CERTIFICATE OF AUTHORSHIP/ ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

Production Note:
Signature removed prior to publication.

Lance Scaife-Elliott

27 February 2013

MASTERS IN DESIGN (RESEARCH) //

THE TRANSITION FROM STATIC TO DYNAMIC WEB DESIGN:

Investigations and directions for the implementation of a low cost cms in educational curricula.

University of Technology Sydney

Chief Supervisor: Darrall Thompson
Co-Supervisor: Dr Gerhard Bachfischer
Student: Lance Scaife-Elliott
Ethics Clearance: 2009-146A
Lance.Scaifeelliott@student.uts.edu.au
01.03.2013

Abstract

Recent developments in open source software are enabling web designers to configure dynamic web sites quickly, easily and cheaply.

The promise of open source CMS software, ultimately is a web environment in which users are programmers. On the surface it would appear that web site creation is moving in the direction of no longer needing expensive custom solutions by professional web developers. These new software systems use a variety of approaches from mere configuration of the system to script implementation. They may require limited programming experience beyond the web designer's CSS and XHTML capabilities. Based on the Open Source platform, PHP and MySQL offer an easily learned and supported framework to web designers. Web designers now have a tremendous opportunity to harness open source content management systems and extend what they are offering to clients and their employers. Increasingly, it will become the duty of web instructors and of tertiary institutions to equip students with these capabilities, so that they can function effectively in the new environment.

This research uses surveys, interviews, and a research workshop with longitudinal results to explore the questions posed. The collation of materials is hosted on a dedicated web site.

The findings of this research indicate that students would benefit from a practical approach to web development through the introduction of a CMS in their studies. These are skills employers consider valuable.

The main conclusions and recommendations are firstly that many design students see themselves as performing an exclusively aesthetic role in the development of web projects, and secondly, that these students, with proper instruction in the usage of open source CMS's, are more than capable of making and implementing decisions concerning the functionality or programming of web projects. As a result of this study it is my belief that design students with a deeper understanding of open source CMS's will be able to occupy a role currently falling between the occupations of graphic design, web development and object-oriented programming. It is my hope that this project will inform and educate so that the wider design community, and in particular design educators, can make more informed decisions about the transition from static to dynamic web design.

CONTENTS

List of Figures 6

Motivation and background 7

01 Research Questions and Methodology

Delimiting the study 9

Structure and methodology 10

The WhichCMS Workshop activities 11

Hosted assistance for CMS implementation 14

Definitions and terms 15

Defining Web 2.0 and dynamic development 16

02 Literature Review

Introduction 24

Static and dynamic web sites 25

Designer or developer? 27

The designer as generalist vs. specialist 29

The rise of web 2.0 31

Perceptual changes in web design from brochure through to dynamic 33

What are the key differences between static and dynamic web sites? 34

What are the drivers in the transition of static to dynamic web sites and the range of considerations? 35

Matrix scoring 36

The current state of play in the CMS community 37

Open Source CMS Market Share Report 38

Content technology vendor maps 42

Assessing CMS usage 43

Schools keeping up with industry 47

What is being taught by institutions in the Sydney basin? 49

What is the range of considerations in teaching a CMS? 50

Analysis of 28 CMS solutions 52

Selecting your CMS 60

start of a section

Original research

Discussion

The web site accompanying this thesis at:
http://www.independentdesign.com.au/whichcms

03 Results and Discussion

Why learn a CMS? 63

The teaching of web design 64

Research design 64

WhichCMS online training 66

Are the outcomes of schools meeting the needs of employers? 69

What are the views of employers regarding the knowledge and skills of web design positions? 70

Interview summaries with local industry 71

CMS Questions 73

04 Conclusions and Recommendations

Conclusions to this study 76

Recommendations for the education sector 79

A way forward for teaching web design 83

Study paths for designers and teachers 83

A personal approach to CMS selection in education 84

CMS market overview 90

Ten CMS selection recommendations 91

Appendix 1 Interview data 93

Appendix 2 Survey Instruments 97

Appendix 3 Web site Screenshots 112

Appendix 4 Bibliography 113

LIST OF FIGURES

Figure 1: Poster editions 1 and 2 11

Figure 2: Two of 70 slides - Workshop CMS PowerPoint 12

Figure 3: WhichCMS workshop in progress with 13 participants 12

Figure 4: Online site - survey Gizmo 13

Figure 5: cPanel interface with Fantastico and MySQL Database Wizard 14

Figure 6: The Internet - authors diagram 15

Figure 7: The Taxonomy of Food 21

Figure 8: The visual schema of development - authors diagram 22

Figure 9: The make up of a CMS - authors diagram 26

Figure 10: Open Source development - authors diagram 28

Figure 11: Flickr.com 32

Figure 12: A development framework - authors diagram 33

Figure 13: Dynamic site structure - authors diagram 34

Figure 14: Frameworks distribution - Frameworks.com 35

Figure 15: A typical CMS - authors diagram 35

Figure 16: CMSmatrix.com 36

Figure 17: CMS and blog market share - Mosher 2008 37

Figure 18: CMS and blog market share 38

Figure 19: Average weekly downloads (Water & Stone, 2009) 39

Figure 20: Installations as per Survey (Water & Stone, 2011) 40

Figure 21: The Alexa One Million (Water & Stone, 2011) 40

Figure 22: Books in Print (Water & Stone, 2011) 41

Figure 23: Brand Sentiment (Water & Stone, 2011) 41

Figure 24: Vendor Technology Map - http://www.realstorygroup.com/vendormap/ 42

Figure 25: Google Trends: Wordpress, Drupal & Joomla! 43

Figure 26: BeBit 2010 - Mark Kofahl from SIteFlex 43

Figure 27: Wappalyzer 44

Figure 28: Wappalyzer - Web sites 44

Figure 29: Alexa comparison of site traffic - June 2011 45

Figure 30: Gartner's Magic Quadrant "Visionaries" Category 46

Figure 31: Courses being taught in the Sydney basin - authors table 49

Figure 32: CMS news related web sites 51

Figure 33: www.sitepoint.com 65

Figure 34: WhichCMS web site - www.independentdesign.com.au/whichCMS 66

Figure 35: Web Traffic to the WhichCMS web site 67

Figure 36: Software rank order 67

Figure 37: Online hosted workshop videos 68

Figure 38: Survey question "Do you feel that educational institutions are teaching the skills to match the demands of employers?" 69

Figure 39: Survey question "Do you feel that educational institutions are teaching the skills to match the demands of employers?" 70

Figure 40: Survey question "Could a CMS influence your desire to code in a web scripting language like HTML, CSS, PHP or Ruby?" 74

Figure 41: Survey question 'WhichCMS did you first select?' 79

Figure 42 : Educational CMS directions - authors diagram 83

Figure 43: CMS directions - authors diagram 89

Figure 44: The CMS Market - authors diagram 90

MOTIVATION AND BACKGROUND

Working as a freelance graphic designer and teacher I have previously created web sites that are static. This means that the finished product is unable to be altered by the client, or content easily updated. Clearly, most clients have a need to keep the information on their web site current for visitors, so this situation is less than satisfactory. I have always wanted the ability to create larger client-maintained web sites, but to achieve that I had to forgo the opportunity to be a developer, involving the loss of both potential income and control of the web site. Now, the field of web design has well and truly changed, with many new software applications in the open source realm. My own early experiments with open source content management systems (CMSs) led me to consider whether the student designers in my classes could improve their understanding of web design, and increase their chances of future employment by extending their skills into web developer territory.

The motivation for this masters research was stimulated by a request from a client with a community-based web site seeking to extend their web capabilities far beyond what I could create. The budget for the project was limited, so approaching a commercial provider and developer was not an option. I looked for cheaper solutions and discovered PHP, a web programming language, which is Open Source and available free. Built on top of the PHP framework are a number of open source tools that enable the configuration of a dynamic web site for a designer. Systems include *Joomla!, ExpressionEngine, Drupal, MovableType* and *Wordpress* to name but a few.

The speed at which web development tools are evolving means that there is also a great opportunity to teach students cutting edge skills that could significantly enhance their employment opportunities. Are teachers of web design missing an opportunity to motivate design students in the direction of dynamic web site development?

> Teaching students open source software applications may encourage them to go further with programming.

Working with open source software might give them a broader range of capabilities without their being hamstrung by a lack of straight programming skills.

Design and development for the Internet is an exciting area, particularly for those teaching it. On graduation, students expect that their skills will meet the skills required by industry for employment. Currently, students studying web generally fall into two categories, designers and developers.

The rapid evolution of the web means that both professional and educational arenas are intertwined and vital to one another. However, the education of designers differs greatly from that of developers who often have computer science degrees or programming qualifications. As new software tools are created employers will be able to exploit these for commercial benefit only if their newest employees have learnt these emerging technologies. Currently however, educational institutions teaching design students naturally tend to avoid the scripting and programming required for dynamic web sites.

In my experience as a teacher, many design students are resistant to learning programming. Design students see themselves as creative practitioners, and often put programming in a category of knowledge that belongs to 'techies' or IT students, and not to them.
A focus on open source software could help to solve this problem, by making them aware of the 'middle ground' in web programming - CMS. This in turn could help to encourage some of them to see themselves as developers not just designers, thus giving them increased agency and potential in their professional careers.

In early June, 2008 I instructed 14 students over four hours in a web design class. The students took a static web site and made it dynamic with the *ExpressionEngine* CMS. With a little scripting *(copy and paste from the manual)* the students were easily able to create a dynamic web site which the client was able to update.

> The response from the students was immediate and overwhelming.

Over subsequent weeks the students built entire web sites in *ExpressionEngine* and were highly motivated. One student then experimented in Wordpress and another in *Drupal* as their preference was for limited programming. Another student was even interested in programming with PHP and an open source PHP based CMS was then investigated. It seemed only logical to investigate this area further.

RESEARCH DESIGN AND DEFINITION OF TERMS

In this section:

Delimiting the study	**9**
Structure and methodology	**10**
The WhichCMS Workshop activities	**11**
Hosted assistance for CMS implementation	**14**
Definitions and terms	**15**
Defining Web 2.0 and dynamic development	**16**

01 RESEARCH QUESTIONS AND METHODOLOGY

This research arises from my concern about the plethora of open source 'solutions' for web design and development, with minimal research about the topic to inform both design businesses and design educators about the transition from static to dynamic web design.

Delimiting the study

The main aim of this thesis is to explore the effectiveness of open source software in assisting designers to enhance their understanding and use of emerging technologies in dynamic web site development. It also aims to provide recommendations and guidelines for possible implementations in tertiary level curricula that may also progress the educational field in this area.

In investigating the transition from static to dynamic web design the following research questions were framed:

1. **What is the current situation with regard to the emergence of dynamic web design?**
2. **What factors influence the selection of CMS solutions for different groups; designers, teachers, employers and students?**
3. **How are current web design teachers engaging in the transition from static to dynamic web design both in practice and teaching?**
4. **What are the factors influencing designers in the engagement of scripting or programming?**

Questions 1 and 2 are partly addressed in the literature review in Chapter 2. Questions 3 and 4 are investigated through surveys, interviews and a workshop designed to specifically explore student responses longitudinally.

In delimiting the research I have:

- **Restricted the study to educational institutions in the Sydney basin area.**
- **Focused on the most popular open source content management systems (CMS)**
- **Restricted the longitudinal study over a 3 month period.**

Structure and methodology

Dynamic web design and development is a rapidly evolving area and this research has been approached from a number of directions to cover the range of questions posed.

This thesis was developed by examining a number of methodologies and then selecting what I felt would give the clearest results to the questions I was seeking to resolve. I started with *explorative research* which assisted me in defining my research perimeters. I felt that the study would be qualitative in nature based upon my experience with students and their availability.

Initially I approached this research with a traditional methodology however I chose to go with a grounded theory methodology as I felt that I already had a good understanding of the topic area, the audience and possible results.

I really like the grounded theory approach as it allows the researcher to work inductively to generate theories strictly from data. (O'Leary 2004). As Urquart points out; "From the time the seminal book, The Discovery of Grounded Theory (Glaser and Strauss 1967), was published, there have been countless applications of GTM (Grounded Theory Method), but also many adaptations and evolutions of the method" (2013, p.3).

My empirical research involved a number of strategies in order to develop my data. I conducted interviews with a number of industry professionals, held a workshop with graphic design students and conducted anonymous longitudinal surveys to follow up this workshop. I also conducted an examination of the extant literature - much of which can be classed as grey literature because of the immediate and practical nature of the information. My methodologies are backed-up by my ethics research clearance 2009-146.

Grey literature here refers to studies published outside widely available journals. As Cooper et al point out: "Grey literature comprises an increasing proportion of the information relevant to research synthesis; it is likely to be more current than traditional literature and channels for its distribution are improving" (2009, p.104).

The literature review has been developed from a variety of materials including topical books and software manuals, academic published papers and more informal but 'of the moment' web blog postings from a range of design and development backgrounds. Additionally commercial organisations publish their own insights as a means of enlightening their customers and these come in the form of reports such as Water and Stones *Annual CMS Report* and the Gartner Report's *Magic Quadrant* to software tools such as *Google Trends* and *Alexia* where the researcher can enter search parameters and draw their own conclusions based on the results.

The research workshop focused on introducing graphic designers to six possible CMS options selected to cover the gamut of software tools. The workshop titled *WhichCMS* ran for five hours and included a CMS overview lecture, self directed video tutorials and a series of longitudinal surveys over three months.

To understand the business perspective I conducted two interviews with industry practitioners whose organisations develop and deploy web sites using both open source and commercial software. The research additionally targeted industry with an anonymous online survey via 280 companies targeted through e-mail located in the Sydney basin.

Educators were contacted via an online anonymous survey to understand how and which content is taught in their courses.

To support the surveys I built a web site designed to inform the support and extend the research collation. Training videos and resources are also available online. http://www.independentdesign.com.au/whichcms

The idea of using a workshop came from the requirement to obtain data about factors for questions 2 and 4. The method chosen was to engage directly with students to instruct them in this area and then follow up three months later to gauge the impact or not of their exposure to the content.

The WhichCMS Workshop activities

In order to acquire more data from students outside the scope of a survey or anecdotally, I designed a series of video tutorials to engage participants in the workshop and longitudinally track their progress. This research has provided a personal insight into the critical choices made when creating a dynamic web site, particularly technology versus creative decisions. Currently CMS solutions are not taught to design students and it is this researcher's hope that this data might influence the educational arena and enable more designers to transition from static to dynamic web sites.

The workshop covered a broad range of CMS development - six systems in total. No web site design or development experience was necessary to participate, as designed XHTML templates were provided. These systems range from simple to complex so it is envisaged that at least one of the systems present would be suitable to all participants. Participants may have some or no experience in CMS development and participants were students of web design, graphic design, architecture, screen animation, fashion, industrial design and web development.

The workshop consisted of:

- One hour CMS overview with examples in each CMS
- Two minutes video slideshow about the CMS
- My First Choice survey
- Approximately 45 minutes of video instruction for each CMS
- Workshop survey
- Post workshop survey – August 1 2010

I created two posters promoting the workshop; the second after receiving a low level of response for participation in the event.

The workshop poster was emailed to department heads and individual lecturers of:

- Billy Blue
- Shillington College
- Martin College
- Qantm College
- Raffles
- University of Sydney
- University of Newcastle
- University of New South Wales - COFA
- University of Western Sydney

Disappointingly only 1 student signed up after the first poster had been displayed for 3 weeks and I had to postpone the workshop. Two contributing factors may have accounted for the poor interest **1)** the workshop was scheduled during student holidays and **2)** the initial poster communication was not effective, and lacked clarity.

Figure 1: Poster editions 1 and 2

Online web site **http://www.independentdesign.com.au/whichcms/** supported the workshop with the following online resources:

- Blog
- CMS official review links
- CMS tryout links
- CMS blog posting comments
- CMS Training links
- Video training links

The workshop consisted of an introductory lecture with slides explaining the basic concepts of a CMS and an overview of the six CMS options available for the practical component.

http://www.independentdesign.com.au/whichcms/whichcms_workshop.pptx

Figure 2: Two of 70 slides - Workshop CMS PowerPoint

Following the workshop, participants undertook a self-directed video tutorial approximately 45 minutes in duration and generally participants tried two or three CMS options of the available six during the workshop. More than 80 hours was spent recording the video training. The training was additionally available online so that participants could further undertake the exercises at their leisure. The titles are located online at the following web addresses:

- What is a CMS? **http://vimeo.com/12526857**
- Things to consider when choosing a CMS **http://vimeo.com/12601525**
- Introduction to *Tumblr* **http://vimeo.com/12575490**
- Introduction to *Concrete5* **http://vimeo.com/12575514**
- Introduction to *ExpressionEngine* **http://vimeo.com/12575535**
- Introduction to *Wordpress* **http://vimeo.com/12575619**
- Introduction to *Joomla!* **http://vimeo.com/12575646**
- Introduction to *Drupal* **http://vimeo.com/12575660**
- FTP uploading **http://vimeo.com/12576719**

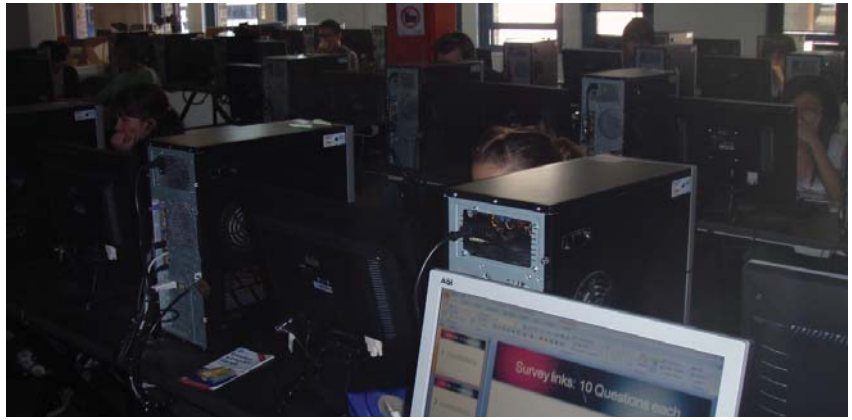


Figure 3: WhichCMS workshop in progress with 13 participants

Online Data Collection

In order to collect data from *WhichCMS*? participants and targeted groups, I created a series of surveys via online specialist - *Survey Gizmo*. Survey data was anonymous and available over a period of six months. I sought online data completion from the following groups:

- 40 — Lecturers from design schools in the Sydney basin
- 180 — Employers selected from the graphic and web design sections in business directories
- 40 — Designers selected from graduate graphic design students
- 13 — Workshop participants who attended and undertook CMS video training

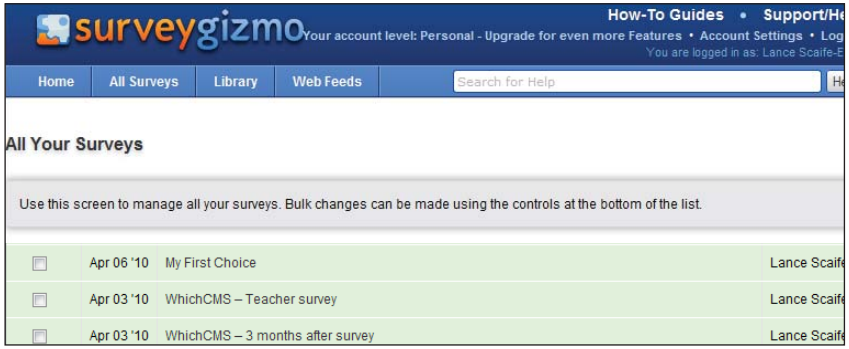


Figure 4: Online site - survey Gizmo

Hosted assistance for CMS implementation

Often users trying to install a CMS experience difficulty with the core component, the database. This is often because of the different server environments provided by *Internet Service Providers* (ISP) and restrictions placed upon users by the ISP because of the users selected plan. For example a user might have access to hosting space with their home connection but that hosting space might not allow for the creation of a database... ..or they might only have one database allowed as per their account with the ISP – the more you pay the more capabilities you get, enter *cPanel*.

cPanel is a web hosting interface associated with the users internet account which provides a control panel to simplify and automate many common activities a user would like to do. Of significance to this research *cPanel* assists in the set-up of databases and even some CMS solutions so that a user with no technical knowledge can install a CMS in a few minutes and be guided through the process.

Components of a *cPanel* interface can depend of the account settings, administrator, reseller and client. When installing a CMS a user would benefit from using the *MySQL Wizard*, a step-by-step database creation tool which would then be called upon by an installed CMS.

Fantastico also available in *cPanel* and is a 'select application/CMS and install' tool. *Fantastico* does not provide every access to every CMS however it does include many of the most popular ones based on the users current configuration and as shown in Figure 5 - *Drupal*, *Joomla*! and *Type3* are all but a click away. If you are planning on operating a CMS *CPanel* is worthwhile looking into.

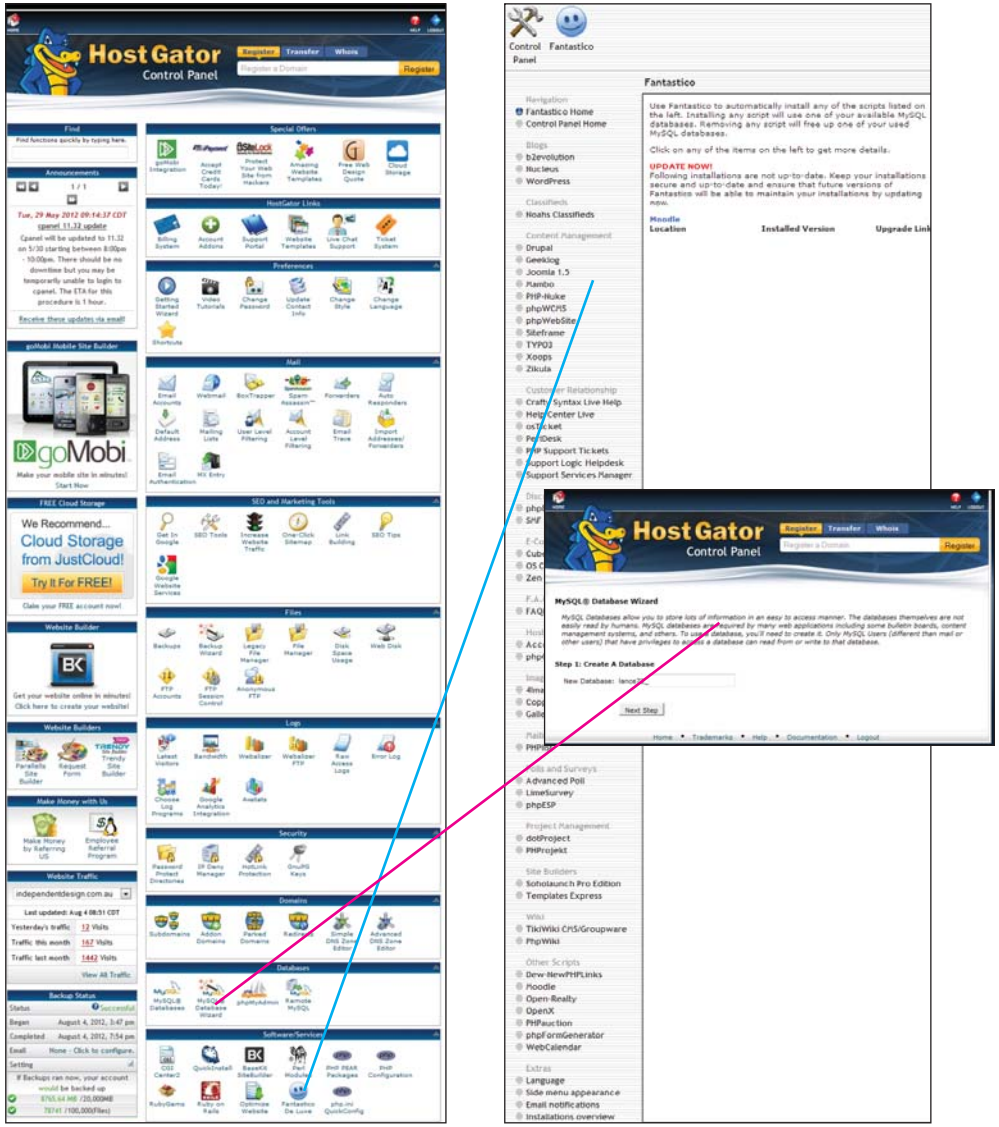


Figure 5: cPanel interface with Fantastico and MySQL Database Wizard

Definitions and terms

There is a great deal of technical language and computer jargon, which necessarily forms part of any discussion of programming or designing for the Internet. In order to be clear about definitions and terms used, the following descriptions will form the basis of clarity as the study proceeds and are based on my interperatations from various sources.

