Day 1 Session Highlights

Keynote Speech by Dries – The takeaway points

Drupal 8 will be designed using the “Mobile First” concept and it will harness the Symfony PHP framework in its Core framework redesign.

- Web consumption is predicted to be heavily mobile – expect 25x jump by 2015. That is less than 5 years from now!
- In order to remain innovative and gain more market share, the Drupal community needs to be ready to tackle mobile-related web development issues and concerns
- Drupal 8 will be designed using mobile first principles and strategies. The core themes will use HTML 5 and be responsive in design out of the box.
- In order to improve performance problems that hinder mobile development on Drupal, the framework is receiving an upgrade: It will be incorporating the Symfony framework. Discussions are still ongoing as to how much of the Symfony framework will be incorporated into Core.
- Integration with Symfony is similar to Core's decision to integrate JQuery for frontend enhancements
- Scheduled release for D8 recently released: Design + Analysis phase is expected to end around May; July and June will be prototyping, mockups, ui + feature decisions; Implementation expected to be completed at the end of November with a Feature freeze in December. Code freeze is scheduled for February. D8 is expected to have a stable release by August 2013!!!!

Video of Keynote Speech: http://bcove.me/ehw8jepn

Session 1: Responsive Web Design: The past, present, and future by Lewis Nyman

Responsive Design should not necessarily mean desktop browsers vs mobile devices, or Blackberry vs iPhone. Responsive Design means responding to different sets of conditions starting with the base, barebones version and progressively enhancing upwards. We all need to participate in the W3C standards that are currently being developed and push browser vendors (desktop and mobile) to adopt them.

- Twitter: @lewisnyman
- First of all, mobile first! This is the dominant phrase. Design the site based on the baseline experience. This means, figuring out the hierarchy of the content (text), basic navigation items, and a BLACK and WHITE version.
- Do NOT use display:none as your responsive design tool! Helpful blog post on progressive enhancement on Planet Drupal
- Starting with the barebones version first gives you flexibility to progressively enhance the experience depending on the capabilities the user has available as s/he visits your site.
- Use Xcode, Opera, and Android simulators for testing

What are some of the different conditions that a site can respond to?

1. **Fluid Grids**: The most obvious condition is with viewport size – you should use media queries to change the layouts for larger and larger viewport sizes (rather than the opposite)
2. **Fluid Images**: Smaller images should be loaded for a small viewport size; a larger image can be served for a larger viewport size. This saves bandwidth – especially important for mobile devices.
3. **Conditional Loading**: Another way to reduce bandwidth waste –
   a. Load different content based on the width using lazy loading by javascript. Example, a site has a sidebar that has a header titled “More about Cats”. For a small viewport size, load only a link for the google search for the keyword “cats.” For a larger viewport size, load the list of search results returned for that query.
   b. Load different resources by relevancy: geolocation resources if relevant, rounded corners for elements if the webkit can render
4. **Touch**: You should not design your “mobile” site for only touch devices! There's also non-touch mobile devices such as Blackberry. So large buttons for links that make it easy to click on an iphone take up unnecessary space on a Blackberry. User modernizr.touch to only load touch screen styling when its needed.
5. **Number of Thumbs Available**: If the user is looking at your site in portrait versus landscape mode, this opens up different opportunities for navigation. Use @media orientation:landscape to detect if you want to experiment with 1 thumb versus 2 thumb experiences.
6. **Network**: W3C is creating a standard to determine the network speed (Network Information API). Just because the network is slow does not necessarily mean a small viewport on a mobile device. It could mean a desktop browser using a USB wifi connection. We need to always assume a slow connection first and then conditionally load using the Network Information API to provide a richer experience. Currently you can only work with Android with this API.
7. **Color**: Grayscale does not necessarily mean a lowend device. Kindle does not have color! But that's on purpose. Designing a site first
with the limitation of just black and white. Then you can make sure the site works in terms of visual hierarchy and contrast when the user
is limited to just grayscale. This also helps with accessibility as well and a plus is that grayscale pixels translate into less data. Use
@media color (instead of monochrome) to progress upwards.

