



|                                     |   |
|-------------------------------------|---|
| <b>Title</b>                        | Introducing web components into the LIS graduate curriculum   |
| <b>Authors(s)</b>                   | Wusteman, Judith  |
| <b>Publication date</b>             | 2017-12   |
| <b>Publication information</b>      | Wusteman, Judith. "Introducing Web Components into the LIS Graduate Curriculum." Emerald, December 2017. <a href="https://doi.org/10.1108/ILS-08-2017-0084">https://doi.org/10.1108/ILS-08-2017-0084</a> .  |
| <b>Publisher</b>                    | Emerald   |
| <b>Item record/more information</b> | <a href="http://hdl.handle.net/10197/9704">http://hdl.handle.net/10197/9704</a>   |
| <b>Publisher's statement</b>        | This article is © Emerald Group Publishing and permission has been granted for this version to appear here <a href="http://researchrepository.ucd.ie">http://researchrepository.ucd.ie</a> . Emerald does not grant permission for this article to be further copied/ distributed or hosted elsewhere without the express permission from Emerald Group Publishing Limited. |
| <b>Publisher's version (DOI)</b>    | 10.1108/ILS-08-2017-0084  |

Downloaded 2026-05-02 00:27:47

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd\_oa)



© Some rights reserved. For more information

## **Introducing Web Components into the Library & Information Science Graduate Curriculum**

Judith Wusteman

School of Information and Communication Studies, University College Dublin, Ireland.

### **Abstract**

This article introduces web components, one of the most exciting and potentially transformative of the technologies that comprise HTML5. Web components provide a standardized method of creating and sharing custom HTML elements. This article proposes that the web development curriculum in LIS graduate programs needs to expand to cover this important topic. Despite the potentially seismic impact of web components on web development, up until this point, there has not been a discussion of this technology within library and information science literature.

### **Keywords**

Web components, LIS curriculum, information technology, HTML5, web development

### **Introduction**

As of July 2017, there are over 91 million websites using HTML5 (BuiltWith, 2017). This fact may lull library and information science (LIS) educators into a sense of complacency: HTML5 is here and the world of the web seems to be largely the same as it ever was. But that sense of complacency is false. HTML5 represents a profound change in the potential of web development and it will facilitate a new generation of websites. The web landscape has already started to change; it will not be business as usual for web developers or end users (Wusteman, 2014). And one of the most exciting HTML5 technologies is web components, a powerful new method of creating custom elements.

Web development is recognized as centrally important in LIS graduate programs and is a core job requirement for many technical librarianship roles. Web development modules in LIS graduate programs need to respond to the changes emerging from the application of HTML5 in general and web components in particular.

This article identifies some of the current discussion points around technology skills within LIS graduate curricula. It goes on to introduce the HTML5 family of standards, before focusing on the subset that comprise web components. It illustrates the potential of web

components to transform website markup, and then gives an overview of their implications for LIS graduate syllabi.

## **Technology innovators or technology users?**

The increasing focus on information technology (IT) throughout the LIS professions has long been a discussion point in the literature (Maceli, 2016; Virkus, 2015; NMC, 2014; Breeding, 2013; Riley-Huf & Rholes, 2011; Audunson, 2007). Kelley (2015) notes the growing specialization of librarianship: generalists are being replaced by new specific roles such as "data analysis librarian, web instruction librarian, emerging technologies librarian, and metadata librarian".

Maceli (2015) highlights the many calls, from both researchers and practitioners, for information professionals to become "technology leaders and innovators instead of being simply users of technology". However, this does not appear to reflect the reality of many library roles. Emanuel (2013) finds that even "digital native academic librarians" are more likely to be technology users than technology creators. They are, for example, more "comfortable" using content management systems (CMS) and web page creation software such as Dreamweaver, than using technical standards such as HTML (HyperText Markup Language) and CSS (Cascading Style Sheets). Neither are they confident with computer programming languages such as C++ and PHP. Despite this, "computer and web programming" are the technologies these librarians would most like to learn.