A oo Content Management System (CMS) may be defined as code which streamlines the process of designing, creating and updating a dynamic web site. CMS's enable users to upload and manage content on their web site, without the need to consult the team that produced the web site (Suh, 2002).

> "Scripting or writing scripts is programming within a program. The purpose of a script is to automate a certain functionality within another program. (Mischook, 2005)

To function properly on the Internet, script must be interpreted by a web browser such as Microsoft Internet Explorer or Mozilla Firefox. A script cannot run on its own – it requires interpretation by a program or application. Programming is the process of translating script into machine language before it can be executed (IVO, 2003). Programming is generally considered more sophisticated and difficult, because it is essentially creating software – writing code which will run independently of an exterior, or parent program.

Web development is built upon a scripting environment and the main advantage is the ease of learning compared to traditional programming languages. Scripting is a much more efficient tool than a traditional programming language, because it uses less code to do the same job. This thesis will sometimes refer to the 'coding background' the web developer has and how it affects their approach to development.

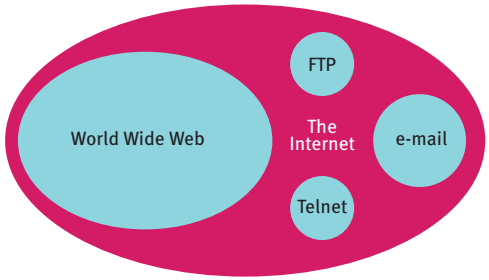


Figure 6: The Internet - authors diagram

In this thesis, SYSTEM CONFIGURATION refers to the process by which a content management system is tailored to the specific needs of THE CLIENT (see below). Any organisation which requests a dynamic web site will have specific expectations of how, where and for whom that web site will function. Occasionally this thesis will refer to SYSTEM CONFIGURATION, which is the process of configuring a content management system to enable the desired functionality and content of a dynamic web site.

> "Software configuration management is an important key technology in software development [where] components, as the integral logical constituents in a system, are managed...and the relationships between components are defined and maintained" (Hong, 2002).

The following definitions are of the authors own making after working in the web design industry for 16 years.

THE CLIENT is the person or business who commissioned the project. *

THE DESIGNER is the individual with an aesthetic focus on the project. Traditionally this is someone with little or no knowledge of how to program the functionality of the web site. One of the main concerns of this research is to identify whether this necessarily needs to be the case, or whether, with training in CMS's, THE DESIGNER can increasingly be someone with both a strong aesthetic vision and definite ideas about functionality, and how these two primary concerns of web development must work in tandem with each other.

THE DEVELOPER's focus is the development of code for the project. Developers tend to have an IT background focused on programming rather than scripting languages.

*Illustrations purchased via istockphoto.com

Defining Web 2.0 and dynamic development

There are several specific characteristics that collectively represent the actual meaning of Web 2.0 (Sharma, 2008). The first of these is 'user-centred design', which means a functionality tailored to the needs of the user, rather than a remote aesthetic and structure imposed by the designer/developer. An example of a user-centred design is iGoogle, a custom home page offered by Google. Also important in defining the Web 2.0 platform is 'crowd-sourcing' – the ability of interactive web sites to draw large numbers of contributor/followers. These web sites have the capacity to quickly achieve an increased profile and relevance in the Web 2.0 environment because of the sheer volume of traffic they get.

A good example of the power of collaboration in the Web 2.0 world is the online encyclopaedia Wikipedia which has far surpassed in profile and content the more standard, information source web sites in the same category, such as *About.com* and *Encyclopaedia Britannica* (Sharma, 2008). Yet another important feature of Web 2.0 is that it is not tied to a particular operating system – and can be accessed by the user with *Windows, Mac, Linux* or a mobile operating system equally effectively. Web 2.0 services tend to be self-sustaining, rather than administrator dependant. A good example of this is Google's self-service platform for advertisement publishing, *Adsense*. Users can employ this system to put advertisements on their blog or web site in a straightforward process which cuts out the need to interface with an administrator for permission. Sharma terms this facet of Web 2.0 'power decentralisation' (2008), and maintains it is an integral part of what separates Web 2.0 from other interfaces.

The term Web 2.0 is often used to describe web applications that allow user interaction, information sharing, flexible operation parameters and user-centered design.

According to Sharma (2008), it is not a single line definition but rather a set of several characteristics that collectively represent the actual meaning of Web 2.0. The following is a list of these characteristics, which form the core philosophy of Web 2.0. The following definitions are of the authors own making after working in the web design industry for 16 years.

User-centered Design

User-centred design is an approach to creating for the web which takes the eventual user's point of view as a starting point. It is web design which aims to meet and anticipate the users needs and even allows the user to change or customise the design themselves. User-centred designs are easy to understand and navigate. They should use the latest in web technologies for example HTML 5 and AJAX for input of content. A good example of a web site which has been created with a user-centred philosophy is *iGoogle*, a customisable Google home page.

Crowd-sourcing

In the current web environment, clients are aware that the volume of traffic to a web site is of paramount importance. Because of the way in which sites generate revenue for their owners, every 'hit' counts. Taking the information, or news business as an example, we can see that because Web 2.0 web sites are able to conscript users as contributors, they are able to build up a much bigger resource base in a shorter time than a more conventional media web site. With millions of users acting as contributors, platforms like *Blogger* and *Wordpress* produce relevant and up-to-the-minute content, and much more of it than web sites which have pursued a more traditional 'viewer as receiver' model. Quality issues aside, the popularity of these platforms means that they now have, in some cases, greater influence than traditional media web sites, and there is no doubt that this is due to the interactive, or 'user as contributor' model.

Web as Platform

Because Web 2.0 technology functions independently on the owner/operator's server, it does not require the user to be locked down to the desktop computer. The user is able to interact fully with a Web 2.0 web site without downloading anything, which makes access possible from any device at any location. Cloud hosting is the term given to this situation, in which the location of the server is unknown to the user and there exists no platform decadency. Web 2.0 applications are also not affected by different operating systems, and will run equally smoothly browser independently on Mac, Windows or a mobile platform such as a tablet or phone.

Collaboration

Web 2.0 web sites such as Wikipedia prove the power of collaboration. Started in 2001, it very quickly outstripped it's rivals, **Encyclopaedia Britannica** and **About.com**, both in terms of user numbers and resources (Sharma, 2008). Although the fact that anyone can edit and add entries means that it must be rejected as a credible source by teachers and academics, the democratic principle on which it operates, and the ease and speed of use hold strong appeal for students. Whether this appeal and presence will be strong enough to pull this resource into a position of prominence and notice within the walls of the academy remains to be seen, but it is certainly a testament to the appeal of the interactivity of Web 2.0. .

Power Decentralisation

A Web 2.0 service is fully automated and self-servicing. An example of this is **Adsense**, a Google service which allows users to place advertisements on their personal blog or web site. There is no administrator to assess requests from users and then allow or reject them. Social book-marking web sites such as **Digg**, **Reddit** and **Stummbleupon** function on the same principle.

Dynamic Content

Because they are much easier to update, Web 2.0 web sites are easier to keep relevant and current. This also adds to their appeal to users, as people are clearly more likely to visit web sites where they will see new information posted more often. As Sharma points out: "In a generation where blogosphere has overpowered the conventional mainstream media, Web 2.0 services have to be highly dynamic and proactive" (2008)

SaaS

Software as a Service (SaaS) is software available as a web service with no platform dependency at all and no knowledge of the back end hardware requirements necessary, the system takes care of all technicalities. *Google documents*, *iTunes* and Wordpress. com are excellent examples of SaaS.

Rich User Experience

A rich user experience uses technologies like XHTML5 , CSS 3 and AJAX, and other rich media producing technologies have helped make web services significantly more appealing to users. Rich user experience involves multiple technologies to provide the most engaging visual (images/ animation/ video), audio and intelligent experience for the user. A satisfying user experience will help to ensure return traffic to the web site.

The Internet

The term *The Internet* is derived from 'Inter-Networking', which refers to the network systems comprised of cabling, switches, gateways, routers and of course computers. The internet adheres to strict protocols or rules to facilitate data transfer. The 'web' is a file sharing network mainly constituting of web sites. 'World Wide Web' uses the Hyper Text Transfer Protocol (HTTP) to make possible to share and communicate information on the Internet.

The Web

The web in this context is independent of the servers upon which the web sites reside.

Back-end web development

The back-end comprises the code used in the web site which incorporates a variety of languages, software and hardware components.

The programmer is responsible for the source code on which a host application runs or a source application which is executed. It is the job of the back end team to write efficient code and test the software for bugs.

Brochureware

The web has been seen for a long time as a marketing tool and companies placed their brochures online. The web site had no further use than as a static unchanging display of the companies brochure. These types of web sites have been catagorised as brochureware and are infrequently updated with new content. These web sites take little advantages of the power of capabilities of the web, such as; video, animation, comments and complex user interaction. Much of the web still exists as brochureware.

Brochureware web sites are suitable for small businesses with a limited number of pages and limited interactive features. Design is an exact translation of the brochure usually executed in static HTML which is difficult for the owner to update. It is easy to identify these web sites as they generally have an extension .HTML at the end of the URL. Examples: www.tilepower.com, www.flowerpower.com.au and www.girlpower.com.au.

Cloud Hosted

Cloud computing describes applications and storage somewhere on the internet. The user does not need to know where this is, they simply use the online service and the transfer is directed to the correct hardware and software location.

Through cloud hosting the user connects to a web site and via the interface and operates the software, uploads files and other activities to the cloud based server. Clouds often appear as single points of access via the interface but may in fact be accessing many different hardware or software installations. This has many advantages in that the user does not need to worry about the technology, cost, capacity, security, maintenance and reliability. The software is already setup and installed on dedicated reducing many initially challenging requirements in CMS hosting. Many CMS solutions offer cloud hosting, however, there is often reduced functionality/ customisation so may not suit all users. Examples: **www.wordpress.com**, **www.speaklight.com** and **www.drupalgardens.com.**

Commercial CMS

A commercial CMS is controlled and published by a company. The type of CMS could be enterprise, open source or custom-built. Recent developments have seen many commercial firms release their software under the open source licence seeking third parties to develop plug-ins and they charge for support of the core installation or some variation of this. ElisLab with *ExpressionEngine* and Concrete CMS with *Concrete5* have successfully implemented this model.

A commercial system is considered “closed source” and the system is not updated without customer notification. This is considered a safer option as developers clients and their business are dependent on the continued reliability of the CMS. The CMS retailer is commercially bound to continued quality development and has a complete, holistic view of the CMS. Concentrated project planning and development are the strengths of a commercial CMS. Examples: *OpenCMS*, *Sitecore* and *MySourceMatrix*.

Configuration

Configuration is usually achieved with ‘extra’ downloaded plug-ins or modules. After activating the plug-in the user is then able to customise the default settings and allocate a location for the features to be displayed.

When the user configures a plug-in or module, various elements of code are automatically installed into the correct location of the web site template without any user input.

Custom-built CMS

A custom-built CMS is designed to provide an exact fit to the client’s development needs. The built CMS can be small, light and implement only the functionality required by the client. Projects can be outsourced and pitched upon by a variety of developers. Selecting this direction is largely a question of the type of content that will be hosted on the web site and the completeness or incompleteness of other non-custom solutions.

Many developers prefer to custom-build their CMS as a bespoke solution, implementing technologies and techniques that they understand so that they can solve any developmental problems as they arise.

There are disadvantages with a custom-built CMS. Firstly, there are the considerable development costs. Why pay if the same functionality already exists for free? Secondly, as it is usually a solo project, the end result can only be as good as the developer themselves, and a reflection of what code they understand and can write. A final consideration is the unknown quantity in the cost/ time-line and availability of the developer themselves, particularly for troubleshooting after the event.

Dynamic web sites

Dynamic web sites retrieve data stored in databases to fill predefined areas in a web site template. The purpose of the web site is content automation/updates capable of tracking thousands of stories on one template.

Dynamic languages include *PHP, Perl, ASP .NET* and *JSP* amongst others. The script language chosen is dependent on the server technology and programmer capability.

A dynamic web site can operate more effectively than a static one, and is easier and faster to update and maintain expand.

Dynamic Blogs

Blogs are the most common form of CMS. They have motivated much of the development of CMS solutions, as amateur publishers of content desired an easy way to get their stories and images online. Blogs form a familiar pattern, usually a post or story listed by date with a headline and images, followed by a comment area.

Recent developments in the evolution of blog tools have meant that users can reorganise pages out of this date format. Plug-ins in these systems add to the basic functionality expanding the feature set into non-blog related activities like displaying calendars, RSS-feeds, weather and advertising. Examples include *Wordpress, Moveable Type, Blogger* and *TypePad*.

Enterprise-level CMS

Large organisations that deal in vast quantities of information will seek a comprehensive solution that performs many functions, not just a web presence. Some of these extra functions required might include scanning, document management, multilingual support and versionising. In organisations such as government departments, newspapers and multinational corporations, these systems will have many contributors. They are defined by the need to capture, manage, store, preserve, and deliver data. Enterprise-level CMS's focus on Business-to-Employee (B2E) rather than the model of most web sites Business-to-Customer (B2C) therefore an Enterprise-level CMS might be too powerful for their needs.

These systems are often very expensive. Depending in the number of users, development and implementation can run into the hundreds of thousands of dollars from vendors like *Microsoft* and *IBM*. Examples include *SharePoint*, *Ektron* and *Vignette*.

Recently new players in the open source (free) environment have started to challenge the commercial CMS solutions as these build a reputation. The main challenge Open Source enterprise CMS solutions have is the lack of technical support, although this is changing. For example, *Alfresco* offers a 'community forum supported' free version and a monthly 'supported help' version of their software. Examples of Enterprise-level open source solutions include *Alfresco, Plone* and *eZ Publish*.

Extranet

An extranet is a web site that is designed for specific purposes and focuses on external parties. It enables customers to connect, usually through a login, for a specific purpose such as a business transaction or educational activity. The web site is usually an extension of the companies existing web site in terms of branding and identification.

Extranets facilitate business-to-business (B2B) interactions in isolation from other internet users and Business-to-Customer (B2C) which involve consumers accessing the web site. In essence there is no difference between the technologies of an extranet, intranet and standard web site.

Front-end web development

Graphic designers usually start their web career by designing the graphics for a web site.

The front-end is used to describe the web site that is displayed in the browser but additionally consists of the user interface technologies that construct the actual displayed page. A front-end designer usually creates their designs in a design application like Adobe Photoshop before transforming it into an HTML web site.

A designer is responsible for cutting up the images and creating the initial HTML layout so a little knowledge of coding standards and that is possible is desirable for a front end designer.

Keeping up with the latest technologies such as AJAX scripting tools a front-end designer must understand the programming principles in order to create designs that are possible within the constraints of the medium.

HTML

HyperText Markup Language is the scripting language used by web browsers to display content.

Intranet

An intranet is simply an organization's internal web site and facilitates many of the same functions and technologies a standard web site can, with a focus on meeting the needs of the staff. The main functions of an intranet are collaboration and communication via record keeping and news.

As understanding of their capacity to implement solutions expands, the use of intranets is growing rapidly, particularly across the spectrum of the education sector.

Open Source CMS

Open Source CMS refers to the freedom to copy and re-use software distributed by a third party. It is often free however some vendors charge a licensing fee for assistance with their system. Under licence the user of the CMS is granted permission to investigate, modify and improve its code.

Open source software can be developed in a very public and collaborative manner with thousands of developers as is the case with *Drupal*, or with just a few, as is the case with *Joomla!*. There are advantages and disadvantages of both methods. Thousands of developers can enable rapid development and quicker releases of the software with perhaps an unfocused direction. A limited or small team of developers can allow for a focused directed outcome, which lacks the speed a large team can offer.

In recent years Open Source software has proliferated from operating systems like *Linux*, graphic manipulation applications like *Gimp* and *Blender3D*. Open source software rivals commercial tools in terms of productivity and cost, but are frequently criticised for their lack of technical support documentation and the perceived unpredictably of future developments and a stable platform. Examples include *Drupal*, *Joomla!* and *ExpressionEngine*.

Partial CMS

Most web pages do not change very often. Only a specific area may need updating on a regular basis such as comments, news, events, e-commerce or a calendar for example. A partial CMS can be deployed alongside a number of static HTML pages generated by a desktop application such as *Adobe Dreamweaver, Microsoft FrontPage* or *Microsoft Expression* to increase the web site's capabilities.

It can be difficult to get a partial CMS solution working when installed technologies conflict. Partial CMS technologies like PHP may not even be available on the current web hosting platform. There are a number of tools available but they don't cover the full range offered with a complete CMS.

A partial CMS is excellent when seeking additional dynamic features to augment an existing HTML web site. Partial CMS solutions are both free and commercial depending on complexity. **Examples include: www. vanillaforums.org, www.cmscalendar.com and www.magentocommerce.com**

Scripting

Scripting is a simplified easily readable language. These "Scripts" are distinct from the core code application, usually written in a different more detailed powerful language.

Scripts are often interpreted from source code and re-rendered by the host application/framework so for example an example a word limit script in *ExpressionEngine*:

```
{exp:word_limit total="30" }
‹p› {summary} ‹/p›
{/exp:word_limit}
```

All *ExpressionEngine* scripts have an {around them} and implementing them is as easy as reading the manual and following directions.

Scripting is a dynamic activity and integrate easily into a dynamic page.

Static

A static web site is written only in Hypertext Markup Language (HTML).The HTML file is stored on the server and sent to the users web browser when requested.

Examples include small personal/company web sites as well as brochure web sites because they present pre-defined, static information to the user. A static web site is no different to a word processing document in it contains all the code inside the file that will be implemented on the web site.

WYSIWYG

What You See Is What You Get (WYSIWYG) describes the process of designing content on screen and the content will display as experienced in the design mode. The first came to prominence with desktop publishing with the on-screen graphics translating to the expected print version. On the web Adobe's software Dreamweaver which attempts to use a Graphic User Interface (GUI) to generate HTML markup is generated automatically.

My research later presented describes how designers/ developers initally creating static web sites and as their skills develop they move onto more complecated scripting languages.

Taxonomy

Taxonomy is a term used in CMS solutions to help categories and sort content. This taxonomy is based upon my literature study, personal experience working as a web designer, my interviews with web professionals and workshop activities with design students.

Some CMS solutions like *Drupal* are heavily reliant on taxonomy in the actual construction and organisation of content in the system. Perhaps more familiar to most users is the term 'tagging' used in *Wordpress* and *Flickr*.

An example of taxonomy might take place with *The Food Pyramid*. We could classify a Banana under Food ›› Fruit ›› Banana.

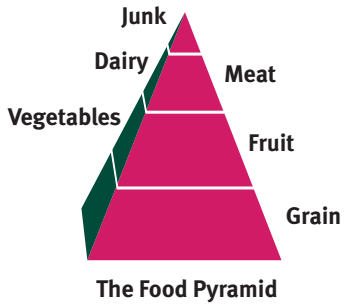


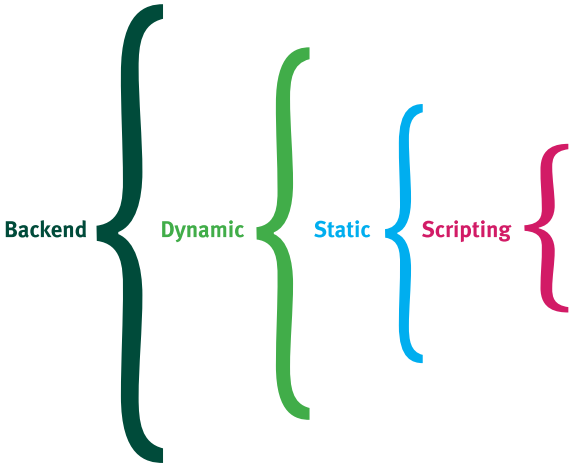
Figure 7: The Taxonomy of Food

XHTML

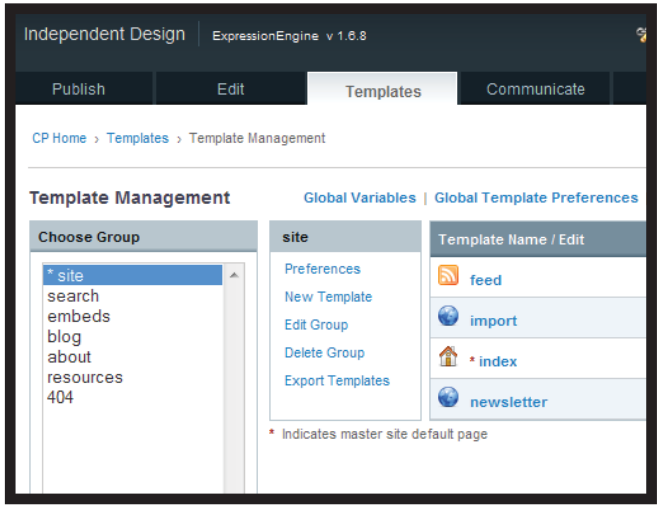
HTML was the first web language and it has since been refined currently to HTML.5

XHTML is an offshoot of HTML and stands for eXtensible HyperText Markup Language and is a more restricted use of code enabling better browser support and device interaction.

Figure 8 illustrates the various components of a CMS. Here the back-end is the actual construction of the web site. The front-end is the finished web site displayed in a browser and the CMS configuration is the administration area where content is entered and the system manipulated.




```
<div class="news">
    <div class="news_pic">
  {exp:weblog:entries weblog="resources" orderby="random" limit="
<div class="cropper">
{exp:imgsizer:size src="{resource_snap}" width="215"  height="215"}
</div>
      <h2>Latest Resource</h2>
      <h3><a href="{url_title_path='resources/page'}" title="{title}">{t
{exp:word_limit total="30"}
<p>{exp:strip_p}{summary}{/exp:strip_p}
{/exp:word_limit}
 <a href="{url_title_path='resources/page'}"><br />Read more</a></p
  {/exp:weblog:entries}
</div> </div>
</div>
 <div class="clear"></div>
  <br />
  <hr />
```



Configuration

Figure 8: The visual schema of development - authors diagram

LITERATURE REVIEW

In this section:

Introduction	24
Static and dynamic web sites	25
Designer or developer?	27
The designer as generalist vs. specialist	29
The rise of web 2.0	31
Perceptual changes in web design from brochure through to dynamic	33
What are the key differences between static and dynamic web sites?	34
What are the drivers in the transition of static to dynamic web sites and the range of considerations?	35
Matrix scoring	36
The current state of play in the CMS community	37
Open Source CMS Market Share Report	38
Content technology vendor maps	42
Assessing CMS usage	43
Schools keeping up with industry	47
What is being taught by institutions in the Sydney basin?	49
What is the range of considerations in teaching a CMS?	50
Analysis of 28 CMS solutions	52
Selecting your CMS	60

02 LITERATURE REVIEW

Introduction

The buzzword in web development for the last few years has been 'Web2.0'. This term is used to refer to a web site where the content is maintained by the people using and visiting it. These web sites are generally structured around a social feedback framework. Examples of the most popular Web2.0 web sites include *Flickr*, *Wikipedia*, *MySpace*, *Facebook* and *YouTube*. These web sites allow visitors to upload content, and leave personal messages. Open source software makes it possible to create a web site that functions in this way with limited programming knowledge. This software generally requires configuration or scripting rather than programming. Open source software is a community-based development environment, and this has advantages and disadvantages, which it is part of the intent of this study to examine. The focus of this investigation will be open source software developments from early 2000 to 2012.