8. **Battery**: How to respond to the condition of low battery. If a user is visiting your site, then they will not want to waste time having to load
all of the bells and whistles if they just desperately need information (not experience). W3C is working on the Battery API to detect to if a
device is charging, low battery, etc so that you can serve different content based on this condition. Not implemented by any browser
vendors so it is up to us to let vendors know we want and need this.

9. **Ambient Light**: We need to provide high contrast in high light situations. In low light conditions, we should switch the display of text to
light text on a dark background to reduce eye strain. Native OS such as Mac laptops already detect and implement these reactions
(Keyboard backlight, dimming screen to reduce eye strain in low light situations, etc). W3C is working on the Sensor API to detect
ambient light, motion or temperature.

---

**Video of Session**: http://blip.tv/drupalcondenver/responsive-web-design-the-past-present-and-future-6036605

**Responsive Design BOF**

Hey! This stuff is time-consuming and hard. But we got to make a serious effort because mobile will soon overtake desktop browsing. Plan for setbacks, reverting to old, tried and aging ways, and pushback from clients because of the increase in time and effort ($$$$.)

- Making a responsive site will take a minimum of 20 – 30% more time and effort. This percentage might decrease as you do more sites in
  this way.
- Mobile is not separate from Desktop.
- Do not hide stuff with CSS – develop/design with the lowest baseline, upwards
- The best testing is with actual devices – make friends. Simulators offer a good first pass, but there’s so much you will discover is not
  working with the actual device
- Don’t waste too much time with your wireframes and mockups. Do an actual HTML/Drupal mockup even if it isn’t pretty. Playing around
  with the layout changes and different functionalities and navigation will prompt better feedback from your client and help them understand
  the opportunities and challenges of responsive design
- If you fall back to your old ways for desktop first then mobile....it’s okay, just try to focus on small changes utilizing mobile first. This is
difficult, time-consuming stuff and you have deadlines for deliverables!

---

**Practical Responsive Development in Drupal**

Use SASS + Compass CSS preprocessing to handle your responsive design. This talk focused on bow to clean up your markup in D7 and techniques for creating interesting layouts that break out of the pattern of responsive design layouts.

- Mobile first = Content first = user first
- Responsive Design ALREADY has “typical” patterns: http://www.lukew.com/ff/entry.asp?1514 – Multi-device layout patterns. This is so
  boring! We need to avoid following this typical layout.
- Drupal 7: Field and Entity APIs means that we can provide “responsive content”
- Responsive content example: www.palantir.net/experience
- The field tpl file is awful! The markup is way too heavy because it assumes the worst-case scenario of: Multi-valued field with Label and
  plain text output. This resulted in several nested divs and no semantic markup. That’s where the module Fences comes to the rescue:
  http://drupal.org/project/fences. Fences provides a way to add semantic (HTML 5) markup to your field. It works very well with Field
  Collections. Refer to theme hook suggestions http://drupal.org/node/1089656 to understand how this module works.
- Layout Markup: Use Custom Cools Layouts – http://drupal.org/node/495654 to define your regions for fields etc (using Display Suite)
- Responsive Images: Take a look at the Adaptive Image module (http://drupal.org/project/adaptive_image) which is based on Matt
  Wilcox's Adaptive Images technique. There is also another module called Borealis (http://drupal.org/project/borealis)
  For more information about: what is the problem with responsive images: http://cloudfour.com/responsive-imgs/ (by Jason Grisby)
- Natural vs. unnatural breakpoints: Don’t think about device dimensions when picking breakpoints; do it based on content.
  (www.palantir.net/blog/re-thinking-breakpoints-responsive-design)
- Layout Buildign techniques: Adjacent sibling rule, The opposing float, The lasso technique, The corset variant, The absolute exception (to
  the adjacent sibling rule), The Violator

---

**Video of Session**: http://blip.tv/drupalcondenver/rethinking-responsive-building-techniques-with-drupal-6036654

**VOIP Drupal BOF**

A Mobile platform that focuses both on “dumb” phones (like mine! – I survive on texting
for info to Google), “smart” phones, and web-based phone calls (Google Voice, Skype, etc). The point is that it is device agnostic. You can contact (call, SMS, record audio) with a dumb phone, smart phone, desktop web browser all with the same result.

AWESOME work done by MIT (Center for Civic Media) to create this platform of 20+ modules. You can launch campaigns (call bursts), call backs, click (link) to call, automated answering service (“press 1 for hours, press 2 for billing, press 3 for general customer service”), and much more!

