

# Evaluation of ‘Liferay’ Portal Container Pilot

---

Version 1.4 Draft rk 05/05/11

Version 1.5 rk 11/05/11 following suggestions by RS

Version 1.6 jr 6/6/11 following comment from Research Board and Web Steering Group

## Contents

Contents .....	1
Summary .....	3
Background .....	4
Report Structure .....	4
Review of Research Websites Pilot .....	5
Websites Created .....	5
The Research Website ‘Package’ .....	5
The ‘Lightweight’ Research Project Site .....	6
Branding (Themes) .....	7
Search Engine Optimisation (SEO) .....	8
Web Content Management .....	8
Unstructured Content Management .....	9
Structured Content Management .....	11
Bulk Content Imports .....	12
Additional Liferay Components .....	12
Shopping Cart .....	13
Asset Publisher .....	13
Nested Portlets .....	13
Web Form .....	13
Extending Liferay .....	13
Existing Compatible Components .....	14

Bespoke Functionality .....	14
Review of the Liferay Product.....	16
Liferay in the marketplace .....	16
Liferay in HEI Globally.....	16
Documentation, Support and Community.....	17
UEA Training Requirements .....	18
Liferay System and Developer Documentation .....	18
ISD Documentation .....	18
Functionality Highlights .....	18
Liferay Social Office.....	18
Direct content access/editing (webDAV).....	19
Technical Aspects.....	20
Authentication and Users .....	20
Authorization .....	20
Standards Adherence.....	20
Robust Architecture .....	21
Choice of Database .....	22
Caching.....	23

## Summary

Overall, feedback on the Liferay pilot from the UEA community has been very positive. Seventeen websites have been created (not all progressed), far more than originally anticipated, and there is continuing demand for additional sites.

One website, created for a research grant application, was created in just a few days by a member of staff with no explicit training or knowledge of the system. This shows the potential for research, and UEA's wider activities.

There remain issues which need to be resolved – such as establishing a robust production service, hooking into UEA authentication, and development of additional functionality. This report outlines the costs and timescales for moving forward.

The pilot has shown that the portal container model could be used to provide UEA web services currently using other systems, including departmental 'flat-file' sites, cPanel hosted micro-sites, The Portal, and ultimately Polopoly and NetCommunities.

Consolidating products would:

- Release licence fees
- Simplify technical support
- Improve security
- Focus development effort and staff skills
- Reduce costs

Proving the concept of 'portal container' is one thing, but is Liferay the right product for UEA? This report considers alternatives to Liferay as product and concludes that Liferay provides a cost effective solution, with scope for moving to a different platform in the future.

The overall recommendations from the pilot are:

- Progress Liferay into full production service for research websites.
  - The service should include providing a consultancy and help for researchers wanting to set up websites.
- Encourage new research websites to use the service, and migrate existing websites to the platform over the next few years.
- Following migration, retire the cPanel service, moving any remaining sites to off-site cPanel (or equivalent) hosting.
- Pilot school websites on Liferay.
- If successful, plan to migrate all departmental content from Polopoly, flat-file and local web services to Liferay over the next few years.
- Following migration, retire Polopoly service.
- Migrate The Portal content (excluding teaching and learning content) to Liferay, and retire the Blackboard Community module from the Blackboard suite.
- Review NetCommunities functionality with a view to migrating to Liferay.

## Background

The Liferay pilot aimed to allow research projects to create websites more quickly, simply, cost effectively and more functionally rich than possible through existing centrally supported approaches.

Currently, if researchers want their website to use branding other than corporate UEA, or functionality not provided in Polopoly, central services can only provide hosting space. This burdens the project with the cost and time of implementing (or buying in) their own content management system. The resulting environment is both problematic and expensive to support centrally, and runs a high security risk. This model also makes it difficult to 'cross-sell' between research websites and the main UEA site, missing out on potential gains in search engine positioning and promotional opportunities.

This pilot aimed to establish an alternative: a single, centrally managed, system that allows for multiple sites to be hosted, with individual branding, a breadth of built in functionality (wikis, blogs etc) and extensibility allowing additional bespoke components to be created.

In 2010, based on initial assessment of popular content management systems, Liferay was chosen as a suitable candidate. Combining a content management system with portal container, Liferay is standards compliant and extensible in a modular way, while offering a smaller learning curve than some competitors.

Ultimately, the vision is that creation of websites for research projects will be an automatic process, which is part of the normal flow of setting up a project.

## Report Structure

This report consists of:

- A review of the work completed during the research website pilot
- An in-depth review of Liferay as a product, taken from the point of view of:
  - The public interface
  - Content Editor or member of a 'community'
  - System Administrator
  - Infrastructure

Each section lists work completed during the project and recommendations for taking the project forwards.

## Review of Research Websites Pilot

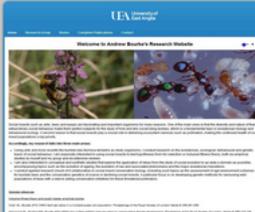
### Websites Created

During the pilot, we created a 'UEA Web Community'<sup>1</sup> site (or 'community' in Liferay speak) to provide an area for anyone working on UEA's web presence to share ideas.

This site includes a gallery of sites created during the pilot (shown above) which shows the scope of projects created.

#### Websites Created During the Pilot

The pilot was originally intended to focus on research websites. Popularity of the system was such that we allowed extra sites to be created beyond the original scope. These sites are all created in Liferay. In many cases, the only input required from central services was to create the empty site, and briefly train the editor. As part of the pilot, we have created a standard UEA theme for Liferay to match the existing site design, and documented the process for building new themes.