Content Management Systems (CMS) may be defined as a system or programme used to order and manage a collection of web material. There exists in print a great deal of material concerning the content, interaction and interface design of CMS's. These aspects of the CMS are not within the range of consideration for this thesis. This literature review will focus on online feedback and web-based data that informs the important issues and questions posed.

There currently exists very little printed material concerning potential pedagogical directions in which this technology can be taken. It is hoped that this thesis will be a useful addition to the educational discourse, particularly for those hoping to teach emergent CMS technologies in web design courses.

This research investigates four key areas. Firstly, it assesses the current situation with regards to the emergence of dynamic web design. Secondly it seeks to identify factors that influence various potential users/groups. Thirdly, the study investigates current practices in teaching web design and factors influencing designers in their engagement in scripting. Finally this research will access critical factors influencing designers in their choice of a scripting environment

Can the advantages of a CMS persuade web designers to make the jump and learn a scripting language in order to retain complete control of their project? Would teaching a CMS encourage students to learn the programming language supporting it? In order to explore these questions, I have created a web site to document online findings and provide a forum for the CMS community, located at: http://www.independentdesign.com.au/whichcms/. The web site hosts the screencast demonstrations of various content management systems that I have developed as part of the study. Screencasts are a video demonstration of software currently called Computer Based Training (CBT). This technique is easily implemented and broadly understood by the design community.

"As of September 2009, Wordpress is used by 62.8 million web sites in the US and 202 million web sites worldwide...when combined with the available plug-ins, it becomes an unstoppable powerhouse of a platform..."
(Worthington, 2010)

Static and dynamic web sites

As stated previously, a static site is one that has the content (text and linked images) embedded within the page, not stored in a database. It is not possible for the owner of the web site to update the web site without web skills. A typical workflow for the creation of such a web site might be as follows;

- The designer creates a static XHTML page in rapid development software such as Adobe's *Dreamweaver*.
- The designer uploads the web site and the client returns to the designer when updates are needed.
- The designer becomes the content updater and the client incurs an ongoing cost for web site maintenance.

When a static web site becomes dynamic the process usually follows this path. One to five templates of various layout options are given to the development team who reassemble it as a dynamic web site in their preferred development tool and the designer is removed from the development cycle.

A dynamic web site calls content from a database and has many advantages in terms of speed and maintenance of content. A CMS, or content management system enables the client to update content through a web interface at any time or place, provided they have an Internet connection. The cost of a commercially produced CMS can be considerable, with a lower limit of around five thousand dollars. Open source content management systems are frequently free of charge, but can incur hidden costs such as commercial plug-ins or developer expenses in order to achieve the functionality required by the client.

Content management systems (CMS) make web site management easier as the owners of the web site can upload text, video and pictures; they can also interact directly with web site users via comments, and e-mail. These non-technical web site owners manipulate their content via a *WYSIWYG* (What You See Is What You Get) editor. The web site once developed is thus easy to manage requiring the owner to have very little technical knowledge.

Wordpress is the 15th most popular web site on the Internet ranking head of Amazon and Microsoft (Alexia, 2010). Further statistics can be gleaned from the commercially hosted blogs of *Wordpress* at: http://en.Wordpress.com/stats/, including its rapid rate of growth in non-English speaking countries, and figures for blog readership and daily usage.

This period of *Free and Open Source Software* termed FOSS began with a sharing culture during the 1960's in computer science laboratories at *Stanford*, *Berkley* and *MIT* (Yang p1 2011). One problem with open source software which needs to be addressed is the hidden cost to a user/designer that can be associated with it. Open source does not necessarily mean 'free', contrary to many users expectations, and indeed, contrary to information provided by *Wordpress*:

> "*Wordpress* is an Open source project, which means there are hundreds of people all over the world working on it. This...means you are free to use it for anything from your cat's home page to a Fortune 500 web site without paying anyone a license fee..." (*Wordpress*, About, 2010).

Open source software is initially free to the user at start-up level, but in order to complete all but the most basic projects the designer will have to purchase commercially produced 'plug-ins' to enhance productivity. Some providers such as Ellis Lab, the makers of open source software *ExpressionEngine*, provide the software free with a service fee. So open source does not necessarily mean completely free of charge, it simply means the code is distributed and accessible. As well as this, there are additional hidden costs that may occur with these open source systems. Paul Boag identifies some of these as including; the cost of training, the cost to quality, the cost to functionality, the cost of redundancy and the cost of commitment (Boag, 2010, p143). The egalitarian nature of open source software however, - the fact that it is available to expert and layman, client and designer alike to experiment with, makes it an interesting subject for enquiry.

At the heart of the communication revolution which is the internet, lies the concept of user-generated content. Open source web software allows everybody to engage in the activity of publishing content on the web and this is the primary reason it has been selected for this study. Eric Raymond elucidates the difference between open source software and commercial software with the analogy of the Bazaar and the Cathedral. In this analogy, commercial software is likened to the Cathedral, because one organisation is in control and access to the inner sanctum is privileged. Open source software is the bazaar, a free market where there are many vendors (Raymond, 2001).

"In a generation where the blogosphere has overpowered the conventional mainstream media, Web 2.0 services have to be highly dynamic and pro-active. If crowd-sourcing is there, then dynamicity follows by default" - **(Sharma, 2008).**

A dynamic web site is made up of a variety of technologies and a wide choice of functions, so that it can meet the clients requirements. Most CMS solutions are constructed so that additional components are easily added in the format of a module or plug-in. These additional components expand the capabilities of the CMS. CMS developers approach this in some interesting ways. *Drupal* for example prefers to install only a 'core' CMS or 'stripped-down' version, allowing the developer to custom load the desired modules or plug-ins. *Drupal* is so stripped down that it's version 5 release didn't even offer image support. This 'bare bones' approach allows a small software footprint enabling the CMS to operate at maximum efficiency. Alternatively *Joomla!* offers a more complete setup catering for most users to install and immediately begin entering content.

The CMS options chosen for the workshop cover the breadth of the CMS market. Figure 9 below illustrates some of the many possible components that make up a CMS solution, each at the discretion of the designer or developer. As new technologies evolve so do new plug-ins. For example a new plug-in for *Wordpress* allows a end-user to click on a Facebook 'like' button and the story feed will then flow through to the end-users Facebook status page.

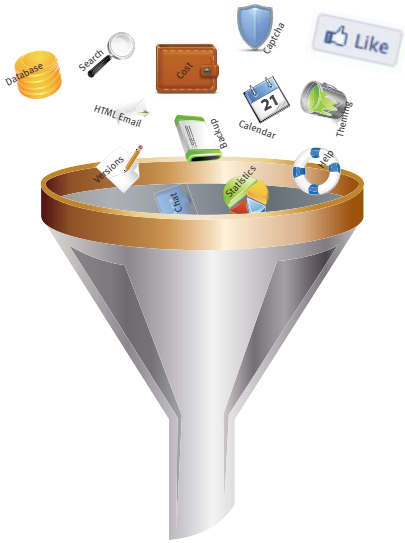


Figure 9: The make up of a CMS - authors diagram

Designer or developer?

In the current environment, a web designer is someone who is adept at using software such as *Dreamweaver* or *Flash* to program a simple *XHTML* web site and to realise the client's aesthetic goals for that web site. They are responsible for positioning, branding, menus and sometimes the content. Their role is likely to be web designer, or graphic designer, either within a design company which makes web sites for clients, or possibly within a large company which has the resources to manage it's own image and promotion. A web developer is someone whose sole focus is programming advanced functionality such as a database driven web site. Typically, a developer will be employed within a larger company to fulfil a specific role, such as a *PHP Developer, Ruby Developer* or *ASP.net Developer*.

> "The terms web designer and web developer are used interchangeably in the media and advertisements. But, they are not the same thing... Unfortunately, there is a lot of overlap and integration needed between web design and web development. When you toss in content creation and usability testing, it becomes a real mess. Can a single individual do all of these things? There are some people who can do all of these things but most people only do one or two of these tasks" (Baker, 2010).

A review of *grey literature* shows that there are three types of web development: Scripting, Programming and System Configuration. Before we examine the question of whether students should be taught open source CMS's a few things need to be made clear. First we ask, what is the process involved in getting a web site live onto the internet?

1. The first type of web development is SCRIPTING, essentially placing the correct code required for a function to work when executed by THE SYSTEM. Scripting requires a low level of understanding and quickly enables THE DESIGNER to create interactivity that normally would be beyond their scope. Designers working within the Scripting paradigm use *HTML*, *Flash ActionScript v2* and earlier, *AJAX*.
2. The second type of web development is PROGRAMMING, which refers specifically to object-oriented programming. This coding process is significantly more difficult to learn and execute than scripting and requires a more intensive and structured learning path. Developers program web sites with languages such as *ActionScript*, *JavaScript*, *Java*, *Rails*, *PHP*, *Python, .Net* and *C++*.

The third type of web development is SYSTEM CONFIGURATION. This involves taking an off-the-shelf content management system (either commercial or open source) and configuring the software to behave in an intended way. Some web development knowledge is needed for these systems. Examples of open source systems which are available for students and designers to download and experiment with include *Drupal*, *Joomla!*, *ExpressionEngine*, *Wordpress*, *Concrete5* and *Tumblr*. As part of the primary research for this study, students and those with limited web development or programming knowledge were invited to undertake a workshop, in which they learned to build a web site using one of the six systems listed above.

These open source tools generally present their content in a template, a concept which is shunned by some designers, because it curtails aesthetic freedom and creativity. An analogy could be made to the early days of desktop publishing when templates were also common. For the client, a template provides a cheap and quick option for developing a web site. From the designers perspective, templates need to be highly customised to become 'aesthetically worthy'. Templates in THE SYSTEM are often very difficult to modify and customise leaving designers who experience them less than positive. THE SYSTEM is usually created by a developer with little regard for the designer's needs during implementation. This is simply a form versus function argument where THE DEVELOPER is all in favour of function and THE DESIGNER favours form.

Gottleib proposes a: "more humanistic, committee-based approach to CMS selection" (2009), which would allow all parties involved in the project to focus on the features which will have real impact on the client, designer and developer:

> "The benefit of this process that selection committee members interactively learn more about their needs and more about the software features as they watch demos and work with prototypes. As a result, their selection criteria become more sophisticated over time and potentially critical information is allowed to enter the decision making process at any time (Gottlieb, 2009)

Clearly, in choosing a solution, it is important for all vested parties to consider interface design, flexibility for updating, available software plug-ins, tool scope, speed of web site creation, learning/tutorial materials, software updates and last and most importantly, ease of learning. Jeremy Epstein, lead programmer at *Eskimo Digital* says that clients often have unrealistic expectations of both price and functionality when it comes to open source projects.

> "We have quite a few clients who have come to us asking us to build a *Drupal* web site. If clients CMS expectations are really out of whack then we talk to them and try to assist them." (Epstein, 2010)

Clearly, if a process of careful consultation is pursued, another key benefit is that everyone involved has a clear understanding of the decisions made, and the reasons for pursuing a particular solution. This should have the effect of lessening client tension of the kind identified by Epstein, where the client has unrealistic expectations, or is disappointed with the solution implemented.

This research investigates how recent software innovations in THE SYSTEM have closed the gap between the two disciplines, web design and web development. THE DESIGNER now has the ability to configure software and create a dynamic web site that would previously have required a web developer THE DEVELOPER.

> Small graphic design and web development companies are now able to take on projects previously outside their scope and at a budget that their clients can afford.

The end result is that there is a greater range of projects and opportunities for individuals and small companies in an ever-expanding Internet environment.

If negotiating the open source realm is proving challenging for established companies and their clients however, it is doubly perplexing for those wishing to self-educate and improve their skills. The area of open source content management systems is evolving so quickly that limited learning materials exist, which makes choosing THE SYSTEM and learning it a difficult and daunting task. Certainly, without some explanation, the problems associated with THE SYSTEM would most likely defeat a student. The novice is presented with an enormous range of software choices and it is almost impossible to make an educated choice.

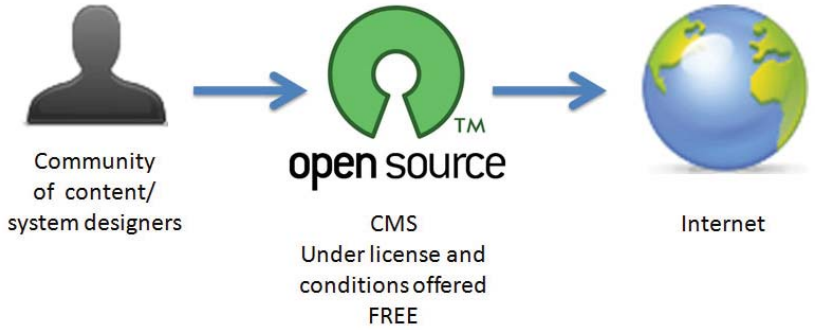


Figure 10: Open Source development - authors diagram

The designer as generalist vs. specialist

Students these days face a daunting task in skill acquisition, balancing their academic requirements and developing their own aesthetic vision. The following list attempts to categorise common areas in the development and maintenance of internet web sites:

- Web Development/ Design: Programmers/Coders, Web usability experts
- Content Developers: Writers, Photographers and graphic designers, audio and video editors
- Web site Marketing: Search engine optimization, search engine marketers

The quandary of generalist vs specialist is something that many web designers struggle with as this statement by Niles (2010) shows;

> "Am I kidding myself thinking that I can carry all of those roles successfully? And if I really am doing all of them well or well enough, why am I still labelling myself a "web designer"?"

Traditionally as design & development studios increase in size, so too does the need for specialisation by the employees. A CMS offers a student the possibility to develops skills so that they are initially generalist but can later be refined. Skills learnt implementing a CMS are easily transferred.

The Internet is a fast moving environment driven by user demand through technologies such as new languages HTML 5, CSS3 and increased browser support for mobile devices. However the development process has not kept pace with the technology advances and many companies still implement outdated approached to web site development with static and table based web sites not uncommon.

> "Gone are the days of the simple brochure web site. Businesses and site visitors have high expectations of what every online presence should deliver" (Admin, 2010)

Web designers and developers still assume that they are coming from two separate directions, primarily because this attitude is still a part of the approach of most tertiary courses in design and design-related fields. Research into the curriculum of courses being offered in the Sydney basin shows that courses with a design focus are mostly teaching *PHP* and *ActionScript*, and courses with a more computer science or programming focus are teaching Java and C++. Clearly this will not produce students who are capable of producing a web site independently.

This leads us to the question, what do employers want? Interviews with employers reveal that there is a desire for graduates who are more well rounded, whose skills cover a wider range;

> "I was taught programming foundations, which I have found essential in helping me learn programming languages that I use these days. Also I had some classes about database theory, which is really helping me these days. I would really have liked some classes about design theory, typography and colour theory because you end up being a jack-of-all-trades in a web design company and it is frustrating" (Epstein, 2010).

As Chad Fowler writes in *My Job Went To India,* the concept of specialisation in many information technology fields is becoming outdated. Fowler (2006 p.26) argues that; "...in this kind of rapidly changing environment, the flexible will survive..." Like Epstein, Fowler re-negotiates the expression 'jack-of-all-trades' as a positive, maintaining that focusing in on one small domain of knowledge is dangerous in the current web environment.

> "The way to become a generalist is to not label yourself with a specific role or technology... If your goal is to be the last person standing amid rounds of layoffs and the shipment of jobs overseas, you better make yourself generally useful."
> (Fowler 2006 p.39)

It is clear that, whilst design and development are still routinely separated at tertiary level study, the current web environment dictates that most students will be required to produce or learn skills from both these areas. It is the intention of this study to show that one way to better prepare them for this requirement would be the inclusion of a unit on open source CMS in design courses.

The rise of web 2.0

As stated previously, the buzzword in web development at the moment is 'Web 2.0'. On a web site which can be classified as Web 2.0, the content is maintained by the people visiting the web site, which is generally structured around a social feedback framework. "...[Web 2.0]...is commonly associated with web applications that facilitate interactive information sharing, interoperability, [and] user-centred design" (Sharma, Core Characteristics of Web 2.0 Services, 2008). Examples of the most popular Web2.0 web sites include *Flickr*, *Wikipedia*, *MySpace*, *Facebook* and *YouTube*. These web sites allow visitors to upload content, and leave personal messages.

The term Web 2.0 was first described by Tim O'Reilly, CEO of O'Reilly Publishing, who first used it at the O'Reilly media conference in 2004 (Graham, 2005).

O'Reilly categorises content management systems and personal web sites as Web 1.0 WIKIs and Blogs are categorised as Web2.0 technologies. The functionality of web 2.0 web sites like *YouTube* and *Flickr* is often supported by a myriad of technologies, both open source and commercial. The interactive component of this functionality is what sets the 'Wiki' apart from a dynamic web site which has been created with a content management system. Thus, content management systems may be deemed Web 1.0, and Wikis as Web 2.0 technologies (O'Reilly, 2005).

The driving philosophy behind Web 2.0 is interactivity. The appeal of these web sites is centred in the fact that anyone and everyone can contribute by uploading content. Flickr, is a Web 2.0 web site where users can upload photographs. Its mass appeal, however, lies in the fact that people can leave comments for others at the web site, which makes it a space for real time social interaction. When people travel to the web site they are not merely viewing still photography, but entering an interactive social domain, where their efforts as an amateur photographer receive notice, and where they can 'associate' with other like-minded individuals. Thus, Marshall McLuhan's famous adage from his seminal work *Understanding Media: The Extensions of Man*; 1964 is once again shown to be prophetic – the medium has become the message.

McLuhan used the analogy of the light bulb, which creates a space for social interaction where previously there was darkness, to explain his now famous statement: "The medium is the message" (McLuhan, 1985). User-generated web content, like McLuhan's light bulb, is transforming us, by making available ways of being and relating to others that were previously unavailable. It is hoped that this research will show that teaching open source software to design students will give them a way to harness the potential of this wave of internet activity and energy, both for themselves and their future clients.

"In a generation where the blogosphere has overpowered the conventional mainstream media, Web 2.0 services have to be highly dynamic and pro-active. If crowd-sourcing is there, then dynamicity follows by default" - **(Sharma, 2008).**

The rise of Web2.0 has brought into focus the opportunities offered by open source software. Early versions of *YouTube*, a web site where users upload video content, and can comment on the efforts of others, were developed on the open source platform. Open source software makes it possible to create similar web sites with limited programming knowledge.

The potential for cashing in on these newly created internet communities of like-minded consumers is not lost on business owners, and it extends an opportunity to young designers prepared to really 'take the bull by the horns'. In this new environment, and with so many small budget projects out in the marketplace, there are clearly more opportunities for the designer who can create in a wide range of tools and can see a project through from start to finish.

> Both large and small companies hire in-house graphic and web designers and generally speaking the smaller the company the broader the skill set necessary.

With good programming skills, a design graduate is in an excellent position to capitalise on these opportunities in the workforce.

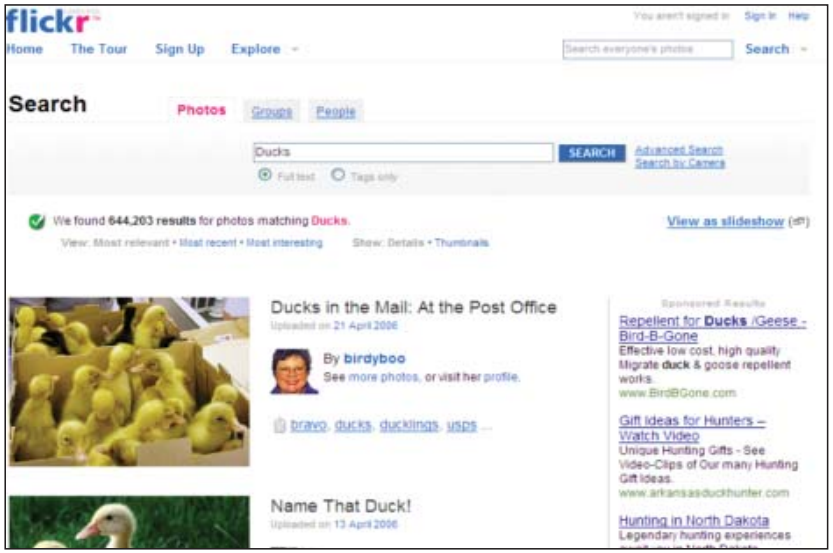


Figure 11: Flickr.com

Web 2.0 web sites are are empowering designers to create web sites that was previously the domain of the developer only working for large miltimedia companies. (Quark, 2010)

Sites like Flickr (see Figure 11) are completely content dependent and 100% of the content is uploaded by end-users.

Perceptual changes in web design from brochure through to dynamic

The primary difference between static and dynamic development is that a static web site has the code and content created and intertwined together. This type of development requires the services of a web developer. A dynamic web site has the code and content separated via a content management system (CMS). The CMS is easily updated by the client who may have limited or no technical skill of web techniques.

In the late 1990's as the World Wide Web evolved, businesses responded and uploaded their brochure content by creating a simple HTML web site. These early static web sites functioned essentially as an online directory, providing the kind of information about a company and it's services which would previously been circulated via brochure or in a phone directory. As Boag points out "Although the web site owner may make updates periodically, it is a manual process to edit the text, photos and other content and may require basic web site design skills and software" (Boag, 2010, p 42)

> "I think higher education institutions first need to understand that web design and web development are two different things, and offer specialized programs in each. They also need to understand that web design isn't the same as graphic design and web development isn't the same as computer science" (Jeoff Croft in Jensen-Inman, 2009)

Once the web became available to businesses as a means of promotion, the natural step was to translate the old advertising format, straight into the new medium. As McLuhan says the old medium becomes the content of the new one (McLuhan, 1965). Print design is a format where the position of graphical elements, text and fonts is relatively unconstrained. In contrast web sites have significant technical constraints of similar elements based on the evolving web language which was not initially designed for graphical pages. Direct translation of brochures to web sites happened in a graphical format with pages being built in table-based layouts primarily because of two reasons. The first is the lack of support by browsers for the WC3 recommended web standards of semantic and cascading style sheets. The second is the designer's inability to understand and implement these web standards. This has been influenced in the educational sphere by books like David Siegel's *Creating Killer Web Sites* published in 1997, which advocated the use of table based design.

Dynamic web sites have assisted in the maturity and adoption of internet standards and the need for content to have predictable standards. According to the web site W3C Web Semantics: "It describes methods and technologies to allow machines to understand the meaning or 'semantics' – of information on the World Wide Web" (W3C, 2010). When the author of the web site is inserting a heading into a dynamic web site they choose 'heading' and the content will be output correctly with no technical understanding required. Dynamic web sites are much better for clients and end-users without web development skills as they are now more in control of their content.

A standard framework for designers

A visual analogy of the development of a dynamic web site might be the simple hamburger. A hamburger consists of a number of elements working together to create a new product. A CMS unites all the components together giving it structure. Beginning with *XHTML* and *CSS* as the foundation then add extras like *AJAX* and *Javascript*. Finally make the web site dynamic by inserting a server side scripting language like *PHP* powered by *MySQL*. Remove the 'extras' and you have a simple static web site. Figure 12 below presents a visual metaphor for the development framework of a modern web site.

