

# What is a Pattern Library?

A pattern library is a collection of reusable patterns that solve common interface problems. This to solve the problem of ensuring that Drupal's interface is of high quality and usable with hundreds of contributors. Pattern Libraries have emerged in early 2003, because other solutions such as approval processes, style guides and standardized templates didn't succeed in the web application space<sup>1</sup>.

"A design pattern is a repeatable design solution, that's been tested, reviewed, and verified"  
– James Refell, UI Design Manager, Ebay Inc.

For Drupal there is a large unknown factor, when we started reworking the interface of Drupal 7 we were immediately faced with the modular space that Drupal lives in. Which means that any module can drastically change and/or introduce new interface elements. Where any typical Drupal site has at least 30 modules installed, the experience of using Drupal changes. We learned by field-testing, that certain interface patterns although working well in isolation – break as they get applied by 5 or more modules at the same time.

Developers started to request a pattern library in early 2008 to accommodate their need for helping hands during the build of an module it's interface. While we invested a lot of time in assisting these developers, we started to see patterns in their needs – which most of them where being addressed by Drupal 7, but not applied consistently.

## Proven interactions

We have done several usability tests on Drupal 7, which now includes a pool of over 30 users we have observed. As we have tested several patterns, we were able to determine the optimal patterns – for many use cases. For the pattern library that we wish to build we are going down the path of only providing interactions that have been proven in usability tests, this to assure a high quality and avoid design guesswork.

## As intended style

When deciding what kind of pattern library we would build, it was important to recognize that we make tradeoffs in applying the patterns in Drupal 7 for specific reasons. However we intend, contributed modules to still follow a preferred optimal experience. As portrait in Nathan Curtis his article<sup>2</sup> there are three groups; As-Is (how it's now), As-Intended (how it was intended), To-Be (how it should be).

Since we will update the patterns as Drupal 7 rolls out, to fill certain information needs and to change certain interface standards. We need to have a form that accommodates this, the As-is form only accommodated the Drupal 7 release point, whereas we would change patterns beyond that. And the To-Be form, is inappropriate for the development process of Drupal

---

<sup>1</sup> [http://www.uie.com/events/roadshow/articles/design\\_patterns/](http://www.uie.com/events/roadshow/articles/design_patterns/)

<sup>2</sup> <http://www.nathancurtis.com/2008/12/31/what-site-does-a-component-library-reflect/>

# Audience

While researching other pattern libraries I was faced with the fact that our Pattern Library was far more a software pattern library, than a general website/application library. Thus I interviewed 14 people from this space for about 30 minutes - 6 Module maintainers, 4 Module users and 4 User experience designers. These took place at an local Drupal camp and remotely over the internet. I varied the conversation, beyond the interview I set out with.

My main focus for the interviews was asking the following four questions :

- When you build an software it's interface, what is your process?
- What are the biggest problems or inefficiencies in this process of building a modules UI today? (only Module maintainers)
- What *\*things\** do you run into using different modules their UI in congestion?
- What interface elements do you tend to use all the time? (in Drupal if applicable)

## Results

The audience I projected is far more diverse then I first estimated, especially the different Module maintainers somewhat surprised me. It also alarmed me that I wasn't interviewing key stakeholders for the build of a pattern library, I held 6 more interviews with the API documentation lead, documentation lead, Drupal 7 maintainer and three Drupal core developers.

From the interviews I was able to determine 4 audiences ; module maintainers, core developers, module users and user experience designers. Within user experience designers I saw no substantial difference in the informational needs of a pattern library between information architects, graphical designers or usability experts. Core developers their informational needs seemed to outgrow the module maintainer group, so I took them as a separate audience.

### *Module maintainers*

Every module maintainer had a specific way of developing their UI, four of them spoke as first working on a code and then granularly getting to interface elements and at the end binding them together. The other two expressed an approach where they would focus mainly on interface facing elements at first, to test different code approaches.

All expressed that most of their UI work evolved as they would release their module on Drupal.org. The common reason for working on the usability of a module, is to decrease the amount of support requests. Especially two developers who worked on somewhat popular modules (30.000+ users) explained that they have a severe need for standardized interface elements – they spoke how other modules would make their module unusable.

A few patterns came forward, the listing page and normal form elements such as select list, radio and auto complete which were applied constantly.

One actually expressed the need for a pattern library, since he felt that he was not an expert at building an interface – but wanted to meet his interface to a certain standard. As I asked more detail, the underlying motivation seemed to be, uncertainty of the current interface standards in Drupal Core after seeing all the usability issues with it.

### *Core developer*

Core Developers form a somewhat special group, they focus mainly on developing for Drupal core – therefore their needs are somewhat different. Drupal 7 will be the first release that will have consistent usage of interface patterns. This was expressed as one of the most important changes in terms of usability by core developers.

They saw the need of improving and developing new patterns as the main purpose of working on the usability of Drupal. The process of developing a new interface, was mostly going from existing testing data or in field experience – to sketch and outline the given solution and take that to the Drupal development process.

A major concern was how the growing number of contributed modules would influence the interface, in all three interviews – they tried to explain how the interface needs to be able to scale, and how our current interface elements don't really accommodate that.

### *Module users*

There has already been extensive research on module users<sup>3</sup>, I interviewed mainly users who have broad experience using Drupal. A trend quickly came forward, which we had also seen in the other research which is basically the overall frustration of using modules in congestion with each other.

What *\*things\** do you run into using different modules their UI in congestion? Was the most prominent question in these conversations. One of the things I learned is that data rich pages such as listing pages and tabs were often pointed out as a place where the interface got flooded by modules.

I learned more about the perception of different Drupal interfaces by users. But I think most of these experiences are already expressed in previous research<sup>4</sup>.

### *User Experience Designers*

These participants had some experience using Drupal and where both senior interaction designers. Their process of designing software was very research driven, and resulted in an pixel-perfect end deliverable that would be used for the design. They haven't designed anything for Drupal yet and where intrigued by design challenges of Drupal – due to the recent activities.

They weren't able to answer any specific questions regarding Drupal, therefore I decided to talk to them about pattern libraries. One shared a particular interest in the impact it could have on the Drupal community. Explaining to me how he used it in a previous job, taking the pattern and laying it against his interface. He spoke how it wasn't merely a comparison tool,

---

<sup>3</sup> <http://beccascollan.com/?p=159> – Drupal developer personas

<sup>4</sup> <http://drupal.org/usability-test-university-baltimore-community-solutions> - Usability test Baltimore 2008

it was most notably a convincing tool against his team of developers – that this was indeed the industry standard they should follow.

The other explained how the use of an interface library would restrict people from thinking of new interfaces – going with and slightly changing existing patterns, where a new pattern would be more fitting.