Maceli and Burke (2016) report the same focus on "end-user" technology by LIS practitioners, with email and word processing being the most common technologies used. Library website creation or management appears 24th on this list, and only 15% of respondents deploy computer programming. But, again, this latter is the skill that most respondents would like to learn, with HTML being the most common "programming language" mentioned, followed by Python, then CSS and then JavaScript.

It is not clear, from the literature, to what extent the library roles of LIS graduates are directed by the skills that they bring from LIS schools, and to what extent the roles are defined by requirements of the organization they are working for. If LIS graduates were more technically skilled, would their job roles change to reflect this? Or are they well trained for the roles that they are entering? Despite the desire to learn programming, a majority of digital native academic librarians in the study by Emanuel (2013) felt that library school "prepared them well to deal with the technology". And Maceli and Burke (2016) comment that "A relatively small percentage of practitioners have technology-intensive roles that would require skills such as programming". However, they suggest that an awareness of the potential of such tools should be a core skill. Further, they define webpage creation as a "skill that bridged intensely technical and more user-focused work".

## Teaching for the Future, Not the Past<sup>1</sup>

In the last decade, there have been a major increase in the number of technology courses taught in LIS schools. Maceli (2015) reports that those offered by North American ALA-accredited LIS graduate programs are dominated by the topics of user experience and web design/development. However, whereas the web design /development skills correlate well with jobs advertised in code4lib, there are a lack of library jobs that required "user experience" skills. She also notes "relatively few" programming courses in the North American programs, and suggests that jobs that require such skills would probably hire "trained software developers with experience in the library domain" rather than candidates with only an LIS degree.

Potnis *et al.* (2016) suggest that LIS graduate programs could provide students with the skills and knowledge needed to allow them to act as "mobile technology consultants for information organizations". These ambitious proposals would require "interdisciplinary coursework in information science, business, and technology", to include web programming, computer networking and mobile application development and management. As of September 2015, no ALA-accredited LIS graduate programs in the US covers the required combination of skills (Potnis *et al.*, 2016).

Despite the "interdisciplinary and transdisciplinary degree programs" that are in increasing hallmark of the more than seventy iSchools world-wide (Cortez, 2016), it is likely that such a computer science-heavy focus as proposed by Potnis *et al.* could only ever be relevant to a small percentage of LIS graduate students. Adding ever more computer science skills to the LIS curriculum does not necessarily reflect either the strengths of LIS graduate students or the roles that they ultimately find themselves playing within libraries and the information professions in general. But, given that web development, and not user experience skills, tend to dominate the relevant job advertisements, it is the former, and not the less technical latter field, that appears to offer the most openings for technical LIS graduates.

Thus, a nuanced response is required to the conundrum of where and how to focus LIS graduate skills within the technology sphere. There should be no fixed model for the LIS graduate technology portfolio: curricula need to be technology-focused but also "flexible and innovative" (Cortez, 2016). And educators need to be continually on the look-out for those emerging technologies and skills for which there is a perceived demand and which complement the strengths of LIS graduates. As Ratledge & Sproles (2017) point out, LIS graduate programs are "generally not geared toward elevating anyone to expert-status in the many technology specializations". However, with new specialist library roles, each of which require a "unique set of experiences and qualifications" (Kelley, 2015), aspiring librarians need to identify and focus on the "the skills they need to advance in their chosen sub-field" (Ratledge & Sproles, 2017). At the same time, core to the LIS curriculum should be

---

<sup>1</sup>(Cortez, 2016)

a broad but thorough overview and awareness of the web development process, and its latest tools and techniques.

## **Web development with HTML5**

An upshot of the advent of HTML5 is that there is now considerably more to understanding HTML than a familiarity with the concept of markup (Wusteman, 2014). HTML5 is larger and more complex than its predecessors, HTML4 and XHTML, and it represents a seismic change in the concept of the website. Formerly static information-dissemination conduits are making way for highly dynamic and interactive environments, often highly social and increasingly mobile. And this new concept requires changes in the way we teach web development in LIS graduate programs.

Despite comments by respondents in the study by Maceli and Burke (2015), HTML and CSS are not, strictly speaking, programming languages; rather, they are markup languages. HTML identifies and marks up the structure of a web page and CSS styles webpages, but neither are executable scripts; they contain no programming logic. Thus, they require considerably less technical ability to master than a true programming language such as JavaScript or PHP, and this makes them accessible to the majority of LIS graduate students.

