

Unobtrusive and Extensible Archival Replay Banners Using Custom Elements

Sawood Alam, Mat Kelly, Michele C. Weigle, and Michael L. Nelson

Old Dominion University

Department of Computer Science

Norfolk, Virginia, USA

{salam,mkelly,mweigle,mln}@cs.odu.edu

ABSTRACT

We compare and contrast three different ways to implement an archival replay banner. We propose an implementation that utilizes *Custom Elements* and adds some unique behaviors, not common in existing archival replay systems, to enhance the user experience. Our approach has a minimal user interface footprint and resource overhead while still providing rich interactivity and extended on-demand provenance information about the archived resources.

CCS CONCEPTS

• **Information systems** → **Digital libraries and archives**; • **Human-centered computing** → **User interface design**;

KEYWORDS

Memento; Archival Replay; Archival Banner; User Interface

ACM Reference Format:

Sawood Alam, Mat Kelly, Michele C. Weigle, and Michael L. Nelson. 2018. Unobtrusive and Extensible Archival Replay Banners Using Custom Elements. In *JCDL '18: The 18th ACM/IEEE Joint Conference on Digital Libraries, June 3-7, 2018, Fort Worth, TX, USA*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3197026.3203881>

1 MOTIVATION

Web archival replay systems express that a user is interacting with a *memento* (an archived representation of a resource) by adding an archival banner. Archival banners provide metadata about both the memento and original resource as well as serving to distinguish a memento from its corresponding original resource. To more accurately replicate the live web experience when viewing a memento, an archive rewrites parts of a memento's contents to ensure the embedded resources and hyperlinks are referenced to the web archive and not the live web. Any component injection in the page makes it different from the original and may consume additional screen real estate. We illustrate this in Figure 1 by archiving `example.com` in three different archives successively, resulting in cascading banners.

We propose an unobtrusive, interactive, responsive, multi-state, and extensible archival banner experience using *Custom Elements*.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

JCDL '18, June 3-7, 2018, Fort Worth, TX, USA

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5178-2/18/06.

<https://doi.org/10.1145/3197026.3203881>

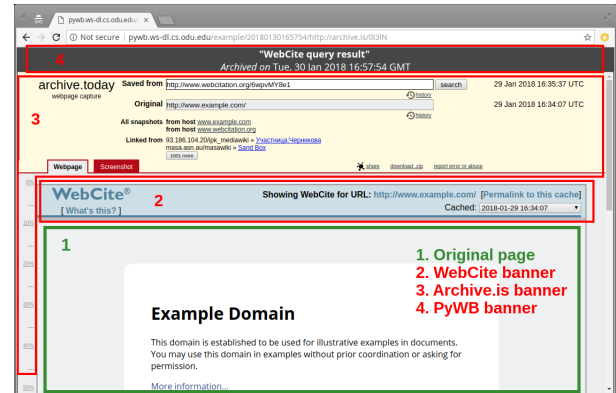


Figure 1: Three Cascading Archival Banners in a Memento

It requires minimal real estate and allows drag-and-drop repositioning in the viewport while hiding itself when not needed. In the on-demand extended mode it provides a set of interactive visualizations and provenance information that are customizable by the archive.

2 METHODOLOGY

There are three primary ways to serve an archival banner with an archived web page that shares the rendering space with the memento. Browser toolbars (e.g., the now defunct *MementoFox*) and archival emulators (e.g., *NetCapsule*) are out of scope of this work.

Inline Plain HTML Banners – This is the simplest and most commonly used approach of adding an archival banner. Many services such as the *Internet Archive*, *Archive-It*, *UK Web Archive*, and *Archive.is* use this method. *OpenWayback*, a commonly used archival replay system, supports this method. In this approach, necessary markup and style for the banner is injected directly into the archived *HTML*. While simple, it poses some serious issues, such as conflicts with the style of the memento or hiding important elements of the page, such as the header of the site.

Frame/Iframe Banners – Using *frame* or *iframe* HTML elements is another common technique to provide archival banners. This can be implemented by 1) serving both the banner and memento documents in separate *frames/iframes* of a parent page, or 2) making the banner document as the outer page and serving the memento inside an *iframe*. For example, *WebCite* uses the first approach while many archives, such as the *Portuguese Web Archive*, use the second. *PyWB*, a popular web archival replay system, uses the second approach by default, but allows using plain inline *HTML* banners. *Iframes* provide full document isolation, both style and origin. Therefore, *iframe* banners do not conflict with the position

or style of other elements of the *memento*. However, their positioning is not flexible enough to place them at any arbitrary location in the viewport. Since these banners must be placed clear of the memento without any overlay possibilities, often less screen real estate is available to render mementos.

Custom Elements Banners – *Custom Elements* [4] is a recent web standard. It allows developers to define their own custom *HTML elements* in *JavaScript* using the same capabilities that native elements have. Using this approach we can hide the implementation details of the banner while injecting a minimal markup in a *memento*. By using the *Shadow DOM* [5] the style is scoped to the custom element, hence, there are no conflicts with other elements of the *memento*. Unlike an *iframe*, the origin of a *custom element* is not isolated from the main document, which has a potential security risk [3]. This can be mitigated by preventing any live-leaks in *memento* reconstruction, such as by using *ServiceWorker* [2]. Native support of *Custom Elements* is about 79% globally (as of Apr 1, 2018¹) and rising. This approach is used by a banner introduced in *Reconstructive* [1], which is used by the *IPWB* [6] replay system.

