



Ode to Our Forgotten Users

Drupal and the content authoring experience

Kristina Bjoran

Content Strategist, Forum One



My name is Kristina - I'm a...well, I'm a lot of things at Forum One. My primary role, however, is content strategist. Which means I get to work closely with our clients to sort out content messes, and often find myself introducing them to Drupal.

Stephen Lucero

Technical Architect, Forum One



who have we forgotten?



Users come first. User experience design, and user-centered thinking has been an important framework for web development, and rightly so. We're right to try to empathize with users, meet their needs, and make sure their experience is a positive one (assuming a positive experience is the goal. For some brands and sites, I'm not convinced...)

Most of the time, though, we're talking about end-users. And while end users are of course important - no argument there - we're leaving out someone pretty important. Content authors.

In our eagerness to accommodate end users, we have, as a result, largely forgotten our content authors. And with Drupal, that's a real problem.

In our day to day, we find ourselves working one-on-one almost exclusively with non-technical clients. These are folks who have either been thrown into a "webmaster" role, or maybe they come from communications backgrounds and often have limited web experience. And since we do a lot of Drupal work, I've grown accustomed to easing these new, non-technical clients into Drupal, because the overwhelming amount of functionality available to them (even with limited permissions), is confusing, and frankly, just not very pleasant.

The point here, by the way, is not to just poop all over Drupal. We're not here to point fingers or complain. This isn't anyone's fault. The Drupal community has focused on building a powerful, flexible, complex CMS, and with some love and attention, there's

no reason we can't turn it into a pleasant experience for content authors, too.

authoring experience (AX)

new and improved buzz phrase!



But before all that, it's probably a good idea to take a step back and survey the land.

We're not the only ones talking about authoring experience. Particularly with regards to CMSs, folks are starting to turn their attentions to accommodating our forgotten users.

There's a real smart guy out there talking a lot about authoring experience in general (not particular to any one CMS), and I highly recommend you check him out. His name is Rick Ya-godich, or [@think_info](#) on twitter. His book, *Authoring experience*, is amazing, and you should read it. And if you want to check out more after this talk, be sure to hunt down Jeff Eaton and Karen McGrane too. They've all done some great deep dives on the matter.

Anyway, let's get some definitions out of the way..

When we say "author," we mean a person using a tool to create some kind of content that will be published. That person is often not a developer, and was hired to handle content and generally manage the website.

We should also always keep in mind, while we're talking about all this, that the primary goal of publishing content is to communicate. An author's job is to communicate. So, it stands to reason, that a good authoring experience leaves an author feeling empowered to communicate with confidence and with ease.

Drupal and AX

the good, the bad and the ugly



- Historically Drupal hasn't been known to have the best AX
 - forms can be overwhelming
 - process can be overly technical
- This is one point where Drupal suffers from being more of a content framework than a pure CMS
 - We have the tools in place to build any content structure we need, but with that flexibility there is a loss in enforcement of principles
- Without intentional effort put toward the AX, the author can easily be forgotten in favor of more flashy features on a site
- As frequent users of Drupal we've become a bit desensitized to many of these common problems as well. When exposing a new user to these interfaces it's common to find that the things we've become familiar with aren't necessarily intuitive.
 - To help highlight this I'll turn to Kristina to walk us through a common use case

[High level Drupal difficulties. What's currently confusing / overwhelming. (focus on Drupal 7)]

so what?



There's less and less resistance to these ideas in the Drupal community. But it bears repeating: the future of Drupal depends on an improved authoring experience. Because, in the end, we're competing to be the best CMS. And in order to be the best CMS, authors need to *want* to use it. We want them to get excited about content authoring, about communicating with their sites and tools.

I want my clients to get excited about using Drupal. I want them to feel freaking awesome about what they create.

And we're positioning ourselves to work toward that goal. But there's still work to do.

Before I turn it over to Stephen for the juicy bits of this talk, I'm going to walk through a couple of examples of good, positive content authoring experiences. Who's doing it well, and what makes it good?

And most importantly, how can we apply those best practices to Drupal?



So what does good content authoring look like? Who's making content authoring positively delightful, to the point that people actually WANT to create content?