Until now, web development in LIS graduate programs has often been translated as a coverage of the basics of HTML and CSS markup, occasionally with a brief reference to JavaScript. This is no longer enough; it will not adequately equip students to take a meaningful part in the creation of the new generation of websites. As well as HTML, CSS3 (MDN, 2017) and JavaScript, HTML5 comprises an entire family of standards including the following:

- Canvas, a drawing surface, which enables 2D and 3D graphics to be rendered on the fly, thus facilitating a new generation of sophisticated interactive images.
- Local storage, which enables HTML5 websites to store large amounts of data in the client. For example, this enables users to access Google calendar events even when they are disconnected from the Web. Local storage is increasingly important in mobile devices.
- Geolocation, which enables the standardized retrieval of the geographical location of a client. Again, this is increasingly important for mobile devices.
- Drag and Drop, which allows users to grab HTML elements displayed in web pages and drop them elsewhere on the page.
- Server-Sent Events, which facilitates the automatic receipt of updates from the server without the client needing to request them.

A description of all HTML5 standards, along with an indication of the level of support provided by current browsers, can be found at <http://html5please.com/>.

To be a fully productive member of a web development team, the future LIS graduate will need to be cognizant of these standards, understanding their function and the implications of their use. And, given that most of these standards are not challenging to implement once

their function is understood, an ability to implement solutions incorporating them would be a distinct advantage.

Among this standards family are Custom Elements, Shadow DOM, HTML Templates and HTML Imports. Collectively, this suite of technologies is known as "web components".

## Web components

The HTML4 standard includes 70 elements<sup>2</sup> (W3C, 1999). HTML5 introduced 32 new elements, including structural elements such as <article>, <header>, <main>, <footer>, the new <canvas> graphics element and a range of new media elements such as <audio> and <video> (W3C, 2014). Deployment of these 32 new elements will undoubtedly facilitate innovation on the web. But what transformations might 100 new elements facilitate? What about 1,000? Or 10,000?

Web components provide a method of creating and sharing custom HTML elements. At this early stage, there are already over 1,000 custom HTML elements available from the *webcomponents.org* shared library alone, and this number is likely to grow exponentially as all aspects of the standard are finalized and become implemented in browsers. The potential of web components is demonstrated in the following example.

Figure 1 illustrates a single WorldCat<sup>3</sup> search result. The HTML markup currently required to produce this display can be seen in Figure 2; it appears confusingly verbose and could be time-consuming to code and maintain. And, yet, the WorldCat markup is an example of current good practice: it is well structured and employs appropriately named CSS classes. Figure 2 simply demonstrates the level of complexity that is often necessary in a pre-web components world.