 <p>Site for funding bid for £4M research project (Paul Wright)</p>	 <p>Centre for Competition Policy (Web Team)</p>	 <p>IDEAL2011 conference (Heidi Prada)</p>
 <p>School of Pharmacy, iPad Trial</p>	 <p>Andrew Bourke Research Website</p>	 <p>UEA The Portal 2.0 prototype</p>
 <p>UEA Main website prototype</p>	 <p>Current Students Mobile App - suitable for iPhone viewing etc</p>	 <p>Student Recruitment Mobile App prototype (Marketing) - suitable for viewing on iPhone and other mobile devices.</p>

### The Research Website 'Package'

The original concept was to produce a one-stop package for researchers wanting a website. The pilot has advanced that idea in some areas, but needs additional work. Initially the package will be implemented through a survey document – an extension to existing cPanel registration documentation which determines the requirements for individual research projects.

<sup>1</sup> Currently at <http://ueaweb12.uea.ac.uk:8080/web/uea-web-community> (will move after pilot to address tbd).

The areas which would be included by a 'package' are:

- Themes – define the overall look of the site.
- Site templates – define a set of default pages which can be used when creating a new site
- Page templates – define default layout, portlets and portlet settings when creating a new page.
- Note – site and page templates can only be selected when creating *new* sites or pages, and cannot be applied retrospectively. This may make future layout changes more problematic.
- Page layouts – define a default set of components for a page.
- Automation – creation of sites, allocation of members
- Additional functionality required:
  - Out-of-the-box Liferay (blogs, wikis, authenticated areas, serving rich media content etc)
  - Ecommerce
  - Research related functionality – developed or implemented at UEA for groups of projects – e.g. surveys, collection and display of remote telemetry data.
  - Project specific, bespoke development.
- Policy issues – DPA, FoI, data retention (including life span and archiving of the site), research project data.

Work completed during pilot:

- Create two UEA branded themes: *standard* and *lite*.
- Create site template *Basic Research Website* featuring pages common to most research websites, all pages using the *UEA default content* page template.
- Created page templates, which replicate the vast majority of Polopoly pages:
  - *UEA default Content* - left hand menu (with view option set to current UEA menu style) and content area, using the Liferay provided page layout '*2 column 30%/70%*'.
  - *UEA Content (RH menu)* – featuring left and right menus and content area.

Further work:

- Fully document the options available, and route through them.
- Automate creation process – e.g. a new research project generates a basic site, using UEA corporate branding, with basic information about the project, including lists of staff, with links back to relevant areas of the main website. Projects wanting more sophistication can use this site as a foundation for creating their own site.
- Agree policy for URLs of sites – where the site sits within [www.uea](http://www.uea) rather than having its own domain.

## The 'Lightweight' Research Project Site

During the pilot, we considered if it would be possible to create lightweight sites for research projects with minimal user intervention. This would be generated automatically, triggered by a state change in PMA (to be agreed), the standard website is a very simple, one page site with basic information about the project. Project owners could use these basic sites as the starting point for a richer site.

Features:

- UEA branding
- The title of the project and any abbreviation.
- A description of the project (if held in PMA)
- A list of staff involved in the project, linked to staff profiles on the main website

- List of project sponsors/funders/partners
- Dates the project is running.
- Links to research group, school or faculty pages listing all projects.

This standard site can be customised as much (or as little) as the owner desires – adding extra pages and content, choosing a new theme, adding additional built-in functionality (blogs, wikis etc) and bespoke development of additional components.

Further work:

- Investigate automatic creation of Liferay sites as a data feed from PMA (via IDMS).
- Data feed of group membership from PMA (via IDMS) [or webservice]

## Branding (Themes)

A key requirement of research websites is the ability to have non-corporate branding. This can range from small departures from the UEA branding to completely different layouts and overall look.

Liferay allows a site (or areas of a site) to look different by the use of *themes*.

Themes can alter not only the visual layout and style of the page through css, but the whole structure of the page in the underlying HTML. This should not be confused with the structure of the web site, which is controlled through adding and organising Pages.

Liferay is supplied with a variety of themes, we have created basic UEA templates during the project, and it is possible to create entirely new themes too. Liferay themes are more sophisticated than HTML templates. Creation requires a little specialist knowledge, but skills that should be in reach of most web design companies. Themes can be created ‘by hand’, or with the aid of Dreamweaver or Eclipse. If required, themes can be made functionally rich by embedding portlets.

A theme can be applied at any point. This allows the structure and content of a site to be created and signed off with a standard theme in parallel to any theme design process. This approach also clarifies the brief for a design company – that their deliverable will be a packaged, signed-off Liferay theme. This removes any confusion (or unexpected work) for UEA editors/developers to take an image, mock up, or plain HTML and rework it into a CMS, which is time consuming and error prone.

Creating websites efficiently is, however, about more than technology, but also its process. That process can be time consuming and confusing for users without in-depth previous experience. A key recommendation in this report is the need to develop a clearly documented process for creating sites. This would help users to understand what their options are, what is required of them, who else should be involved, and what the costs might be.

As part of the pilot we have:

- Created a set of basic themes to suit common needs – UEA corporate template, UEA minimal template, etc.
- Made available a set of Liferay community developed themes
- Documented the process of creating a Liferay theme, including a full local test environment setup with Eclipse.
- Worked with an external design agency to create a theme which can be dropped into Liferay with no involvement from the web team.

Further work:

- Establish a group of ‘recommended suppliers’ who can work with non-technical UEA staff to produce Liferay themes.
- Develop a documented process for developing new web sites.
- Expand the range of basic themes, including addition options to existing themes.
- A mechanism to allow approved designers to upload themes as required (by default this requires administrator rights to Liferay, which is a security risk).

