers would no longer be able to benefit from the convenience provided by this code. We know from a 2020 Cloudflare report that web page authors rarely update a link to a JavaScript library link once a page is published. We have no reason to believe that links to robustlinks.js and robustlinks.css will be any different, so we decided to investigate the possibility of minting persistent identifiers for our robustlinks.js and robustlinks.css URLs.

Considering that funding agencies did not fund our project infrastructure indefinitely, running our own persistent identifier server would eventually lead to the same link rot for our browser code that our project is trying to defeat. We considered using DOIs (and their handle variation), PURLs, and W3IDs. We decided in favor of DataCite DOIs because DataCite is an organization with a rapidly growing membership base, values regarding openness that we share, and a strong commitment to the long-term usability of their identifiers. Our constructive interactions with DataCite staff in the context of this effort gave us further confidence in our choice.

What was still needed was a party - a Datacite member - to mint the DOIs assigned to our code and manage them going forward, for example, for updating the correspondence table, in case the code's location changes. We found an agreement with collaborators from Old Dominion University (ODU) to take on this role. Our agreement benefits from the fact that ODU holds memberships in both organizations Datacite and IIPC.

Thus, to robustify the Robust Links browser source code, we developed the following solution after much collaboration with third parties:

1. Ensure that an entity with longevity hosts our code, in our case the IIPC.
2. Engage a collaborator to mint and manage a DOI so that URL changes do not break web page access. In our case ODU registered 2 DOIs provided by DataCite pointing directly at our JavaScript and CSS resources.
3. Use DOIs in all instructions and other code references so that the underlying URLs may change, but authors do not need to update their pages.

A Journey to Combat Reference Rot

Reference rot, the union of link rot and content drift, is a function of the dynamic nature of the Web and the increasing frequency at which we link to web resources in our publications. Its detrimental effects have a negative impact on all web-based inquiries that require accuracy of referenced resources, yet, thus far, we have not seen a comprehensive approach to effectively address it. Our Robust Links approach changes the status quo by utilizing publicly available web archiving infrastructure to proactively create mementos in different archives at the time of linking. Robustified links offer several fallback mechanisms in case the original linked resource and its pro-actively created memento become unavailable. Our Robust Links web service provides a simple user interface for authors to create robust links and conveniently obtain the HTML that embeds the Robust Links JavaScript and CSS in linking pages. The underlying API offers the same functionality to machines, for example, to accommodate batch processing of many links.

Making robustified links actionable in a browser requires the inclusion of robustlinks.js and robustlinks.css into the web page. With an eye on an ever-changing technological environment, we were motivated to establish a solution that allows for continued modification of the code while also helping to ensure its long-term availability and accessibility. Our approach entails hosting the code in the community-supported IIPC GitHub repository and minting persistent identifiers (DOIs) that redirect to the code as served by GitHub Pages. This setup enables the by-reference inclusion of the JavaScript and CSS code using their DOIs.

Robustifying Links takes effort. While our web service makes this process more convenient for humans and machines, we can envision more automated workflows. For example, content management systems such as WordPress and MediaWiki, word processing platforms such as Microsoft Word and Google Docs, or blogging platforms such as Blogger and Medium could integrate Robust Links functionality. At the end of the day, however, it still takes humans' commitment to combat reference rot.

As is true for all web archiving based approaches, we work under the assumption that "good enough" mementos of linked resources can be created but we realize that existing archiving and replay technology does not always satisfy. However, that is a challenge beyond the scope of this effort that many talented people are working on.

Acknowledgments

The authors would like to thank everyone who contributed to the Robust Links concept and service. The Robust Links work was initiated in the Hiberlink project, a collaboration between the University of Edinburgh and the Los Alamos National Laboratory, funded by the Andrew W. Mellon Foundation. Michael L. Nelson from Old Dominion University contributed to Robust Links, both during the Hiberlink project and the recent effort to improve the solution as described in this paper. We appreciate the assistance of DataCite staff Helena Cousijn, Martin Fenner, and Matt Buys for supporting the solution that uses DOIs to accord a persistent URI to Robust Links browser code. We also appreciate the service of Karen Vaughn at Old Dominion University for managing those DOIs. Thanks to Mark Graham from the Internet Archive for answering questions about their collaboration with Wikipedia. Finally, we also thank the International Internet Preservation Consortium for agreeing to ensure the browser code's longevity.

About the Authors

Shawn Jones is a graduate research assistant working for the Prototyping Team at LANL's Research Library. Shawn is also a PhD student at Old Dominion University and a member of the Web Science and Digital Libraries research group, led by Michael Nelson. He has contributed tools, data, and analysis to projects and frameworks such as Memento, Signposting, and Robust Links. In addition, Shawn is the lead of the Dark and Stormy Archives project, an initiative to summarize web archive collections through visualization techniques common in social media. More information about Shawn is available at: https://www.shawnmjones.org/

Martin Klein is a scientist and lead of the Prototyping Team in LANL's Research Library. In this role he focuses on research and development efforts in the realm of web archiving, scholarly communication, digital system interoperability, and data management. He is involved in standards and frameworks such as Memento, ResourceSync, Signposting, and Robust Links. Martin holds a Diploma in Computer Science from the University of Applied Sciences Berlin, Germany and a Ph.D. in Computer Science from Old Dominion University. More information is available at: https://orcid.org/0000-0003-0130-2097

Herbert Van de Sompel is Chief Innovation Officer at Data Archiving and Networked Services (DANS) in The Netherlands. He has played a major role in creating the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), the Open Archives Initiative Object Reuse & Exchange specifications (OAI-ORE), the OpenURL Framework for Context-Sensitive Services (ANSI/NISO Z39.88-2004), the SFX linking server, the bX scholarly recommender service, info URI (RFC 4452), Open Annotation (W3C Community Group specification), ResourceSync (ANSI/NISO Z39.99-2014), Memento "time travel for the Web" (RFC 7089), Robust Links, and Signposting. He holds degrees in Mathematics, Computer Science, and Communication Science from Ghent University, Belgium. More information at https://hvdsomp.info