**Figure 1: A single WorldCat search result**



<sup>2</sup> Number does not include deprecated elements

<sup>3</sup> <https://www.worldcat.org/>

**Figure 2: HTML markup required to produce the single WorldCat search result illustrated in Figure 1**

```
<tr class="menuElem">
  <td class="num"><input name="itemid" id="itemid_846177884" value="846177884"
  type="checkbox"><label for="itemid_846177884" style="display:none">1. Principal
  component analysis</label></td>
  <td class="num">1.</td>
  <td class="coverart">
    <a href="/title/principal-component-analysis/oclc/846177884&referer=brief_results">
    </a>
    </td>
  <td class="result details">
    <div class="oclc_number">846177884</div>
    <div class="item_number">1</div>
    <div class="name">
      <a id="result-1"
      href="/title/principal-component-analysis/oclc/846177884&referer=brief_results"><str
      ong>Principal component analysis</strong></a>
    </div>
    <div class="author">by I T Jolliffe</div><div class="type">
      &nbsp;<span class="itemType">Print book</span><a
      href="/title/principal-component-analysis/oclc/846177884/editions?editionsView=true&
      p;referer=br" title="View all held editions and formats for this item"> View all formats and
      languages »</a>
    </div>
    <div class="type language">Language: <span class="itemLanguage">English</span>
    &nbsp;</div><div class="publisher">Publisher: <span class="itemPublisher">New York
    Springer 2010</span></div><div class="type database">Database: <span
    class="itemDatabase">WorldCat</span></div>
  <ul class="options">
```

```
<li class="elinkstatus av" id="et1" style="display: none;"><a class="tab" id="el1"
href="javascript:void(0);" title="View this item">View Now</a></li>
<li> <a
href="/title/principal-component-analysis/oclc/846177884/editions?editionsView=true&am
p;referer=br" title="View all held editions and formats for this item"> View all editions
»</a></li>
</ul>
<div class="panel hidepanel" id="elpanel1"><p class="closepanel"><a
href="javascript:void(0);" title="Close">Close</a></p></div>
<div id="slice">
<span class="Z3988"
title="url_ver=Z39.88-2004&amp;rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Abook&
&amp;rft.genre=book&amp;req_dat=%3Csessionid%3E&amp;rfe_dat=%3Caccessionnumber
%3E846177884%3C%2Faccessionnumber%3E&amp;rft_id=info%3Aoclcnum%2F846177884
&amp;rft_id=urn%3AISBN%3A9781441929990&amp;rft.aulast=Jolliffe&amp;rft.aufirst=l&a
&amp;rft.aunitm=T&amp;rft.btitle=Principal+component+analysis&amp;rft.date=2010&amp;r
ft.isbn=9781441929990&amp;rft.place=New+York&amp;rft.pub=Springer&amp;rft.edition=
2.+ed.&amp;rft.genre=book&amp;rft.identifier=QA278.5.J65+1986&amp;rft_dat=%7B%22st
drt1%22%3A%22Book%22%2C%22stdrt2%22%3A%22PrintBook%22%7D"></span>
</div>
<!-- Add "opacSaveMode" and "opacBypassMode" opts to Debug urls on the Brief
results page.
This helps with finding test records in WCLocal: Click on a lot of Debug
links then grep the DR logs for your magic string in the saved opac
response files.
-->
</td>
</tr>
```

## [End of Figure 2]

Consider, instead, the use of web components to create three new custom HTML elements. Let us call these elements `<search-result>`, `<resource-thumbnail>` and `<resource-details>`. Figure 3 demonstrates how these elements might be instantiated to produce the display shown in Figure 1. Behind the scenes, these three new elements might still use markup such as that listed in Figure 2, but the verbose, repetitive and confusing code is hidden from the developer inside the web component implementation. As well as hiding the complexities of HTML coding, web components can also hide related JavaScripts and CSS.

### Figure 3: HTML markup required to produce the WorldCat search result in Figure 1, using custom elements search-result, resource-thumbnail and resource-details

```
<search-result item-id="846177884" label="Principal component analysis">  
  <resource-thumbnail resource="oclc/846177884" src="+-+933226193_70.jpg"/>  
  <resource-details oclc-number="846177884" title="Principal component analysis"  
author="I T Jolliffe" type="print" lang="English" isbn="9781441929990"  
rft-id="QA278.5.J65" publisher="New York Springer 2010" source="WorldCat" />  
</search-result>
```

## Web components vs widgets

Prior to the emergence of web components, the creation of custom elements was possible via approaches such as web widgets. The latter is the term used for a small application that can be installed and executed within a web page. Widgets are widely used within library websites to perform useful tasks such as incorporating Google-sourced book jackets (Back & Bailey, 2010), or journal tables of contents<sup>4</sup> in OPAC pages. However, whereas web components are on track to become W3C standards, widgets may be implemented using many different and incompatible methods. Further, whereas browser vendors are producing browser native implementations of web components, this is not the case for widgets.

Another advantage over widgets is that every web component is self-contained. Any CSS styling or JavaScript effects belonging to the custom element will not affect or be affected by the rest of the web page, unless the developer wishes this. It is theoretically possible to ensure this encapsulation with widgets, but the CSS required can become very complicated and the performance of the resulting page may be affected.