Project page: http://drupal.org/project/voipdrupal
Documentation: http://drupal.org/node/1078710
Sandbox: http://www.voipdrupal.org/

- I have a Fact sheet that I need to scan that has a lot of information
- The current supported VOIP providers are: Twillo, Tropo, Plivo/Freeswitch.
- It is possible to implement an audio tour where a user can call or text a VOIP number and be served info by your Drupal site.
- It is possible to implement an interactive call where a user calls the VOIP number and can interactive with the site at the same time while the call waits for the interaction. Think interactive phone-based tutorials

Fact Sheet:

You need flash player installed to preview ppt and pdf files

[VoIP Drupal Fact Sheet!^Voip_Drupal_factsheet.pdf]

This is related to my experience in the Tropo Hackathon because the idea we worked on was an Emergency Response Buddy System using Tropo to support SMS. The goal is to mitigate the overflow of calls to a university in times of crisis. In an emergency, the site would mass SMS all registered users with an emergency response message. The message might be simply informational OR it might require a response from the user: "Are you Ok? Text back 'Yes'". For the situation where the user is prompted for a response, the buddy system will be activated. If the user responds back with "Yes", then the site will send SMS messages to all buddies of that user notifying them that the user is ok. Users create Buddies with information like: name, phone number, relationship. Tropo (SMS) is used to verify both the user's own phone number and buddy acceptance/rejection ("Person A registered you as their emergency contact. Text back:'Yes' to accept"). "Campaigns" (a separate content type) are used to distinguish the different emergency notifications. Each campaign is auto-assigned a different Tropo number upon creation. The assumption: users already have an account on the Drupal site.

Drupal contrib modules:

- http://drupal.org/project/tropo
- http://drupal.org/project/sms_tropo
- http://drupal.org/project/smsframework
Overview

Are you working with communities that do not have easy access to the Internet? Would you like to provide your clients with automated phone access to their orders and personal information? How about organizing a phone and SMS campaign in your neighborhood?

VoIP Drupal is a versatile open source communication toolkit that adds the power of voice and Internet-telephony to Drupal websites. It can be used to build hybrid applications that combine regular touch tone phones, the Web, SMS, and other channels in a variety of ways, facilitating community outreach and providing an online presence even to those who are technologically challenged, or who do not have regular access to computers.

Although VoIP Drupal has been used primarily for voice and SMS-based communication, its architecture can be easily extended to support Twitter, IM, email, and other media. Sample code and a sandbox to help developers play with the different aspects of the platform may be found at http://voipdrupal.org/

Key VoIP Drupal features

- A plug-and-play interface (API) that interoperates with popular Internet-telephony providers, dramatically reducing the learning and development costs associated with the construction of unified communications systems. VoIP Drupal currently works with the Tropo and Twilio services. It also supports the open source PiVX (FreeSWITCH) framework, which enables you to run phone service in developing countries and in areas that are not covered by other telephony companies.
- A simple dialplan scripting language for the creation of interactive phone calls featuring voice menus, conference calls, speech generation, SMS handling and more.
- Support in the dialplan scripting language for hybrid interactions combining voice and text communication channels. For example, a user may send a text message to the system and receive a voice call in return. This feature is particularly useful in regions where phone calls are expensive or where a large number of users subscribe to pay-per-use phone programs.
- Portability across different VoIP services that provide different features, programming languages, coverage, and pricing structures. VoIP Drupal enables developers to program once for all those different services without having to learn their specific languages.

Applications

Potential applications of the VoIP Drupal platform include:

- "Get Out to Vote" campaigns
- 2-1-1 and 3-1-1 community hotlines
- Call centers
- eCommerce phone and SMS access
- Phone- and SMS-based surveys
- Group communication
- Story recording / playback
- Group voicemail
- Audio speed dating services
- Language training
- Audio tours
- Adventure games
- Interactive community radio programs
- Emergency announcements

Key benefits

- Facilitates the construction of unified communications systems integrating SMS, email, Web, and voice
- Makes Drupal accessible from any phone — no data plan required!
- Enables the expansion of community outreach initiatives (aka. as "community plumbing") beyond the web
- Is open source and free — you are in control

Benefits for administrators

- Easy installation and configuration — no programming required
- Fully customizable — enable only the features you need
- Run as part of the Drupal system itself with support for features, fields, notifications and more
- Enhance user interaction with ready-to-use audio blogs, click-to-call fields, phone recorders, audio announcements, etc.