## Search Engine Optimisation (SEO)

SEO is about creating pages designed to rank well in search engine results. Liferay has a number of features aimed at SEO:

- Automatic generation of Google/Yahoo site map.
- Friendly URLs – Search engines rank pages better where the web address contains relevant keywords. Liferay has several mechanisms which allow the editor to setup friendly URLs.
- Page Title and meta tags can be specified for each page.
- On-page optimizations. The existing content guidelines will continue to be relevant to writing good copy for the pages.
- Portlet which supports Google analytics tracking.

Further work

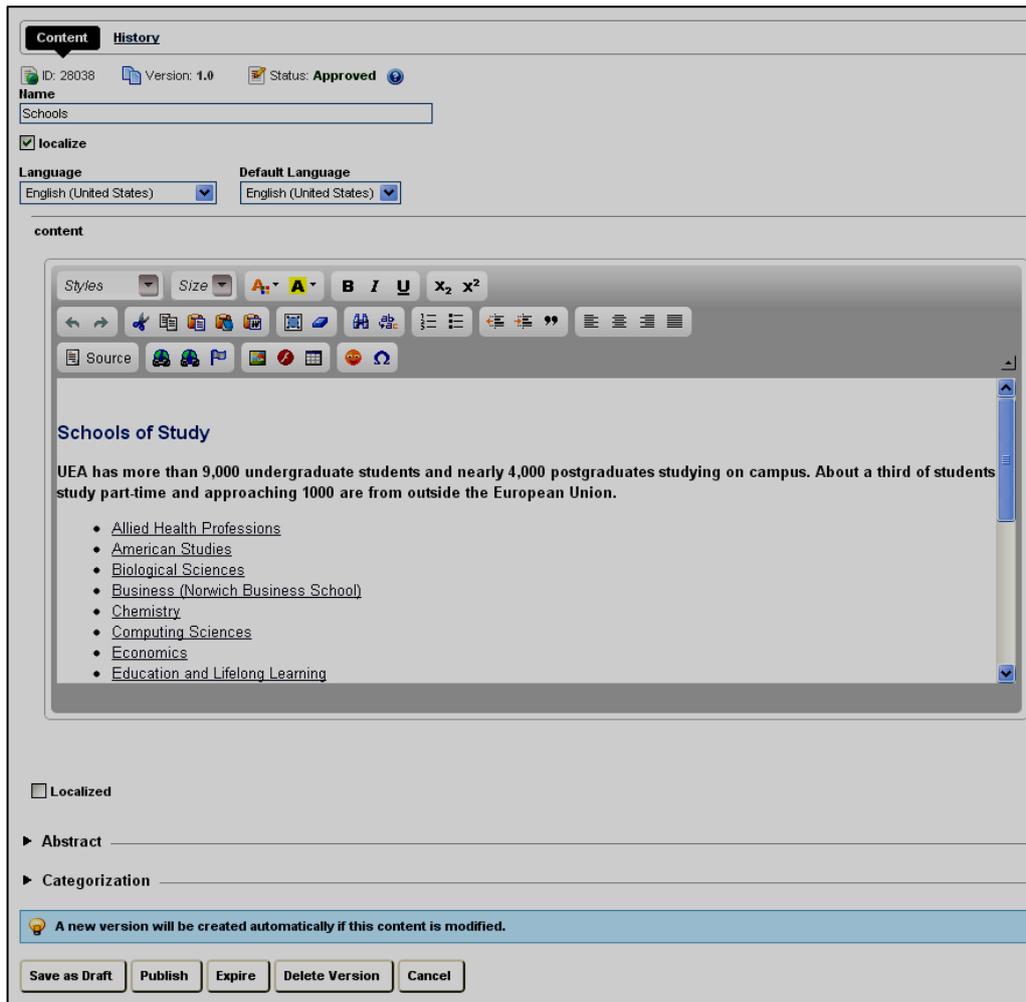
- Review SEO features provided by Polopoly, and comments from SEO audit.
- Implement any outstanding features in Liferay.

## Web Content Management

Web content management is a significant part of the required functionality. Unlike Polopoly, in Liferay content management is a component. The inbuilt web content management portlet could be replaced by another content management system. The main alternative promoted by Liferay is Alfresco.

Overall, content management is similar to Polopoly, with one key difference – pages are not the same as content. That is, a content portlet must be specifically added to a page. Pages can be created that contain no content portlets at all, or that contain multiple content portlets. This has considerable advantages in flexibility in many ways, but could frustrate users used to creating a page and content at the same time. A page template ‘UEA Content’ has been created to simulate the Polopoly approach.

Editing content in Liferay in ‘on-page’. Unlike Polopoly there is no separate editor application. Editors navigate to the page they want to edit, sign in, and – assuming they have suitable permissions – click the edit icon for the relevant content. Editing is not ‘in-place’, but in a separate page. Overall editing content is intuitive and straightforward.



Liferay adds a toolbar to the top of the current page for authorised users, which makes adding new portlets to a page easy, and allows quick access to the ‘control panel’, which is Liferay’s administrative area. It also allows the user to switch off editing controls, to

get a precise preview of the page for viewers.

## Unstructured Content Management

Liferay’s default content management element is called a journal article. This is simply a section of HTML, and equates to Polopoly’s Standard Article.

This is unstructured content (in the sense that the data is not separated into individual fields).

Journal articles provide the features expected of a content management system – versioning, categorisation, workflow etc. These features are discussed in more detail in the following sections.

## Version Control

Liferay keeps previous versions of content when changes are saved. The *History* tab list previous versions of the content with who made the change, providing an audit of changes, which can be viewed, deleted or copied.

Polopoly has significant issues in its version control handling. Frequently, viewing content creates a new version – even though no changes have been made. Frequently the listed user is incorrect – and changes depending on who is viewing it. This makes it impossible to use Polopoly’s version control history as an audit trail.