## Sharing Web Components

Web components were first proposed by Google (Russell, 2011) but the standard is now being progressed by the World Wide Web Consortium (W3C, 2017). It is not yet finalised but Polymer<sup>5</sup>, the Google implementation of the standard, is a useful half-way house. Until all browsers fully support the emerging W3C specifications for web components, standard polyfills will be required. A polyfill is a piece of JavaScript that ensures support for a technology, in this case web components, in browsers that are not sufficiently modern to provide such support natively.

---

<sup>4</sup> <http://www.tictocs.ac.uk/>

<sup>5</sup> <https://www.polymer-project.org/>

Web components can be easily shared, and the development and sharing of components for many common project requirements has already started. Polymer provides free access to an increasingly large code library of web components. And *webcomponents.org* provides an even larger catalogue of reusable web components from many projects including Polymer. Library-related organizations and open-source communities are likely to benefit greatly from these shared web component libraries. There may be even be potential for such organizations and groups to develop their own collections of library-specific web components at some point in the future. The resulting reduction in complexity of web development may facilitate a wider range of librarians to take greater control of library web development. Further, the combination of web services (Wusteman, 2006) and web components opens up exciting possibilities for libraries. In addition, web components should facilitate the implementation of universal design principles (Riley-Huf, 2012).

## Implications for LIS Graduate Programs

The web development syllabi in LIS graduate programs need to cover the HTML5 standards family, particularly those standards that comprise web components. Although this will add somewhat to the cognitive load for students, these extensions to the curriculum should not move web development beyond the ability of the majority of LIS graduate students. Nor do they risk converting LIS graduate degrees into computer science degrees by another name. The necessary syllabus changes are limited and achievable without their monopolizing the curriculum.

Coverage of the technical aspects of HTML5 standards, particularly web components, fits most obviously into the syllabus alongside HTML, CSS and JavaScript. But their implications will, in time, spill over into other technical topics, such as information architecture. The application of web components offers the potential for a more direct mapping between the theory of information architecture and the practice of web development. For example, the elements, such as <search-result>, listed in Figure 3, could be derived from a system's information architecture, whereas the elements in Figure 2, such as <div>, could not. Thus, web components facilitate the translation of user needs, identified during user research, into technical solutions. This means that the outcomes of student web development projects could become more ambitious, as students use standard web components to create more realistic sites.

With time, implications for other modules could be conceived; for example, modules in digital libraries, digital curation, research data management, systems analysis and design, user experience and interaction design could all benefit from an awareness of standard web components.

## The downside?

As with all technical advances, there are potential downsides to the introduction of web components to the LIS graduate curriculum. It could be argued that it will lead to an increasing focus on existing website patterns, thus stifling individual student creativity. Further, it may encourage students to shoehorn user needs into inappropriate existing solutions. However, a crucial lesson in information architecture which students struggle to learn is the importance of standardization in website structuring, labelling and navigation. Creativity is generally best left to the content. As information architect, Donna Spencer (2010) advises,

"Make your content interesting and your features innovative, but don't try to break new ground with your labels....The best labels are dull, boring and completely obvious"

And the same could be argued for structuring and navigation: students must first learn to follow the rules before they explore how to break them.

## **Conclusion**

To date, there appears to have been no discussion of web components in the LIS literature. But this technology is likely to become increasingly significant in web development. Although web components are still evolving, the concept and the techniques involved are sufficiently stable that an awareness of their existence, plus an exploration of their potential impact, could be introduced into LIS graduate syllabi immediately.

Unlike many programming techniques, the concepts behind web components are not academically challenging. A technically-minded LIS graduate student could learn to create their own web components, thus allowing the student to act as a technology innovator. A less technically-minded student should at least understand how they are used and how to incorporate them in a web page, thus facilitating their role as a technology user.

The technology focus in the LIS graduate curriculum will, no doubt, continue to expand and incorporate new topics, and the discussion will continue as to which topics fit within the LIS remit and which are a step too far. But an understanding of web components should be a core skill for LIS graduate students. And, far from burdening an already bulky syllabus, web components have the potential to simplify it.

## **References**

Audunson R. (2007). Library and Information Science Education—Discipline Profession, Vocation?. *Journal Of Education For Library & Information Science*, 48 (2):94-107.

