

Table of Contents

DrupalCon Vancouver 2006	1
Actions and workflows	1
Ad-hoc Event Discussion	2
Session Notes	2
Ad-hoc session on event-related modules and topics led by ??	2
Current Event-related modules:	2
Wishes:	3
Potential Issues:	3
To Do:	3
Backgrounder: spatially enabling CMS content	3
Beginning coding for Drupal	6
Better AJAX integration	7
Decisions module	8
Drupal administration user experience	8
Drupal community processes	10
Drupal Enterprise-wide	11
Session Notes	11
Using Drupal as a web application framework for enterprise-wide application development and deployment led by Chris Johnson	11
Primary Topics	11
Drupal guild and the open business model	12
Drupal roadmap	12
Drupal, enterprise-wide	13
Motivations	13
Session Topic Outline	14
File API	14
Improving Drupal documentation	14
Integrating Drupal with CiviCRM	15
Presentation Slides and Resources	16
Support and documentation resources for CiviCRM	16
Slides from the presentation	16
Agenda: OSCMS '06 - Integrating CiviCRM with Drupal	16
1. What the *!#k is CiviCRM?	16
2. CRM Core	17
3. Optional Components	17
4. Integration Approaches	17
5. CiviCRM Profiles	17
6. Read-only access via simple PHP snippets	18
7. Read/write module integration (Richard Orris / CivicActions)	19
8. Two-way integration w/ hooks (Rob Thorne)	19
9. Future directions for the platform	19
Session Abstract, Resources and Expected Outcomes	19
Session Abstract	19
Resources	20

Outcomes	20
Inter-site functionality	20
Solution components	21
Distributed authentication and shared user accounts	21
Enable sites to discover and register with each other.	21
Collectively manage content and taxonomies among different sites	21
Workshop outline	21
Large project workflow improvements	22
pre-session discussion	22
Media support in Drupal	23
Welcome to the Media Drupal meeting.	23
Relationship API	23
Revamping URL aliasing	23
Theming system enhancements	24
Usergroups.drupal.org	25

DrupalCon Vancouver 2006

Notes from the Drupal Conference in Vancouver, February 6 - 11, 2006

Actions and workflows

Moderators: Matt Westgate/John VanDyk

Now that we have the ability to create Drupal actions and workflows, it's time for a paradigm shift. Aren't configurable actions a lot like forms with stored procedures? Aren't workflows (moving through a series of states) a lot like a wizard path of execution? Why don't we glue them together with CCK to make a configurable wizard wizard (sic)? As you can see, we have more questions than answers.

Who said that workflows need to be restricted to nodes? We hope to map out a course that will introduce workflows to the user and comment lifecycle as well. And we need to plan how to move from Drupal's current checkbox-based "workflow" settings to the real thing.

Because so many people have signed up for this session, we plan to divide the time into an initial 5-minute overview of actions and workflows followed by smaller working groups focusing on the following topics:

MIGRATION STRATEGY WORKING GROUP

Currently Drupal has a hardcoded two-state workflow (unpublished to published). How do we get from current code to workflow-in-core? What does the new workflow system look like? Core currently uses checkboxes for node options, but workflow uses a drop-down field. Can we have both? What are the usability issues surrounding this?

Outcome: a strategic plan with defined milestones.

ACTIONS WORKING GROUP

Go through Drupal's core modules and enumerate the possible actions we need to write, then write the actions. Examples:

- blocking a user (user.module)
- creating a node (node modules, focus on CCK)
- sending a ping (ping.module, drupal.module)
- enabling a block (block.module)
- changing a workflow state (workflow.module) eek! recursion!

Outcome: a list of actions for each module.

FORMS INTEGRATION WORKING GROUP

Develop plans to use the workflow engine to drive form-based wizards.

Outcome: description of the approach.

ACTION API WORKING GROUP

What is the lifecycle of an action? Do we need batchable actions? How do we solve the recursion issue? What about branches and looping?

Outcome: a refined actions API.

PERMISSIONS AND WORKFLOWS WORKING GROUP

How do workflows integrate with access control?

Outcome: enumeration of different approaches to the problem.

PREPARATION

To be prepared to participate in this session, you should

1. Grab a copy of HEAD
2. Install the cvs versions of actions and workflow modules
3. Watch the FOSDEM 2005 video where John VanDyk presents actions and workflows if you are unfamiliar with the concepts (video is at homepage.mac.com/drupal/drupalconf-workflow.ogg and requires something like VLC)
4. Get familiar with the new Form API

Ad-hoc Event Discussion

Session Notes

Ad-hoc session on event-related modules and topics led by ??

Current Event-related modules:

- Event (does have an API - let's you view a themed calendar with passed list of events) 4.6 works
- Event finder (requires Event? and Location. has lots of cool stuff, e.g. zip code, metro area, etc.) 4.6 works
- RSVP
- SignUp - 4.6 works, used for OSCMS
- Event Repeat (4.7 release - Chad)
- Volunteer
- Event Filter (not yet in CVS, from Owen/UK)
- Moshe: I added an 'all day events' module to events Contrib. 4.7+ only

Wishes:

- Pull in event feeds from other sites - iCal import
- User Calendar
- Private Events - Flexinode (Aaron)
- E-Commerce / Event - Signup (4.6 contrib for signup or ecommerce - Chad: signup_ecommerce)
- Find events by Venue

>>> Much of this has been implemented by GoJoinGo (Angela Byron, it *is* a project checked into CVS, it uses Organic Groups and more)

>>> gsearch module (Ankur, has a lat/long location table, part of GoJoinGo)

Potential Issues:

- Event Repeat / iCal export
- Memory usage issue with Event module? related to node system?
- Themeing

To Do:

- Install GeoCoder locally

Backgrounder: spatially enabling CMS content

The following is an outline of a background document on creating spatially enabled content with a CMS. Charles and Dan put this together after the 2006 Vancouver OSCMS.

