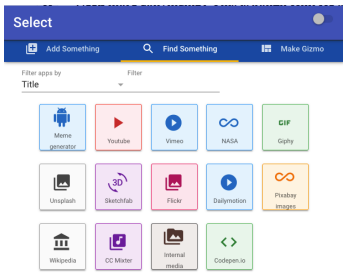
HAX, Web components and the distributed NGDLE future

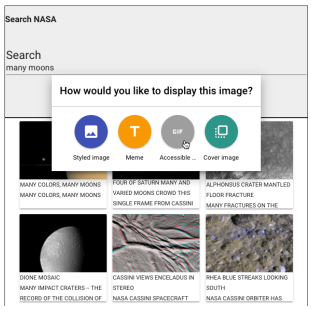
Bryan Ollendyke, M.S.
The Pennsylvania State University
http://haxtheweb.org/

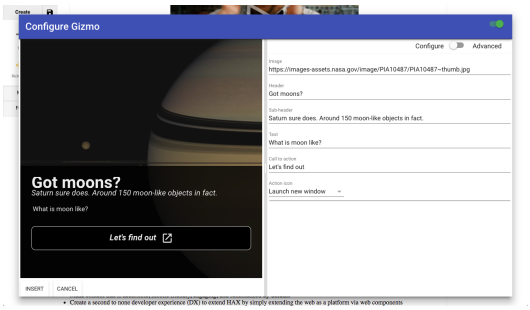
1 Introduction

The Headless Authoring eXperience (HAX) is a next generation WYSIWYG editor built on the W3C web component standard which by design decentralizes content authoring and knowledge production. HAX is a visual building tool that authors HTML that works anywhere but without the need for faculty to understand HTML. HAX can be integrated into existing systems by wrapping HTML content in a simple <hax-body> tag.

HAX allows faculty to build complex, visually engaging content, without a need to worry about accessibility or system lock. Its ease of integration means the ELMS:LN team can build desktop, mobile apps, and ELMS:LN's core authoring experience from the same codebase. It also means that any other platform on the web is able to integrate HAX with similar ease. Developers can extend HAX by augmenting existing web components in minutes. HAX transforms authorship so faculty can focus on instruction rather than HTML in order to accurately represent their expertise.







2 Access

Because HAX can be integrated into any platform, it means that faculty no longer will select solutions based on their own user experience on the content and media production side. This frees solution selection so that faculty can pick the best way to teach rather than just what has the least clicks. Web components operate in any browser based environment and can do an incredibly broad array of things.

As examples, there are web components that are HAX capable to present VR in the browser, reference material from Wikipedia, accessibly and responsively display video, generate self-check questions, present flash cards, layout material, present instructional objectives and more. With HAX, the web as a platform becomes our playground for producing instructional experiences. We can produce new HAX capable elements in minutes when it used to take weeks and guarantee accurate display on any device.

The standard also allows organizations to collaborate and share methods of content presentation and experience development, not just reducing the time it takes to build online experiences but also increasing their quality through collective mindshare. Experiences produced with HAX / web components can focus on instructional experience instead of UI considerations, allowing learners to have a single, familiar cognitive load as to how to engage with HAX / web component generated material.

3 Affordability

The HAX development methodology allows for the more efficient transfer of resources. Because assets unpack client side, this leads to greater server efficiency in delivery on every transaction with a server. This provides greater energy efficiency for servers and time to get access to materials being reduced. For instructors, it also increases the array of options available while removing the barriers of HTML and code in general.

The interface has been built around making faculty and instructional designers job as effortless as possible. As a basic example, utilization of any video from any source into a HAX capable page would take faculty and staff members seconds instead of minutes to produce. Those minutes add up quickly when trying to represent material in a digital form.

Being future proof focused, elements currently produced by the ELMS: LN team are also OER Schema aware. OER Schema increases discoverability of materials by automatically schematizing assets embedded within content. This provides instructional significance to search engines as well as setting up for a more remix capable future. This future sees OER and non-OER leveraging the schema to break apart previously produced works with ease while maintaining instructional and copyright details.

4 Accountability

Cost savings come in multiple forms. Systems fundamentally take less time to develop when HAX is implemented. HAX solves so much of the presentation and authoring aspects of putting media and content on the web, that it reduces what developers need to add into solutions built or bought. It also reduces the strain on faculty and staff to produce high quality online content, currently being outsourced to publishers because of barriers to produce.

The HAX in desktop tool can be thought of as a glorified OER Publishing app for faculty that allow for the easy projection of one's work on any platform. By reducing the difficulty of faculty to author, remix and publish content we can increase OER and reduce costs for students in the form of knowledge purchasing from publishers.

5 Adoption

We have multiple faculty members in two colleges currently using HAX to author content. HAX is the default authoring experience of ELMS: Learning Network, an open source NGDLE that many courses at Penn State use to deliver content and other student experiences. As a result, many faculty and instructional designers have begun using HAX as a result of this platform's implementation.

HAX also has integrations for non-edtech platforms like GravCMS and Drupal so it will allow non-traditional CMS based usage to also be things made leveraging HAX. HAX is headless, as in implemented exclusively on the front-end, so it requires very little resources in order to scale and organically grow in adoption. The HAX desktop app also allows people to produce content offline or in hybrid development workflows by storing content in cloud storage solutions. All of these methods of integration mean that HAX is positioned not just to transform educational technology but the web in general.

6 Organizational Learning

HAX is built on web components which allow developers to share custom HTML elements across projects. It may be hard to grasp, but this effectively means that any web site built with web components is capable of augmenting and extending the visual capabilities of HAX with minimal developer effort. This translates to faculty and staff sharing knowledge in different ways as well.