## Categorisation (Tagging)

Liferay allows for categorisation of content in three different ways:

- Type – Articles can be assigned a type – ‘Blog’, ‘News’ etc. New types can be added.
- Categories – Articles can be assigned to one or more categories, from an admin-definable tree, similar to Polopoly.
- Tagging – Articles can be associated one or more tag. Tags can be defined by adding a new string into the box, but the interface allows the user to choose from previously defined tags, and even suggest appropriate tags for the article.

An article’s categorisation can be used to limit the content returned in searches and in Asset Viewer.

Further work:

- Agree approach to categorisation. Tagging may be a good option for research sites, but categorisation may be more appropriate for central content.

## Scheduled Publishing

Liferay allows editors to specify a date/time the content will be displayed from, and optionally to.

Liferay also allows for a review date to be specified (a feature requested but not added to Polopoly).

Content can also be expired, which removes it from public view, without deleting the content.

## WYSIWYG Editor

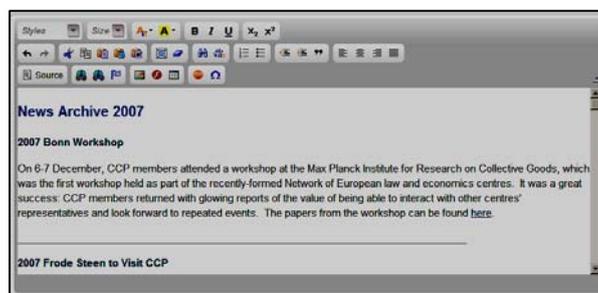
Liferay content is edited using a WYSIWYG editor. During the pilot, we used the default CKEditor, the successor to the FCKEditor used by Polopoly. The WYSIWYG editor is ‘pluggable’, and could be replaced with other commonly available editors, such as TinyMCE.

However whichever editor control is used, work will be required to extend the basic functionality to match the sophistication of UEA’s flavour of Polopoly’s editor control.

One frequent complaint with the Polopoly editor control is that the editor is not completely ‘what you get’, and content may look different in preview. This is due to un-resolvable styles between the Polopoly Editor and web site styles. During the pilot, Liferay’s editor gave much better results, but there may still be some differences.

One aspiration of research websites is to allow academics to edit content themselves, with a minimal amount of

This means that the editor control needs to be highly intuitive, and understand the problem of and pasting from Word of ‘junk’ formatting to problem, and needs



effort and training. editor control needs to not require any HTML. The issue of cut adding large amounts HTML will remain a further investigation.

As part of the pilot we have:

- Created, edited and imported content using the inbuilt editor control.
- Encouraged pilot users to work with, and feedback on the inbuilt editor.
- Investigated a number of potential HTML filters: HTML Purifier etc.<sup>2</sup>

Further work required:

- Agree editor control to use.
- Add functionality to:
  - Reduce the range of controls – e.g. preventing font colour changes, encouraging the user to rely on CSS
  - Validate, sanitise and strip ‘junk’ formatting (as added by Word). Ideally implement as an internal web service.
  - Enforce a level of accessibility.
  - Restrict HTML source editing to selected user roles.

## Workflows

Liferay ships with a default workflow engine called Kaleo. It can generate and reference roles scoped of Organisations, Communities and the whole portal. Kaleo can be replaced with an external workflow engine e.g. jBPM.

Kaleo predefines a workflow called ‘single approver’ which emulates the Polopoly ‘short’ workflow. Additional workflows can be created, however the ‘long’ workflow in Polopoly is seldom (never?) used, and the ‘public’ workflow arguably should be abandoned for audit reasons (i.e. although practically in most situations the editor is the approver, that should be an explicitly given right, not assumed).

Further work:

- Agree and implement any additional workflows required at UEA.
- Investigate if Liferay workflows can integrate with other UEA systems – e.g. a research project created in PMA triggers the creation of a site; a new course in SITS triggers a request for Admissions staff to add additional web information.

## Structured Content Management

Liferay allows for structured content – i.e. where content is separated into fields, mimicking a database. In Liferay, a *Structure* defines the fields to input, and a *Template* defines the presentation of the data – corresponding to the *InputTemplate* and *OutputTemplate* in Polopoly.

*Structures* are defined in XML, *templates* can utilise Velocity (web script language), CSS or XLS. The model allows for nested and repeated items. Changes are made via LR Control Panel, and can be applied without restarting.

---

<sup>2</sup> <http://htmlpurifier.org/comparison> compares 7 html filters. Not surprisingly, their own filter rates best.

Structured content could be used to replicate functionality such as People Pages and Course Catalogue. However, a web service approach may be more appropriate. This allows the data to continue to be stored and maintained in Polopoly initially. In the longer term, we would migrate the data from Polopoly to more suitable data stored (IDMS for people data, stand alone or third party for SITS).

Further Work Required:

- Investigate how to globalise structured content so it is available to multiple communities.
- Implement a web service based approach to People pages and Course catalogue.

## **Bulk Content Imports**

UEA has a significant amount of existing web content both in CMSs and 'flat-file'.

Manually transferring data across would be time consuming. The operation is much more complex than cut-and-paste of content. Typically Polopoly content contains 'junk' HTML – invisible formatting tags, mainly created when a MS Word document is pasted in to Polopoly. This has a negative impact on page rankings. Any hyperlinks, image references, or links to document files (PDF, MS Word etc) in content will point to their original location. Where that content is also being migrated to Liferay, any documents/images must be migrated link must be updated in the content to include both the new location and name (the file name is system generated).

During the pilot we investigated options for importing content automatically from Polopoly.

We made some progress in importing individual content automatically, which needs further work to scale up to exporting whole departments. Ultimately though, a fully automated process is unlikely to be viable. However this may provide a good opportunity to re-evaluate the structure and volume of content published, which may in turn provide SEO benefits.