How to use this document? Some people may only want to add spatial tags to their nodes (and use canned mapping sources like Google Maps). They will find Sections 5 and 6 most interesting. Others may be interested in getting their own base map data and even serving it. We hope they will find the early sections useful.

We should really use a Wiki tool for this organic document/book. We will look into this.

Index

0. Other Sources of Information
 1. Data: Base Maps
 2. Data: Base Geocoded Data
 3. Data: Client Map Data
 3. Data: Other geocoded data
 4. Location: IP-based Location Information
 5. Location: Spatially Enabling Nodes
 6. Bringing it all together in Drupal: work to date
 - GMap, Carto, Plazes, Location, Gig

7. Interested People

View full node to see content...

=====

0. Other Sources of Information

- PlanetGeospatial

Planet Geospatial is an aggregation of public weblogs written by GIS Professionals and Hobbyists. The opinions expressed in those weblogs and hence this aggregation are those of the original authors. Planet Geospatial is maintained by James Fee

1. Data: Base Maps

- free

- satellite raster data, GIS vector data

- e.g. search geoconnections [<http://www.geoconnections.org/>]

- e.g. sat data [<http://glcf.umiacs.umd.edu/>]

- better yet, free but someone else is custodian and server!

- GoogleMaps

- Yahoo

- e.g. BC Government (Integrated Land Management Bureau)

- geocoded [standards: <http://www.opengeospatial.org/>]

- raster

- vector

- open source geocoding / data processing tools

- GDAL (raster) [<http://www.remotesensing.org/gdal/index.html>]

- OGR (vector) [<http://www.remotesensing.org/gdal/ogr/>]

- FWTools [<http://fwtools.maptools.org/>]

- PROJ.4 (projections)

- map data viewers / GIS tools

- OpenEV (FWTools)

- uDig (java tool)

- QGIS

2. Base Geocoded Data

- e.g. National Geospatial Agency (US)

- name servers

- geocoder.us; geocoder.ca

- Brainoff.com [<http://brainoff.com/worldkit/geocoder/>]

- Yahoo does this too

- * assumption that using WGS84 (same as GPS system)

- MaxMind

-

3. Data: Client Map Data

- security

- storage

- MySQL 5 [<http://dev.mysql.com/doc/refman/5.0/en/spatial-extensions.html>]

"MySQL implements spatial extensions following the specification of the Open Geospatial Consortium (OGC).

- PostGIS [<http://postgis.refrations.net/>]

"PostGIS adds support for geographic objects to the PostgreSQL object-relational database."

- serving [standards: OpenGIS]

- Mapserver

- Geosever

- Do we need to store data in OGC specific formats?

- No and Yes... Right now ok to have geographic information in node module

- later may need tables

- for some clients/problems, you may want to store the data in tables that have the OGC

4. Location: IP-based Location Information

- Module that does

- Localities (IP-based plus DB behind)

5. Location: Spatially Enabling Nodes

- adding spatial locations to nodes or other information

- mapping metadata to metadata (e.g. P-CODE to lat/long)

Client Applications

- e.g. polling station monitoring

- dial in asterix server, voice

- entl groups

- tracking news posts and pictures on logging or entl

6. Bringing it all together in Drupal: work to date

- task 1: viewing base maps

- task 2: overlaying client data

- GMap module [<http://drupal.org/node/33591>]
- original concept: display a map in a node
- xmap
- defined by macro
- any module that wants a map, could make call to this module
- possible to generalise even further...
- "maps" variable with all info, defn, overlays, + external call
- include a google include or yahoo...
- explorer uses
- Carto module [<http://drupal.org/node/24727>]
- Plazes module [<http://drupal.org/node/37281>]
- Location API, module [<http://drupal.org/project/location>]

7. Interested People

(we could start a 'spatially enabled' mailing list...)

Dan Karran <http://www.dankarran.com/contact/> dan@karran.net

Charles Burnett <http://www.mapswest.com> cburnett@uvic.ca (setaceous@drupal.org)

James (webgear@drupal.org)

Eric Gunderson eric@developmentseed.org

Beginning coding for Drupal

Moderators: Aaron Welch / Amanuel Tewolde

Assuming some basic knowledge of PHP, the session will focus on coding a Drupal module. There is no predefined area of discussion for this panel, so depending on the overall interests of the participants the discussion may spend more or less time on certain areas of Drupal development.

To start off, we will ask some general questions to try to determine the depth of knowledge and areas of interest in Drupal development to help guide the panel.

Hopefully we will cover the following general areas during the session:

- A general definition of a Drupal module
- How the module system works in Drupal
- Hook systems and how they work and interact with modules
- Nodeapi and how its different than the node hooks
- Forms API. Bring a pen.
- Theme system and themeing best practices
- Best development practices: where to go for help and resources, how to be a valuable contributing member, how to deal with core patches

- CVS: WTF? I mean really. Does anyone know how it works really? Really.

NOTE: This is a presentation/Q&A, unlike the other moderated sessions. The areas defined above may change during the session, or be presented in a different order. Questions/clarifications/minor details are encouraged. Bring a list of things you would like to cover or share to get the most out of the session. Bring a pen and paper, PDA or laptop to take notes and record urls of valuable resources mentioned during the talk.

NOTEv0.2: On the suggested PHP experience level:

We will probably not be talking strictly in phpspeak for the whole session, but more in broad terms about how modules work and how they interact with Drupal, so there may be useful information even if you aren't completely comfortable with php yet. Just be aware we may wade into deeper waters occasionally. Some time will also be spent talking about best developer practices, how the drupal development community works, and some other such things so that may be valuable as well.