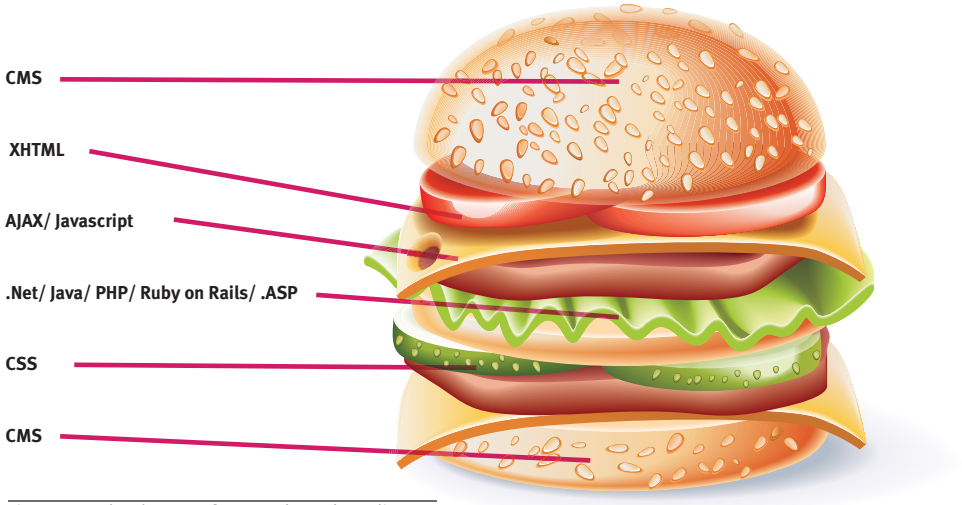


Figure 12: A development framework - authors diagram

What are the key differences between static and dynamic web sites?

A static web site is simply *HTML* and content written code that resides on a computer server and this is how the first web sites were structured. The web site is probably small and only able to be updated by someone proficient in *XHTML* web site development. These web sites are small and cheap to develop. According to Boag (2010) differences can be separated by:

The advantages of static web sites

- Quick to develop
- Cheap to develop
- Cheap to host

The disadvantages of static web sites

- Site requires web development expertise to update web site
- Site is not as useful for the user – static sites essentially function as a visual version of the Yellow Pages on the Internet. They let potential clients know that the business is there and provide contact information.
- Content can get stagnant and out of date

Dynamic web sites are much more complicated and usually more costly to develop, but they have numerous advantages (Boag 2010). In their most simple form they can replicate a static web site with the added advantage that the client can update the content without any knowledge of *XHTML* or requiring *FTP* transfer of files. This means that once the web site has been created there will be less ongoing costs incurred by the client.

> Dynamic web sites store their content or data in a database and retrieve it when the page content is requested by the web browser.

Some examples of interactive features possible on dynamic web sites include e-commerce, bulletin boards, intranet facilities, and the ability for users and administrators to upload documents and create content (dynamic publishing).

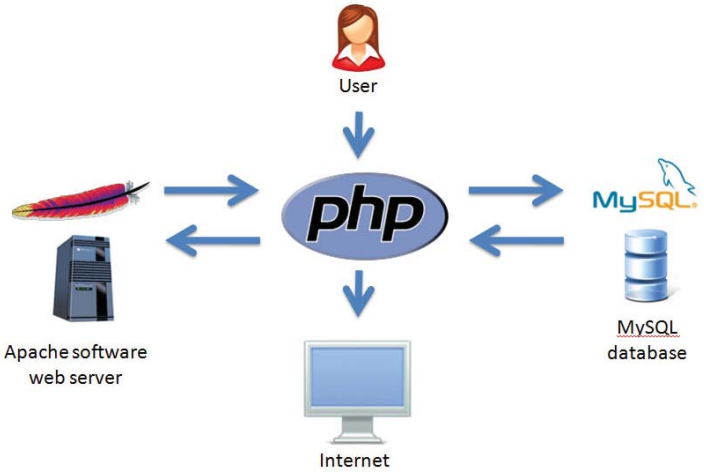


Figure 13: Dynamic site structure - authors diagram

What are the drivers in the transition of static to dynamic web sites and the range of considerations?

There are several key drivers that would assist the designer in their transition from static to dynamic web sites. The first is an understanding of the actual capability the system offers. The capabilities include client editing, image uploading, syndicated news feeds, e-commerce, image uploading and re-sizing, video compression as well as commenting and approval systems. These capabilities make it easy for the designer to create a powerful web site, extending their capabilities beyond simple *XHTML* and *CSS*.

Also responsible for motivating this change are the designer and the developer. As the designer is able to get more control of the system they are able to offer the client more options. The developer with the use of plug-ins is able to leverage upon existing code and offer quicker cheaper development. Another factor is the adoption of web standards across all browsers giving web sites predictable results.

The final key driver is the actual development framework. The framework allows templates to be built and THE SYSTEM populated with content. The framework controls the development cycle and is the key technological decision driving development. "Many frameworks exist and each framework can typically have another framework built on top of it" (Frameworks, 2010). The framework is the software supporting the web site, i.e a web site built on *Drupal* CMS, which then functions on top of the *Apache* operating system.

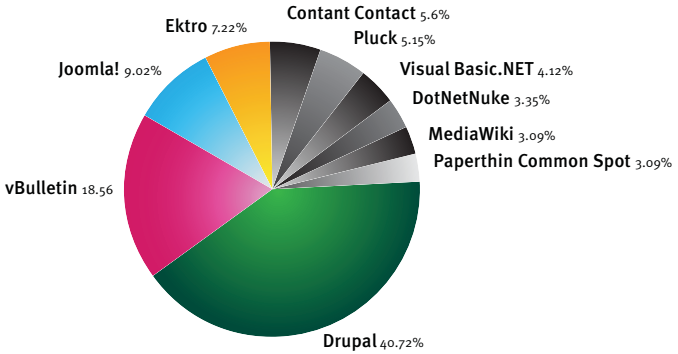


Figure 14: Frameworks distribution - Frameworks.com

Low cost or Open source software solutions can help the designer take the next evolutionary step forward. The *grey literature* concludes that the designer in learning the system is able to:

- Create a large web site where all content is current and is maintained by the client.
- Set up advanced functions such as Blogs, Calendars and Content Archiving themselves.
- Earn more money as they are able to generate more work
- Keep control of their web sites by not handing them over to developers
- Provide more technical options to their clients.

Small graphic design and web development companies are now able to take on projects previously outside their scope and at a budget that their clients can afford. The end result is that there is a greater range of projects and opportunities for individuals and small companies and an ever-expanding Internet.

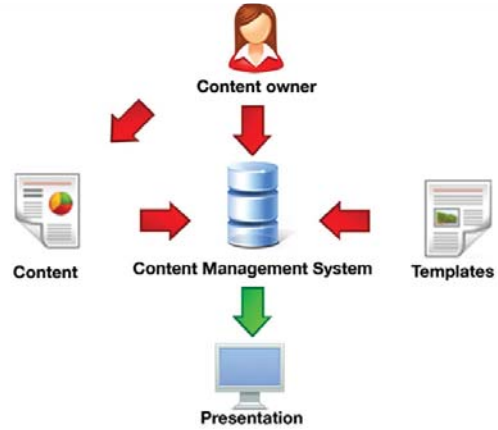


Figure 15: A typical CMS - authors diagram

Matrix scoring

One way to go about selecting an open source CMS is to consult one of the web sites that provide online matrix scoring. On these web sites, CMS's are ranked and scored according to a range of variables, however, there are several reasons why the matrix scoring method fails to accurately direct a user to the right solution. The criteria chosen, and the weight attached to them often seem arbitrary and subjective, and the scoring system obscures information that is important - particularly to a first time user. The idea quantifying this 'data' in a total score is also a less than useful approach in my opinion. Users need to be made aware of the compromises or trade-offs that are involved in choosing one CMS over another. And finally, the criteria chosen for these matrixes appear in many cases to be of unequal importance - for example, we frequently see broad criteria like *usability* compared with something specific, like *SSL on the login page*. Comparing software with hardware, or comparing either of these with user experience is a difficult proposition. Other scholars, when investigating this, have identified similar problems: "A typical example is where a user finds a critical (to him or her) feature totally unusable but that is overshadowed by excellent ratings in a majority of less important features. (Gottlieb, 2009)

A better qualitative measure might look for interface design, flexibility for updating, available software plug-ins, tool scope, speed of web site creation, learning/tutorial materials, software updates and last and most importantly, ease of learning. When discussing CMS selection with Jeremy Epstein, (Epstein, 2010) he said that "... we have quite a few clients who have come to us asking us to build a *Drupal* web site. If clients CMS expectations are really out of whack then we talk to them and try to assist them."

Gottlieb proposes a more humanistic, committee-based approach to CMS selection.

> "The first benefit of this technique is that it keeps the focus on things that have real impact on the users of the CMS — forcing the group to think through the implications of specific aspects of the solution. This is better than having a people register their concern in the sparse format of a low numerical score and then just move on." (Gottlieb, 2009)

Gottlieb goes on to explain that although a committee-based approach is more time consuming, it is worth the effort, as team members learn more about software features, and about their own needs as they watch demos. Down the line, this committee-based approach can also help, if there are problems with the implemented solution, as there is a process of analysis to refer back to. (Gottlieb, 2009).

The area of open source content management systems is evolving so quickly that limited learning materials exist, which makes choosing THE SYSTEM hard, and learning it a daunting task. Certainly, without some explanation, the problems associated with THE SYSTEM would most likely defeat a student. The novice is presented with an enormous range of software choices and it is almost impossible to make an educated choice. Comparison web sites like http://www.CMSMatrix.com, (Figure 16) compare key features and tools in a scoring matrix based but these too can be problematic as they do not give a wholistic summation of the user experience ofter critical to the CMS being adopted or used by the owner.

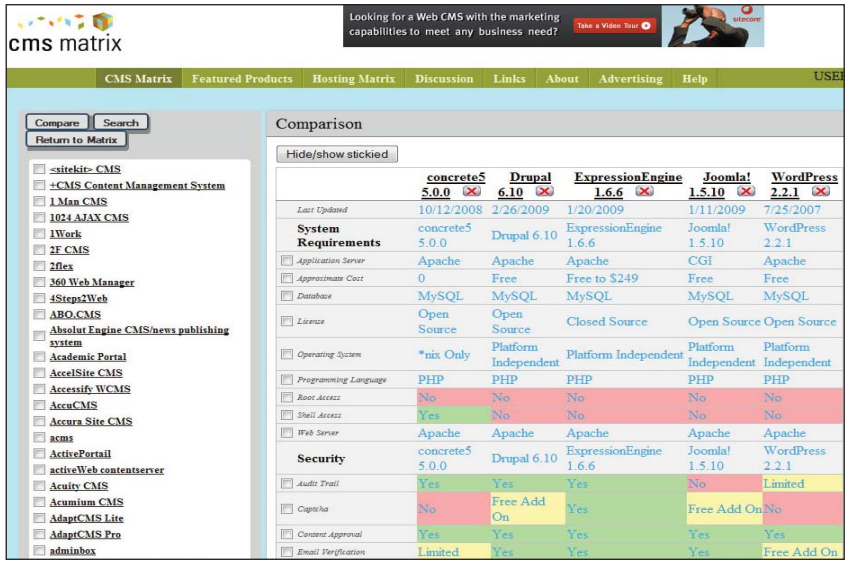


Figure 16: CMSmatrix.com

The current state of play in the CMS community

The criteria mandated for this research is low cost, or open source content management systems. The scope of this project therefore, does not include commercial offerings and larger, 'enterprise' CMS solutions like *SharePoint* and *Vignette*. A quick look at the cost of CMS solutions in this market place shows that they are well beyond the reach of students or smaller companies. The following figures show the cost of the system per one thousand users, in the first year:

- EMC/Documentum -US$ $863,938
- Open Text -US$ $637,304
- MS SharePoint -US$ $318,738
- Alfresco -US$ $33,500 (Mosher, 2008)

For an idea on what platforms are being used most frequently, *BuiltWith* Internet Technology Usage Statistics provides free information regarding the most popular technology used on the web (BuiltWith, 2010) This web site provides data scanned from the 1 million web sites which are most frequently visited. This market research is problematic for two reasons. The first is that low traffic web sites don't count in their survey - it only looks at popular web sites and the second is the way it categorizes the content. The *BuiltWith* survey groups Blogs and CMS's separately when in fact there is a significant blurring of these two in terms of capabilities. Traditionally a blog has content posted by date only and gives the user very little control over customised content fields. This is changing – for example *Wordpress 3.0* will have custom post types which will make it a fully fledged CMS.

According to founder Matt Mullenweg: "*Wordpress* now runs an estimated 8.5% of all sites on the internet. This is a staggering amount – almost one in every ten sites is sitting on what was once just 'blogging software' (Shepherd, 2010). Judging from this statistic, it is safe to generalise that *Wordpress* has greater market adoption or usage than *Drupal*.

Here I present two separate areas worthy of consideration:

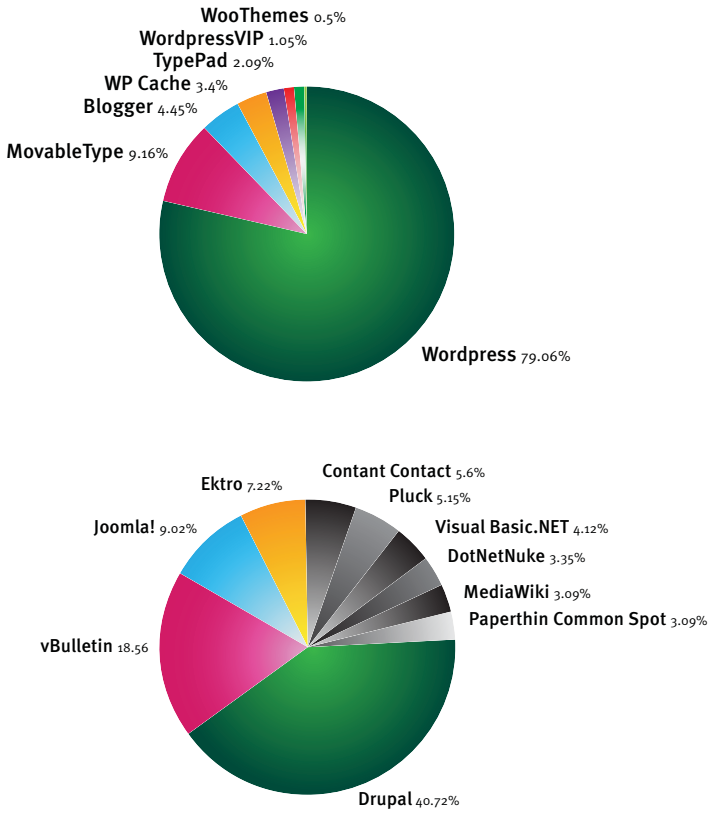


Figure 17: CMS and blog market share - Mosher 2008

Open Source CMS Market Share Report

Water & Stone (http://www.waterandstone.com) a web consultancy firm has surveyed hundreds of open source users since 2008 and produced an their annual Open Source CMS Market Share Report. This research measured some important indicators, including brand strength, popularity, download rate and rate of adoption. Water & Stone limited their open source selection to the 20 content management systems excluding partially commercial CMS's. Those included in the survey have included: *Alfresco, CMS Made Simple, DotNetNuke, Drupal, e107, eZ Publish, Jahia, Joomla!, Liferay, MODx, OpenCms, phpWeb site, Plone, SilverStripe, Textpattern, TikiWiki, Typo3, Umbraco, Wordpress* and *Xoops* (Water & Stone 2008). Over the last four years the considered software list has changed after the research has been gathered.

The research by Water & Stone is particularly relevant to this research because it highlights CMS's chosen for this study, that were featured in the *Which CMS? Workshop - Drupal, Joomla!, Wordpress* and *Concrete5*. Unfortunately *ExpressionEngine* whilst open source is not entirely free (users incur a 'low-cost' service fee), so it was excluded form the survey. *Tumblr* is free to use but not considered a CMS.

Water & Stone's research measured four key factors as indicators of a particular system's popularity. These four factors were; evaluations and trial usage, current usage, third party support, and average download rate.

Figure 18: CMS and blog market share

Although download data for open source CMS's would appear on face value to be a simple and straight forward indicator, Water & Stone state in the survey results that: "… the download data for open source CMS products reveals much less than one would hope" (Water & Stone, 2009). They note that this was because of variables which interfered with the consistency of the results, such as spikes in the weekly download rate created by the release of a new product, and the unavailability of data on many systems (Water & Stone, 2009). Interestingly, this research indicates that brand strength and loyalty are already important factors which affect the success of open source CMS's.

> "Brand sentiment and adoption rates imply that user dissatisfaction may be beginning to have an effect on several systems. Brand recognition and familiarity is a major challenge facing all systems outside the 'big three'. For example, *phpWeb site*, *Textpattern* and *TikiWiki* face significant challenges in terms of brand strength and market share" (Water & Stone, 2009).

The research review included CMS software which operated on the three main web platforms, .NET, PHP and Java.

The 'big three' referred to in the above quote are *Wordpress*, *Joomla!* and *Drupal*, which were found to dominate the market, leading the other candidates in both adoption rates and marketing profile by a comfortable margin (Water & Stone, 2011). The research also found that PHP-based systems are universally more popular than both Java and .NET systems, with Alfresco claiming the title for most popular Java-based system, and DotNetNuke the most popular .NET based system.

An examination of the 2011 report shows a continuation of trends observed in previous years. There are twenty open source CMS's assessed in the report, which examines a selection of variables, including rate of adoption, sentiment and brand strength. The results are tabulated in metric form, making comparison across the range quick and easy. This report makes it easy to identify the market leaders, and understand how the software is being received in the marketplace and why. The report concludes that *Wordpress*, *Joomla!* and *Drupal* remain in command of the market, although *Joomla!* records significant decline in some key areas. Of these three, *Wordpress* leads in brand strength and market share, after a strong year. On the .NET platform, *DotNetNuke* is leading the market, closely followed by *Umbraco*. There is more close competition in the market for Java based solutions, with *Liferay* and *Alfresco* on relatively equal terms, and the 2011 report names *Concrete5* as 'one to watch', as it had a big year. Finally, the report names systems that are possibly at risk, including *e107*, *Movable Type*, *Textpattern* and *Xoops*. These reports are clearly an invaluable resource for the open source novice, particularly a teacher who is faced with deciding upon an educational tool (Water & Stone, 2009).

The 2009 survey reviewed the web sites http://www.elance.com and http://www.guru.com with over 42 thousand registered users in 2009 and Water & Stone were able to offer an overview below.

	Average weekly downloads	Source of data
WordPress	433,767	WordPress Download Counter
Joomla!	189,429	JoomlaCode.org
Drupal	62,500	Drupal Team
Umbraco	5,670	Umbraco Team
eZ Publish	5,612	eZ Publish Team
CMS Made Simple	4,903	Cmsmadesimple.org
SilverStripe	2,500	SilverStripe Team
e107	2,242	e107.org
Xoops	1,209	Xoops.org
TikiWiki	373	Tikiwiki.org
phpWebSite	347	Phpwebsite.appstate.edu
Typo3	100	Sourceforge.net
Alfresco	57	Alfresco.com

Figure 19: Average weekly downloads (Water & Stone, 2009)

The data from Water & Stone's 2011 report shows a significant lead for *Wordpress*. The graph blow lists list the installations of software, ie software installed by participants over the last two years. The research reveals that *Concrete5* has had a significant increase in installations and that *Joomla!* installation rate is down whilst *Wordpress* and *Drupal* remain constant.

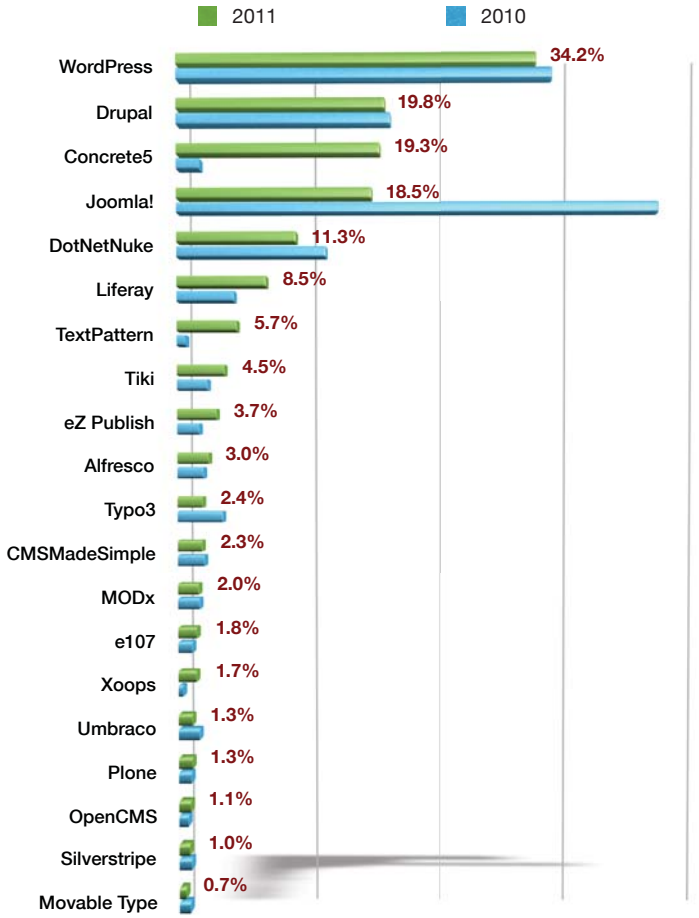


Figure 20: Installations as per Survey (Water & Stone, 2011)

The table below gives us an insight into which systems are being used by the most popular web sites. Rather than re-invent the wheel, the report has accessed the research of other third party sites, to achieve maximum accuracy and representativeness. The web site *W3Techs9* analyses the top one million web sites as rated by *Alexa* (the renowned web site monitoring and ranking system). Included in the *W3Techs9* analysis is an examination of content management systems, a number of which are those featured in the CMS market share report. The table shows their assessment of CMS market share amongst the *Alexa* top million web sites.

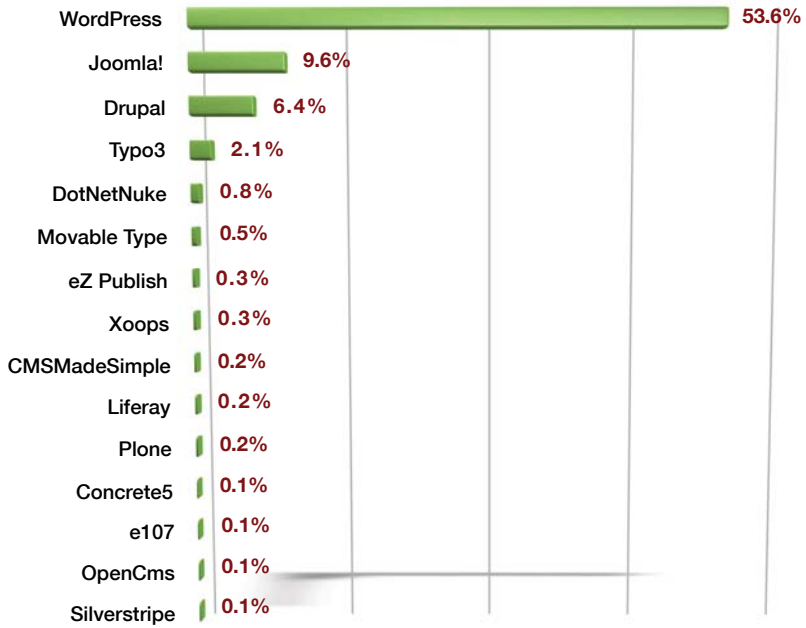


Figure 21: The Alexa One Million (Water & Stone, 2011)

Book publication is another area to consider when accessing usage, popularity, and adoption of a CMS. Books are a critical learning tool for students and a key determining factor in selecting CMS software. Open source software is often criticised for having poor documentation so this area is an important consideration.