During the project:

- Migrated 50 pages to Liferay for CCP. The basic migration was done manually, did not include improving HTML quality or fixing links and images, and took 2 days of copy/paste.
- Developed a process to import single-pages from a Polopoly URL and apply purify HTML.
- Migrated content for NBS site to Liferay.

Further work required:

- Export content on a per-department (research project/team etc) basis
- Functionality to sanitise the HTML
- Automatically re-linking hyperlinks to equivalent pages in Liferay
- Automatically transferring images and documents.
- Investigate importing data from 'flat-file' pages.

## **Additional Liferay Components**

Liferay ships with 60+ portlet components, which provide a wide range of functionality from content management to image gallery. This section will concentrate on the most potentially useful components to research projects. Requirements for research websites have been explored in the document 'Research Websites Functional Specification'.

## Shopping Cart

Liferay provides a Shopping Cart portlet, which was investigated during the pilot. However, we were not able to integrate it with UEA's eCommerce merchant.

We also explored EKMPower as a potential provider. However, this was not able to deal with multiple accounts (i.e. unable to include the account code in the transaction details, making it impossible for Finance to allocate incoming funds to the correct account code). This is being followed up with the supplier.

Further work required:

- Follow up EKMPower issue, if successful implement EKMPower portlet.
- If required, find alternative provision for shopping carts.

## Asset Publisher

*Asset Publisher* is a portlet which queries content and displays the match items as a list. This is similar to Polopoly's publishing queues, but has a user interface aimed at editors. This would be useful for listing News or Events items, staff associated with a research project etc.

Queries can be based on an article's tags, categories etc. List items can also be manually selected.

Asset publisher is build on top of Liferay's Asset Framework, which provides the categorisation and tagging functionality seen in content editing.

## Nested Portlets

The *Nested Portlets* portlet acts as a container for other portlets. Editors with suitable permissions can drag and drop new portlets into this container to make complex page layouts.

An example of where this might be useful would be to allow maintenance of one area of a page to be delegated to a promotions company, who then has great control over how that area looks, with no possibility of altering the rest of the page layout.

## Web Form

Provides similar functionality to Polopoly web forms, however the user interface is much more intuitive. In addition to the ability to send data submitted via email, data can also be written to a data file/database which can be viewed/downloaded by authorised users.

This would provide a basic tool for collecting research data in the form on online questionnaires/surveys. More sophisticated options could be developed as plug-ins.

Forms can require user authentication by placing them in an authenticated (private) area of the site, and Liferay can allow users to self-register, which would allow non-UEA people to take part.

## Extending Liferay

Research is about innovation, and research websites should reflect that by offering diverse functionality related to that research. That diversity can be used to stimulate public and media interest in research projects, and in turn, boost SEO across the whole UEA web presence.

Examples could include displaying live telemetry from remote probes, presenting research data in visual and interactive ways, or specialised surveys which collect research data.

Currently projects wanting to innovate only had the option of hosting their own site in cPanel, and developing the entire site as a bespoke CMS. Liferay allows additional components to be 'plugged in' to the existing CMS, which means any development can focus on the project's specific requirements, rather than reinventing the wheel over and over. Overall this approach will be much more cost effective if widely adopted.

This section reviews different approaches for extending Liferay.

The Liferay *Control Panel* allows administrators to upload and automatically deploy additional components. For components downloaded from Liferay's component repository, the system can also detect newer versions, which can be installed with a few clicks.

Components are self-contained, and can be added and removed without restarting the server. The risk of a component destabilising the system as a whole is low, but cannot be ruled out.

## **Existing Compatible Components**

Liferay can consume several classes of existing component. The basic mechanism is the standard JSR 168/268 portlet, but there are portlets available which will act as containers for other component types such as Google gadget/Open Social – i.e. any Google gadget.

There are a few small libraries of JSR 186/286 portlets available. During the pilot we experimented unsuccessfully with compiling and installing those components in Liferay. JSR portlets have been the output of a handful of centrally funded HE projects, from which we may be able to gain some benefit, so we recommend further investigation in this area.

## **Bespoke Functionality**

Liferay provides both a developer API and an IDE (integrated development environment) based on the Eclipse platform. This makes developing new portlets considerably more efficient than developing additional functionality in either Polopoly or Blackboard.

The modular 'plug-in' model means that components can be deployed, or re-deployed. This contrasts with the Polopoly approach where the entire application has to be recompiled, tested and the server restarted for many deployments.

Often a component provides most but not all of the required functionality. With Blackboard building blocks, the only option is to write a new component from scratch. This has the additional problem that the code has diverged from the supplier's codebase, and for each system release, the diverged component needs to be recoded.

Liferay's development model avoids this problem by allowing new functionality to be 'hooked' into existing components. During an upgrade, generally bespoke component will not even need recompiling, as the component code just hooks into the new core code.

During the pilot we created several portals, including:

- RSS feed prototype portlet – replicating functionality in The Portal. Based on the Liferay RSS component, extended for UEA requirements.
- eVision portlet, replicating functionality in The Portal. Written from scratch.

Liferay also offers a 4GL called Site Builder which allows users to create simple database driven portlet applications. However, this type of component will not be readily transferable to other portal containers if needed. This applies for portlets built using any Liferay specific API or extension methodology.

Although Liferay is OpenSource (so the source code is available to developers) so far we have not needed to modify the core code and recompile the product itself.

## Review of the Liferay Product

### Liferay in the marketplace

The market for content management systems is extremely diverse, with hundreds of potential products. The market for Portal container/CMS hybrids is much smaller, and smaller again for java based solutions.

