



# Drupal Technical Evaluation for new UWS CMS

---

*University Website Programme*

*ITS087*

*Document version 1.2*

*23<sup>rd</sup> May 2012*

## Table of Contents

1	Background .....	1
2	Summary of results .....	1
2.1	Contributions to cost/benefit analysis.....	1
2.2	Recommendations for future work .....	1
3	Multiple sites and local customisation .....	2
3.1	Common modules and style elements. ....	2
3.2	Scenarios .....	3
3.3	Support.....	3
4	User Interface for Editors.....	3
5	Content Sharing .....	4
5.1	Advantages of views .....	4
6	Module management .....	5
6.1	Consistency of UI across modules.....	5
6.2	Upgrade management .....	5
7	Technical Architecture .....	6
7.1	Technology.....	6
7.2	Resilience, capacity and scalability .....	6
7.3	Browser compatibility.....	7
8	Other HEI's using Drupal .....	7
9	References .....	7
A.	Potential Advantages of Drupal .....	8
B.	Recommendations .....	9
	Functionality .....	9
	Technical Architecture .....	9
C.	Document Management.....	10
	Contributors.....	10
	Version Control .....	10

## 1 Background

The University website (UWS) is currently implemented on a local implementation of the Polopoly content management system (CMS). A review of the platform is underway and one suggestion is to use the Drupal open-source CMS.

There are several existing installations of Drupal in the university already. These are mainly small and of limited relevance for evaluating the suitability of Drupal as an enterprise-level platform for the university web site. Hence a feasibility study was created in order to assess whether Drupal will satisfy the technical requirements for the University's web presence.

This feasibility study was a lightweight exercise, using a combination of a desk-based investigation exercise and a workshop incorporating representatives from the University Website team, IS Applications and an external Drupal consultant. The aim of the study was to review the available information to assess whether a more substantive evaluation would be worthwhile and to identify issues that require in-depth investigation.

This study concentrated on the technical aspects of Drupal and did not compare Drupal with other CMS systems. As such, it forms just one part of an overall consultation and procurement process organized by the University Website team. That overall process will need to consider the results of this study alongside other criteria, such as support arrangements, functionality, cost of migration, and so forth. More background information on the broader process is available on the UWS support wiki.

## 2 Summary of results

Based on the information that was collated at the time by the project team, the study has found no reason to rule out Drupal as a potential platform for managing the university's web presence. For several key issues, the study has identified approaches to implementing Drupal that are likely to meet the university's requirements. The high-level recommendation from this study is that if the University wishes to progress with the potential use of Drupal, then the next phase should be a project to test these approaches in practice using a proof of concept installation and involving a broad community of the groups that would be involved in running the service.

### 2.1 Contributions to cost/benefit analysis

The study was not specifically tasked with the production of a detailed cost/benefit analysis for a change of CMS, but as part of the technical evaluation it was necessary to consider a number of scenarios for the potential deployment of Drupal. As a result of these considerations, the study has identified a number of potential advantages for adopting Drupal as the university's CMS. These are listed in Appendix A as contributions to be considered as part of the broader consultation and procurement process.

### 2.2 Recommendations for future work

Recommendations for future work are listed throughout the body of the report and are collected in Appendix B. They constitute a set of issues that need to be investigated in the context of a project to build a proof-of-concept installation, as noted above. These issues all seem achievable from the

information available but they need to be confirmed in practice and most of them have many details that will need to be resolved before a full go-ahead can be given.

The study team are not aware of anyone in the university with experience of an enterprise-level Drupal installation. Such experience as there is within the university is scattered across many departments. A proof-of-concept project would serve as a learning laboratory for IS staff as well as evaluating the system.

*Recommendation: Project staff should attend Drupal training events before beginning the proof of concept work*

*Recommendation: Any further work should engage Drupal consultants to give advice and guidance.*

---

On the basis that the proof of concept proves to be successful, consideration should then be given to building a pilot installation to enable a representative website to be built and maintained, before considering moving fully into a full production environment.

### 3 Multiple sites and local customisation

One of the motivations for considering Drupal is that it might provide a common technical platform for a range of web sites, from the enterprise-scale central university web site through to small sites for research groups or other specialist groups. Drupal offers several options for providing multiple sites while sharing aspects of an installation. The study examined a number of scenarios and reviewed the implementation options.

#### 3.1 Common modules and style elements.

For Drupal, a key concept is that of a *module*. This is a reusable component that can be added to an installation of Drupal to provide extra functionality. Every installation of Drupal must choose which modules it requires beyond the small Drupal core. The study team envisage that the relevant teams in IS (Applications Division and the University Website team) will select and maintain a common set of Drupal modules.

Drupal also has the notions of *themes* and *content types*. These can be used to provide a common look and feel across the pages of a site, as part of an agreed information architecture. They also allow presentation to be tailored to specific types of content. The study team note that the tasks of defining and maintaining these common themes and content types will require appropriate resources.