As for some resources for learning php, it depends on your general level of programming experience. If you know any other programming language, there are several good O'Reilly books on php (I generally only buy O'Reilly technical books, just from experience they seem to be better for me):

This one is a beginner php programming book, but assumes a general knowledge of basic computer programming:

Linky

Since php and the mysql database go hand in hand (and Drupal is commonly run with mysql) I would recommend this as well once you are more up to speed:

LinkyV2

Also, for a free option, you can try the official php website tutorial: <http://us2.php.net/tut.php>

This may be of dubious value depending on your level of knowledge about web architecture already.

Better AJAX integration

Moderator: Jeff Robbins

AJAX is the web buzzword of the year - well... last year. None of it is new technology, but these days we can expect that most web browsers will support it. It has a high "wow" factor that many of our clients are demanding. But at the same time, proper etiquette prescribes that our pages should continue to work JavaScript-free down to the lowest common denominator.

Some AJAX functionality has been written into Drupal core, but what else can be done? How can AJAX benefit the Drupal user experience? How can AJAX benefit Drupal site design? What do the existing AJAX libraries such as Prototype/Scriptaculous, Dojo, and xajax have to offer? Should we be building our own JavaScript or

contributing back to one of these projects? How can we develop a transparent API that can allow Drupal developers to integrate these powerful features without getting bogged down in JavaScript coding?

We will show some examples of the work that is being done within Drupal and elsewhere and attempt to answer at least one of the questions above. Maybe two.

Decisions module

Moderator: Omar Bickell

Session Description:

A bunch of activists and Drupal enthusiasts from Quebec (aka Koumbit.org) have decided to create a module for Drupal which is aimed at enabling groups to take decisions online in a manner that replicates and augments what is possible in face-to-face meetings.

This includes but is not limited to modular decisions mechanisms, quora and representativity, vote delegation and retraction, issue-based categorisation/controls, vote weighting and spoiling, e-mail notification/integration, auditability and secrecy, etc.

We have already elaborated a number of the specifications for this project and plan to have some working code by the time the conference takes place.

Drupal administration user experience

Moderator: Kieran Lal

A couple of months ago we conducted a Drupal administration user experience survey[1] with 877 respondents. Here is a summary of the outstanding work to be done in response to the results of the survey. Much of the work involves interviewing users, doing surveys, and writing documentation about how to use Drupal. If attendees of the workshop are interested I would like to work with you to prepare some of this work in advance of the workshop so we can make the workshop time more productive. <http://civicspacelabs.org/home/survey/categorization>

We will also explore some of the innovative work with the Forms API to create multi-tabbed forms to create basic, advanced, and expert forms. We will also explore some of the new Form Wizards.

Summary of administration problems

Drupal is missing an overview of the site for administrators. We need to allow administrators to visualize and deliver a cohesive, compelling, and highly dynamic user experience to their end users.

ACTION: David Reed has developed the administration module[2], with screenshots[3], and designs by Isaac Horton. He is now working with Earl Miles.

TODO: We are looking for people to help with the development and testing of an administer module with guidance from our information architects, David, and Earl

Top administration tasks

1. Administer content such as comments-83% of respondents

ACTION: Jeremy Andrews, for CivicSpace Labs(CSL), has added multiple comment administration to make administering comments easier.

2. Install new modules and learn new features- 83% respondents

ACTION: Wrote 81 administration help handbook pages for Drupal Handbook : the Drupal Docs team

ACTION: Committed administration help for 31 modules in Drupal core and dozens of contributed modules.

ACTION: Developed project module to allow categorization of modules on Drupal.org. Developed by Nedjo Rogers, CSL, and Dries.

ACTION: Developed project module to report to Drupal.org what modules are being used and infrastructure to identify if modules have updates(security review pending). Nedjo Rogers and CSL.

TODO: Test the new project module categorization at

<http://scratch.drupal.org/project/Modules>

TODO: Card sort contributed modules to identify categories for Drupal modules

Difficult administration tasks

1. Manage inconsistency in themes- <http://drupal.org/node/37156>

ACTION: Wrote managing inconsistencies in theme with community

ACTION: Conducted 6 theme developer interviews to identify theme development tasks for Drupal theme help documentation

ACTION: Launched Drupal theme mailing list, and launched #drupal-themes

2. Find pages that are deeply linked:

ACTION: Change content type administration to be 3 links deep for content types based on flexinode

3. Configure blocks for roles

TODO: Configure blocks for roles-Dries

4. Understand Drupal terminology and synonyms- Need to do study

TODO: Conduct survey of CMS standard terminology and user expectations of terminology

5. Update site to a new version of Drupal

ACTION: Added UPGRADE.txt with basic instructions

ACTION: Ported CivicSpace upgrade instructions to

<http://drupal.org/upgrade/tutorial-introduction>

6. Compare Drupal files

ACTION: Developed Drupal and project module infrastructure to send messages potentially including module version information and updates.

7. Administer categories with the taxonomy module

TODO: Interview users about taxonomy

ACTION:<http://civicspacelabs.org/home/survey/categorization>

8. Order and organize content.

TODO: Interview users about how to order and organize content

TODO: Categorize modules that order and organize content

9. Administer structure of content.

TODO: Research capabilities for modifying structure of content

10. Administer Menu's

ACTION: Assigned Richard Archer menu maintainer

TODO: Card sort default Drupal menu items

TODO: Separate administration menu from navigation menu to be more situationally relevant

TODO: Remove or make optional user name label for navigation menu

11. Administer site navigation

TODO: Research capabilities for global, local, contextual, and situational navigation

Links:

1. <http://www.surveymonkey.com/DisplaySummary.asp?SID=1425065&U=14250658155...>

3. <http://cvs.drupal.org/viewcvs/drupal/contributions/sandbox/der/admin4.PN...>

2. <http://cvs.drupal.org/viewcvs/drupal/contributions/administration>

Drupal community processes

Moderator: Robert Douglass

The way Drupal is developed and evolves is highly influenced by the technical details of our infrastructure. From the issue queue to CVS to the mailing lists, the technological tools that we implement and use shape the way people contribute to Drupal.