In the last year, presumably helped by the acquisition of Sun by Oracle, Liferay has become the most deployed java-based CMS system. It still trails traditional CMS such as Joomla, Wordpress and Drupal considerably.<sup>3</sup>

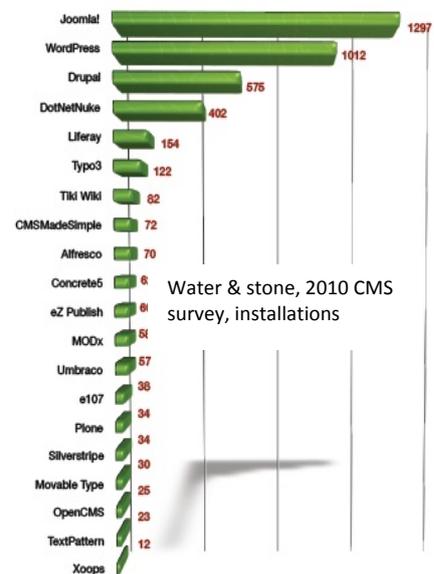
W3 Techs 'Usage of content management systems for websites'<sup>4</sup> still rates Liferay with <0.1% market share. The top of the market is dominated by PHP products (Wordpress 55%, Joomla 10%, Drupal 6%). However, the site also notes that 75% of sites are running PHP (23% ASP, 4% java), and only 0.6% of sites run Tomcat. PHP is standard in most hosting services, where as java (and Tomcat) is non-standard and expensive. The unanswered question is: How many sites are running PHP heavy-weights because they want to, how many are running them because it's most cost effective with their hosting provider.

"Recently Gartner (news, site) placed Liferay in their 2010 leader/visionary portion of the Horizontal Portals Magic Quadrant, making the company the only focused, open source, independent portal vendor in that portion of the quadrant among larger players who sprawl across numerous technologies."

Liferay's client list includes: French Ministry of Defence, Fox (Germany), Lufthansa, Sesame Street, T-Mobile (Czech Republic), Vodafone (Betavine).

### Liferay in HEI Globally

Liferay's website lists 26 HEI users who are using the product to various degrees, but we are yet to find another HEI using Liferay to the extent proposed at UEA. However this may be more of a statement on the fragmented nature of the CMS market, and lack of any specific functionality for HE. Liferay is well used in commercial organisations.



<sup>3</sup> <http://www.waterandstone.com/sites/default/files/2010%20OSCMS%20Report.pdf>

<sup>4</sup> [http://w3techs.com/technologies/overview/content\\_management/all](http://w3techs.com/technologies/overview/content_management/all)

Higher Ed		
<ul style="list-style-type: none"> <li>Abilene Christian University</li> <li>Adelphi University</li> <li>Arteveldehogeschool</li> <li>Central Queensland University</li> <li>eCollege</li> <li>Excelsior College</li> <li>Harvard Business School</li> <li>Indian Institute of Technology (Delhi)</li> <li>La Cite Collegiale</li> </ul>	<ul style="list-style-type: none"> <li>Louisiana State University</li> <li>Penn State University</li> <li>Point Loma University</li> <li>Queen's University</li> <li>Universidad Complutense de Madrid</li> <li>Universidad de Alcalá</li> <li>Universidad de Etea</li> <li>Universidad de País Vasco</li> <li>University at Manitoba</li> <li>University at Albany</li> <li>University of Bolton</li> <li>University of California San Diego</li> </ul>	<ul style="list-style-type: none"> <li>University of Nebraska</li> <li>University of Portland</li> <li>Victoria State's Education Dept in Australia</li> <li>Virginia Commonwealth University</li> <li>York University</li> </ul>
		<p><b>K-12</b></p> <ul style="list-style-type: none"> <li>Discovery Education</li> <li>Madrid School System</li> </ul> <p><i>This is not an exhaustive list. Please email <a href="mailto:pr@liferay.com">pr@liferay.com</a> for more information.</i></p>

Central Queensland and York (Toronto) are both using Liferay for student portals.

York University (Toronto, Canada)<sup>5</sup> has been doing some interesting work in linking Liferay to their SIS and other admin systems. As part of the Pilot, we contacted York, but have not yet received feedback.

## Documentation, Support and Community

Liferay is an OpenSource product with a commercially supported option called Liferay Enterprise Edition.

Support for the open source product is via a forum on the Liferay website, on which we posted questions during the pilot. The response was mixed. In some cases we had dialogues with key developers at Liferay, in other cases the question went unanswered. A reason for this may be due to the nature of UEA's involvement. To foster a good community, we need to 'give back' into the development community, rather than just expecting answers, through contributing to support forums, an donating some code back.

There is also an issue tracking system, which we used to request SAML2 SP functionality, a wiki and community events.

*Recommendation:* UEA commits to contributing into the Liferay community. This could be through blog entries and contributing selected, and non-UEA specific documentation and code.

Support and licensing details for the commercial products have been requested from Liferay UK.

Commercially, Liferay has over 80 global partners providing support, consultancy and development services. Cignex, Fulcrum and Nous Infosystems provide Liferay support with UK based offices, plus there a number of other companies who offer Liferay consultancy/development without being full partners.

*Example Rates:*

MagicWebSolutions (<http://www.magicwebsolutions.co.uk/liferay-development-services>)  
 £480+VAT per day for a technical lead/senior consultant £320+VAT for a developer

<sup>5</sup> Case Study [https://www.liferay.com/c/document\\_library/get\\_file?uuid=0f9f9d16-a421-43a7-965b-e333a0d073f9&groupId=14](https://www.liferay.com/c/document_library/get_file?uuid=0f9f9d16-a421-43a7-965b-e333a0d073f9&groupId=14)

## **UEA Training Requirements**

Compared to CMS systems used at UEA, Liferay is a more intuitive product, aimed at a less technical audience, and consequently should require much less training investment for end users.