Given these assumptions, the study team expect that Drupal could support the scenarios outlined in the following section.

*Recommendation: Decide on a process for selecting modules, considering issues such as functionality, stability, conformance to agreed standards, and effort required to support.*

*Recommendation: Agree a common set of modules.*

---

### 3.2 Scenarios

**All websites within a centrally managed CMS.** The main university web site will use the common IS modules, themes and content types, running on IS-supported hardware. Web editors will have access to the pages they need, e.g. those within their school or business unit. This is directly equivalent to the current platform.

*Recommendation: Confirm the best method for specifying access rights by user, page and/or role.*

---

**Style variations.** If desired, certain sets of pages on the central web site could use specific themes or content types, while still using the centrally-supported modules and other centrally-supported style elements.

**Functional variations.** A unit or group may decide to run their own instance of Drupal, adding modules that the central service chooses not to support. In this scenario, the expectation is that IS will provide the common set of modules and the users will choose the extra modules that they wish to install. This should simplify the support burden by providing a common core across all instances.

In order to protect against possible security breaches, the study team envisage that this scenario will require a separate instance of the CMS on different (virtual) servers. These servers and/or extra modules could be managed by IS (by arrangement) or managed by the users themselves. .

**Independent sites.** It will remain possible for users to install their own web site systems, should they wish. If they choose to use Drupal, they will at least have the advantage of using the same technology as the rest of the university.

### 3.3 Support

The university benefits from the existence of core web and technology support groups. Under the scenarios above, these teams will continue to support the central web site, with the degree of support they can offer to other sites diminishing as the sites diverge more from the centrally-supported offering.

*Recommendation: Investigate the options, processes and resource implications for supporting local variations from the centrally-supported package.*

---

## 4 CMS User Interface

Many university staff who use the current CMS complain that the user interface provided to them is clunky and slow. It is essential that any revision of the university's CMS provides a user interface that is easier and faster to use. This user interface should still enforce or encourage certain levels of editorial consistency and best practice.

Some editorial guidelines can be enacted using Drupal out-of-the-box. For example, an instance of the text editor can restrict the tags permitted, which could be used to ban inline links, if a separate instance were provided to enter lists of URLs. Drupal can also require that all images have a text

alternative, which enforces that particular accessibility rule. Further investigation would be required to examine which other guidelines can be supported and at what cost to implement.

***Recommendation: Use the proof-of-concept installation to investigate options for providing a better user interface for university staff.***

---

The university's implementation of Polopoly has also constrained the system to present a hierarchic structure of articles, whereas both Drupal and Polopoly "out of the box" let articles and nodes be accessed in many different ways. Drupal does provide an option to construct menus and submenus, which can provide a hierarchic view, although not an exclusive one.

***Recommendation: Use the proof-of-concept installation to investigate opportunities that might be offered by multiple ways of accessing pages.***

---

## 5 Content Sharing

The university's implementation of Polopoly makes it easy to share content between different parts of the central web site. The study concluded that it should be possible to provide very similar functionality in Drupal. In considering possible use cases where content was or might be reused, it would be useful to analyse cases in the current CMS to understand which examples have been most useful.

An advantage of using the same CMS for the central web site and for smaller sites is that it should be easier to share nodes from different instances of the CMS. This would require more work to implement than the simple case of sharing content within a single installation and will require more technical knowledge to use.

It is good practice that the owner of shared content should be aware that it is being shared and can maintain that content appropriately. Currently we manage this process by hand. We have not investigated whether Drupal could provide better support than our current implementation but we note this as something a follow-up activity could investigate.

***Recommendation: Use the proof-of-concept installation to investigate possible mechanisms for sharing content.***

---

### 5.1 Advantages of views

Drupal also has a notion of *views*, which provide additional advantages. A view can dynamically combine items into lists. For example, a view can show all articles that have been tagged with a given tag. Example uses might include an automatically-updated A-Z list of services, or a table of contents, or a list of news items sorted by date. An example occurs where a page includes a list of news items which are automatically updated. In our current implementation, this is only possible using a technically complicated RSS-based system that is unsatisfactory for this purpose. Drupal views make this easy.

## 6 Module management

As mentioned in the previous section, a Drupal installation will need a number of modules to augment the core Drupal functionality. This raises issues of consistency, between different modules in the selected set and between different versions of the same module during an upgrade.

### 6.1 Consistency of UI across modules

A key question raised for Drupal is whether different modules have compatible user interfaces. This is greatly improved in Drupal 7. The core modules are supposed to have a consistent UI and this is encouraged for contributed modules as well. Well-written modules provide a standard set of API “hooks” that allow an installation to customize the user interface without having to modify the module itself. The conformance to this API should be a factor in deciding whether to use a module. In cases where an otherwise desirable module still has an incompatible UI, it would be necessary to write an intermediary module that accesses the desired functionality while conforming to our chosen UI standards. It may also be possible to contribute UI changes back to the author of the module.