This is how we're defining good versus bad. In the simplest definition, great content authoring experience means:

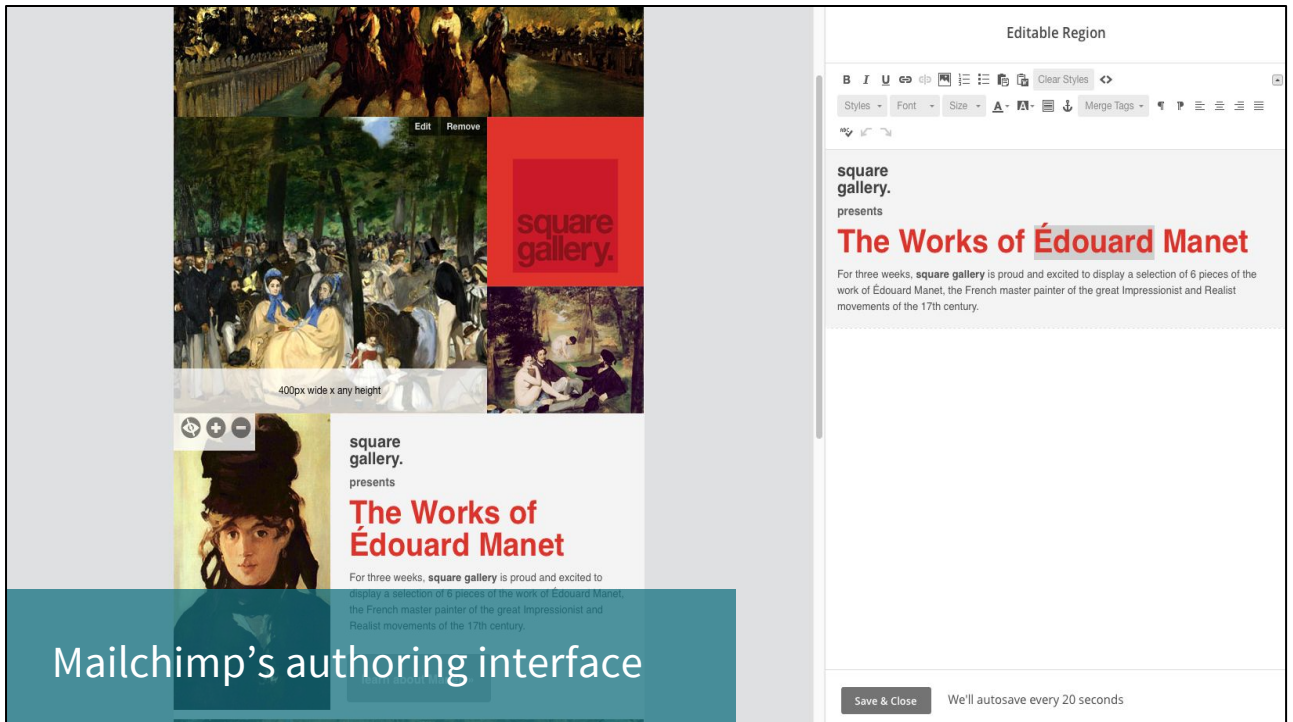
- authors can do what they need to do
- authors feel good about what they're doing when they're doing it
- authors feel good after content is published

Some folks are nailing this. The following examples aren't CMSs, but the principles behind the content creation are ultimately the same, and we can learn from them.

MailChimp



First is my favorite. MailChimp. They did the impossible - they made creating and managing email campaigns not only easy, but, honestly, kind of delightful.



This is their part of their authoring interface. The whole creation process is worth discussion, but I'm focusing here only on the actual screen for building the email.

MailChimp takes the stress out of figuring out a tool, and throws that out the window in many ways, allowing content authors to focus on the hard part - building a nice looking email.

They use clear language, never throwing any jargon at the author. And they recognize that sending emails out to a bunch of people can be stressful (to say the least), so they do their best to speak with the right voice and tone in order to ease that pain.

It's a one-two-three punch:

- Users can see what the content will look like while they're creating it (or, contextual editing)
- Through their use of language, users feel like they're being guided through a process.
- Users are never bogged down with the crazy technical gymnastics that allow MailChimp to be as powerful as it is.

So it shouldn't be a big surprise that MailChimp has as many brand evangelists as it doesn. I've yet to meet someone who hates MailChimp.

GatherContent



GatherContent is another personal favorite of mine. It's a little bit different than others on this list. GatherContent acts as what I like to call a "pre-CMS." It's a tool that allows you to build out the structure of a page, and use that structure as a form in order to plan out and collect content for a website.

Really, it has two layers of authoring (and it's nice being on both sides).

The screenshot displays the GatherContent authoring interface. On the left is a dark sidebar with a 'Template name' section showing 'DAI Longform Chapter Page' and a note 'This template is active on 7 items.' Below this is a 'Fields' section with a list of field types: Text, Attachment, Guidelines, Checkboxes, and Multiple choice. At the bottom of the sidebar is a 'Description' section with the text 'Standard template for a DAI chapter page.'