For implementing at UEA, the following courses are recommended for project team members:

- Introduction to Liferay
- Liferay system administration
- Liferay development

Training and support for the wider community needs to be considered. Any training would be able to be much lighter touch than for Polopoly (a 1 day course).

Liferay includes some user documentation (e.g. a user guide and introductory videos). We should avoid creating a UEA version of this documentation. Instead we should rely on Liferay produced documentation in the main, only creating documentation to explain UEA's implementation of the system. For example, how to tag and classify content, or how course catalogue should be used. Interactive approaches – screen casts, lunchtime workshops etc – should be used to encourage users to engage with the system.

We would require a wide program of training to adopt the system, however, the expected training requirement is far less than that for Polopoly, and less that retraining for the next version of Polopoly.

During the pilot, feedback on the product, and its usability was very positive. Everybody rated it much more usable than Polopoly. Several editors were able to create rich sites with a minimal input from the web team, far more quickly than could have been achieved with existing routes.

## **Liferay System and Developer Documentation**

Developer/ SysAdmin documentation is available from the website, but not comprehensive enough. Setting up a development environment proved extremely time consuming, but is now documented. Documentation of the Liferay api is lacking, although as various elements (such as portlet development) are 'standards based', so additional documentation is available on other sites. There is also an amount of example code, including examples of themes and portlets.

## **ISD Documentation**

Further work:

- ISD project closure documentation:
- DR plan
- Security Review and Risk log
- Equality and Diversity
- BC checklist
- Service Level Definition.

## **Functionality Highlights**

This section lists potentially interesting functionality which warrants further investigation.

## **Liferay Social Office**

Social Office is a social collaboration solution for the enterprise. Functionality includes:

- Documents: Shared document repository with Microsoft Office integration
- Communication: Team calendars, webmail, task management and contact management
- Collaboration: Integrated threaded discussions, blogs, and wikis with related content and federated search
- People: Searchable LinkedIn-style User Profiles with user tagging, user search, friends, and integrated chat

A user complaint regarding Polopoly is the difficulty of integrating Word documents, so we recommend investigating this functionality further.

Future work:

- Explore possible benefits of Liferay Social office.

### **Direct content access/editing (webDAV)**

Liferay provides the ability to edit documents and images directly through the webDAV protocol – which presents a chunk of web space as a desktop folder. This may allow for several features requested by editors including:

- Bulk upload of images
- Direct editing of Word (and other MS Office) documents in CMS
- Sharing documents within groups (for administrative and teaching purposes)
- Providing Dropbox-like functionality (storage and file size limited by available SAN space)

Future work:

- Investigate feasibility including security, storage and usability implications.

## Technical Aspects

### Authentication and Users

Research projects typically involve partnership or close collaboration with colleagues in other institutions, industry, and in some cases the general public. This may be as part of the research – a member of the project, a survey posed to the public - or through in the publicity of the project. So the portal container needs to be able to authenticate both UEA accounts, and non-UEA accounts. UEA Web Strategy is moving towards ‘authentication as a service’ using SAML2, which means web applications don’t authenticate users themselves, but pass control over to a central service called an Identity Provider (IdP). UEA’s chosen IdP is the Eduserve product OpenAthens.

Further work required:

- Add SAML2 service provider functionality to Liferay. This is available as a Liferay patch, but needs testing.
- Establish the OpenAthens service.
- Enable the OpenAthens service to authenticate not just UEA users, but external users. Where possible, use existing identities – e.g. for a user from within the UK federation, we can ‘trust’ their home institution’s authentication credentials. – e.g. For FaceBook, Google etc accounts we ‘trust’ and delegate authentication to the provider. These users would typically be treated with a lower ‘level of confidence’ (LoC) to UEA users, the extent of their permissions can be set per site.
- Provision Liferay user records for authenticated users automatically.
- Prevent current Liferay practice of storing hashed passwords in the Liferay database.

### Authorization

Liferay uses a role concept for authorization to areas, functionality and editing rights i.e. the user is assigned to a role, and roles given permissions. This is similar to the role-base security in Blackboard and Polopoly.

Liferay’s authorization model can be a little difficult to understand, mostly due to lack of documentation of the built in roles. However the model is flexible and extensible. We are able to provide better authorization control than Polopoly – e.g. users can be assigned to multiple area of the site, without the need for second user accounts (CMS security risk log).

### Standards Adherence

Liferay uses the following standards:

- AJAX
- iCalendar & Microformat
- JSR-168
- JSR-127
- JSR-170
- JSR-286 (Portlet 2.0)\*
- JSF-314 (JSF 2.0)
- OpenSearch

## Robust Architecture

For the pilot, Liferay was installed with the application tier and database on a single server. This is not a model generally considered best practice, but allowed for proof of concept.

Ensuring a robust environment is vital to providing high availability of the website. This ensures continuation of the service under the following conditions:

- Unplanned service outages – database corruption, loss of database server or connectivity, loss of network connections (on campus, or campus link), loss of application servers, and security incidents (i.e. hacking).
- Planned outages – Low level backups, database security patches, OS patches, server moves.
- Upgrades – system upgrades, component upgrades (themes, portlets etc).

It's important to note that, as with the current CMS, the level of service possible under these conditions is likely to be less than 'full' service. The key will be to degrade the service gracefully. To do this well UEA needs to identify and prioritise key information and functionality, and provide suitable alternatives.

An example might be in the event of failure of the off-site link, where we would aim to switch over to an externally hosted version of the site. The externally hosted site may only provide a limited number of non-personalised pages. Functionality such as searching of course catalogue data or signing up to an open day may not be possible to support. So the relevant pages should show suitable alternatives – a telephone number for open day enquiries, or a link to a PDF of the prospectus.