Many proposals have been put forth for improving the cooperation between developers, the efficiency of the patch queue, and people's ability to find support on Drupal.org. This session will discuss one or more of these suggestions, analyse the technical requirements to support them, and explore a battle plan for implementation.

- Drupal Enhancement Proposals (DEPs) to foster coordination on large-scale projects, to build consensus on important ideas, and to better convey a sense of direction in development to the interested public, Adrian Rossouw has proposed that Drupal adopt a formalized procedure for handling new ideas.

Modelled on the JEP system used by Jabber: <http://www.jabber.org/jeps/jep-0143.html>

- Project module, CVS module, Issue queue Many good ideas have been put forth on how to improve issue tracking at Drupal.org. These ideas must be put into code before they can benefit anyone, yet few people use or understand the Project or CVS modules, the existing code responsible for code and issue tracking at Drupal.org. How do we make these modules, central to the past and future success of Drupal, become more accessible to others so that the pace of development on them can increase?

Please contact me with suggestions concerning this session.

Drupal Enterprise-wide

Session Notes

Using Drupal as a web application framework for enterprise-wide application development and deployment led by Chris Johnson

Primary Topics

1. Authentication
 - Single Signon - !! key
 - LDAP Usage:
 - Existing LDAP module, written by Moshe, modified by Bryght
 - questions about module quality and robustness (maybe only does auth yes/no and does not import any user data, for instance)
 - import into Profile
 - Another module which does something similar? [research]
 - Import or use Roles from another authentication service or database (e.g. LDAP)
 - "shut off" default Drupal auth -- make login / acct creation / acct editing screens go away, and don't let user edit their data in Dupal's database, as it comes from central location.
 - account synchronization with external database?
2. Multi-Database support
 - current ability to share one or more tables between sites
 - Kieran / Amazon working on documentation - get with him to help
 - ability to use other non-Drupal databases in applications built on top of Drupal.
3. Inter-release / backward compatibility, and upgrade/update path support.
 - 4.7 has .install files which tell automation how to upgrade the specific module the .install file is associated with.
4. Performance - fast page generation for each user
 - Kieran / Amazon working on documentation - get with him to help
 - there is a performance/scalability forum at drupal.org
5. Scalability - ability to support large numbers of users and ability to use conventional scaling techniques, e.g. load-balancing, database clusters, etc.
 - list of modules which don't scale well now.
 - autopath (auto path alias urls)
 - flexinode
 - caching for logged in users
 - optimize / pursue the lowest hanging fruit first
 - good system admin / system architecture tuning for generic application scaling, e.g. tuning the database, the server, etc. first.
 - Kieran / Amazon working on documentation - get with him to help.

6. Document existing tools and techniques to make Drupal function as such an enterprise web application framework. Much can already be done, but it's not obvious or clear just what all the pieces of the puzzle are.
 - Controlling various features, turning off, etc.
 - How to scale.
7. Content Domain Models, e.g. ERP vs. hospital, etc.
8. Shared Content
 - see John Van Dyke's Publication and Subscription matched-pair of modules. It needs a defined, documented API.
 - Aggregator 2 can do much of it for ordinary nodes (not events, e.g.)
9. Human Resources -- finding an adequate number of qualified Drupal developers.

Drupal guild and the open business model

Gregory Heller, Civic Actions to talk about Drupal Guild Concept and the open business model.

CivicActions is Professional Services firm specializing in Drupal deployment for progressive political and non-profit organizations. We are building a business model tailored to open source development and the open licensing that is at the core of our business.

We see the "Drupal Guild" concept as a way for similarly organized development firms and independent developers to organize in a manner consistent with the philosophy behind open source development and that will foster the growth of the open source ecology around Drupal.

In this session, Gregory will briefly present the CivicActions business model, introduce the concept of the Drupal Guild and open up a discussion with interested developers about business models and the Drupal Guild.

(please review the [Drupal Guild Wiki Page](#))

Drupal roadmap

Moderator: Dries

Given the way Drupal is developed, we can't set things in stone. Code is contributed, or it is not. Hence, the goal is not to define a formal or strict roadmap, but (i) to get a good feel of where we are going, (ii) to compile a list of priorities for Drupal 4.8 and onward, and (iii) to align ideas and visions. Really, this session is going to be a bit of a "potpourri of things".

1. What are key features that are currently missing?
2. What are the key performance problems?
3. What are the key usability problems?
4. What are the short-term priorities?
5. Who is working on what?
6. What is core-worthy?
7. What are the various dependencies?

8. Who should be working together instead?
9. What makes a Drupal 5.0.0?
10. What is of strategic importance?
11. What is merely nice to have?
12. What are potential pitfalls?
13. What is important for Drupal's future?

Drupal, enterprise-wide

Start: 2006-02-07 14:30

End: 2006-02-07 15:45

Location: one of the rooms

Moderator:: Chris Johnson

Drupal can be more than just another single web site content management system (CMS). Can and should it be used as an enterprise-wide web application framework?

Due to its modular design and well thought out interfaces, extending Drupal's functionality is straight forward -- and more importantly, actually makes sense, as it can be done cleanly and efficiently. Drupal also has the basic framework for shared authentication and distributed data through features like XML-RPC, RSS and more, as well as more mundane features like a database abstraction and sessions.

Discussion will focus on the kinds of features needed by an enterprise web application framework, how Drupal can provide those, and where Drupal falls short.

Please register to sign up for this event.

Motivations

My initial concept for this session is centered around using Drupal as a framework for building a variety of web applications within the context of a business, since that is part of what I'm currently doing at my place of employment.

