

Drupal Accessibility is allowing this CMS to dominate education and government sectors

By Mike Gifford of [OpenConcept Consulting Inc.](http://openconceptconsulting.com)

Slides available at <http://slides.com/mgifford/csun15>

37% of all .gov websites are Drupal

The entire Australian Govt is adopting Drupal (and are offering this to all all state and municipal websites)

[Web Experience Toolkit](#) is a toolkit to help govt develop accessible, usable, and mobile friendly sites.

27% of all .edu sites (though W3text.com indicates that WordPress has a larger deployment) but of the 100 largest universities 80 are on Drupal. Very prominent in universities in Canada as well.

What is Drupal

A content management system built on PHP, very small, tight core build around an API.

Scalable to very large projects. Large ecosystem and being used for public, crm tools, intranets, for headless Drupal (data engine). Flexible system used for a lot of different things. Popular in universities because of the extensible permission system and a strong commitment to security.

A goal of meritocracy (shared ideas) therefore there are a lot of debates on a lot of ideas. This can mean that it can take longer to make a decision, but generally a better idea results.

If there are problems (such as accessibility) look upstream to see if it's a problem with a library.

Culture Change and Core

Drupal has become a lot more accessible because of the support of the leadership. There are various issues (how fast it loads, security, usability, accessibility) that need to be addressed before code is released. One of the items that held up the release of Drupal 7 was accessibility.

Long term thinking

One of the upsides and downsides of the Drupal release cycle is that it takes a long time to get things out (4 years so far for Drupal 8). But there are two supported versions 6 and 7. Three months after 8 is released 6 will be dropped. Supported means that security updates are released. The advantage for institutions is that they work on longer time scales as well. So a an 8 year life cycle is a decent amount of time for the support of a product. Commercial support is also available (Acquia and Pantheon).

Journey not a Destination

Accessibility is a journey. It's an ongoing effort just like security. We need to continue to work with people to see what works and doesn't work with them. Browser keeps changing. It's great for features, but it impacts user expectations and support. Websites are being used for many many different things (desktop apps, presentations, etc.)

Beyond our silos

Drupal wants people to not think in silos and to think beyond ourselves such as jQuery.UI and CKEditor. With both we've been able to contribute accessibility into them so they could contribute to Drupal. CKEditor's admin interface was improved. PHP 4 support was dropped in Drupal 6 to help encourage adoption of more modern versions. Drupal also is reliant on Symfony. It's interesting how Symfony is shaping the PHP community. There is also work on WordPress and with their security teams.

Drupal 8 from Drupal 7

In Drupal 7 we weren't able to get ARIA in (a little, but not much). But this was fixed in Drupal 8, a lot of ARIA roles and landmarks, and ARIA Live. We're doing as much as we can (labeled-by). We're adding it where it makes sense in context. The feedback so far is that it's good. We're not only removing barriers to accessibility but making it a pleasurable experience, including with an aural interface. I think we've come further than any other CMS to make it a pleasurable experience. We hope it will be released this year, once we've addressed some identified elements. There will be micro releases, like 8.1, so people can see small improvements over time. We also adopted HTML5 (which is tricky, particularly since not all the elements in HTML5 spec are supported by the browsers.) We had to re-write some of the code and needed to ensure that accessibility was still supported. Drupal 8 will look a lot better, and easier for both new & old users.

Getting to the specifics

HTML5, CSS3, ARIA, WCAG 2, ATAG, elements were brought into core (such as HTML5's autocomplete), and error structure (custom errors that are accessible as possible)

Core Themes

Three core themes in Drupal 8 that are very accessible include Bartik, Seven & the sub-theme Classy.

CSSLint

A module to clean up your CSS now checks that hover and focus on par with hover - <http://csslint.net>

Core Systems

As part of creating forms and labels in Drupal, when you create a form using the WYSIWYG and forms API they're accessible. You have to work to make it not accessible. Tables are quite accessible in Drupal 8 now. Caption and summary. Login interface is accessible. Pagination and breadcrumbs, and taxonomy improved in Drupal 8.

Core Modules

The biggest core module is Views. Views is a powerful module that allows you to create lists but render them different ways (bullets, calendar, grid, tables, etc.) It was part of a contributed module, but is now brought into core and therefore the accessibility of it improved greatly. Drupal is addressing the accessibility of the backend (install, administer, and contribute content.) Internationalization (not French OR English, but French AND English). This is complicated especially for accessibility (it's in WCAG, but not well respected.) Drupal 8's not perfect but we're doing a lot better. The help module in Drupal 8 now includes documentation for ATAG – accessibility tool for authoring guidelines), to make it easier for content authors. Using a lot of the HTML 5 fields (phone, email, range, number, search, url, etc.)

Common Needs

We want to make sites and the ecosystem as accessible as possible. With Drupal 8 we are looking to improve Quail.js and leverage things like the WAVE API. We want to include people with disabilities into the open source communities. Implementing ATAG is extremely complicated. Standardization is also a large need (to share best practices, common conventions, etc.) Want to move organizations to a common approach to deal with accessibility issues.

Accessibility by Default

CSS display:none; - it's easy for developers to throw in code, but it's tough to do it in a way that doesn't impact design and accessibility. If you have it in a central location and they know it's there and they can leverage it, that's great, but it takes a long time to get people to know and do this.

Free as in kittens

It takes resources when you use open source, no project will survive long without them.

Why is it being adopted

Because accessibility is hard, but it's a worthwhile effort. Legal risks are daunting. Solid reputation.

Accessibility queue of open issues for Drupal: <http://goo.gl/Dgb7W9>