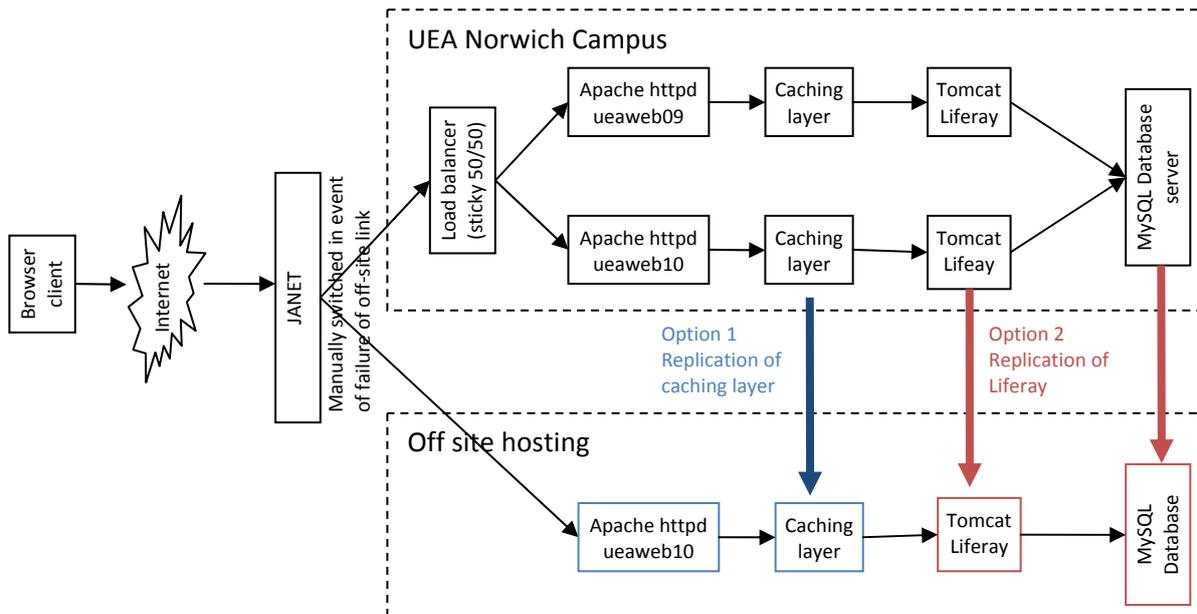
For comparison, Polopoly's current architecture consists of two load balanced web servers, a single application server with warm standby and an oracle database server. Polopoly operates a front-end cache system which allows read only web pages can be available (if cached) in the event of a back-end outage.

It could be argued that the complexity of a J2EE architecture increases rather than reduces risk of outage.

The target for the website should be 100% uptime. Unlike internal systems where scheduled downtime is tolerated, the website is UEA's public face. From the point of view of reputation and confidence of prospective students and investors, the website should be continually available. The goal is that scheduled maintenance, including upgrades and low level backups can be completed without website outages. Clearly, retaining the full functionality of the site, with personalised content or write operations depending on a database, isn't always feasible during maintenance or failure of the connection to the database. But we should aim for a subset of 'important' pages to be available continually.

This should also include the event of a failure of the off-site internet link, which implies creating a copy of the website externally, and employing switching at Janet to redirect traffic to it quickly (rather than re-pointing DNS, which can take hours/days to filter around the internet). Initial investigations with Janet have determined that this would only be possible following the planned upgrade of DNS/DHCP (currently July 2011).

With all this in mind, the long-term architecture would be as shown below.



Further work required:

- Consultancy from Liferay
- Upgrade to DNS/DHCP by Network team
- Agreement with Janet, and work by Network team to set up switch
- Investigation into best approach to offline site (a mirrored page cache, or DB replication)
- Implementing off-site version of web site.

## Choice of Database

During the pilot, a MySQL database was used. MySQL performed adequately. However following discussions with CIS's DBA team, MS SQL Server has been chosen as the live database platform. The managed instance of MySQL does not offer the library required for transactional backups, which would mean nightly backups would be the best that could be achieved. MS SQL Server is running in a clustered environment, which should ensure high availability of the database layer.

MySQL databases will still be used for local development. Liferay offers a mechanism to transfer databases between platforms. This is a significant advantage to Polopoly, where moving data from the Oracle database to another database (for testing purposes) is not possible in the current version at UEA.

As with Polopoly, Liferay is database agnostic, but unlike Polopoly, Liferay relies on standard, proven, libraries such as Hibernate to interact with the database, which ensures database independence without compromising performance.

Further work required:

- Create test and live database instances on MS SQL cluster.
- Migrate pilot MySQL database to live, and copy to test.
- Agree backup regime, and calculate ranges of RPO and RTO's for SLD.
- Investigate replication of Liferay database.

## Caching

One of the major benefits of Polopoly is its caching mechanism, which allows a reduced, read-only version of the site to run without database availability.

Liferay offers opportunities for caching at several levels. However, the documentation is not especially forthcoming as to which option(s) would be best for UEA, so we recommend using a Liferay consultant.

The options include:

- Caching whole pages using a front-end cache such as squid. This would be suitable for public pages with no element of personalisation.
- Mirroring pages to an off-site failover service. Again only suitable for public pages with no personalisation.
- Caching database objects, to ensure continuing service in even of loss of database. Suitable for a read-only view of public and personalised pages.
- Caching portions of the page, including portlet contents. Flexible, but complex and needs support from the application layer.

As with all caching, content must be accessed to be in the cache.

Further work required:

- Liferay consultancy to advise on best route for caching.  
Solution should ensure performance at least as good as the Polopoly solution – i.e. web pages continue to be served in the event of database outage. The service should degrade gracefully, so 100% of the website functionality will not be available