I've spent 6 years building various web applications for our extranet using PHPLIB (phplib.sourceforge.net). When I started, I needed sessions (this was before PHP had sessions itself) and user authentication. PHPLIB provided these features, as well as a nice database abstraction.

I'm continuing to build new functionality and web applications, including such things as connectivity to other systems via proprietary TCP socket connections as well as via SOAP interfaces. I've ended up with a hodge-podge of things, and no consistent user interface. I've also pretty much coded everything from scratch, other than what PHPLIB gave me. This meant more work than it would require now using some good libraries, classes and/or framework.

So I've looked for a better way to leverage my time and to integrate everything under one consistent GUI. Drupal looks like one possible good answer. It provides many of the basic services (authentication, database, user data handling), puts it all under a nice looking theme, and is extremely modular and well written, making it easy to extend.

This session is about this aspect of using Drupal -- not just as a CMS engine for a single site, or even a community of sites -- but as a tool to build bigger, more varied applications. There are some areas where Drupal is weak in providing needed support. If enough people are interested, perhaps we can get some development into core that will make it a better framework for such purposes.

Or perhaps I'll learn it was not such a good idea after all! :-)

Session Topic Outline

Requesting Session Participant Input!

Outline of Topics

- Secure distributed authentication
- Performance tuning
- Multiple database support
- Your...
- ...Ideas...
- ...Here.

File API

Increasing performance and security, and files as first-order content in Drupal

Improving Drupal documentation

This session was held as a round-table discussion where people raised issues important to them, most of which surrounded making Drupal.org more easily accessible and identifying areas which were lacking in Drupal's documentation as a whole.

Feature requests

- Overview of how things connect (diagrams) -- Dale is attempting to spear-head this effort
- Screen casts/screen captures to provide visual explanation on how to accomplish tasks (Development Seed has offered to do work in this area)
- Improved search on Drupal.org (coming with 4.7 upgrade)
- Version control on documentation; better able to see what version of Drupal page corresponds to
- Ability to import/export documentation (Djun doing lots of work in this area)
- Ability to gauge how helpful pages are with some sort of ratings system as well as statistics analysis -- determine who has been helped by a page, who's finding what they need, who's not.

- Open up documentation process so people can easily contribute a quick note/correction here or there.
- Modularity of documentation to allow customization of Drupal docs for client installations, allow templating (for customization)
- Add ways of encouraging contributions... hall of fame, sponsor page, economic incentives
- Teasing apart drupal.org into separate sites for developers, themes, support, usergroups?

Areas for improvement

- Better explanation/overview for how theming works; we are currently turning off designers by making things too technical
- Lack of editorial vision; there is lots of content coming in every day, but it seems hap-hazard; brittleness of information architecture
- Should there be a separate site for newbies? (lots of agreement on this point, but it carries its share of cons)
- The term 'Handbook' is not immediately recognizable as support documentation. Suggestions were to rename this tab to something like "Documentation"
- Simplify the "Support" tab. Wayyy too many links here.
- Navigation is difficult; more links directly to deep into the handbook to key areas of interest
- There is 'friction' within the documentation working group, which can make potential contributors intimidated or annoyed
- A FAQ: Compile list of questions in the forums/ mailing lists that are asked all the time in one place
- A 'tour' or 'start here' document
- Improve Drupal.org homepage? "River of News" is not seen as attractive. Not obvious for people to know where they should go when they first arrive.
- Hard to get a sense of how the project and contributing to it work... where do you start?
- Have checklist of goals that can be viewed 'at a glance' to see where progress has been made in the documentation, what else is left to do, and where people can help out (issue tracker is not good for this; most people who contribute documentation are not developers so working with the concept of 'bugs' is foreign)
- have division between writers/subject matter experts; team up people who can write with people who know what needs to get written
- Right now, the site has a very internal view rather than external view (data centered, not user centered). Need to define target audience for each section and then always write toward that target audience. There are many ways to break down target audience:
 - skill/familiarity
 - task orientation
 - background
 - role
 - type of site they're trying to develop

Integrating Drupal with CiviCRM

Presentation Slides and Resources

Support and documentation resources for CiviCRM

Project Home

<http://www.openngo.org>

Online Documentation

<http://objectledge.org/confluence/display/CRM/CiviCRM+Documentation>

API Doc and Snippets

<http://objectledge.org/confluence/display/CRM/CiviCRM+Public+APIs>

Mailing List (support and discussion)

<http://lists.objectledge.net/mailman/listinfo/crm-dev>

FAQs

<http://faq.openngo.org/>

Downloads

<http://downloads.openngo.org/civicrm/>

Roadmap

<http://objectledge.org/confluence/display/CRM/OpenNGO+Roadmap>

Issue Tracker

<http://objectledge.org/jira/browse/CRM>

Slides from the presentation

Agenda: OSCMS '06 - Integrating CiviCRM with Drupal

1. What the *!#k is CiviCRM?
2. CRM Core
3. Optional Components
4. Integration Approaches
5. CiviCRM Profiles
6. Simple PHP snippets - Read Access
7. Module integration
8. Two-way Integration w/ Hooks
9. Future Directions

1. What the *!#k is CiviCRM?

Constituent Relationship Management Platform
Non-profits, NGO's, Advocacy and Political Organizations
Fully Internationalized - Current Translations in

- * Spanish
 - * German
 - * French
 - * Portuguese
- Multiple CMS - Drupal & Joomla/Mambo
LAMP, PEAR libraries, PHP gettext for i18n

2. CRM Core

Data Store, API's, UI, Import and Export

- * Contacts
- * Locations
- * Relationships
- * Static and 'Smart' Groups
- * Activities / Activity History
- * Custom Properties for Contacts, Groups, Activities, Contributions