3 RECONSTRUCTIVE BANNER

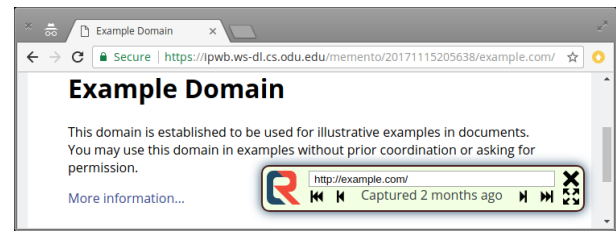
The three approaches described in Section 2 only address the implementation of the markup. However, the user experience of all three has some issues. We want to keep the experience as close to the original as possible, while still providing some means to interact with the memento and to display provenance information in an unobtrusive manner. Many current implementations load the metadata for every page, even when a user may not be interested. This results in a slow page load, poor user experience, and resource wastage of both the client and server. We only bring more data when a user asks for it. We propose the following two-state banner called *Reconstructive Banner*. It is implemented using *Custom Elements* with some UI behaviors (such as being responsive, unobtrusive, and expandable) to improve the user experience.

Floating Action Bar (FAB) – A floating bar with basic archival information and quick action buttons is placed in the bottom-right corner (as illustrated in Figure 2(a)), but draggable to any place on the page. The button shows a branding logo of the archive to highlight that it is not part of the *memento*. A similar approach was earlier used in *Mink* [7]. After a set duration of inactivity on the page, the *FAB* hides itself and reappears upon a user interaction such as mouse move or page scroll.

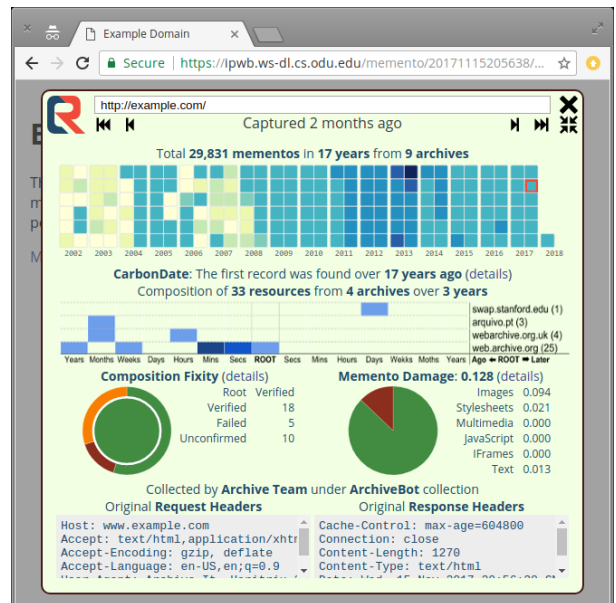
Expanded – When the *expand icon* in the *FAB* is clicked/tapped, it expands into a modal window with extended archival provenance information, interactive visualizations, and other metadata (as illustrated in Figure 2(b)). The *expand icon* changes to a *collapse icon* to allow restoring the *FAB* state. The expanded state does not hide or collapse automatically without an explicit user action. Both states have a *close icon* to completely remove the banner from the UI. This state may make additional resource requests to fetch some visualization libraries or data.

4 CONCLUSIONS AND FUTURE WORK

We proposed an unobtrusive and extensible method to revamp the archival replay banner UI. We implemented *Reconstructive Banner*,



(a) Draggable Floating Action Bar (FAB): Brief Information and Quick Actions



(b) Expanded: Metadata, Provenance, and Interactive Visualizations

Figure 2: Reconstructive Banner Modes

a responsive, multi-state, and details-on-demand archival banner using *Custom Elements* to be both more interactive and more informative to the user. We anticipate exploring the newfound and extensible abilities to archival replay banners as applicable in other archival systems.

5 ACKNOWLEDGEMENTS

This work is supported in part by NSF grant III 1526700.

REFERENCES

- [1] Sawood Alam. 2018. Introducing Reconstructive - An Archival Replay ServiceWorker Module. <http://ws-dl.blogspot.com/2018/01/2018-01-08-introducing-reconstructive.html>. (2018).
- [2] Sawood Alam, Mat Kelly, Michele C. Weigle, and Michael L. Nelson. 2017. Client-side Reconstruction of Composite Mementos Using ServiceWorker. In *Proceedings of JCDL*. 237–240.
- [3] John A. Berlin. 2018. *To Relive The Web: A Framework For The Transformation And Archival Replay Of Web Pages*. Master's thesis. Old Dominion University, Department of Computer Science.
- [4] Domenic Denicola. 2016. Custom Elements. <https://www.w3.org/TR/custom-elements/>. (2016).
- [5] Hayato Ito. 2017. Shadow DOM. <https://www.w3.org/TR/shadow-dom/>. (2017).
- [6] Mat Kelly, Sawood Alam, Michael L. Nelson, and Michele C. Weigle. 2016. Inter-Planetary Wayback: Peer-To-Peer Permanence of Web Archives. In *Proceedings of TPDL*. 411–416.
- [7] Mat Kelly, Michael L. Nelson, and Michele C. Weigle. 2014. Mink: Integrating the Live and Archived Web Viewing Experience Using Web Browsers and Memento. In *Proceedings of JCDL*. 469–470.

¹<https://caniuse.com/#feat=custom-elementsv1>