*Recommendation: Take the user interface into account when choosing the core set of modules and investigate the work required to provide and maintain a common user interface.*

---

### 6.2 Upgrade management of core software and 3<sup>rd</sup> party modules

Drupal modules are maintained by a world-wide open-source community. Indeed, when selecting which modules to use, an important consideration is the size of the group supporting each module and the quality of support provided. A result of this approach is that modules may be upgraded more frequently than in a commercial product and that upgrades may mix security fixes with functionality enhancements. This means that the university will need a robust process for upgrading modules.

This upgrade process must include provision for testing new modules against existing web content, to check that the upgrade does not break existing web pages. For the central web site, this will be provided by a Test installation that parallels the Production environment, as is standard for IS services. The process must also take into account the support of other sites that use the centrally-supported modules. These use cases have been explored on the project wiki (see the references section).

Another requirement is that it must be possible to upgrade a module with minimal loss of service. Here, the choice of multiple load-balanced servers in the technical architecture may contribute to a solution. It may be possible to restrict traffic to one server while the other is upgraded, then to switch across while the first server is upgraded in turn.

*Recommendation: Use the proof-of-concept installation to investigate options for upgrading modules with minimal loss of service.*

---

## 7 Technical Architecture

### 7.1 Technology

Drupal is developed using MySQL as the database technology and the vast majority of Drupal sites use MySQL. By contrast, the University mostly uses the Oracle database for its central services. Although there is an Oracle module for Drupal, this module is not maintained by the core Drupal project and is not widely used. Therefore the recommendation is to use MySQL to give the best supported platform.

The University has recently used MySQL for other new services, in particular the Moodle and WordPress systems deployed for the Distance Education Initiative. This experience has shown that it is possible to deploy MySQL in a manner that provides the backup and disaster recovery functionality that meets the university's requirements.

Another difference from the university's existing CMS is that Polopoly stores all files in the database, Drupal stores them in a separate file store. This partially simplifies the database infrastructure, while requiring a resilient file system that can be accessed from multiple sites. Again, this is functionality that the university has recently implemented for other projects.

The university uses CoSign to implement its reduced-sign-on system and our in-house ID management system to provision accounts. We routinely integrate these with third-party systems and expect no problems doing so with Drupal.

***Recommendation: Use the proof-of-concept system to check that the database and resilient file system can be accessed and backed up, and confirm that integration with EASE and IDM are practical.***

---

### 7.2 Resilience, capacity and scalability

Any implementation of a new web platform must at least match the performance of the university's existing web service. It should also provide the opportunity for extending this capacity if required and must be resilient in the event of hardware failure.

The university's Polopoly system is estimated to receive approximately 200,000 page hits per day, divided between two servers. The number of pages currently published is approximately 40,000, which is stored as about 10GB of data. Up to 250 website editors (out of 550 or more) have used the CMS system simultaneously. The average response to a web page request is 0.4 seconds, whilst the average response for CMS website editors (over a variety of activities) is 0.8 seconds.

The recommendation is that to offer the most horizontally and vertically scalable architecture we should use the virtual hosting infrastructure offered by IS ITI, in conjunction with a load balancer. Virtualised hosting for Drupal sites would allow resource allocation to be easily matched to demand. Using multiple servers on different sites would meet the requirements for scalability and resilience. A caching technology such as Varnish should be used to give the appropriate level of performance. It will be necessary to investigate how to structure the caching technology with the resilient file system and how both of these can be configured to best support the notion of locally-managed servers discussed in Section 3.2.

*Recommendation: Run performance tests on the proof-of-concept installation and work out the best arrangement of caching technology, load balancers and servers.*

---

### 7.3 Browser compatibility

Drupal will support a variety of browsers, including Firefox, IE, Chrome, Safari, and Opera. This meets the university's requirements.

## 8 Other Institutions using Drupal

Few UK HEI's manage their web presence with an integrated site-wide CMS as we do, although many aim to move towards this model. This makes it hard to assess the use of particular technologies as there is a small pool to assess. This pattern is repeated across the world, so although many HEI's use Drupal for a significant part of their web presence, the number of HEI's attempting our integrated approach is relatively small.

There are many other sites using Drupal for business crucial sites. These include The Economist, NASA and the White House. Commercial sites such as those rarely have the distributed characteristics of Higher Education. The nearest likely models are likely to be found in other public sector institutions.