3. Optional Components

CiviContribute

- * Configurable online contribution pages
- * Plug-ins for PayPal and Moneris
- * Online and offline contribution tracking

CiviMail

- * High-capacity broadcast email
- * Bounce-handling, open, click-thru, forward tracking

Coming

- * CiviSMS
- * CiviMember

4. Integration Approaches

CiviCRM Profiles

API's - invoke local PHP functions

- * Simple PHP snippets in blocks
- * Module Integration (read/write)
- * Two-way Integration (CiviCRM hooks)

Remote invocation via SOAP "alpha"

5. CiviCRM Profiles

Expose search, listings, and insert forms w/o coding

Configure sets of core and custom fields

Built in pages or standalone forms

Linked to user registration, user account, or other modules/components

6. Read-only access via simple PHP snippets

*** Get sorted info for members of a 'named' group using search API ***

```
<?php
$myGroup = 'Newsletter Subscribers';
$results = getGroupMembers($myGroup);
if ($results) {
    echo '<h3>' . $myGroup . '</h3>';
    foreach ($results[0] as $myContact) {
        $string = 'ID: ' . $myContact['contact_id'] . ' ' .
            $myContact['sort_name'] . ', ' . $myContact['email'] . ',
Phone: ' .
            $myContact['phone'];
        echo $string . '<br />---<br />';
    }
}
else {
    echo 'No group found with that name.';
}
function getGroupMembers($group_title) {
    // Check if CiviCRM is installed here.
    if (!module_exists('civicrm')) return false;
    // Initialization call is required to use CiviCRM APIs.
    civicrm_initialize(true);
    // Pass group 'title' to crm_get_groups to retrieve the group_id.
    $groups =& crm_get_groups(array('title' => $group_title));
    // Pass 'group' => group.id into search API to find members of that
    group
    // Only one group with that title, so we can reference
    // first object in returned array from get_groups.
    $params = array('group' => array($groups[0]->id => 1));
    // Define which contact fields should be retrieved
    $return_properties = array('sort_name' => 1, 'email' => 1, 'phone' =>
    1);
    // Define how we want the results sorted - also as an array.
    $sort = array('sort_name' => 'ASC');
    $contacts =& crm_contact_search( $params, $return_properties, $sort );
    if (!$contacts) return false;
    return $contacts;
}
?>
```

7. Read/write module integration (Richard Orris / CivicActions)

Why do it? example of either event-finder or petition module

Demo / discuss key integration points

Code example(s) and tips (do's and dont's)

8. Two-way integration w/ hooks (Rob Thorne)

Why do it?

Demo / discuss key integration points

Code example(s) and tips (do's and dont's)

(disclosure on current state/hacks...)

9. Future directions for the platform

On our radar

- * Finish/tweak hook implementation

- * Better doc and more examples

- * Remote invoke via SOAP

- * Search improvements (UI and API) - esp flexible query building w/ AND, OR, more operators...

- * Command-line import (large datasets)

- * Report-writer integration

- * Configurable core forms and listings

Contacts and other CiviCRM Objects <=> Nodes

Taxonomy Integration

Session Abstract, Resources and Expected Outcomes

Session Abstract

CiviCRM is an open source, internationalized, constituent relationship management (CRM) application, designed specifically to meet the needs of advocacy, non-profit and non-governmental organizations. CiviCRM is licensed under the AGPL, and comes with well-defined open APIs.

The purpose of this session is to introduce the APIs to the Drupal development community so that Drupal developers can start integrating CiviCRM objects such as contacts, profiles, relationships, groups and activity history into their modules.

CiviCRM is currently integrated with the e-commerce, event finder, location, mass mailer, organic groups, user, and volunteer modules. We will look at how to write modules to detect if CiviCRM is running, and use code examples to illustrate how developers can use CiviCRM's APIs to add the benefits of a structured CRM repository to their modules. Several individuals will be presenting integration examples, including brief code walk-throughs and tips.

We will also be soliciting discussion and input from the group on needed enhancements and future directions for CiviCRM API development.

NOTE: This is a presentation, not a moderated session. However, we will keep it informal and leave plenty of time for Q & A and discussion.

Resources

The Public APIs section of the CiviCRM wiki is currently the best resource for learning about available APIs. There is a growing set of code snippets in this section as well as a page of links to modules which use the APIs.

Outcomes

- Attendees should leave with a better idea of the core CiviCRM entities, available APIs and examples of how those APIs can be used to allow modules to share a structured integrated data store of contact and contact interaction data.
- Ideas generated for improving and extending CiviCRM APIs and API architecture and deepening the opportunities for integration with existing Drupal features.
- More full examples and code snippets generated and posted as resources for new implementors.

Inter-site functionality

The vision: International organizations can implement networks of sites that freely and securely share selected data (user accounts, categorization, content). Networks of drupal sites can discover each other through shared features and focuses, connect in webs of sites, and pool knowledge and contacts to weave isolated initiatives into strengthened movements for change.

If Drupal's fame for enabling collaborative communities was built largely on a single-site model, there's a good case to be made for the claim that it's connections among and between sites that will be the next big step.

In this session we will map out concrete next steps in extending the ability of discrete Drupal sites to link up and share information with each other - pooling user accounts, nodes, taxonomies.

Of course, we already have some good beginnings for inter-site pooling of information: syndication, blog-based protocols, the rudiments of remote authentication and networks of sites (drupal.module), and even Drupal-specific XML exchanges.

But there's a distance to go. Inter-site collaboration could include:

- Tightly integrated content categorization, lending itself to, e.g., organization/network-wide syndication feeds by category.
- Single user identity across organization/network.
- Pooled content/collective databases, reducing duplication.

Solution components

In the workshop we'll look at (at least) three different components of an overall solution. Here's an initial list, with some notes on existing solutions that may be relevant. How could each of these be adapted or extended to serve as part of a full-featured inter-site solution?