Anything on the web becomes something that can visually represent information via HAX. Some initial elements for HAX also allow for presenting instructional objectives in visually appealing ways for "fun facts" and other visual offsets. At first glance, this would appear to be giving faculty more visual options but if these visual assets are actually expressing sound pedagogical concepts (awareness, learning objectives, additional resources for exploration) they are effectively institutionalizing the sharing of quality instructional practices through accessible visual design.

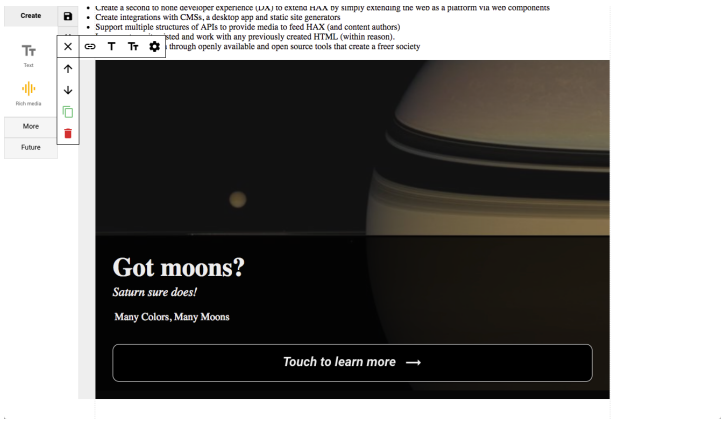


Figure 1 - Focus on content, allow good design to happen

7 Quality

HAX reduces cognitive load both on the part of those producing instructional materials and those consuming them. On the production side, HAX is allowing people to stop thinking about how to convey concepts and instead focus on what they want to convey. This allows for development time to be focused on instructional planning and concept mapping rather than worrying about the time to physically produce and integrate those materials.

For learners, it means that material across the institution will be presented in a high quality, uniform manner. This reduces cognitive load, every video looks like I'm about to engage with a video, every objectives area an objectives area, etc. without eliminating creativity the way a traditional LMS would (HTML templated content is not engaging).

8 Interoperability

HAX leverages web components, a meta-specification of the W3C, which means that any visual or functional asset of a page (example: <video-player>) can work with or without HAX. One of the goals of HAX is to edit things exactly like an advanced developer could and then on save, disappear, leaving no trace throughout content that it was ever produced by the WYSIWYG. HAX achieves this through proper implementation of the web component standard as well as having no reliance on any specific elements or library in order to function.

If an element fires an event on attached life-cycle then HAX is aware of how to talk to it. HAX also leverages JSON Schema in order to dynamically build front-end validated forms. JSON Schema is easy to emit both from elements as well as back-end systems and is a standard. Elements we've written for HAX also automatically wire themselves for OER Schema. OER Schema is an emerging schema.org style extension for machine learning and search engines to better understand the instructional significance of content.

Leveraging these standards and using JSON as a transmission mechanism, we've been able to rapidly integrate third party media solutions with HAX, allowing HAX users to search Youtube, Vimeo, Kaltura, Giphy, Drupal, NASA, Sketchfab, Fickr, Dailymotion, Wikipedia, CC Mixter, and Codepen (just to name a few) directly from inside the HAX editor and quickly integrate content from those sources visually via web components. Early conversations are happening about the potential of web components to help augment or provide a successor to the aging Common Cartridge standard due to their larger standards body adoption, portability and HAX making them easy to produce and visually understand.

9 Innovation

HAX is a first of it's kind decentralized knowledge production technology. It takes CKEditor / TinyMCE style "WYSIWYG" and advances in development 10 years; producing higher quality / accurate code from a developer's perspective, while increasing transparency for developers to build and extend the platform through alignment with existing standards. HAX is a game changer technology because of its ease of integration into multiple platforms and ability to provide a consistent authoring experience across platforms. HAX brings about order and visualization to the NGDLE paradigm plagued by lack of open solutions or actual physical implementations to look to. HAX not only makes NGDLE possible for anyone, it makes it understandable due to how transformative the implementation is.

HAX does not directly influence learning and research theory, but it opens the door to advanced research capabilities by making the web incredibly approachable. Where it can get into these areas is that when thinking of HAX just as a way of editing web components (fundamentally it provides this) then web components don't need to be purely visual. We are working on a <citation-element> which would provide all the copyright details under the hood, that HAX could edit and make approachable. From a data science stand point, a <learning-analytics> element could help ensure that vendors who prefer tracking via xAPI or Caliper could both play nicely while sharing development efforts on the bigger problems then structures of JSON objects.

HAX also represents the beginning of radical reduction in the cost to produce and transfer knowledge between individuals. It's auto-wiring of OER Schema also ensures a bright, open, remix-able, transparent future for the production and re-production of learning materials over the life of the internet. It's adherence to the web component standard also guarantees it will never fall to the wayside like Flash; Web components are here to stay for the long haul of the internet as a platform.

The implications for HAX extend beyond traditional educational technology into any web-based application, desktop (via Electron) or mobile app (via Ionic). Brigham Young University has also expressed the potential of HAX, if taught the right elements, to become website-tonight software the likes of WIX or Squarespace, but completely open source and decentralized. We've made the web easier to approach, construct and consume. Because of this, faculty and staff training on how to use and build complex content is incredibly minimal, seconds even. The reactions to HAX from all user bases, advanced developers down to those afraid of the web can be summed up in one word: stunned.