> "For this metric we sought to learn three things: First, who has the largest number of books in print; second, which systems have been the subject of publishing activity in the last 12 months and finally, which systems are currently the subject of books that have been announced, but not yet released. The search was restricted to English language books only." (Water & Stone, 2011)

	Total in Print	Released in 2011	Announced
WordPress	83	23	4
Drupal	64	22	7
Joomla!	65	13	1
Tiki	7	6	0
Plone	14	3	0
Alfresco	9	3	0
Concrete5	2	2	0
DotNetNuke	15	1	0
Liferay	11	1	1
Movable Type	5	1	0
Xoops	4	1	0
eZ Publish	4	1	0
CMSMade Simple	2	1	0
e107	2	1	0
MODx	2	1	0
SilverStripe	1	1	0

Figure 22: Books in Print (Water & Stone, 2011)

In marketing, the term 'sentiment' refers to the way in which a brand is perceived by the consumer in a total sense, including the way in which it behaves within the market, and engages in profile-raising activities, such as advertising. It is an aggregate, therefore, of the consumer's attitude towards the product itself, and the behaviour of the company which markets that product and the messages it sends in the marketing process. Water & Stone reviewed brand sentiment below.

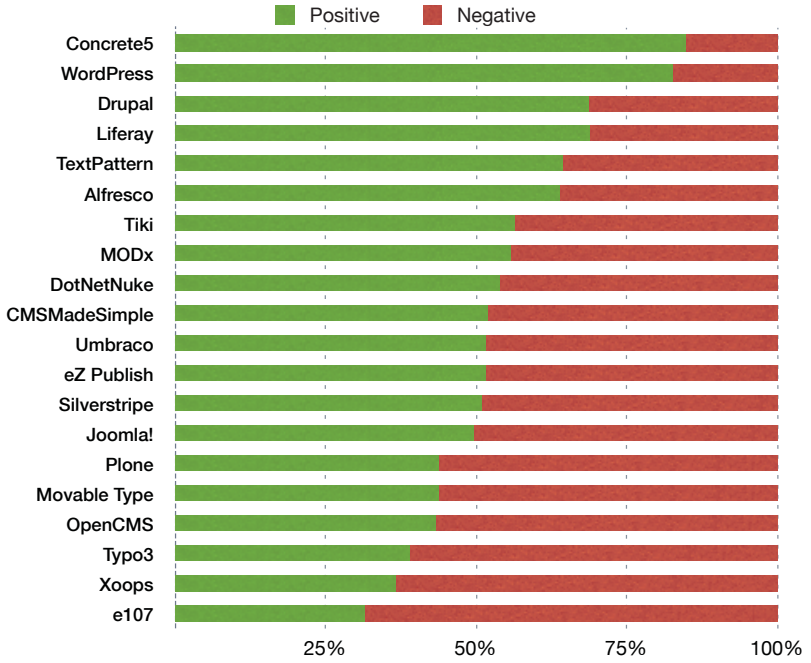


Figure 23: Brand Sentiment (Water & Stone, 2011)

Water & Stone reports contain a wealth of information and are a good starting point for CMS comparisons. **Read all of Water & Stones reports:** http://www.waterandstone.com/book/white-papers

Content technology vendor maps

The following diagram is provided courtesy of *Information Architects*, Japan. It illustrates many of the main vendors and channels in the CMS and web arena.

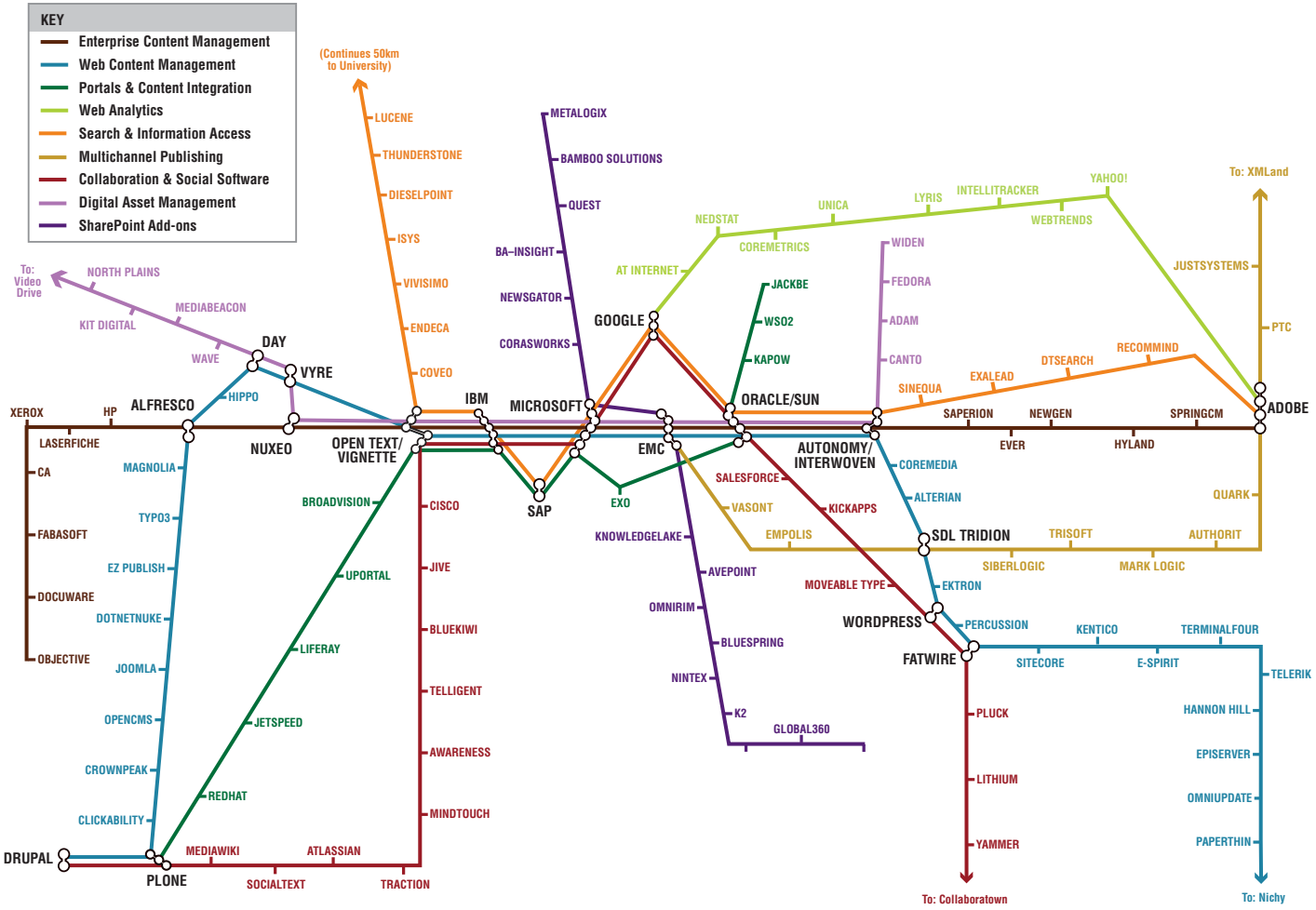


Figure 24: Vendor Technology Map - http://www.realstorygroup.com/vendormap/

Assessing CMS usage

There are a variety of ways to assess CMS usage. On its own, each of them is flawed in some way so it is useful for the first time chooser of an open source CMS to avail themselves of more than one type of information.

Briefly I will examine five here:

1. Google Trends
2. Trade shows
3. Browser plugins: Wappalyzer
4. Alexia software CMS download rankings
5. Garter's Magic Quadrant

One way to gather some casual data on the level of interest in the available versions of THE SYSTEM is by using *Google Trends*, which provides information on how often a particular topic has been searched on. This service could be used to track the growth of a CMS over time, although only in short range or recent terms, since the web site only dates back to 2004 (**http://www.google.com/trends**). By simply entering the search terms *Wordpress*, *Drupal* and *Joomla!* you are presented with an analysis of search activity in the form of a line graph and a list of the most popular clicked on items.

Figure 25: Google Trends: Wordpress, Drupal & Joomla!

> One of the big challenges in assessing a CMS is that each system version is constantly receiving development upgrades.

The system that existed in 2004 may no longer be available so printed written material or screencasts of this edition date very quickly. Reviews of a CMS written in 2007 may be completely different to those in 2009 once software upgrades have taken place.

Trade shows like *CeBit Sydney,* 2010 (http://www.cebit.com.au) provide vendors with an opportunity to demonstrate their product in a live interactive medium. On May 28th 2010 at the *Cebit Sydney* web site, *Flex* demonstrated their own CMS and a presentation titled '*Selecting a web CMS*'. Some of the features identified by *Flex* as important include usability, ability to integrate with other services, how 'search-friendly they are, hosted vs local and open source vs proprietary (Kofahl, 2010).

Figure 26: BeBit 2010 - Mark Kofahl from SIteFlex

Also available to help users assess web technologies is *Wappalyzer*, a *Firefox* add-on (http://wappalyzer.com/stats/cat/CMS). This service provides a summary of the key players, and will provide numerical data on the most popular applications based on the number of times they have been installed in the last 30 days.

Wappalyzer is an software add-on for the internet browser Firefox. *Wappalyzer* inspects the loaded page and to understand which CMS technology was used to construct the HTML markup. http://wappalyzer.com/stats/cat/CMS

Wappalyzer even provides a nice summary of the implemented CMS technologies. Figure 27 is an analysis over a 7 day period of CMS solutions and helps you understand the technologies of the pages you have been browsing.

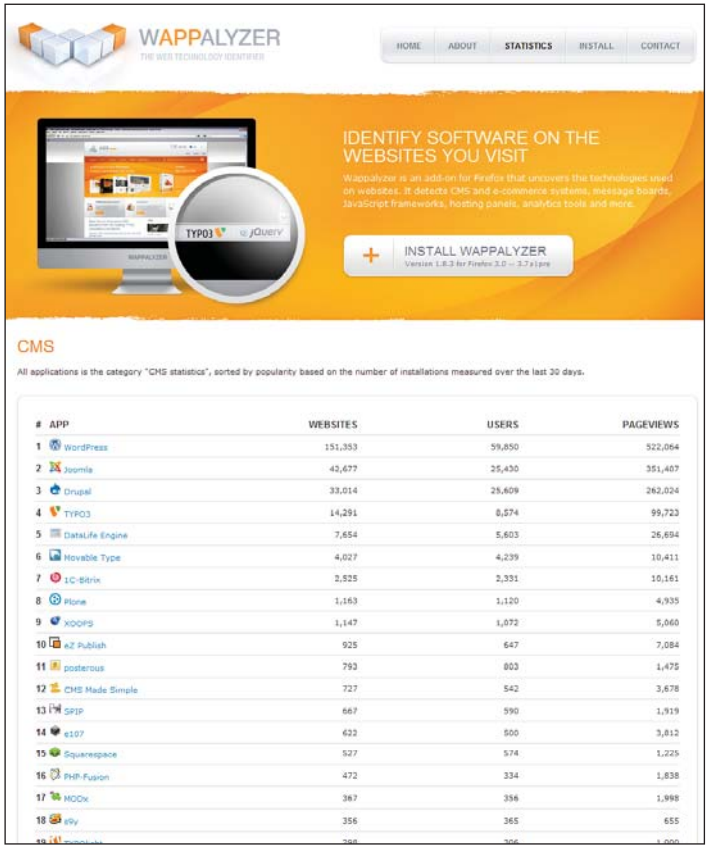


Figure 27: Wappalyzer

Looking at these statistics gives a general perspective on the size and popularity of the CMS market.

> "There are over 22 million *Wordpress* publishers as of February 2010: 10.6 million blogs hosted on *Wordpress.com* plus 11.4 million active installations of the *Wordpress.org* software... According to *Quantcast*, around 250 million people visit one or more *Wordpress.com* blogs every month, and they view over two billion pages on those blogs" (*Wordpress*, Statistics, 2010).

There are numerous ways of measuring web site traffic and obtaining demographic data. Another key player in this field is *quantcast.com* which has analysed *Wordpress.com* over a 5 month period, and recorded 22.6 million visitors to the web site (see: http://www.quantcast.com/p-18-mFEk4J448M, 2010). *Wordpress* has two web sites; *Wordpress.org* which delivers server-based installations, downloads and forums (free to the user, with additional plug-ins available for a fee), and the commercial version *Wordpress.com*, where the user connects to a hosted solution.

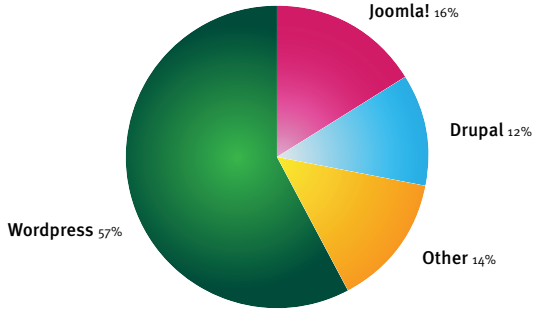


Figure 28: Wappalyzer - Web sites

Lastly, it is worth mentioning *Alexia*, which is another useful resource for designers to learn about web sites. Provided it is also tracking your web site, *Alexia* will allow a comparison of traffic across other web sites. Of the five CMS systems trailed in the *WhichCMS?* workshop, five were present, ranked in the following order as of June 2011:

- *Tumblr* 146th
- *Wordpress* 184th
- *Joomla!* 338th
- *Drupal* 547th
- *ExpressionEngine* 7,729th
- *Concrete5* (not ranked)

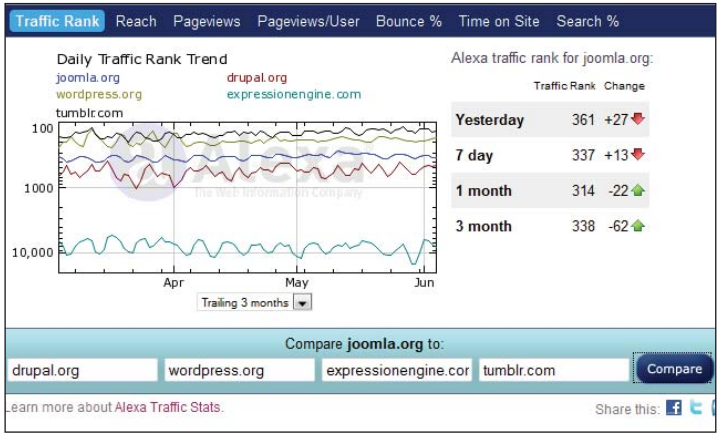


Figure 29: Alexa comparison of site traffic - June 2011

Since *Wordpress* has been identified as a market leader, it is interesting to compare it's ranking on some of the previously mentioned web traffic monitors. On *Qantcast*, *Wordpress.com* was ranked 24th, *Alexia* placed it 16th and on *Ranking.com* it is also 16th. On the same three web sites, *Wordpress.org* was ranked 2,154th, 184th and 477th respectively. Of the five systems listed above *Tumblr* is the only CMS to live host their 'development' web site so visitation includes developers and the broader community browsing complete web sites. These web site ranking tools tend not to include very small CMS operations in their data at all. These statistics only serve as an indication of popularity. They do not provide any information on the rate of adoption/installation or the experience of the user.

There are many other factors to consider when selecting the CMS, such as core functionality, user interaction, multiple web site support, multilingual support, managing assets and ease of customisation to name a few (Boag, 2010 p156).

> "I should note here that I would apply the same recommendations for evaluating commercial software if I could. However, software companies do not expose who the brain behind the technology is, how helpful the tech support is, and how the organisation tends to respond to turnover" (Boag, 2010 p167).

Of the five systems listed above, *Tumblr* is the only CMS to live host their 'development' web site so visitation includes developers and the boarder community browsing complete web sites. These web site ranking tools tend not to include very small CMS operations in their data at all. These statistics only serve as an indication of popularity. They do not provide any information on the rate of adoption/installation or the experience of the user.

A fifth option would be to look into awards a new area in this fledgling sector. *Packt Publishing*, publishers of web-development related books recently awarded *Wordpress* the "2009 Overall Best Open Source CMS Award" there were "...over 12,000 nominations and over 23,000 votes received across its five categories" (PacktPublishing, 2009) additionally "Hall of Fame Award in the 2009 Open Source CMS Awards. With this award, *Drupal* has won two categories this year, the other one being Best Open Source PHP CMS."

Another direction might be to follow the market trends and industry insights by *Gartner Research*, considered the world's leading information technology research and advisory company. *Gartner* provides a report called the 'Magic Quadrant' for Web Content Management.

> "Magic Quadrant will help CIOs [Chief Information Officers], and business and IT leaders that are analysing their Web strategies to assess whether they have the right WCM [Web Content Management] offering to support them. Because the technology has changed so much in recent years, we strongly advise organizations that have WCM technology that is more than four years old to re-evaluate their WCM strategies" (Gartner, 2009).

Gartner uses a patented methodology (the structure of which they are quite secretive about), to position a CMS within one of their four quadrants. According to their assessment a particular web tool is categorised as a leader, Challenger, Visionary or niche player. (Wikipedia, Magic Quadrant, 2010)

If we look at Figure 30 we can see *Drupal* occupying the 'visionaries' category, and others that are moving across from being 'niche player's. Although this assessment may seem somewhat esoteric, to a seasoned programmer it gives a rather interesting indication of the enthusiasm with which they should potentially be approaching these products, and which are worthwhile keeping an eye on.

In this sense,as Byrne points out (2009), the criteria that Gartner have developed here are more immediately useful to investors and larger development companies, than users o rindividual buyers, however, for the perspective it provides, this work is certainly worth touching on, not least because of the general dirth of available information on these products out there, which has already been mentioned. It is certainly apparent that the open source community is not significantly represented when compared to its commercial cousins here. This appears to be because *Gartner* analysts are in communication with clients at the enterprise level and do not hear 'noise' in the open source community (Byrne 2009).

While *Gartner* has steadily improved their methodology over the last few years, it is far from perfect. One problem is that there is not complete transparency when it comes to all criteria, especially the ones based on the analyst's subjective opinions. (Sagecircle, 2008)

So Gartner seems to only offer enterprise-level review comparison listing very few open source CMS solutions in its review. It seems that an organisation has to be in direct communication with *Gartner* analysts to increase their profile. "By maintaining relationships with all of these analysts, and showing them all the great work we have done, we can get *Drupal* to the next level in terms of enterprise adoption." (Buytaert, 2009)

Finally, you could visit web forums and acquire a personal perspective from users. Most CMS providers will have a forum hosted on their web site. Often a topic such as "*Joomla!* vs *Xoops* vs *Typo3* vs *Drupal*" (ajwagner777, 2007) is an attempt to draw responses from interested parties. The postings are always interesting reading and solicit passionate debate. Learning a CMS is something that takes a significant personal investment in time; once you have learnt one system most users are reluctant move onto another, so there are always those ready to defend 'their' CMS.

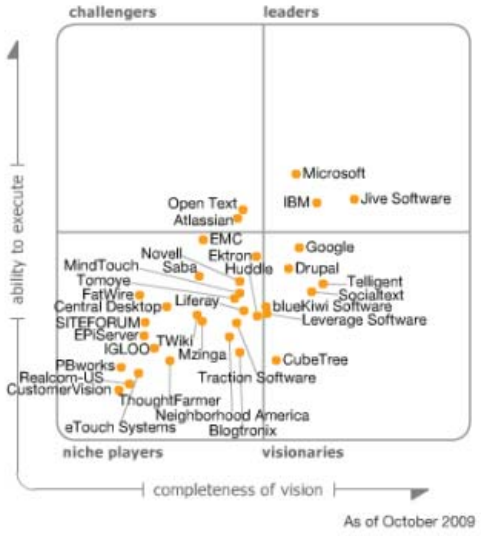


Figure 30: Gartner's Magic Quadrant "Visionaries" Category

Schools keeping up with industry

In some ways the goal of keeping students up to date with what is most current in web technology is fundamentally incompatible with the way in which curriculum is decided upon and implemented in most academic tertiary institutions. The process of implementing curriculum changes in tertiary institutions is, rightly, a laborious and carefully measured one, and not something which is undertaken lightly, or quickly. For this reason, Leslie Jenson-Inman (2009) believes it is arguable that tertiary providers who are teaching in the area of web design should have, built in to their curriculum, a process of review, in order to consider recent developments at regular intervals. In her article on how web design is taught at University level, teacher and designer Jenson-Inman quotes James Archer, of design studio *Forty*:

> "The culture of large educational institutions has, in my experience, consistently proven itself unable to cope with the demands of such a varied and fast-moving industry. I know many good people are trying, but I've yet to see anyone come out of a university program knowing what they'd need to know in order for us to hire them" (Jensen-Inman, 2009).

The teaching of open source software could form part of a unit dedicated to current trends in industry. As McEwan points out: "Although it is not always possible to use technology really well the first time, it is important to make a start" (2010, p. 18). This acceptance of the concept of at least dipping a toe in the waters of unknown territory, if not wading into the deep end, is an important one. We want our students to be innovators, and we will not achieve this by teaching them to adhere to that which is already known, whether it be method or knowledge. From this perspective, teaching students to take on new and problematic software, and to feel that the knowledge gained is worth a less than perfect project outcome first time around, could be a better result than getting them to produce something perfect with outmoded tools. As McEwan puts it: "Schools should be aiming to develop a culture where staff members are happy to share and take risks with new technology..."(2010). Making students comfortable with the concept of continuing experimentation and self-education after they are employed, could do a lot for their career prospects in the long run.

There is a consistent problem across the spectrum of education with keeping teachers up to date with advances in digital technology. The demand that the digital age has placed on teachers to keep their skills and knowledge up to date is considerable, but nowhere is this task more daunting, or more necessary than in the realm of Information Technology teaching.

> Design teachers may face the added hurdle of not being currently employed within their own industry whilst they are teaching.

There are, however, some excellent resources on the Internet to help teachers of design and their students stay current.

Knowing where to go for answers when they are needed is important for teachers and students to feel secure and engaged. Teachers and students might work together and pool their resources onto a *Diigo.com* or a *Symballoo.com* web page/s, so that a collection of places to go when problems arise can be created and maintained.

There are many excellent resources available online to assist students and teachers who want to stay in contact with what is most current in web technology and design. The *Web Standards Project* (WaSP) is an initiative designed to ensure equality of access to web technology. A task force attached to this group was created in 2005 to work with schools in the U.S., and some of the resources they created, such as their coursework book Information: *Architecture to Accessibility*, amongst others, are available at their web site: http://www.webstandards.org/.

Also available here is the *WaSP Curriculum Framework* (WCF), designed to help teachers create lesson plans for the incorporation of new web technology which will remain compatible with the stated learning outcomes of their existing courses.

> "Departments need to create a culture of learning that requires faculty to stay abreast of new topics. Schools should make it a priority to send faculty to conferences and training programs to ensure they're not falling behind" (Jensen-Inman, 2009).

The *Web Standards Project* identifies one of the leading arguments for turning to open source software into the tertiary realm – the question of equity and accessibility. If students of web production are restricted to the superficial aspects such as design, their ability to act as innovators and contributors to the web environment will be affected. Cathedral-style web software is beyond the price range of most start-up companies and freelancers, and is becoming rapidly more expensive. Clearly the ultimate effect of this is a reduction in the number of people who have the means to publish. Over time, this could result in control of our web environment being divided between a small number of key players - much the same as has already happened in the music, television and film industries. Keeping web students on top of new technology is not just important for enhancing their job prospects, therefore, but is important in a wider socio-political sense also. It may play a role in the fight to preserve democracy in cyberspace.

> "The Web Standards Project introduced a practical message that practically nobody wanted to hear. If *Netscape* and *Microsoft* persisted in building evermore incompatible browsers, the cost of development would continue to skyrocket, tens of millions would find themselves locked out and the web would fragment into a tower of digital Babel." (WaSP, n.d.)

Another online initiative in this direction is *Opera Education*, a web site which offers students the opportunity to experiment with and learn more about the web, and which provides step-by-step scaffolds for various projects, including designing a campus hub. In association with *Yahoo!*, *Opera* has created a Web Standards Curriculum. This is a set of tutorials which take students through all the basics of web design, including Javascript, HTML and CSS. The web site has been constructed so that students have the choice of dropping in for individual units of learning, or following the set of lessons from start to finish. Each of the units of learning is, in turn, divided into separate articles, maximising its usefulness to educators.

