Digital Signage Client Side Report
August-September 2019

Purpose

To determine the feasibility of using client side digital signage to replace the library's current workstation. Since the library would have only a small number of display units at one location, licensing a digital signage package intended for commercial use might not be cost effective. With one digital signage workstation per floor, it should be possible to substitute a web browser for the central editing and control programs and media player software. Since both poster images and events information are available from online sources, the purpose of this evaluation was to demonstrate whether they could be integrated successfully into a web page.

Objectives

1.) To set up a prototype web page that would integrate both poster images and events information.
2.) To test and evaluate as many events outputs as are available from LibCal.
3.) To determine whether an integrated web page would be robust enough to meet the ongoing needs of the library.

Methodology

A three panel prototype web page was developed. The left panel consisted of a rotating carousel of digital posters for library events; the top panel consisted of a heading for the screen; and the right panel consisted of a listing of library events derived from the Springshare LibCal system. This structure made it possible to test bringing in poster images from the library's website and the RSS data feed from LibCal.

There are four versions of the prototype: two for computers that use 1920x1080 (HDTV) monitors and two for older computers that support 1440x900 displays. Of each pair one version uses two lines per event entry; the other four lines per entry. Here are the addresses:

1920x1080
https://sites.uni.edu/caswell/epl/dsg0.html

https://sites.uni.edu/caswell/epl/dsg2.html

1440x900
https://sites.uni.edu/caswell/epl/dsg1.html

https://sites.uni.edu/caswell/epl/dsg3.html

LibCal Outputs

Most of the LibCal outputs are formatted HTML pages, which are generated by PHP processes on the LibCal server. The main version of the calendar (https://boone-lib.libcal.com/calendar/programs) can stand independently or be embedded in a web page by means of an iframe. It consists of a calendar at the top, a list of filters underneath; and a list of events beneath that. The embedded calendar and the mini-calendar can also be embedded in web pages by means of an iframe. Each contains a monthly calendar and a list of events below. All three of these outputs are intended to be used interactively, that is, having users click on hyperlinks to narrow the date selection or bring up details for specific events. Thus they are not suitable for a digital signage application which is for display purposes only.

The Events API provides a well-formatted list of events without a calendar and looks good when embedded in an iframe. However, there are no external formatting options for the presentation of the events. Whenever graphics are included in the description field, the balance of the display is affected.

API Endpoints is a new output service provided by Springshare, which requires a paid subscription and the establishment of a security key. Unlike the four options above it is a true data feed without HTML formatting. It returns a list of events in JSON format, which requires programming in JavaScript or another language in order to present them in a web page. It will be worthwhile following the development of this approach, but it could be costly to implement, depending on the tools and staff required.

RSS Feed
This is a free service which provides a list of events in XML. It adheres to the RSS 2.0 specification, which allows for the provision of extra fields called namespaces. These extend the content beyond the core RSS fields of title, description, and link, and make it possible to select, convert, and display them with great precision. Thus the date, time, and location of events can be presented without having to resort to work-arounds such as are required in feeds without these fields.

Converting XML into HTML uses a function built into all modern browsers called XMLHttpRequest. JavaScript is used to select and prepare the fields for presentation. Models for these processes may be found at the W3Schools training site. However, one has to be aware of minor differences in the way browsers handle these processes in order to be successful in converting the XML and displaying the results.

There are also security issues in obtaining and converting the XML. Because of the exploitation of past vulnerabilities web browsers do not allow material from different origins to be mixed. In addition, some server environments place restrictions on downloading external content. There are some work-arounds such as using operating system utilities to download feeds on a regularly scheduled basis. Overcoming these obstacles is a necessary prerequisite to developing a production-level.system.

As of the end of September the four prototypes that have been developed are updated daily with a week's events. The University of Northern Iowa's Network Services graciously consented to permit automatic updating, even though they don't normally allow it.

Poster Images
Poster images are generated on the web, saved to a local drive, and then uploaded to the library's website. The website runs on Plone, which is an open source content management system that can also provide some web services. Plone stores all information in an object-oriented database called the Zope Object Database. This database assigns a unique ID (UID) to each object, which makes it possible to call each object by that ID as well as by the object name. New objects can be uploaded into Plone as replacements for existing objects. In order to call them without having to change the coding in the external applications, several objects with fixed names were created. This created virtual or persistent URLs which allow the content behind those addresses to be changed. The new objects are uploaded on the Edit page by clicking the option for replacing existing images and navigating to the new file on a local drive. Thus, when a poster image for a particular event expires, the object at that address can be replaced with a poster image for another event. In this way the poster images uploaded for the website will also serve the external application.

Summary
Developing a low cost web alternative to a full-blown digital signage package for events information is definitely feasible. Web browsers are much more powerful than they used to be and contain the functions needed to transform and format imported information. In full screen mode they provide as good an image as dedicated digital signage applications do. They also

cache the external output locally, so that they do not place a burden on the network connection. Refreshing the browser screen is all that is needed to reload images or events information. The challenges lay in the mechanics of obtaining the external data, integrating it into one application successfully, and in dealing with the security requirements of various server environments. Now that those issues have been addressed, the library can evaluate whether this will be a suitable replacement for the existing digital signage workstation.

Jerry V. Caswell
Revised: 04 Oct 2019