The main area on the right is the editor, titled 'Content' with tabs for 'Figures & Images' and 'Add a tab'. The editor shows a structured layout of fields:

- Title of page + Subtitle of page:** A header field with a rich text editor toolbar and a word limit of 0 words.
- Big hero image:** A large image field with an attachment icon and the instruction 'This is the full width image that will lure readers in.'
- Most relevant for?:** A field containing a bulleted list: '[keep these brief]' and '[keep these in bullets]', with a word limit of 0 words.
- What will readers walk away with?:** A final field at the bottom with a rich text editor toolbar.

This is what the first author experience looks like - this is how an author builds out the structure of the page.

It's super simple. There's a very basic (read: purposefully limited) grouping of field options. Authors can drag and drop the field types into the form they're building, and choose a few settings for that field.

At this level of authoring, GatherContent succeeds by:

- empowering authors to do what they need to do, and do so quickly - including being able to customize the names of the fields
- real-time content building - there's no need to worry about what this will look like when done - it will look exactly the same
- and finally, keeping unnecessary options and menus out of the way. The content the author is creating is the only thing that matters on the screen.

Save

*****Hi! Read Me First!*****

Front Page

Overview

Stakeholders

Regulatory Issues

Strengthening Ecosystems

Using Digital Finance in USAID Operations

Programming

DFS by Sector

Resources

Model SOWs

Market Diagnostic Tools

Contents of a Solicitation for a Project Using DFS

Content

Figures & Images

Title of page + Subtitle of page

Big hero image

Upload file(s)

This is the full width image that will lure readers in.

Most relevant for?

What will readers walk away with?

Introduction to section

A central message of this handbook is that a digital financial services ecosystem is a key means to many ends. In addition to helping to expand financial inclusion, it can also be used as a channel to achieve other development

The second layer is similar, but even cleaner. At this point in the process, GatherContent is serving to...well, GatherContent. Because of that, they remove any other visual stimuli so that content creators focus only on the fields at hand. It's not asking for anything from the author, other than the author's primary task.

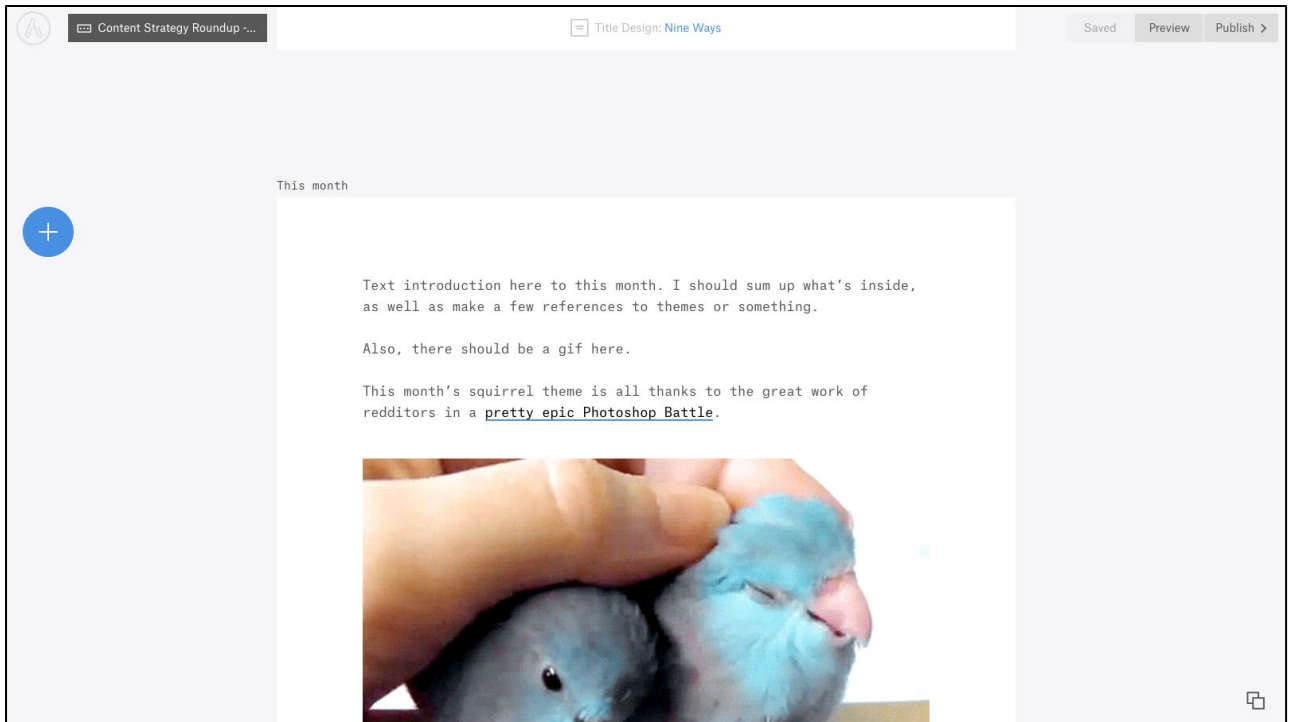
It does a fantastic job of just staying out of the way.

