A Survey of Archival Replay Banners

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 surveyed various archival systems to compare and contrast different techniques used to implement an archival replay banner. We found that inline plain HTML injection is the most common approach, but prone to style conflicts. Iframe-based banners are also very common and while they do not have style conflicts, they suffer from screen real estate wastage and limited design choices. Custom Elements-based banners are promising, but due to being a new web standard, these are not yet widely deployed.

1 INTRODUCTION
Web archival replay systems express that a user is interacting with a memento (an archived representation of a resource at a URI-M) by adding an archival banner. Archival banners provide metadata about both the memento and the original resource as well as serve to distinguish a memento from its corresponding original resource. These banners may also contain various controls and toolbars to interact with the archive and the memento.

There are many ways to include an archival banner, both from the code and the user interface (UI) perspectives. An archival banner can be part of a standalone native archival application (e.g., WAIL [5]), a browser toolbar (e.g., now defunct MementoFox), or directly included in the markup of the served memento. It is the latter that is the focus of this survey, i.e., the banners that share the viewport and rendering environment with the memento (injected by the server or a client-side script/extension). Banner injection in a memento is generally obtrusive (it makes the page look different from the original) and may consume additional screen real estate. We illustrate this in Figure 1(a) by archiving example.com.

1.1 Inline Plain HTML Banners
Inline plain HTML is the simplest and most commonly used method of adding an archival banner in which necessary markup and style are injected directly in the archived HTML. While simple, it poses some issues such as vulnerability to attacks [4], conflicts with the style of the memento (as illustrated in Figure 1(b)), or hiding important elements of the page (e.g., the header of the site).

Many services such as the Internet Archive, Archive-It, and UK Web Archive (Figure 1(b)) use this method. OpenWayback, a commonly used archival replay system, supports it. The Archive.is banner, highlighted as number 3 in the Figure 1(a), is different from the above mentioned archives. It flattens the rendered memento markup, removes all the JavaScript, and injects it into a page that is surrounded by the banner markup. Oldweb.today uses a similar technique, but it utilizes server-side rendering that is emulated on a canvas element using Virtual Network Computing (VNC). Mink [10] is a Chrome extension that injects banner markup in a page on-demand (as illustrated in Figure 1(c)). It uses Shadow DOM [8] to isolate the style of the banner from the page.

1.2 Frame/Iframe Banners
Using frame or iframe HTML elements is another common technique to provide archival banners. Iframe banners provide full document isolation, both origin and style, hence do not conflict with the position or style of the 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.

This can be implemented by 1) serving both the banner and memento documents in separate frames/iframe of a parent page, or 2) making the banner document as the outer page and serving the memento inside an iframe. For example, WebCite (highlighted as number 2 in the Figure 1(a)) uses the first approach while many archives, such as the Portuguese Web Archive and National Records of Scotland, use the latter. PyWB, a popular web archival replay system, uses the latter approach by default (highlighted as number 4 in the Figure 1(a)), but allows using plain inline HTML banners.

1.3 Custom Elements Banners
Custom Elements [7] is a recent web standard. It allows developers to define their own custom HTML element in JavaScript using the same capabilities that native elements have. Implementation details of the banner can be hidden, allowing a minimal and clean markup injection. By using the Shadow DOM the style is scoped to the banner, hence, there are no conflicts with other elements of the memento. This method allows both flexible design and placement choices like inline markup banners and style isolation like iframes.

2 BANNER COMPARISON
Table 1 compares three primary techniques to serve an archival banner markup with a memento used in archival replay banners of various archival systems. Below are their brief descriptions and how are they used in various archival systems.

Table 1: Comparison of different archival banner types

<table>
<thead>
<tr>
<th>Features</th>
<th>Plain HTML</th>
<th>Iframe</th>
<th>Custom Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation</td>
<td>Simple</td>
<td>Difficult</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Markup rewriting</td>
<td>Messy</td>
<td>None</td>
<td>Clean</td>
</tr>
<tr>
<td>Compatible browsers</td>
<td>All</td>
<td>All</td>
<td>Modern</td>
</tr>
<tr>
<td>Isolation level</td>
<td>None</td>
<td>Document</td>
<td>Style</td>
</tr>
<tr>
<td>Positioning</td>
<td>Anywhere</td>
<td>Edges</td>
<td>Anywhere</td>
</tr>
<tr>
<td>Element overlap</td>
<td>Likely</td>
<td>Unlikely</td>
<td>Likely</td>
</tr>
<tr>
<td>Draggable &amp; floatable</td>
<td>Possible</td>
<td>No</td>
<td>Possible</td>
</tr>
<tr>
<td>URI-M visibility</td>
<td>Clear</td>
<td>Hidden</td>
<td>Clear</td>
</tr>
<tr>
<td>Origin isolation</td>
<td>Limited</td>
<td>Possible</td>
<td>Limited</td>
</tr>
</tbody>
</table>
This approach is used by a banner introduced in Reconstructive [1], which is used by the IPWB [9] replay system. Reconstructive Banner [3] is an unobtrusive, interactive, responsive, and extensible multi-state archival banner. It requires minimal real estate in the Floating Action Bar (FAB) state (as illustrated in Figure 2(a)) and allows drag-and-drop repositioning in the viewport while hiding itself when not needed. In the on-demand Expanded state (as illustrated in Figure 2(b)) it provides an extensible set of interactive visualizations and provenance information that are customizable by the archive. To prevent from any live-leaks (or zombies [6]) it utilizes ServiceWorker for client-side reconstruction [2] of composite mementos.

3 CONCLUSIONS AND FUTURE WORK
We surveyed various archival systems and described different techniques used to implement an archival replay banner. Inline plain HTML injection is the most common approach used by many systems, but prone to style conflicts. Iframe-based banners are also used by many archival systems and while having style isolation, they suffer from screen real estate wastage and limited design choices. A more promising approach is Custom Elements-based banners, as used in the Reconstructive and IPWB.

REFERENCES