Distributed authentication and shared user accounts

- FOAF module
- Drupal module's authentication for external sites

Enable sites to discover and register with each other.

- The core Drupal module enables sites to register with a central server. Although used initially for maintaining a directory of Drupal sites, this functionality could be used to allow groups of sites to register with each other.

Collectively manage content and taxonomies among different sites

- SIOC module
- node import/export has an XML encoding for node data
- blogapi
- ditto for taxonomy import for taxonomies

Workshop outline

Our workshop follows directly on the related topic "Drupal Enterprise-Wide", and a number of people will participate in both. We will take advantage of this to continue discussions begun in DEW.

As we're a large session, we'll do most of our work in small groups.

1. Introduction and overview (5 minutes)
2. Organizing into small groups by topic (10 minutes)

Suggested topics:

1. Site network discovery and registration: how do networks of sites find out about each other and connect?
 2. Distributed authentication: how to users gain rights on multiple sites? how can a user logging into one site share data with another site?
 3. Pooled data: how can content and taxonomies be shared across sites?
3. Small group working sessions by topic (45 minutes)

For each group we need:

- *facilitator*: facilitate discussion, ensure balanced participation
- *summarizer*: summarize relevant discussions from Drupal Enterprise-Wide
- *recorder*: take notes, ideally on laptop
 - ensure you have everyone's contact information
 - pay particular attention to who is taking on what piece

Small group discussion suggested format:

- *quick* introductions: name, hometown, groups/companies/projects worked for, and 1 sentence on a question or interest you bring to the discussion. Example: "I'm Nedjo Rogers, I'm from Victoria, Canada, I work with CivicSpace, and I'm interested in how we can extend the Drupal module to be used for connecting up different sites."
 - Problem definition: agree on a basic description of the intended outcome of the solution component.
 - Status overview: discuss existing options or new initiatives, with an emphasis on what they do but what else would be needed.
 - Next steps: From the list of existing options or new initiatives identified, choose one or two for further development.
4. Report back (15 minutes)
 5. Report back from each small group, focusing on action items
 6. Wrapup (5 min)
 7. Whole group discussion of conclusions and next directions

Large project workflow improvements

pre-session discussion

i'm hoping to get a little discussion going here prior to the actual session. see here for a summary what the current outlook/goals for the session are.

i'm looking for ideas on what specifically we might want to cover in this discussion. What are our roadblocks to making big changes in drupal happen more easily?

Here's some initial things i think would be great to address:

1. The patch queue -- Woefully underpowered for bigger projects. What are our alternatives?
2. Test sites -- Bigger changes might greatly benefit from having a test site set up. It should lead to more testing as it eliminates the barrier of creating the test environment. Can we put a system in place for this?
3. Encouraging small group development -- one person writing a 100K patch is simply inefficient. What systems can we put into place that will not just offer the tools for small

group development, but actually encourage it??

Please throw in your ideas/comments, and I'll try to form a coordinated session plan out of it all... :)

Media support in Drupal

Welcome to the Media Drupal meeting

Over 30 of us registered so far already. After learning about each other at the beginning of the meeting and some general discussion we may need to split into a few splinter groups to stay productive. Seems there are at least four possible special interest groups:

- Folk who want to integrate flash and other media into the look and file and GUI of their sites
- Media bloggers who want to learn how to set drupal up for Vlogging, Appcasting etc.
- Developers working on media modules and core support for media
- Media-oriented business models

We will reconvene to summarize the smaller meetings.

One of the main results I am hoping for from the meeting is a roadmap for future media drupal work and some guidance for the core developers for any base features we will need to move media plumbing forward.

Here was my last stab at an agenda - please amend or change as you like:

Metadata, mpeg-7, tagging

Syndication formats, e.g. Media RSS, iTunes Store

Media workflow

Licensing and IP

State and progress of existing media modules: audio,media,video,podcast,image*,acidfree

Media in the Drupal GUI and themes, e.g., flash

Relationship API

Picking the best of the current implementations and further developing it.

Revamping URL aliasing

The current system scales poorly Even with 4.7 enhancements, there is a lot more than can be achieved. And it's not really trivial to administer 100K+ aliases. How do we move to a mapping approach while retaining the current flexibility?

Callbacks should not rely on `$_GET['q']`, only on parameters handed to them. This creates much more reusable code which is a boon in itself.

As a second step we could make the `url()` and `l()` functions 'intelligent'. Instead of handing over a simple string, we hand over a type and an id. In most cases this is as simple as splitting at the first slash, but what about the urls that do not look like `node/12345`?

To continue the system, for each type, we have a function, chosen previously on an admin UI, which creates the path from the type and the id. Incoming URLs can simply be handed by the current menu system: every mapping system adds a menu entry, and from the parameters it finds out the relevant callback and calls it. For example, pathauto paths could look like `http://example.com/pathauto/user/username`.

To sum up, our tasks are

- make all callbacks clearly parameterized.
- Remove excess `arg()` usage.
- Decide on the exact mechanism on how we could make the `url()` and `l()` functions 'intelligent'.

Theming system enhancements

Moderator: Adrian Rossouw

A common problem among people who have just started using Drupal is that creating a new theme is too hard.

A large part of this is due to the flexibility of the themeing system, but unfortunately a lot of this complexity is not completely explained and well managed to the end user.

This session will be about better organising the themeing system, to more easily allow for sites that don't all look like Drupal sites.

A part of this will be about introducing a flexible 'Layout' layer into drupal core, that allows for different site / section layouts to be more easily configured, using the powerful regions support that has recently been added to core.

This will also allow us to ship with a seperate (configurable) admin layout, which will in turn drastically reduce the amount of work required to create a distributable theme. The goal is to simplify the theme configuration interface to allow it to be more easily understood.