Benefits for developers

- Well defined API that can be extended to other VoIP services
- 20+ sample scripts that can be customized
- A suite of 20+ modules such as click2call, phonerecorder, voicemail, and others that implement commonly needed features — no need to reinvent the wheel
- Simple, yet powerful, PHP-like scripting language with a short learning curve
- Visual programming language for novice developers and fast prototyping

VoIP Drupal Fact Sheet

Collection of modules (n=30)

- Make and answer phone calls
- Organize conference calls
- Send SMS
- Play recorded audio file "script"
Case studies

The first beta version of VoIP Drupal was launched in March 2011. We are currently promoting the platform and building more systems on top of it. Here are a few interesting projects we have built:

- **My Dot Tour**, a system that helps people organize participatory neighborhood tours using web, sms and even touch tone phones (http://mydottour.org/).

- **What's Up**, a local information system that enables access to web content via offline channels such as low-cost digital signs, customized flyers and posters, and an auto-generated community hotline. By dialing or SMS texting to a central number, folks might find out info about ongoing events, get reminders, and be redirected to local organizations (http://civic.mit.edu/whats-up).

- **VozMob** (http://vozmob.net/), a platform for immigrant and low-wage workers in Los Angeles to create stories about their lives and communities directly from cell phones.

In addition, we have been collaborating with different grassroots organizations to implement hotlines for their programs. Examples include:

- **Domestic Workers United** (http://www.domesticworkersunited.org/), an organization that promotes the rights of nannies, housekeepers, and elderly caregivers: +1 (646) 699.3989

- **OneVille’s Parent Connector’s Network** (http://oneville.org/bilingual-parents-as-connectors-for-other-parents-2/), an initiative that provides local school information in English, Spanish, Portuguese and Haitian Creole: +1 (617) 209.9094

- **VoxBox** (http://myvoxbox.org/), a service that provides voice-based bulletin boards for grassroots organizations such as PTAs, churches, local sports teams, and informal groups. By dialing or texting a VoxBox number, community members might find out about important weather alerts, meeting reminders, poems, stories, and announcements of all kinds. Users might also subscribe to different VoxBoxes and, with that, receive email and SMS notifications whenever something new is added to the system.

Future directions

**Short-term goals**

- Promote community adoption and ownership of the platform
- Drupal 7 port

**Longer-term goals**

- Integration with Drupal Commons, CiviCRM, OpenScholar, Open Atrium, Commerce, and other widely adopted Drupal-based systems
- Expand beyond MIT

References

Source code: http://drupal.org/project/voipdrupal/

General documentation: http://drupal.org/node/1078710

API information: http://drupal.org/node/1155572

Sandbox: http://voipdrupal.org/

Discussion group: http://groups.drupal.org/voip-drupal

45-minute video introducing the platform, its applications and benefits: http://groups.drupal.org/node/215969 (recorded at the March 7, 2012 webinar) with slide deck at http://slidesha.re/yN3jLy

5-minute video recorded at the VoIP Drupal launch at DrupalCon Chicago in March 2011: http://www.voipdrupal.org/node/152

Additional information

VoIP Drupal is an initiative of the MIT Center for Civic Media (http://civic.mit.edu/). For additional information about the project, please contact Leo Burd (voipdrupal@medialab.mit.edu)

How to create an application like Google Text SMS service? for VCSB.

- **Philo, tropo, phyno** (supported by VoIP)
- Interactive phone call
- Call 1000 users w/audio file inviting them to a rally
- **Dial 1 for NO, 2 for YES, etc.**
- **Example:**

  ```javascript
  $script = new VoipScript();
  $script->addSay('Hi, welcome');
  $script->addHangup();
  $script->addSay('Call "TUTORIAL" to help desk.');
  $script->addHangup();
  $script->addSay('Call "JUICE" to get more details on JUICE and the campaign against KFC.');
  $script->addHangup();
  $script->call($phoneNumber);
  ```

→ Click to call: # enter your number.

Script is executed to call user.