Atavist



The Atavist is...just fantastic. While on the surface it's just an article builder, what you can essentially do with it is create your own self-contained magazine. It's pretty similar to Medium, but Atavist is more super-charged. And in my opinion, much cooler.

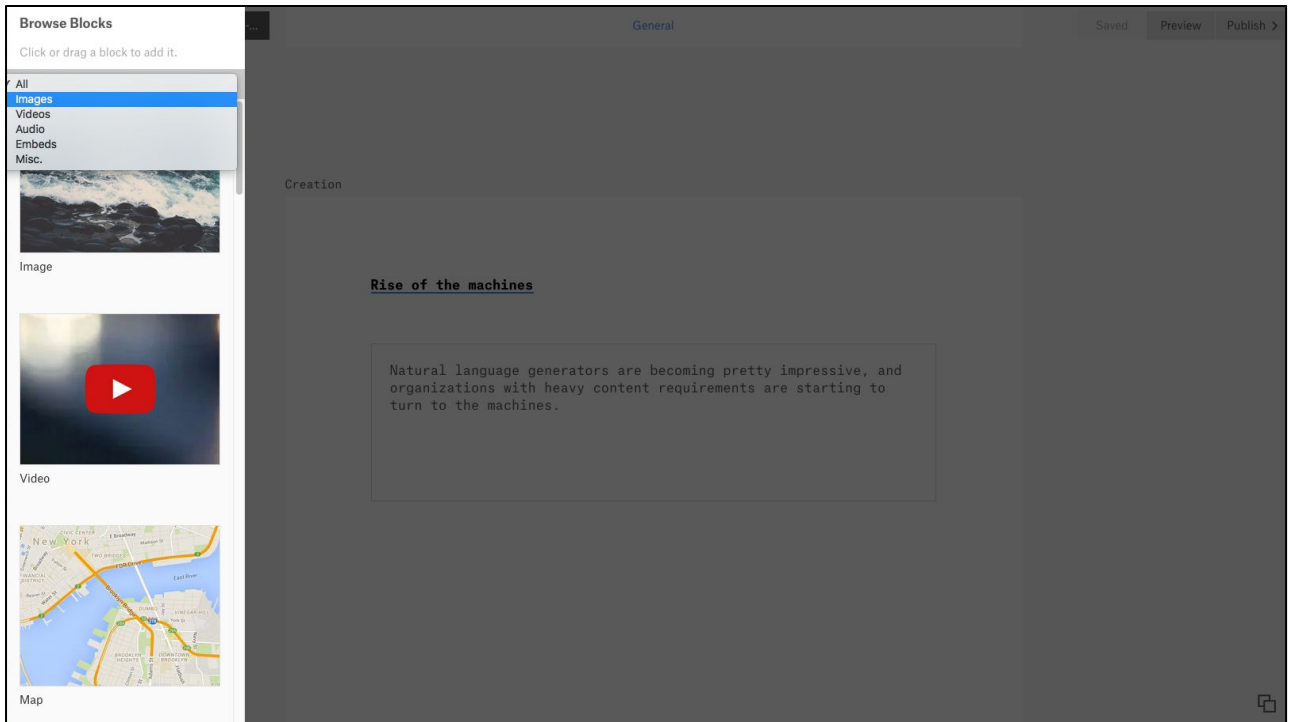
And sure, it's nothing so complex as, say, a 3000 page Drupal website with lots of taxonomies and complex data models. But it still provides an experience we can learn from.



Here's an example of their editing interface. First off, it's just clean. It feels nice. And the default content authoring styles (like font size, image treatment, etc) are pleasant, which makes content authors feel like what they're creating is quality.

Like gathercontent, it stays out of the way. Way out of the way. Menus are hidden until you seek them out, all in favor of the meat of the work.

And a big one - it inspires confidence. The admin side of this content looks largely like it will when it's published. There are no major styling surprises.