*Recommendation: Identify other institutions that are using Drupal in a manner similar to that we intend and ask them for advice and guidance.*

---

## 9 References

More background information on the consultation and engagement process is available on the UWS support wiki:

- [CMS and website review - engagement and planning](#)

Use cases for upgrading modules on multiple installations are available on the Project wiki:

- [Technical use cases for a CMS](#)

Drupal 7 as an Enterprise CMS for a Higher Education Institution

- [Report by John A. Vieth, University of Wisconsin-Platteville](#)

A further list of HEI's using Drupal:

- [Drupal in Higher Education](#)

## A. Potential Advantages of Drupal

This study was not specifically tasked with the production of a detailed cost/benefit analysis for a change of CMS, but in considering scenarios for the potential deployment of Drupal the study has identified a number of potential advantages for adopting Drupal as the university's CMS.

The project brief noted that the perceived advantages of an open-source CMS such as Drupal include the following:

- The availability of many plug-in modules from the global user community
- As Drupal is better known by web developers, it is expected that the University will be able to harness this knowledge and experience and take advantage of it in a coordinated manner
- Being written in PHP, Drupal is more easily customisable by web developers

The study found that the range of modules available for Drupal confirmed these perceptions.

This study reviewed a number of scenarios in which different university web sites could make use of common technology, as summarised in Section 3.2. This approach would have the following advantages:

- Knowledge of Drupal could be shared between the university's web developers to the advantage of each site.
- Specialist sites could use parts of a centrally-supported set of modules, while adding their own customisations
- The central site could benefit from experience gained with smaller sites, for example when using a new module.
- A variety of support arrangements would be necessary

The study also found a number of Drupal features and modules that could be employed to improve the university's web platform. A particular example is the Views module, described in Section 5.1. Views provide an easy way to dynamically generate and maintain lists of related items, such as news items or A-Z indexes. This feature would simplify parts of our current web site and offer other opportunities for enhancing content.

The study noted the findings, albeit pro-Drupal, of the report relating to the University of Wisconsin-Platteville, which illustrates a number of the technical aspects in this report

## B. Recommendations

If the University wishes to progress with the potential use of Drupal, then the next phase should be a project to test these approaches in practice using a proof of concept installation.

As with any other project, this should begin with the clarification of the terms of reference and the governance arrangements. In this case the governance arrangements should include the process for agreeing to progress with the evaluation at each stage. These arrangements should also identify a user group.

The project team should be identified and should attend training events on Drupal (c.f. section 2.2). It is expected that the project would engage one or more Drupal consultants to give advice and guidance.

The project should then identify its business requirements. This should be able to use the extensive work already done by the University Website team. These requirements will guide the project team in the choice of Drupal modules (c.f. Section 3) as well as in the tasks of the project itself.

The project team should decide on a process for selecting modules, considering issues such as functionality, stability, conformance to agreed standards, and effort required to support. The team should then agree a common set of modules.

The project team should identify other institutions that are using Drupal in a manner similar to that we intend (c.f. Section 8). They should seek advice from these institutions when relevant and especially at the outset.

The bulk of the project time would be spent investigating the remaining issues identified in the body of the report. These fall into two groups.

### Functionality and usability

Confirm the best method for specifying access rights by user, page and/or role (c.f. section 3.2).

Take the user interface into account when choosing the core set of modules and investigate the work required to provide and maintain a common user interface (c.f. Section 6.1).

Use the proof-of-concept installation to investigate options for providing a better user interface for university staff (c.f. Section 4).

Use the proof-of-concept installation to investigate opportunities that might be offered by multiple ways of accessing pages (c.f. Section 4).

Use the proof-of-concept installation to investigate possible mechanisms for sharing content (c.f. Section 5).

### Technical Architecture

Use the proof-of-concept system to check that the database and resilient file system can be accessed and backed up, and confirm that integration with EASE and IDM are practical (c.f. Section 7.1).

## Drupal Technical Evaluation for new UWS CMS

Run performance tests on the proof-of-concept installation and work out the best arrangement of caching technology, load balancers and servers (c.f. Section 7.2).

Use the proof-of-concept installation to investigate options for upgrading modules with minimal loss of service (c.f. Section 6.2).

Investigate the options, processes and resource implications for supporting local variations from the centrally-supported package (c.f. section 3.3). This will require significant involvement from the relevant support teams.

### C. Document Management

#### Contributors

Role	Organisation	Name
Author	Applications Division	Dave Berry
Project Manager	Applications Division	Andrew Stewart
Reviewer	Applications Division	Bill Lee
Reviewer	Applications Division	Iain Fiddes
Reviewer	UWS Project	Miles MacCalman
Reviewer	UWS Project	Neil Allison
Reviewer	UWS Project	Joseph Farthing

#### Version Control

Date	Version	Author	Section	Amendment
27/04/12	1.0	Dave Berry	All	First version
30/04/12	1.1	Dave Berry	All	Revisions from review meeting
23/05/12	1.2	Dave Berry	All	Revisions from review meeting