Leslie Jensen-Inman has put her classroom web exercises online at http://www.teachtheweb.com/ also available on the web site are her interviews with over thirty leading practitioners. In these interviews, she asks them what is essential knowledge for students hoping to gain entry into the field of web design and implementation. For some of these industry leaders, the response was a wish-list of programs, and programming languages, such as CSS, MySQL and XHTML (Jensen-Inman, 2009). Interestingly, almost all of those interviewed included open source resources in their list.

> A course which focuses exclusively on practical skills, however, whilst it may prepare students to be effective workers, does not prepare them to be art directors.

If tertiary courses were designed in this way, from a pick-list of software, they would simply be white-collar apprenticeships. Others that Jensen-Inman spoke to emphasised the value of theory and aesthetic principles. The question is, given the relatively short amount of time that tertiary institutions have to prepare web design students for their future profession, and given the tension which already exists in these courses between design theory and principles, and practical skills, is there room for one more unit on open source CMS?

Another problem that Inman-Jensen identifies is that of Universities favouring those with qualifications over those with industry experience when it comes to choosing those who will teach their students. Many of those interviewed highlight the problem of teachers who have failed to keep up with current trends and are therefore teaching their students out of date practices. Jensen-Inman's interview with Greg Storey, principal of *Airbag Industries*, highlights this problem:

> "...I was contacted by a large university about teaching web design and was quite interested. Then they found out I had no graduate-level degree. So instead, they hired a retired Java programmer to teach 'web design'" (Jensen-Inman, 2009).

What is being taught by institutions in the Sydney basin?

In order to better understand what was being taught to design students in the Sydney basin I researched by speaking to the academic teaching staffand created the following table - Figure 31. The research focuses on diploma and degree courses taught in the Sydney basin. All courses with a design focus deliver static HTML/CSS based instruction.

Institution	Course	Web Address	Focus
TAFE: Ultimo	Diploma in Information Technology (Web site Development)	http://www.sit.nsw.edu.au/courses/search.php?cid=26013&area=courses&Media_Index_ ID=169	Development – Java/database
TAFE: Enmore	Diploma in Graphic Design and Communication Course	http://www.sit.nsw.edu.au/courses/search.php?cid=25654&area=courses&Media_Index_ ID=169	Design – HTML/CSS
University of New South Wales - COFA	Bachelor of Digital Media	http://www.cofa.unsw.edu.au/degrees/undergraduate/bachelor-of-digital-media-bdm/	Design – HTML/CSS
Billy Blue Design College	Bachelor of Applied Design (Digital Media)	http://www.billyblue.com.au/courses/digital-media-design	Design – HTML/CSS
Martin College	Diploma of Graphic Design	http://www.martincollege.edu.au/courses/diploma-of-graphic-design-advertising-multimedia-course.aspx	Design – HTML/CSS
Qantm College	Bachelor of Creative Media (Major in Graphic Design)	http://brisbane.qantm.com/en-gb/course/2507/Bachelor_of_Creative_Media_(Major_in_Graphic_Design)	Design – HTML/CSS
Qantm College	Bachelor of Creative Media (Major in Interactive Media)	http://brisbane.qantm.com/en-gb/course/2506/Bachelor_of_Creative_Media_(Major_in_Interactive_Media)	Development – HTML/CSS/PHP/ MySql ActionScript 3
University of Newcastle	Bachelor of Visual Communication Design	http://www.newcastle.edu.au/program/11253.html	Design – HTML/CSS
University of Newcastle	Bachelor of Information Technology	http://www.newcastle.edu.au/program/11497.html	Development – HTML/CSS/PHP/ MySql ActionScript 3
University of Western Sydney	Bachelor of Information Communications Technology	http://future.uws.edu.au/ug/arts/bachelor_of_arts_combined_information_communications_technology	Computer Science
University of Western Sydney	Bachelor of Design (Visual Communication)	http://future.uws.edu.au/ug/creative_and_communication_arts/bachelor_ of_design_visual_communication	Design – HTML/CSS
University of Canberra	Bachelor of Information Technology	http://www.canberra.edu.au/courses/index.cfm?action=detail&courseid=322AA	Development – HTML/CSS/PHP/ MySql ActionScript 3
University of Canberra	Bachelor of Graphic Design	http://www.canberra.edu.au/courses-units/ug/design-architecture/bgd	Design – HTML/CSS
University of Technology Sydney	Bachelor of Science in Information Technology	http://it.uts.edu.au/course/undergrad/local/bscit.html	Development – HTML/CSS/PHP/ MySql ActionScript 3
University of Technology Sydney	Bachelor of Design in Visual Communication	http://datasearch.uts.edu.au/dab/courses/details.cfm?spk_cd=C10059&sp k_ver_no=8	Design – HTML/CSS
University of Wollongong	Bachelor of Information Technology	http://coursefinder.uow.edu.au/coursefinder/CourseLevelDetail.aspx	Development – HTML/CSS/PHP/ MySql ActionScript 3
University of Wollongong	Bachelor of Creative Arts (Graphic Design)	http://coursefinder.uow.edu.au/coursefinder/CourseLevelDetail.aspx	Design – HTML/CSS
University of Wollongong	Bachelor of Digital Media	http://coursefinder.uow.edu.au/coursefinder/CourseLevelDetail.aspx	Design – HTML/CSS
Wesley Institute	Bachelor and Diploma of Graphic Design	http://www.wi.edu.au/courses/creative-arts-courses/graphic-design-courses/graphic-design-course-information	Design – HTML/CSS
Raffles	Bachelor of Design	http://raffles.edu.au/index.php/Bachelor-of-Design/Graphic-Design.html	Design – HTML/CSS
Raffles	Bachelor of Arts (Visual Communication)	http://raffles.edu.au/index.php/Bachelor-of-Arts-VisComm/Multimedia-Design.html	Design – HTML/CSS

Figure 31: Courses being taught in the Sydney basin - authors table

What is the range of considerations in teaching a CMS?

Firstly, prior knowledge and the need for prior knowledge must be assessed. Just as it is a mistake to assume that all people over the age of fifty are computer illiterate,

> It is wrong to assume that design students will come to tertiary study already aware of developments in the open source realm.

Depending on what kind of background they come from, and what kind of school they attended, design students will have varying degrees of knowledge as to how a dynamic web site functions. Most of them will be users of this technology already, but this does not imply an internal or working knowledge. Most of us use a clock each day, without knowing how to put one together. It is therefore necessary to ascertain a student's level of coding experience at the beginning of a course.

This could offer a potential obstacle to the successful implementation of a unit of study based around open source CMS.

> Without at least some prior knowledge of coding, or at least of how coding works, teaching the students a course in open source software could prove problematic.

If the material is not authentically seated within the total learning experience of the student, and does not build on and add to the skills they are acquiring, it could prove superficial. Any new unit must effectively enhance what the students take away from the course. In order to get the most out of a study of open source software, and to be capable of incorporating it into the web sites they are making, some prior knowledge of generic HTML and CSS is needed. This could be a problem, depending on how the student's course is structured, since at some design schools, coding is optional.

> "There are pedagogical and even philosophical reasons why the CS [Computer Science] faculty should encourage students to learn about the open source software movement. Open source software is a model of software development that competes with commercial models. Universities are the ideal place for students and faculty to consider and debate the advantages and disadvantages of competing models. Lively discussions can sharpen student thinking both about these competing models and about deep issues in computer science" (Wolf et al, 2002)

A second consideration for the teacher who wants to use open source software within the context of web design classes, is which CMS to choose, from the considerable array of options available at the present time. In making a choice, the teacher of design must consider the desired student outcomes. In formulating an introduction to dynamic content delivery systems, the teacher must consider that to canvass all available options will be too perplexing for someone encountering open source for the first time. The question of how can teachers make informed choices about technology and tools in such a rapidly changing environment must be can only be answered with a systematic approach to trial and research. Up-to-date information is available on CMS review web sites. Some of these web sites, such as http://www.cmscritic.com, require the user to subscribe but others such as http://www.cmsdesignresource.com, provide information and impressions of usability for free.

Ideally of course, the teacher should make time to trial some of the available software themselves. There are currently over 240 web sites on the Internet which host live demonstrations of open source software. These web sites do not require installation and provide students and teachers with the opportunity to get in and interact straight away.

> The site http://www.opensourcecms.com/ has demonstration versions of over 70 open source content management systems.

The following is a table containing some of the more useful open source review web sites on the Internet, which have been assessed for recent activity. As Gotleib points out there are also online blogs for industry experts:

Site	Updated Regularly	Information accessible to a non-expert
http://www.cmscritic.com/	✔	✔
http://www.cmswatch.com/	✔	
http://www.cmsinfo.org/	✔	
http://www.cmsreview.com/	✔	✔
http://cmsreport.com/	✔	
http://www.cmswire.com/	✔	✔
http://www.cmsdesignresource.com/	✔	

Figure 32: CMS news related web sites

> "Having several people actively posting answers is a sign of a strong community…Browsing through the bug tracking system will tell you how active the software being tested is, and how efficiently issues are being resolved. Do not assume that having lots of issues in the bug tracking system is a bad thing. It means that the software is being used by people that care enough to work with the community to improve it" (Gotleib, 2006).

The challenge for teachers is becomming aware and being willing to try and experiment with new CMSs. Demonstration, non installation sites like those listed in Figure 32 provide the perfect opportunity to 'dabble' with numerous CMS most of which provide industry commentry and the latest versions of the software.

Analysis of 28 CMS solutions

There are thousands of CMS solutions in the market place. Here I present an overview of 28 CMS's in order to better understand the subtle differences between the software options. This information has been presented in the format of a table for quick reference and easy reading.

An analysis of these differences illustrates the variations between products. Included here are researched user profiles for each of the products highlighted.

The research here has been derived from the literature review, discussion with individuals using the systems and learning many of the CMS solutions either via the providers site or the online site:
http://php.opensourcecms.com/

Drupal

WhichCMS workshop choice
Open Source
PHP
drupal.org/

Drupal is suitable for individuals or non-profit organisations that need a cheap solution. It can be used to create a web site which is very client-friendly, but it does require a web developer with knowledge at the implementation stage.

Part of the appeal of this CMS is the ethos behind open source software contributing. This allows the user to feel part of a community which can help with problem solving, and to contribute custom developed extensions of their own. A leading contender in the CMS market *Drupal* can do most things required on a web site.

Standout Features

- Many modules are available to enhance utility
- High level of control makes personalisation possible with mimimal tool instalation. This is refered to as the 'core' installation.
- Limited base install offers a small footprint
- Created over a number of years and maintained by a large community of over 900 developers
- Not recommended for a first time CMS user

Joomla!

WhichCMS workshop choice
Open Source
PHP
E-Commerce
Joomla.org/

Joomla! provides an excellent range of well designed templates for the developer to choose from. Based on PHP, it is suitable for someone who does business in more than one locale, and therefore needs an established platform for developer uptake and capability.

With a minimal amount of guidance most clients will find *Joomla!* very easy in terms of maintaining and updating content.

There is also a good supply of social network plug-ins for instant implementation.

Standout Features

- Easy installation
- Developed over a number of years and maintained by a small team of dedicated programmers
- Flexible configuration
- Free plug-ins for all commerce options
- Well renowned ease of use with e-commerce
- A good quality collection of templates

ExpressionEngine

WhichCMS workshop choice
Commercial low cost
PHP
expressionengine.com/

ExpressionEngine (EE) is a CMS that allows small companies and freelancers to offer their clients a powerful CMS, and thus provide functionality that would normally be the domain of a web development company, and come at a much higher price tag.

This CMS comes with good community support via forums and will reinforce basic XHTML skills without requiring the developer to become expert in PHP. *ExpressionEngine* has a 'code-centric' approach which gives the developer complete control over every element.

For those purist who shun template design, this CMS will allow the user to create unique designs, unlike *Wordpress* which uses default templates.

Standout features include:

- Custom fields
- Easy scripting and excellent documentation
- Good base installation of plug-ins for functionality
- Template files can saved
- Powerful built in search engine
- Image re-sizing and thumbnails within the system
- Vibrant design community

Concrete5

WhichCMS workshop choice
Open Source
PHP
concrete5.org/

Concrete5 is easy to implement and will impress clients with its ease of use.

Using this CMS it is possible to take a basic brochure site, custom designed in *Adobe Dreamweaver*, and convert it to a site where clients can edit their own content, with only a few extra lines of code.

For the non-expert client, editing in *Concrete5* is essentially just like using a word processor. They simply navigate to their desired page then press EDIT and edit the content.

Standout features include:

- Complete base install
- Easy templating system
- WYSIYWG editing system, in context page editing
- Flash, sound and video friendly
- Easy form generation
- Very client and designer friendly

Wordpress

WhichCMS workshop choice
Open Source
Specialist blog, can be forced into a CMS
PHP
Wordpress.org/

Wordpress offers a vast array of free and commercial plug-ins to extend a web site's capabilities. It is also possible to add social network buttons for content sharing.

This CMS would suit someone who is a blogger, but is not really interested in learning any code. Essentially it is an easy 'Lego-like' solution where users can configure their site and immediately publish.

Initially a site made using *Wordpress* can be hosted at www.Wordpress.com but for those who seek more customisation it is possible to download a version to expand at www.Wordpress.org

Standout Features

- Most popular blog platform
- 5 minute setup and installation
- Very easy upgrade path
- Huge community of plug-in developers
- Build in Search Engine Optimisation (SEO)

Tumblr

WhichCMS workshop choice
Developer-hosted platform
Specialist blog
Ruby on Rails
tumblr.com/

For students, or anyone who likes sharing and engaging with others online, *Tumblr* provides a wealth of inspiration. Any 'liked' content the user locates can appear directly on their site at the click of a button. Uploading visuals is easy, and it provides the facility for others to 'follow' and post comments.

Tumblr provides endless distractions as it feels like Facebook meets Twitter meets Flickr all in one site, where content is automatically updated.

Standout Features

- Simple free sign-up process
- Controlled template design system but not extendable via plug-ins
- Limited and controlled environment focused on the 'Tumblelog'
- Constantly changing content and peer followed content
- Great community feeling
- Followers review and can comment

Blogger

Developer-hosted platform
Specialist blog
.NET and Java base
blogger.com/

Blogger is ideal for those who would prefer to be creating content than learning code. Since its inception in 1999, thousands of contributors have used *Blogger* to publish online. The learning process is intuitive, and although it does not offer complete aesthetic control, it is certainly one of the easiest ways to get your web contribution out there.

Google uses *Blogger* for its own blog: http://googleblog.blogspot.com/

Standout Features

- Simple free sign-up process
- Great entry level into blogging
- Allows for customisable templates with a drop and drag interface
- Limited ability to expand system functionality

Moveable Type

Open Source
Perl
movabletype.org/

Moveable Type is an excellent next step for someone who has experienced blogging, but wants to move into ecommerce.

This solution provides the capacity for basic marketing and transactions, as well as the ability to run a poll and collect feedback.

Standout Features

- Extremely easy to use
- Easy custom fields
- Simple templating
- Long development history so interface and plug-ins work well
- Pro and Enterprise account level for more advanced features but base is still very complete

Text Pattern

Open Source
PHP
textpattern.com/

Textpattern has evolved from a blogging application to a general purpose CMS suitable for the development of a wide variety of sites.

This is not the most powerful CMS solution available, but it is relatively easy to learn, and provides ample capacity to run a small, from-home business.

Standout Features

- Has its own custom scripting environment called Textile which is a text to HTML converter allowing publishing without knowing HTML.
- Has a built in CSS editor
- Template system in which the developer builds reusable blocks of code
- Has slow development cycle and team

Pligg

Open Source
PHP
pligg.com/

Pligg provides users with the ability to track categories of information on the net. It will create a site where an internet community can catch up on the latest information in their area of interest, as well as uploading their own contributions and voting on popular stories.

Similar to the popular web site *Digg*, *Pligg* offers traditional voting and a 5 star rating method.

Standout Features

- Dedicated social networking CMS
- Uses the popular 5 star rating system + the voting system
- RSS support
- Ranking system for high volume users
- Good collection of plug-ins to extend functionality
- Built in member to member messaging
- Good built in Search Engine Optimisation (SEO)

Light CMS

Hosted
Commercial
lightcms.com/

For a business which is located online, *LightCMS* provides many useful features, such as direct client billing, easy install procedure and re-skinable branding.

Because it is commercially supported users are instantly able to have any questions answered.

One drawback of this CMS is that it has limited customisation in terms of new functionality.

Standout Features

- Ease of use
- Excellent commercial technical support
- Excellent design control
- Total package with complete control over content

Umbraco

Open Source
Developer focused
.NET platform
umbraco.com/

Umbraco is suitable for a net developer who needs a highly customisable CMS, rather than an 'out-of-the-box' solution.

Even for those with some experience, this CMS will offer a steep learning curve at first, but coding purists will like the fact that *Umbraco* Is not plug-in related. It also comes with an impressive online community, allowing the developer to contribute to and draw upon the knowledge of others.

Standout Features

- Focused on developers
- Customisation because it is not tied to plug-ins
- Excellent subscription based training with Umbraco TV

CMS Made SImple

Open Source
client editing focused
php platform
cmsmadesimple.org/

CMS Made Simple has focused on an easy user interface, by simplifying the user administration area. The CMS is really focused on a lightweight site with static content and very client-friendly usability.

CMS Made Simple has all the usual tools such as polls, blogs, shopping carts and forms. Administration interface could be refined with some more up-to-date graphics.

Standout Features

- Very easy install
- Installs with dummy content that makes it easy to understand but must be replaced with client content
- Good collection of modules for standard site activities
- Help is limited to the online community

Wix

Free
Hosted platform
Flash based
wix.com/

Wix creates Flash driven web sites with an intuitive interface with easy drop and drag functionality. Essentially it enables the client to create their own simple sites.

Whilst *Google* and other search engines can now read such sites, a Flash-based site is always going to have limitations compared to standard HTML, primarily because of Flash's lack of adherence to WCAG 2.0 accessibility. *Wix* was clearly intended to appeal to users of *MySpace* in terms of the level of graphical customisation offered.

Standout Features

- Very popular with over 8.5 million sites created (Wix web data)
- Easy to use and learn
- Has animated Flash capabilities
- Ships with *MySpace* templates
- Completely free (with ads) however has a premium option up to $25 per month for more features
- Has a variety of templates from blogs to e-commerce

Flash Moto

Commercial
Self-hosted Platform
Flash based
motoCMS.com/

Moto CMS (previously *FlashMoto*) is the most powerful Flash CMS available and its online editing tools feel just like a desktop application.

MotoCMS is aimed at designers and developers with pre-built templates to allow them to deeply customise the template using *Actionscript3* skills. The development framework is very specific and not too flexible. The CMS is capable of playing video, MP3 files and photo galleries, building forms, and file management with a browser/uploader.

Standout Features

- Very attractive and professional GUI great for clients
- Excellent WYSIWYG with immediate results
- Installation can be difficult and there is a lack of supported hosts
- Has a commercial cost of $200

Xoops

Open Source
Self hosted
PHP
xoops.org/

Xoops, or: 'eXtensible Object Oriented Portal System' has moved beyond being a portal CMS since its inception in 2002. The current edition can be seen as a web application platform.

Xoops has a smaller learning curve than its leading competitor *Drupal*. Its administration interface, called the 'Backoffice', is well designed and re-skinnable, with hundreds of themes available online.

Xoops is capable of a reduced installation with its 'news' module functioning as a blog which can be installed on web sites. *Xoops* is extendable like *Drupal* and *Joomla!* however the system has had periods of limited development and many modules have not been updated in many years.

Standout Features

- Capable of small to large sites
- Worldwide volunteer network of several hundred
- Powerful permissions system

ModX

Open Source
PHP
modxcms.com/

MODx, is a CMS and web application framework (WAF). It is compliant with web 2.0 technologies like *AJAX* and *Javascript* libraries and suitable for more than a simple brochure site.

MODx has a small development community which may hinder growth in the future if the software is not updated.

MODx has a global community and its interface has been translated into many languages. Recently *MODx* received the 'Most Promising Open Source Content Management System 2010' award from Packet Publishing.

Standout Features

- A strong CMS for web designers to implement client sites - ease of use is its strongest feature
- Easy to understand hierarchical organisation which is logical and easy to understand.
- In-text editing
- Built in SEO

Radient

Open Source
Ruby on Rails
radiantcms.org/

Radiant is one of a small number of open source CMS solutions for the Rails platform. *Radiant*'s focus is on making a good experience for web designers and editors with an elegant back-end interface.

Ruby on Rails can be difficult to install as this platform is not as widely supported as PHP.

Radiant uses its own simple scripting language called *Radius*.

Standout Features

- A limited installation core keeps setup simple
- Uses snippets of text for repeated items
- Relatively new to the CMS market
- Administration area is split into 3 areas: snippets, pages and layouts
- Currently the most complete choice on the Ruby platform
- Over 200 extensions available

Rail Frog

Open Source
Ruby on Rails
railfrog.com/

Railroad frogs are the joining structures where trains change direction. They guide the train between the rails.

Railfrog is still in its infancy and announces it's intention of being 'the CMS for the rest of us'. Developers of this CMS have stated that they plan to keep user simplicity foremost with a small base install and later provide extensions for extra features.

Standout Features

- Still in 'development'
- Currently the only competitor to Radiant in open source Ruby on Rails CMS marketplace
- Makes standards compliant pages
- Web 2.0 integration such as AJAX

Pixie

Open Source
self hosted
php platform
getpixie.co.uk/

Pixie is designed to be a CMS for small web sites. The software has focused on easy installation and Graphical User Interface (GUI).

In 2008 Packt Publishing gave *Pixie* its award for most promising open source CMS.

Pixie treats theming the same way *Wordpress* does with a custom setup for core and CSS pages. In fact this CMS feels like a simplified version of *Wordpress*. It limits what is presented to the editor to avoid confusion.

Pixie development has also focused on microformats so it easily imports and exports *RSS*, *Twitter*, *Flickr*, *Delicious* and other social bookmarking tools.

Standout Features

- Ease of use for clients and developers
- Simple install
- Custom built in WYSIWYG editor - CKEditor for text areas
- Search engine friendly with clean URLs
- New themes arriving all the time

Moodle

Open Source
self hosted
php platform
moodle.org/

Moodle is focused on e-learning sometimes know as a Learning Management System (LMS). It is a free online application that helps teachers create online courses.

Moodle is the leading choice in the LMS sector for free software. It is able to handle large deployments with thousands of students and conduct courses fully online.

According to the *Moodle* site there are over 45,000 communities making up 32 million users in 205 countries.

Moodle has benefited from a diverse open source community who actively contribute development to *Moodle*.

Standout Features

- Fully built-in WYSIWYG editor
- Forums, journals, chat surveys and assessment submission
- Build in tools to make online learning fun like quizzes, multimedia and a Wiki

MediaWiki

Open Source
self hosted
php platform
mediawiki.org/

A Wiki is simply a web-based application for editing content on a web site. Wiki's are specifically designed to allow communities to upload and edit content. The idea behind a Wiki is that the content is read by many and that many experts discuss and approve final content.

Wikis have become a powerful direction in CMS solutions and are perhaps the easiest way to groups of people to collaborate together on content. *MediaWiki* is the most popular and widly used Wiki. It is used on some of the more well known sites like *Wikipedia* and *Wikinews*.

Standout Features

- Free
- Good for large and small projects
- Large amount of extensions
- User interface ported to over 300 languages
- Highly configurative
- Over 1600 extensions

Cushy

Closed Source
Commercial
hosted
cushycms.com/

CushyCMS is a very easy to implement and edit CMS focused on small business. The best thing about this CMS it that it is so easy for clients to edit and graphic designers to setup the pages.

It is best suited to static pages and is not extendable or customisable apart from basic CSS and HTML.

To implement admin adds class="cushycms" to a DIV box which will then be editable with a WYSIWYG editor - easy.