And ease of use. This thing is so easy to use. Those menus, the ones that stay out of the way, once they're revealed, they provide a series of content blocks that can be dragged into the document. They call them blocks, and they do what the Paragraphs module does, but do so in a more constrained, explicit way, which, in this case, makes for an empowered author (even though many configuration options are in fact, hidden).

best practices



Now, I could list off nice experiences all day. A few other notable examples that come to mind are Medium and Squarespace (and even TurboTax), but honestly, we continue to find similar patterns.

Get out of the way!

Strike a tone

Constrain (but don't be obvious)

Good content authoring experiences do these three things.

They get out of the authors' way by empowering them to focus on their content

They use clear, relevant, nontechnical language in their designs, and sometimes go as far as to have a conversation.

And they provide a lot of constraints by hiding unnecessary settings, menus, and configuration options.

With these things in mind...



So, about Drupal...

We do a lot well with Drupal. A lot. And on the near horizon, we've got a lot more coming.

The not so bad



We currently have all of the tools to deliver a solid user experience, but much of the responsibility to carry across a good AX is left to the site builder. These tools are well defined even if they're not well known, so an important step forward is familiarizing yourself with them.

- [Well-defined UI standards](#) (even if not particularly well known)
 - Drupal core sets a good example
 - read standards and apply rather than mimic
- Flexible content structure options
- Extensive form API
 - Standard widgets readily available
 - Ability to create and share new widgets
 - Implement and follow modern interface standards via contrib

What we have now



Well-defined UI standards



<https://www.drupal.org/ui-standards>



Extensive form API



Community support



- [Well-defined UI standards](#) (even if not particularly well known)
 - Drupal core sets a good example
 - read standards and apply rather than mimic
- Flexible content structure options
- Extensive form API
 - Standard widgets readily available
 - Ability to create and share new widgets
 - Implement and follow modern interface standards via contrib

What we can do *now*



So, awesome - it sounds like we do have some capacity to deliver a better authoring experience. How? What can we do?

In core now



Write field labels and help text



Hide the unnecessary



Configure shortcut links



Order related fields intuitively



- Field labels and help text
 - Content type descriptions
 - Field labels
 - Field help text
- Hide the unnecessary - limit permissions
- Configure shortcut links - Toolbar and shortcut modules
- Order related fields intuitively - Field UI
 - Enter most important or best-known content first

With some contrib help



Hide unnecessary configuration



Arrange fields in groups



Create and expose contextual links



Structure with reusable content pieces



- Hide unnecessary configuration
 - Simplify module
- Arrange fields in groups
- Create and expose contextual links
 - Views - 'Include contextual links'
 - Custom contextual links module
- Structure with reusable content pieces
 - Field collections
 - Entity references
 - Embedded forms
 - entity relations and creation sequence

What's coming soon



Well that's all exciting. And I know Drupal 8 is just around the corner and its Authoring Experience has received a lot of attention. So what do we have to look forward to there?

New in Drupal 8



WYSIWYG editing



In-place field edits



Reorganized edit forms



Responsive by default



Out of the box

- WYSIWYG editing
 - CKEditor built into core
 - Built-in filtering to remove common stylesheet overrides
- [In-place field editing](#)
 - More intuitive workflow for authors making quick edits
 - More obvious separation of content and configuration
- Improved content editing forms
 - [Multi-column form layouts](#)
 - Consolidation of related components
 - More visual focus on most important elements (content)
 - Consolidated publication options alongside save button
 - Important actions are all grouped together
- Responsive out of the box
 - Responsive administration
 - Responsive toolbars
 - Mobile-friendly editing

Still a lot of work to do...



We're clearly on the right path. But there's still a lot of work to do. And we need everyone in the community to pitch in to make Drupal genuinely delightful to use.



So you want to help?

First and foremost, everyone can contribute by:

- Weighing in on feature discussions in the issue queues
- Test new patches and features
- Report bugs and request new features in issues

Authors, you can help by:

- Complaining! Complain often!! Let everyone know about pain points. **Don't assume you're wrong or stupid**
 - Report bugs
 - Request new features in issues

Developers:

- Tackle the issue queues
- Update existing modules
- Create new modules for missing workflows and features

Drupal needs our help. Let's make Drupal the best CMS for everyone.

A black and white photograph of a dark studio. In the center, a crab is wearing a black top hat. The background is a dark wall with several spotlights hanging from the ceiling. The floor is a light-colored, reflective surface.

THANK YOU!

Kristina Bjoran
@bjoran_identity
kbjoran@forumone.com

Stephen Lucero
slucero@forumone.com

Photo sources, in order of appearance

- [frustrated writer](#)
- [LoZ tshirt!](#)
- [Redditor /u/how2post](#)
- [happy writer](#)