Breeding, M. (2013). Library technology: The next generation. *Computers in Libraries*,33 (8), 16–18. Retrieved from <http://www.infotoday.com/cilmag/oct13/index.shtml>

BuiltWith (2017). Websites using HTML5 DocType. *BuildWith*. Retrieved from <https://trends.builtwith.com/websitelist/HTML5-DocType>

Cortez, E. (2016). Teaching for the Future, Not the Past. *Journal of Education for Library and Information Science*, 57 (3), 221-225.

Emanuel, J. (2013). Digital Native Librarians, Technology Skills, and Their Relationship with Technology. *Information Technology and Libraries*, 32 (3), 20-33.

Back, G. & Bailey, A. (2010). Web Services and Widgets for Library Information Systems. *Information Technology and Libraries*, 29 (2), 76-86.

Kelley, K. (2015). *The Myth and Magic of Library Systems*. Chandos Publishing.

Maceli, M. & Burke, J.J. (2016). Technology Skills in the Workplace: Information Professionals' Current Use and Future Aspirations, *Information Technology and Libraries*, 35 (4), 35-62.

Maceli, M. (2015). Creating Tomorrow's Technologists: Contrasting Information Technology Curriculum in North American Library and Information Science Graduate Programs against Code4lib Job Listings. *Journal of Education for Library and Information Science*, 56 (3), 198-212.

MDN (2017). CSS3. *Mozilla Developer Network*. Retrieved from <https://developer.mozilla.org/en/docs/Web/CSS/CSS3>

NMC (2014). NMC Horizon Report, 2014 Library Edition. *The New Media Consortium*. Retrieved from <http://cdn.nmc.org/media/2014-nmc-horizon-report-library-EN.pdf>

Potnis, D., Regenstreif-Harms, R., Deosthali, K., Cortez, E. & Allard, S. (2016). Skills and Knowledge Needed to Serve as Mobile Technology Consultants for Information Organizations. *Journal of Education for Library and Information Science*, 57 (2), 187-196.

Ratledge, D. & Sproles, C. (2017). An analysis of the changing role of systems librarians. *Library Hi Tech*, 35 (2), 303-311.

Riley-Huf, D.A. (2012) Web Accessibility and Universal Design: A Primer on Standards and Best Practices for Libraries. *Library Technology Reports*, Chicago 48 (7), 29-35.

Riley-Huf, D.A. & Rholes, J.M. (2011). Librarians and Technology Skill Acquisition: Issues and Perspectives. *Information Technology and Libraries*, 30 (3), 129-140.

Russell, A. (2011). Web Components and Model Driven Views. *Fronteers 2011*, 6-7 October, Amsterdam. Retrieved from <https://fronteers.nl/congres/2011/sessions/web-components-and-model-driven-views-alex-russell>

Spencer, D. (2010). *A Practical Guide to information Architecture*. 2nd edition, Australia: UX Mastery.

Virkus, S. (2015). "Change and Innovation in European Library and Information Science Education". *BiD: textos universitaris de biblioteconomia i documentació*, 35. Retrieved from <http://bid.ub.edu/en/35/virkus.htm>.

Wusteman, J. (2014). Web Development Skills for Information Architects. *UXMatters*, June 23rd. First presented as invited lecture th at Architecture de l'Information - Colloque International, Écoles Normales Supérieures de Lyon, November 19–20, 2012. Retrieved from <http://www.uxmatters.com/mt/archives/2014/06/web-development-skills-for-information-architects.php>

Wusteman, J. (2006). Realising the potential of web services. *OCLC Systems and Services: International Digital Library Perspectives*, 22 (1) 5-9. Retrieved from <http://hdl.handle.net/10197/8071>

W3C (2017). Web Components Current Status. Retrieved from [https://www.w3.org/standards/techs/components#w3c\\_all](https://www.w3.org/standards/techs/components#w3c_all)

W3C (2014). HTML5: A vocabulary and associated APIs for HTML and XHTML. W3C. Retrieved from <https://www.w3.org/TR/html5/>

W3C (1999). W3C Recommendation, HTML4 Index of Elements. W3C. Retrieved from <https://www.w3.org/TR/html4/index/elements.html>