Standout Features

- No programming knowledge needed - great for a brochure-ware site
- Developer can define on a static page what is editable and what is not
- A free or Pro plan ($28p/m)
- Owners never see the 'backend' of template design
- Clients 'get it' and find editing extremely easy
- All database backend site install is done by the host

OS Commerce

Open Source
Self hosted
PHP
oscommerce.com/

osCommerce, as the name implies, is focused on e-commerce. For the novice business owner, a small commitment of capital to fully develop and maintain a site using this CMS will be needed, but once created, ease of use for the client is assured.

osCommerce is an e-commerce specialist application that is also a CMS in its own right. *osCommerce* is not a plug-in solution, so users need to select one of the supplied templates for the setup. No knowledge of PHP or MySQL is required by the developer for the install-wizard setup, and there is a large online community to take advantage of, including many useful tutorials.

Standout Features

- Dedicated e-commerce solution
- Extremely popular with over 12,700 web sites using the solution according it its founder
- Installs with *Fantastico* web hosting
- 6,400 'add-ons' extending functionality

Alfresco

Open Source
Self hosted
Java
alfresco.com/

Alfresco is an Enterprise Content Management (ECM) solution that avoids the high cost, complication and limited user control of previous solutions in this category.

Alfresco is the leading open source ECM providing more features than a standard CMS and capable of delivering the functionality of solutions that would usually cost tens of thousands of dollars.

Standout Features

- 2 installs, Alfresco Community Edition which is free and Alfresco Enterprise Edition which is commercially supported and expandable.
- Built with Java
- Low cost alternative to Sharepoint
- Simple scripting to further enhance the system
- New direction focus on social content development

Sharepoint

Open Source
Self hosted
.NET
sharepoint.microsoft.com/

Sharepoint has long been considered the de facto standard for Enterprise Content Management (ECM) solutions in some circles.

This system allows for intranet portals, extranets, document management, collaboration, web application and, of course, web sites.

Initially provided free, a premium edition features greater functionality and expands the capabilities of the ECM. Users will find the document management features, which facilitate sharing with colleagues, one of its best features.

Standout Features

- Requires dedicated Microsoft hardware
- New integration into Microsoft Office 2011
- Categorises content into '6 Pillars' - sites, composites, insights, search, content and communities
- Focused on data storage 'in the cloud'

Vignette

Open Source
Self hosted
Java and .NET
vignette.com/

Large companies like *Disney*, *FoxNews*, *National Geographic*, *NASA* use Vignette as their Enterprise Content Management (ECM) solution of choice.

Vignette is an established ECM provider however it has struggled recently in this highly competitive marketplace to remain at the forefront of the sector.

Similarly to *Sharepoint* , one of Vignette's strengths is its document storage and retrieval operations.

Vignette ships with various products: Vignette Content Management, Vignette Portal, Vignette Community, Vignette Video, Vignette Recommendations and Vignette Records/ Case Manager.

Standout Features

- World class enterprise content management system
- Vignette is expensive, very expensive and a typical deployment can cost over $250,00. It is difficult to acquire an actual cost as the
- Recognised as a leader in the Gartner Report's Magic Quadrant

Coroflot

Free - Closed system
Portfolio focused
Hosted solution
coroflot.com/

Coroflot is the largest free hosted portfolio service catering to architecture, Industrial Design, Illustration, and Print.

Since its inception in 1998 *Coroflot* has focused on industrial designers. It now supports the largest and most established body of creative portfolios on the web. Unfortunately it is not possible to customise the appearance of her desired template.

Standout Features

- Very professional minimal design templates
- Very easy to use
- Custom URL for portfolio
- No coding necessary
- Jobs seem to be mostly USA-based
- Anyone can join

Behance

Free - Closed system
Portfolio focused
Hosted solution
behance.net/

Behance is a company split into a few divisions. It offers creative clients the ability to host their portfolio on behance.net, an invitation-only portfolio space which aims to keep quality high and maintain a specialised sphere.

Behance.net has great customisation features for uploading text, images, video and comments.

Unfortunately there are banner ads on the right hand side but this is unlikely to bother an emerging artist or photographer getting a free service.

BehancePro site is a new initiative removing the banner ads and providing with a lot of customisation from $11p/m http://prosite.com/

Standout Features

- Very professional minimal design templates
- Very easy to use
- Custom URL for portfolio
- Unlimited hosting
- Unlimited projects
- No programming required

Carbonmade

Commercial
Portfolio focused
Hosted solution
carbonmade.com/

Carbonmade is a service focused on easy-to-use portfolios for photographers, illustrators, graphic designers and artists. The site offers a free 35 image upload option or for $12 (USD) a month you can have 500 images + 10 videos. Carbon made offers very limited customisation, and no templating options. It is a good choice for creating a professional looking designer portfolio where the owner has no web development skills.

Standout Features

- Quick and easy portfolio solution
- Hosted solution
- Fee or $12 per month
- Images swf and video
- No template options
- No setup
- No learning curve
- Not able to extend - is a locked system

Figdig

Open Source
Portfolio focused
Hosted solution
figdig.com/

Those with a portfolio to get online will find the free service *Figdig* provides invaluable for promoting their work, even though there are ads on the portfolio pages.

The layout lacks customisation however it does provide the opportunity for viewers to comment on the work they see.

This solution makes it easily to be able to upload multi-page documents and magazines with custom thumbnails.

Standout Features

- No customisation
- Strict template format production
- It is possible to upload high resolution photos
- Has a good or bad limitation to the default orange and grey colour scheme

Selecting your CMS

It is important to start the selection process with a clearly defined set of requirements to know the limits of what the CMS is required to do. Otherwise, it is easy to be sidetracked by sophisticated functions that are not necessary to the purpose of the web site. How then do you establish your list of requirements? The following is a checklist of ten things that are particularly important (Boag, 2010, p. 156). when selecting your CMS.

1. Core functionality
2. Ease of editing
3. Asset management
4. Search function
5. Aesthetic and functional customisation
6. User interaction
7. Roles and permissions
8. Versioning
9. Multiple-site support
10. Multilingual support

These attributes are worthy of considered reflection and whilst no absence of a single item in the checklist would exclude the CMS of consideration are are a good starting list of significant factors.

> CMS's fall into several broad categories. Although individual systems vary within these categories, each has certain general characteristics. Matching the established requirements of the web site to one of these broad categories, will help to reduce the list of suitable options considerably.

The categories are as follows:

Partial CMS's

Most web pages rarely change. Only specific sections need updating on a regular basis. There is no reason an entire web site needs to be built using a CMS (Boag, 2010). It is possible to combine a desktop application with partial systems to deal with specific functionality. Excellent tools are available for managing news, events, communities, and search.

Blogs

Blogs are the most common form of CMS. Their core functionality is to publish a series of pages sorted by category and date. This is ideal for managing regularly updated news. Recently, blogs have become considerably more powerful, letting you order pages into hierarchical structures.

Enterprise-level or Commercial CMSs

Enterprise-level CMSs are comprehensive solutions that offer the complete suite of functionality from versioning to multilingual support. They often carry a hefty price tag, although open source solutions are also available (Boag, 2010). Most of these CMS's meet my criteria, but the quality of implementation can vary tremendously. It is necessary to look at the options and compare ease of use, power, and flexibility. These full-blown CMS's are ideal for organizations that require a wide range of functionality and have regularly changing content produced by a large numbers of contributors.

In March of 2009 there was a competition between the interested parties of *Drupal*, *Joomla!* and *Wordpress* to build a web site with a set of specifications.

The web site 'CMS SHOWDOWN' claims: "You are the judge in an unprecedented apples-to-apples comparison between the leading open source content management platforms". (http://www.cmsshowdown.com 2009) Each team was made up of specialist developers, all developing the same web site. The web site was to be a community leadership program platform. The goal of the competition was to provide viewers of the competition with a 'decision-making tool' to use when they choose which of the three top open source CMS solutions might be suitable for their project. Up to 90 hrs was spent developing the web sites and the judges declared that there was no winner – they were all equal and winners! Renowned designer Mark Boulton said...

"It does indeed look like all the teams were not limited by their choice of CMS in realising the design..." (Showdown, 2009).

The competetion demonstrated that each CMS was equally flexible and capable when in the hands on an expert with significant experience. This is really a demonstration of product maturity and not the sutability of each CMS to new learners.

RESULTS AND DISCUSSION

In this section:

Why learn a CMS?	63
The teaching of web design	64
Research design	64
WhichCMS online training	66
Are the outcomes of schools meeting the needs of employers?	69
What are the views of employers regarding the knowledge and skills of web design positions?	70
Interview summaries with local industry	71
CMS Questions	73

03 RESULTS AND DISCUSSION

This section will examine the information that has been gathered, with a view to addressing the questions that initially prompted this research (ch.1). The data generated was derived from student feedback provided during the initial workshop and subsequent longitudinal surveys. Discussion with industry representatives added a professional perspective on what is required in tertiary design courses to prepare students for their professional life.

WHY LEARN A CMS?

Based on the research presented and my own experience as a teacher, students are taught to build static web site design as part of most undergraduate web courses. If educational institutions were to teach design students dynamic development via a CMS, they would be more likely to improve their coding skills, which would greatly enhance their employment options.

> Web students will benefit from being taught open source CMS's, because it will encourage them to investigate programming further, and the instant success they can achieve with open source will encourage them to see themselves as a developer, not just a designer who hands the development process over to someone else.

In the current market, there are many opportunities to create web content for small clients who don't have the money to pay a team of developers. It is the way of the future and tertiary institutions training the designers of tomorrow should consider making it a requirement that their students graduate with skills that are up to date in a rapidly changing environment – skills that will put them on an equal playing field with their competitors.

Today's students are already the main users of web 2.0 technologies. Web design and development students should be introduced early in their education to existing and immerging CMS solutions if they are to be properly engaged in web site development. This could be done through integration to their existing course or a CMS unit taught as a complete separate unit.

The nature of this discipline is practical. Learning a CMS will give students the skills that are most sought after by business. In order to be of maximum benefit, the delivery of such a course would focus on practical rather than theory based learning. This ensures that the outcome for students is a skill set they can take into industry. A CMS course therefore has the potential to turn student designers into independent learners and 'on-the-job problem solvers. Employees that can keep their own skill set current with trends in web development are more valuable to employers than those who require input from outside to do so.

> It would be entirely possible to introduce a CMS module into the current curriculum of most web courses at tertiary institutions.

An extended internship in the industry would make the introduction of this material more meaningful to students because they would have a chance to see CMS problem solving in action, and to learn it from industry professionals (*Wordpress*, 2010).

The teaching of web design

There are many different structures available to those teaching web, in terms of the types of knowledge and practical skills that are emphasised. Tertiary educators must make choices. It is simply not possible to canvas all the technical possibilities within a three or four year course. What can be agreed upon, surely, are the more holistic educational outcomes, that will determine the kinds of thinkers we are producing. Firstly, the concept of change as a given must be normalised for our students.

> No institution should be graduating students to industry with the expectation of a static environment, or that a stable skill set will suffice.

As long as our students are entering the professional realm with a passion for continuing research and self-education, perhaps we can comfortably abandon the idea that undergraduate tertiary can do it all. For this reason it is necessary that teachers in this field abandon the notion of themselves as being able to define the parameters of knowledge for their students, and embrace the idea that they act as guides and mentors, helping their students to navigate and make their own choices.

Research design

To gain an insight into whether students are interested in CMS as a study area I proposed and undertook a 1 day (6 hour) workshop. I selected six CMS solutions based on their complexity and accessibility. Participants completed a series of surveys during the workshop and over a 3 month period. The surveys were designed to establish attitudes to learning dynamic web sites and if the engagement of the CMS solutions altered their directions in learning web technologies. The workshop consisted of six video tutorials (screencasts) which were designed to ensure that all students received exactly the same amount of instruction - approximately 45 minutes each. On average the 15 participants completed two CMS options during the time allocated for the workshop. These screencasts were designed to allow students to work at their own pace. The workshops began with a 45 minute introductory lecture, outlining what a CMS is, and providing a general overview of the six options that were available for selection.

A student's skill capability has a direct outcome on their ability to gain employment.

> The CMS options chosen for the workshop were selected to assist the participants in identifying their current level of expertise and aim them towards the skills employers are seeking.

As part of the study, employers were also asked to provide information about what they are looking for in graduate employees. When asked what web design and development skills they were seeking when hiring graduate students, the responses made it clear that employers are seeking the following: CSS, HTML, Javascript, AJAX and PHP in that order. It became clear from what employers said that CMS is an area where graduate knowledge is considered highly valuable, that is not currently taught in the tertiary institutions covered by this study.

Given the recent economic downturn, should web designers/developers specialise or be general in their approach to finding employment? According to Schwabel: "Being a specialist and a generalist simultaneously is the best route to being successful in a good or bad economy" (2009).

> It should be noted that employers do not consider graduating students 'specialists' even from more focused courses. Employers seek the passion that with experience may turn the student into a specialist. Companies need a mix of both in order to be flexible and skilled enough to meed industry demands.

Traditionally a university education is a broad and the student gets a taste of everything. In Bruinsma's view (1998), the ideal situation would be for students to be constructing their own courses by choosing from a range of both general theory-based and specialised skill-based units. He emphasises that general knowledge should not be underrated, because it produces designers with rigorous, academic thinking and problem solving capabilities, which will enable them to tackle complex, conceptual difficulties.

> "The way it is organized, certainly in Europe, design education delivers neither specialists nor generalists. Design academy alumni know the basics of their trade, but they are not seasoned typographers, book or exhibition designers, or Web whiz kids" (Bruinsma, 1998).

Whilst specialised skills are also important in the current environment; "...theory, criticism, and practice should be linked in a more meaningful way than they are now"(Bruinsma, 1998)

The popular web design publishing company *Sitepoint* says:

> "In conclusion we have seen a significant shift in the way designers are educated, education has moved from a generalist to a specialist focus however the student upon entering the workplace is not a specialist".

Sitepoint provides web training directly to students and has proved popular amongst many of the students I have spoken to.

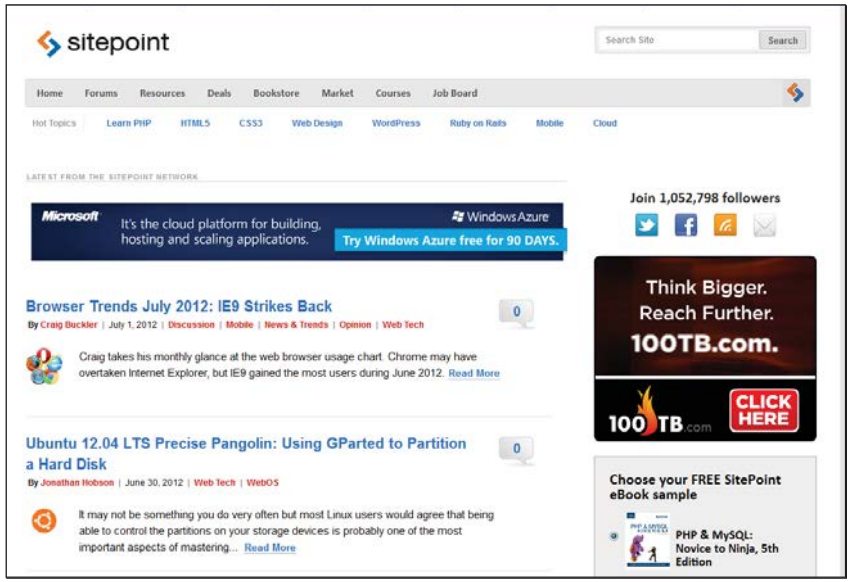


Figure 33: www.sitepoint.com

WhichCMS online training

The practical project component of this thesis is an online Blog with two focus areas.
http://**www.independentdesign.com.au/whichCMS**

1) The blog *WhichCMS?* details my internet research findings, and provides an opportunity to for others to contribute to the discussion. The blog has been created in *Wordpress,* which was one of the CMS options available to students doing the workshop.

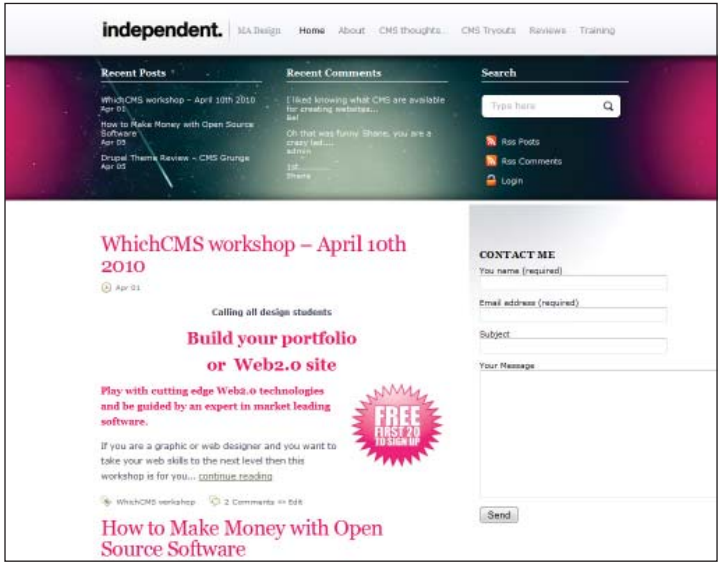


Figure 34: WhichCMS web site - www.independentdesign.com.au/whichCMS

2) The second area of the blog is an online, user-demonstrated CMS video comparison, or 'screencast' of the different solutions. Currently there is a vast array of quantitative data on the Internet in the form of feature comparison at web sites like **http://www.cmsmatrix.com.** The data merely lists the features and many systems appear equal when analysed in this manner. A qualitative analysis from experienced users of their preferred CMS would be on enormous benefit to the community. The screencasts offer viewers a personal account of the CMS in demonstration. Screencasts of Computer Based Training (CBT) have gained a wide acceptance and an alternative method to traditional printed material and provide instant visual feedback.

> Currently it is possible to download or even trial live demonstration versions of blogs but this is reliant on the skill of the user to learn a new system.

This area of the web site provides visitors with the opportunity to set up a CMS of their choosing and creating a web site. In order to match the various features and design options of the different CMS's, the screencasts look and function differently. The screen capture process is of course dependent on the creator owning screen capture software. I have provided a link to a 30 day fully functional trial version of this software on the blog web site.

CMS screencast demonstrations, a growing community

In order to further the reach of the *WhichCMS?* workshop I uploaded the screencast videos to web host, *Vimeo*. *Vimeo* is a free video hosting service for artists, film makers and not -for-profit organisations. *Vimeo* also supports high quality video and does not overlay the video with advertising as *Youtube* does, so it is a suitable platform for a number of reasons. The screencast videos are also connected and displayed within the *WhichCMS?* web site. It is hoped that having the videos online will enable this research to engage with a broader audience. The videos are located at the following web address:

http://vimeo.com/user1404741/videos and **http://www.independentdesign.com.au/whichcms**

Both online offerings have the provision for comments, however at the time of this writing there has been very little feedback. This may simply be the ration of users visiting the web site being very low. The web sites usage spiked on days when I sent out request for survey completions with the busiest day being August 9th 2010 with 184 unique views.

Figure 35: Web Traffic to the WhichCMS web site

What are the preferred technologies based on student feedback from the WhichCMS workshop ?

The *WhichCMS?* workshop offered students a choice from a wide variety of CMS options to investigate. Software was specifically ranked according to 1) the Level of complexity and 2) the Learning Curve. The scope of the tools selected was deliberately broad as to allow students easy differentiation.

The students seemed interested in acquiring a functional CMS as soon as possible and selected the system they though would offer that outcome. It is apparent that students were not being taught CMS in their current schools although some said that they had had experience in one or two, primarily *Wordpress* and *Tumblr* because of web design magazines. The students enjoyed being exposed to new CMS options and some later contacted me to tell me that they had built client web sites with their new skills, which was pleasing. PHP and open source has many limitations not necessarily understood by students when learning PHP. The selected CMS options are in keeping with industry recommendations and usage.

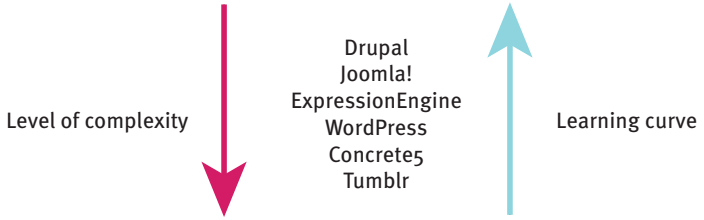


Figure 36: Software rank order

Each CMS screencast has its own comments section allowing viewers and other developers to comment, providing an ongoing conversation. I have provided the first 3 screencasts and will advertise the web site for audience participation. There are of course significant issues of neutrality and bias on behalf of the presenters and I will endeavour to create a framework to minimise these.

In order to host the videos I am using a free online video streaming service as dedicated options are very expensive and outside the scope of this thesis. There are a number of options available and these have been documented on my blog web site: **http://www.independentdesign.com.au/whichcms.** The choice of VImeo means that registered members are able to upload screencasts and e-mail my blog the link. I simply add the link to a blog posting on my web site and the screencast is displayed within my web site ready for comment. There are already screencast demonstrations hosted at web sites like **http://www.demodemo** and **http://www.wolfdemo.com** however these web site restrict their demonstration to *Drupal* and *Joomla!* exclusively. This thesis is interested in broadening the debate to include all open source systems, not just the most popular ones.

Figure 37 is a screen snapshot from the online host of the WhichCMS video training.

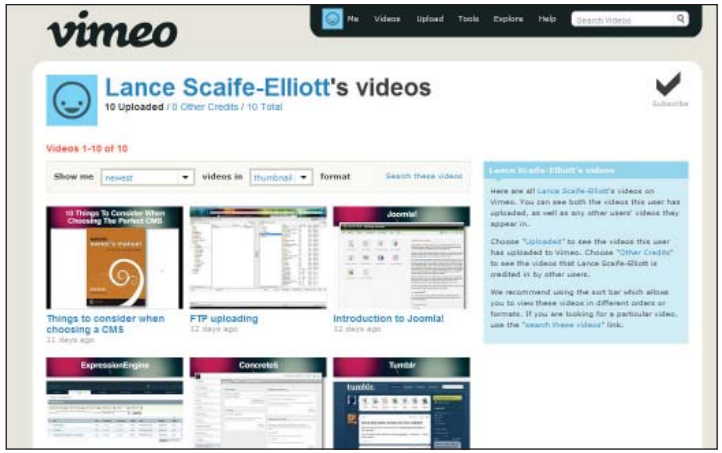


Figure 37: Online hosted workshop videos at:

http://www.independentdesign.com.au/masters/whichcms

Are the outcomes of schools meeting the needs of employers?

This research is a question extracted from the *WhichCMS?* workshop survey.

Do you feel that educational institutions are teaching skills which match the demands of employers?"

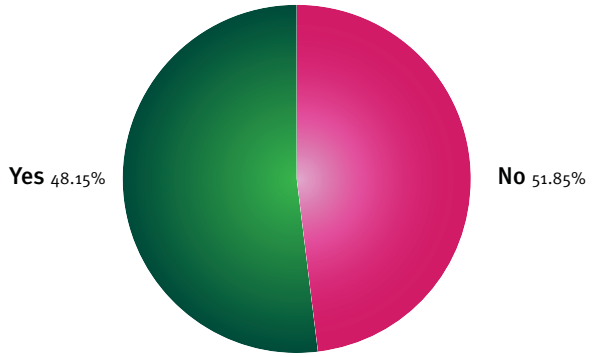


Figure 38: Survey question "Do you feel that educational institutions are teaching the skills to match the demands of employers?"

In the extended version of this graph the responses included:

Yes

- Most of the time
- Mostly
- Some, I am also a part time teacher in the TAFE system, and I feel the standard is quite good, however could do with some higher level programming, providing the resources.
- The internet changes faster than institutions can adapt, but for the most part the gap in skills is the student's responsibility.
- Yes and no. It would be better if students understood the practicalities of design and development in a business context.

No

- Completing projects to deadline is an ongoing concern
- Need real practical experience
- They do not have any sense of SEO! most importantly, we do not need beautiful design but enquiry creating web design / development ! think about it
- They don't teach CMS
- Not sure about programmers, not in multimedia/creative's
- Too much emphasis on tools instead of the underlying technology
- They only touch the surface

What are the views of employers regarding the knowledge and skills of web design positions?