Another goal will be to better integrate with `phptemplate` now in core, as a lot of the other theme engines are based on `phptemplate`, and we can also stand to simplify `phptemplate` itself to allow for better performance. Introduction of meta-data alone will greatly enhance the themeing experience.

Additionally the implications of the forms api, especially the changes in 4.8, need to be discussed. As they will allow for more flexible templating, more specifically allowing for automatic template loading and a much simpler implementation of a fully integrated template editor.

Lastly, we should discuss style guidelines and possibly start compiling a proper styling guide for module and theme developers, to allow for more themeing to be done on a purely css level.

Usergroups.drupal.org

Meeting Notes - Usergroups.drupal.org â Moshe Weitzman â OSCMS - Tuesday, February 7, 2006

User groups for the Drupal community: What is it?

Will provide the ability for global communities using Drupal to attend local meetings and get together periodically to collaborate. Like âMeetup for Drupal.â These are not the same as workgroups centred around a specific topic, but rather geographical user groups.

Usergroups.drupal.org will provide one central place for all user groups to coordinate and be found, rather than each group having to maintain their own site.

Demo of the usergroups site

Demo site with stock OG installation was displayed and the site was run through to briefly explain the features:

- When creating a group, you select a country (taxonomy-based). Can choose whether to show group in a global directory and whether to show a selection for group on registration form.
- Listing of groups, each showing # of subscribers, group manager, and description. You can also view a list of subscribers and their role in the group.
- Groups are nodes, so things like taxonomy work as expected; can use modules such as Taxonomy View or Taxonomy Browser to sort user groups into categories or view by country, etc.
- Groups are containers for nodes that get added to them (child nodes). These nodes can be of any type (forum topics, events, etc.) and can be broadcast to multiple groups (âaudienceâ) or available publicly.
- Group management: groups have one owner, and an arbitrary number of admins. Admins can control membership, promote/demote admins, and edit/delete content posted to a group.
- Ability to invite a friend to join group.
- Users can subscribe to groups to obtain nodes via e-mail, or subscribe to RSS feeds of each node posted.
- Group context is preserved â when it detects you're looking at a node that's part of a group, will automatically populate blocks, etc.
- Group home page provides a 'mission statement', options to view content as 'River of news' or group by content type. Respects sticky posts.

- Basically, OG is almost like a Drupal site inside a larger Drupal site. A group's page can be themed. Admins have node administer permissions **without** having node administer privileges. ;)

Blocks:

- Block for content creation options
- 'My Groups'
- 'New Groups'
- Recent subscribers (who's new)
- Others via contrib (Group events block)

Use cases/Feature requests for usergroups.drupal.org

- Meet/Network with other Drupal developers
- Encourage more people to try Drupal/Promote Drupal
- Event calendars, both group-level and global-level
- Collaboration between groups, announce issues so people can all help each other
- RSS syndication between groups
- Event-enabled groups, with "sub-events" to run conferences on Drupal
- Provide information about Drupal in local language- reduce barriers to entry.
- Tie into central site "cross-promote events between two sites (pub/sub)
- Promotion of local user groups through central site (Drupal.org).
- Face-to-face time with other Drupallers, where stuff gets done!
- Can use user groups in lieu of piggybacking to every OS conference.
- Helping Drupal users in their own local language
- Private groups focusing on individual areas
- Get developer/user support locally
- Mail integration
- Hierarchical groups
- Roles within groups
- Associate 'tags' with groups
- Use for DEPs/working groups on particular goals rather than locality.
- A way to find developers, consultants, themers, etc. in group profile "self-promote"
- Allow community to aggregate its knowledge
- Communities of support, expertise, etc. -- how to get this out of OGs and into Drupal.org
- Support for custom theming: colour picker, choose-your-own stylesheet
- Possibility for role-based groups; eventually could move to multisite install with shared users

Next steps

- Find domain to put this on (Moshe)
- Launch site (Moshe)
- Theme (Kieran)

- Clean up event block (John)
- Browse/Find groups (Angie)
- Location/Maps (Dan)
- Hierarchical groups: write up use-cases (Aaron)

Questions/Comments

Move blocks to front page, rather than just sidebars â provide lots of information.

Can blocks be configured per-group?

One property when you create a group is what theme they use. All groups can look different. Plumbing is there for it, but no interface â possibility for a contrib module?

How do roles correlate?

Right now, they don't. But someone could write a contrib module to sync the two.

What about access control? How can I limit access to content?

Four different types of groups:

- open
- moderated
- invite-only
- closed

Integration with CiviCRM groups?

Yes, Gerhard has written this module. Overkill for user groups, unless you want to have an organization that controls all of these groups (more for 100s, 1000s of people).

Permissions for creating closed groups?

No; we want to maintain that flexibility.

Can you promote content out of a group to the site at large?

Yes, you can mark a post âPublicâ in addition to it being available for the group itself.

Languages? Will German group have to use English?

We can use locale to change the interface, but overall haven't thought through this functionality.

Enable the option to house own website off-site, and point usergroups site to it.

A couple options:

- Ability to enter remote RSS feed, which auto-populates nodes
- Add field to group form to supply external link to parent organization

What about breaking it down further by region (hierarchical relationships between groups? e.g. New York City, Brooklyn, etc.)

- Can do this with taxonomy...
- But there is no admin of higher-level group can be admin of lower-level group functionality.
- Write out use cases. What happens to posts? Etc.

Can you create a group within a group?

No, just posts within group - no hierarchical relationships.

Why two different sites?

OG starts off node_access stuff, and this could cause problems with existing nodes on Drupal.org.

Does OG scale?

- For 4.7, OG should be fine; if it isn't we have the skills and knowledge to fix it.
- Under 4.6 you will want to patch your node_access() if running a large site. That patch is already in 4.7