To answer this question I asked employers:

What are web design and development students being taught in educational institutions?

Their responses were:

- Learn to use version control software.
- All web design/development courses are pretty basic (even the ones called advanced). They think OOP is useless, never heard about SQL injections etc. etc.
- They're being taught tools, not skills. Teaching students to problem solve, research, handle client and communicate are just as important as anything else.
- Students could be better prepared for the commercial world by understanding their role as a team member delivering a service to meet client needs -not their own.
- Should there be a CMS knowledge base?
- They should've been taught other skills related to being a designer/coder/programmer, other than just design skills. For example, time management, handling clients, self discipline, integrity, etc.
- We have had some wonderful design grads and undergrads over the years. Coders have been more of a disappointment. For example teaching flex before one attains high level as3 is, in my opinion, short-cutting vital skills in favour of learning a layout framework. Too much focus on software and not enough on core skills and problem solving.
- Technology is always changing. Teaching HTML/CSS/Javascript/AJAX is a minimum then they learn the rest on the job.
- Developers have a mismatch on what they thing their skills are and what they actually are. They expect way too much money and are rapidly driving development off shore. There will be fewer and few project completed using local developers, particularly back-end. 2. Develops often do not have enough business acumen to marry business requirements with technical deliver requirements. 3. Core skills are often weak. They have had insufficient practice and have not honed their skills.
- I think that marketing or user experience/information architecture needs to be taught along with the standard course modules.
- They should ALL be encouraged to do internships in real working environments, solving real problems, learning what it's really like servicing clients. If they are freelancing themselves out directly to end clients, they must be taught why NOT to underprice -as it reduces the value of the skill set a properly structured business MUST charge to stay sustainable. So if they work out their contractor rate to an agency is $30ph, any direct client's MINIMUM rate should really be $120ph. Obviously, the rates are always linked to their knowledge and how much they can output in that hour.

When asked: **For commercial projects do you use open source software and why?** Employers overwhelmingly said Yes with 78% implementing open source. Value, ease of use and stability were chief attributes mentioned. It is worthwhile mentioning that over 54% of respondents said that they used their own Custom made in-house CMS.

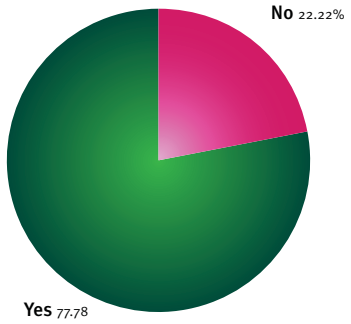


Figure 39: Survey question "Do you feel that educational institutions are teaching the skills to match the demands of employers?"

Interview summaries with local industry

In June of 2010 I conducted interviews with two leading web design/development studios based in Sydney's Surry Hills. The selected studios were *Eskimo Digital* - digitaleskimo.net/ and Mass Media www.massmedia.com.au/

> The studios were selected because of their different web site development strategies.

Eskimo Digital develops most of their web sites with open source software for mid-sized clients, primary on the PHP platform. *Mass Media Studios* develops their web sites for large clients primarily on the .NET platform. Both organisations employ their own programmers to provide additional support and programming capability to the software they employ. Their approaches to development are very similar and each company separates front-end and back-end design, allowing the software solution to be customised.

Jeremy Epstein — PHP Developer, Eskimo Digital

Jeremy is a Computer Science graduate from the University of Technology, Sydney who stumbled into web development at the end of his degree. *Drupal* was Jeremy's first CMS as it was one of the earliest and the largest open source CMS options at the time.

Eskimo Digital is an environmentally sensitive company, which strives to create an ethos of community, support and development. Many of the company's clients are not-for-profit so the open source environment fits into their budget with limited development costs.

Jeremy has contributed code back to the community and feels that this has assisted him in becoming a better programmer, particularly having to adhere to the standards set by the development team. Aware of the shifting marketplace and his own dependency on PHP as a platform Jeremy has recently been learning *Python* and *Djagio*. He recommends *Wordpress* as a CMS which is suitable to get started with, because of it's popularity, learning resources and capability.

"I have never had any formal training in PHP and pretty soon after stumbled into Drupal. Before I knew it I was a Drupal developer." - **Jeremy Epstein**

Adam Quirk — CEO, *Mass Media Studios*

Adam is a design graduate from the University of Newcastle where he also lectured there for a number of years so he is very passionate about teaching students the appropriate tools for business.

As a business *Mass Media* supports open source mainly through the implementation of *Wordpress* for client blogs. *Mass Media* finds PHP difficult to support because the platform can change and update at any time without their knowledge, thus affecting a client's site. Their preferred platform is the commercially supported Microsoft .NET and they have built their own custom CMS which runs all their client web sites.

Adam is cautiously optimistic about developments in the open source realm, but generally not keen to recommend this route to his clients, some of whom are attracted by the economic benefits of using a 'free' CMS. He concedes that there is a good range of modules and plug-ins, and that it can facilitate building and implementation of sites quickly and cheaply. Experience has shown him, however, that often, at a later stage, clients with open source based sites will request a feature that is not supported or which is implemented poorly in the software they have chosen. According to Adam, this usually necessitates the expense of new development which the client was trying to avoid. He feels that it is easier to go with an integrated, bug-proof system in the first place, and sometimes, ultimately cheaper also.

Adam feels that open source is not a real money saver in the long run as eventually additional development is usually required. He worries that resting the success of the company's work on a changing platform like PHP might be too risky. If client sites break, or even experience small scale problems on a regular basis, it could prove financially disastrous for his company.

"If the environment you host on (PHP) gets updated...then we have things drop out and fall over ...that just doesn't happen with Microsoft." - **Adam Quirk**

CMS Questions

From the 28 CMS's researched in the literature review I selected six to study which were included in the *WhichCMS?* workshop and my research methodology. Based on the feedback from the research and workshop, I developed a six questions that the novice or student might ask themselves before selecting a CMS.

Drupal

A excellent CMS for developer who already has some CMS experience.

Do you?

1. Feel happy to jump into PHP to make modifications?
2. Have a large web site that is expecting a lot of traffic?
3. Experienced with other CMS before?
4. Want to search out solutions when development problems arise?
5. Consider yourself a developer?
6. Are you keen to develop your own plug-ins?

Joomla!

A very good CMS with plenty of documentation, The templates are relatively easy to install in order to get it up and running.

Do you?

1. Seek an e-commerce focused CMS?
2. Have a community project and seek a ready to roll out of the box solution?...
3. Have a basic idea of what you seek but desire a safety net of available plug-ins?
4. Like a large range of templates?
5. Seek only to configure a web site?
6. Seek a market leader?

ExpressionEngine

A very powerful CMS where design and customisation over content is paramount.

Do you?

1. Want granular control over every element?
2. Desire to improve your XHTML skills?
3. Like the idea of a complete solution?
4. Have a spare $200 for the commercial version?
5. Have patience to develop a web site from the ground up?
6. Want a separate front and back-end?

Wordpress

The world's most popular blogging platform can be 'bent' to behave as a CMS.

Do you?

1. Have no experience in a CMS and just want to get posting?
2. Like plug-and-play module/CMS configuration?
3. Have limited time and seek and instant solution?
4. Mind 'bending' the blogging aspect to make it a complete CMS?
5. Have and experience with PHP?
6. See blogging as your focus?

Concrete5

A fantastic CMS for font-end client editing. A new and very promising solution.

Do you?

1. Seek an front-end editable CMS?
2. Have an HTML template to begin with?
3. Like the idea of purchasing commercially supported plug-ins?
4. Like an elegant user interface?
5. See yourself using the next generation on CMSs?
6. Want to avoid PHP?

Tumblr

A great 'micro blogging' tool, easy to set up and start posting.

Do you?

1. Enjoy social networks?
2. Enjoy contributing to a large creative community?
3. Commenting on other peoples posts?
4. Spend all your time surfing for web inspiration?
5. Seek a limited functionality social networking web site?
6. Want a hosted free solution?

When students were asked: **Would you be encouraged to improve your coding skills if you learned this software completely? Participants responded:**

1. I'm using EE at the moment for my web site and learned a lot of coding through this.
2. The same for WP for my Portfolio
3. Definitely
4. I'm happy to learn more, because I'm the person keen to learn more new system.
5. it does. I am not completely satisfied with the limitation of any software, and would add additional scripting to make the best of comcrete5 framework.
6. its like learning another language
7. Probably 111.11% to help give more functionality.
8. Yes
9. yes for greater control and creativity

At the conclusion of the workshop participants were asked: **Did the workshop give you a good introduction content management systems (CMS)?**

1. Also into open source options
2. Good understanding
3. Met all my expectations, even more
4. Quite a lot of information
5. Very good
6. Yes
7. Yes and i will use it in the future
8. Yes. I would like to learn and use every time another CMS.

Students were then asked: **Could a CMS influence your desire to code in a web scripting language like HTML, CSS, PHP or Ruby?**

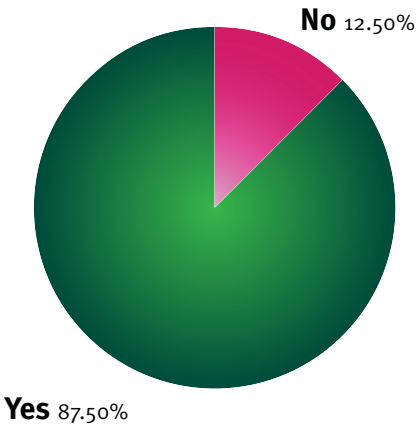


Figure 40: Survey question "Could a CMS influence your desire to code in a web scripting language like HTML, CSS, PHP or Ruby?"

Additionally student were asked: **Will you investigate any of these CMS solutions further?**

- Tumblr 75%
- ExpressionEngine 37%
- Wordpress 100%
- Concrete5 62%
- Drupal 37%
- Joomla! 75%

CONCLUSIONS AND RECOMMENDATIONS

In this section:

Conclusions to this study	76
Recommendations for the education sector	79
A way forward for teaching web design	83
Study paths for designers and teachers	83
A personal approach to CMS selection in education	84
CMS market overview	90
Ten CMS selection recommendations	91

04 CONCLUSIONS AND RECOMMENDATIONS

Conclusions to this study

It has become clear, throughout the course of this study, that choosing and engaging with an open source CMS will be a challenging task for both teacher and tertiary student, but that it is nonetheless a worthy and important educational goal for anyone planning a career in web design and development. It is becoming increasingly clear that in order to survive and thrive as professionals in the current web environment, our students can no longer treat the 'back-end' of web as a mystery - something to be handed over to programming boffins. Indeed, the two terms, design and development should no longer be viewed as separate activities, and we should be encouraging all our students to view themselves as web generators. It has also become clear that choosing and engaging with the tools available in this realm is more about asking the right questions than expecting a 'known quantity' outcome.

1. What is the current situation with regard to the emergence of dynamic web design?

Whilst a majority of design students are interested in producing for the web, most of them see themselves as performing an exclusively aesthetic role in the development of web projects. In the current web environment, and moving forward, this makes them less useful and less employable, because this kind of division of labour only occurs within large companies. Many graduating students will be working freelance for themselves, or in smaller environments that favour those with a broad skill base. The Internet environment is changing rapidly, and the profession of web needs practitioners who are ready to be flexible and embrace that change. It is the finding of this study that web students, with proper instruction in the usage of open source CMS's demonstrate the capacity and the motivation to build a dynamic web site. The research indicates that students would benefit from a practical approach to web development through the introduction of a unit on CMS to their existing curriculum.

> It is the author's belief that many benefits would flow from using open source content management systems to teach students of web design a more integrated and less specialist approach.

The days where creating an image for a web site and then later making decisions on the useability and interface of the web site are gone. More than perhaps any other design medium, it is in the generation of material for the web that function and aesthetics are most closely meshed, with information and format together creating a reading path for the viewer. This study has identified other benefits for students in embracing a knowledge of open source CMS. Since open source software is a collaborative project, it would help students to realise and embrace the benefits of web-based communities in their own professional lives, both as contributors and receivers. An evaluation of the cost of open source versus cathedral-style software will also naturally lead students to consider the political economy of the Internet, and issues of accessibility and fairness in this media environment. Students who have grown up used to thinking of the internet as the scion of information democracy may come to realise in the next few decades that it will be at least partly up to them to try and preserve that democracy.

> Perhaps, most importantly, empowering students to author their own interactive web sites for the web is clearly going to increase their confidence and a saleable skills base.

Once students can see that they are in charge of all aspects of a project, not just the window dressing, they will be excited to realise their projects without needing to wait for a developer, or an employer's vision.

Web design students need to graduate with an understanding of Web 2.0 and the forces that power it. One only has to look to the experience of *Wikipedia*, versus the *Encyclopaedia Britannica* online, to see the difference in success between the new paradigm of community ownership and maintenance, and the old paradigm of intellectual ownership and a proprietary attitude to development.

Open source operates within the new paradigm, where users have the ability to modify and improve what they encounter, feeding back these improvements for public use. As teachers preparing students to be industry it is the educator's responsibility to make sure students understand this new paradigm.

2. What factors influence the selection of CMS solutions for different groups; designers, teachers, employers and students?

Selecting a CMS is a problematic task and not easily undertaken. Students need to be able to identify what their current capabilities are in order to select an appropriate CMS. As this study has shown, there is no one solution that fits the needs of all clients, employers and students. Development history, popularity, ease of learning and capability all play key factors in deciding which CMS to select.

Traditionally open source CMS's have suffered from a lack of training materials. The availability of good training videos and screen-casts will certainly influence a student's choice, and thankfully, more training material is appearing online to cater to an increased demand. My own contribution to this area via the online site and CMS screen-casts will hopefully assist students in making their CMS choice. Longitudinal responses from the research were disappointing with only 25 of 200 surveys completed. 18 lecturers were emailed at the main design/development institutions and only two responded. 15 students undertook the workshop and 5 completed all the survey material in the three months allocated. On the surface it would appear that web sites are moving in the direction of no longer needing expensive custom solutions by professional web developers. However it is clear that there is a split and that as yet, free open source software does not hold all the answers. Many development studios and their clients still prefer a commercially supported product.

The research conducted with trial groups of students shows that most were not intimidated by this new area of knowledge, and had subsequently attempted to use an open source CMS in the three months following their *WhichCMS?* workshop research session.

3. How are current web design teachers engaging in the transition from static to dynamic web design both in practice and teaching?

Whilst design, and web design have become an established part of the tertiary academy, there exists in most institutions which offer degrees in this area, a tension between theory and practice. There is some division of opinion over the extent to which each of these should be emphasised in order to produce a good practitioner. Many courses are already crowded with units of study, and some educators in this field would probably argue that there is not time in a degree course for students to do more than dip a toe in the many practical requirements of this profession. The list of vocational skills which they may be required to produce after study however, in addition to understanding the principles of design, is certainly already long, and getting longer. This means that it is a fact of life for web designers that a lot of learning must take place 'on the job'. It is, however, the finding of this study that there is a need for educational institutions in the Sydney basin to teach CMS development in order to better engage students of web and graphic design in their understanding of scripting for the web. Currently, most web design courses incorporate at least some coding and programming languages, which, at least in my own experience, are often shunned by students that see themselves as designers or artists. Perhaps these courses could be made more 'real' and dynamic for the students, if they incorporated an investigation of the open source realm.

Students with high aesthetic ideals often treat programming courses as peripheral, and engage with them on only a superficial level. If these units were taught in conjunction with CMS development, the students would have a worthwhile, 'real-world' goal to inspire them to get to know their CSS and XHTML. This goal would be the ability to realise their own aesthetic designs as finished web products.

> Content management systems are not currently taught at institutions within the Sydney basin and there is evidence that they are failing to meet the expectations of employers.

In the current environment, employers could well have an expectation that a student will have some experience with CMS 's, and graduate students working as free-lancers are now often required to implement a client managed web site, which means implementing a CMS.

Web design and web development are currently offered in two separate courses at most tertiary institutions. These courses typically exemplify an art approach via design departments and a programming approach via computer science departments. A closer connection between these two departments would significantly benefit both educators and students. Educators need to maintain their skills in an environment of constant change, in order to impart relevant skills to their students. The *WhichCMS?* workshop demonstrates that in a high pressure environment, where demands on students to learn new software are high, software screencasts are an excellent way for students to independently learn at their own pace. As shown in the workshop, teachers can prepare these tools so that students can preview software options and then complete tutorials as a means of augmenting their current skills.

> "Not surprisingly, design may emerge as the quintessential twenty-first century profession, a vital, form-giving, way finding, meaning-making synthesis of art, technology and social science"
>
> (Jensen-Inman, 2009).

4. What are the factors influencing designers in their engagement with scripting or programming?

The research conducted here demonstrates that students are very excited by the range of tools when they understand the capabilities and are easily able to differentiate between various CMS's before undertaking learning and implementation.

Students should start their web development education with XHTML and CSS. After that there are a number of tools available depending on the students own desire and ability to continue coding.

CMS options start with installation and configuration without code and then progress into more complicated and powerful tools as the student gains confidence and ability.

When asked the question in the *WhichCMS?* surveys: **What are the web design/ development skills you are currently seeking when hiring graduate students?** It was clear from the online survey and conducted interviews that employers are seeking: CSS, HTML, Javascript, AJAX and PHP in that order.

> CMS is an area that employers now consider valuable that is currently not taught in educational institutions.

It is my belief that design students with a understanding of open source CMS's will have an increased knowledge of the importance of usability and clarity of reading pathway in a web site, and will therefore be better web designers. Whilst static XHTML will always be the web foundation implementing a CMS can contribute significantly on the owner/ administrators and users of the web site.

> " Design academy alumni know the basics of their trade, but they are not seasoned typographers, book or exhibition designers, or Web whiz kids." (Bruinsma, 1998)

What Bruinsma is saying is that whilst we cannot teach our students everything before they enter the work place, it is our job to turn them into self-educators and give them the tools necessary for them to continue their enquiry unassisted upon graduation.

> The inclusion of a course on open source CMS would offer students a solid foundation for the correct approach to usability, as well as opening the door to further scripting and programming, by reducing the 'fear factor' traditionally associated with these areas for designers.

One big benefit of using these tools to educate is the minimal cost involved.

> Open source offers educational institutions a unique and affordable opportunity to teach students skills which are transferable from the 'bazaar' to the 'cathedral' realm.

My own experience of open source systems is that they encourage experimentation and ingenuity on the part of the user, albeit at the cost of some frustration along the way. In doing so they offer a way of thinking which is invaluable to any designer. Open source CMS technologies are ideal for education and continue to develop and evolve putting pressure on commercial and enterprise CMS solutions.

It should be noted that technologies evaluated in this study may have changed at the time of reading as software improvements are continually evolving. Some CMS solutions develop quicker and others stall. On average commercial software has a major version release upgrade every 18 months. Open source development is not based on the same commercial imperatives and development can be a few individuals or a few thousand. Additional software releases and updates of plug-ins, modules and extensions all contribute to capabilities, brand value and user uptake.

As a result of this study it is hoped that the analysis of CMS's, discussion and recommendations will inform the approaches for both the education and business communities. Web technologies are moving at a rapid rate driven by social media, new mobile technologies with adaptive development, improved browser interpretation and quick adoption of internet languages HTML5 and CSS 3.

Recommendations for the education sector

Matrix web sites like CMS Matrix are really unsuitable when CMS comparisons because they list features and the real insight comes in your own adaptation to your personal requirements. Based on my experience and the feedback from the *WhichCMS?* workshop, it is necessary for students to download and engage with this software, or at least to view a video tutorial - to dip their toe in the water, rather than trying to make sense of the written assessments of others, often more experienced programmers. The review below is intended to guide students in selecting their first suitable CMS. Research evidenced in Figure 41 reveals that there was no pronounced bias towards any particular CMS.

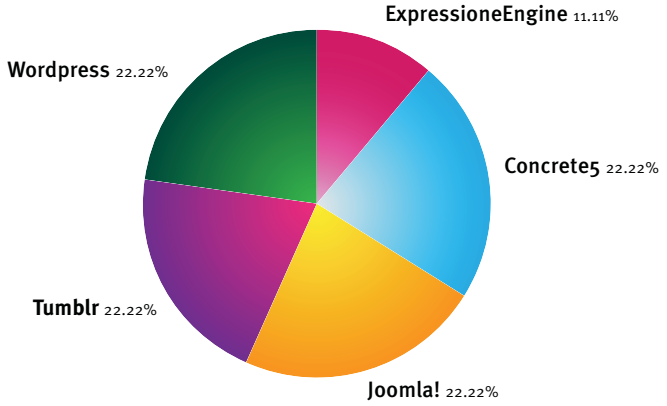
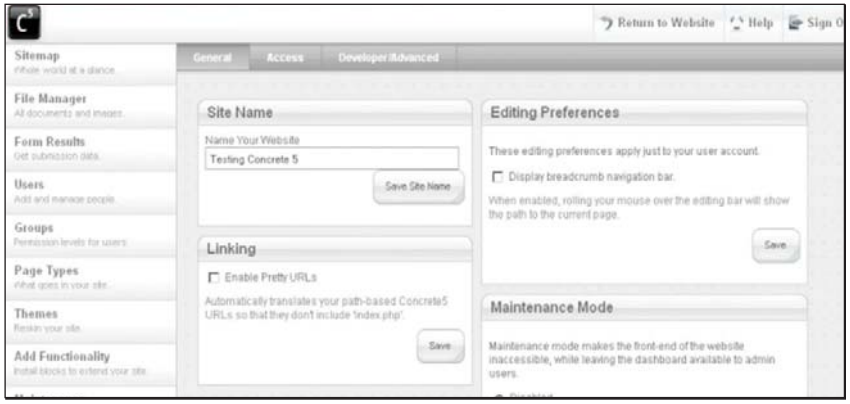


Figure 41: Survey question 'WhichCMS did you first select?'

Participants were presented with a brief 5 minute interface and functionality overviews before selecting their 'First Choice CMS'. It is notable that *Drupal* was not selected by any participant, this may have been because of the setup time required and technical skills needed. Some considerations students were presented to assist them in their selection included; Availability of books, Coding skills required Community, E-commerce, Flexibility, Framework, Learning Curve, Resources, SEO, Social Network, Templates and Tutorials .

The following are insights gained from the workshop, the online surveys and my own understanding after experiencing each CMS.

CONCRETE 5



Concrete 5 is a good choice for someone who only needs a static web site, and who is not interested in programming. Its key feature is in-line editing of pages. For the client, changing content is as easy as clicking on it - a popup window appears for editing.

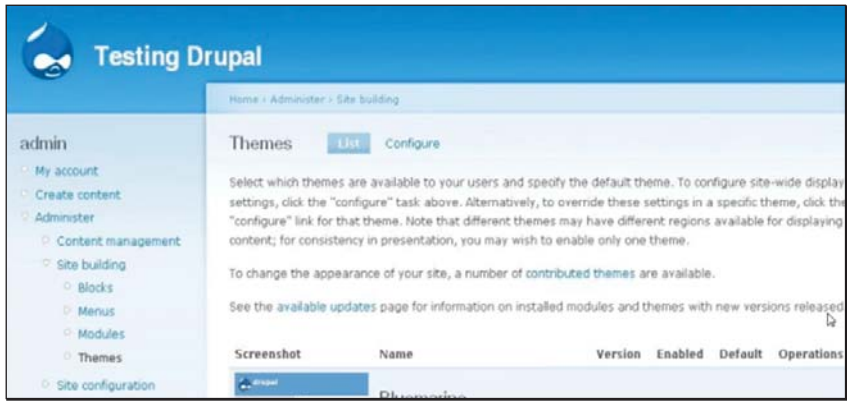
Advantages

- Great for client to edit 'live' pages
- Has versionising
- Extremely simple to use
- Blocks can be moved around easily.
- Themeing is extremely easy

Disadvantages

- Very new, version 7 announced in 2011 with limited plug-in support
- Limited community, 9000 registered members
- Only about 30 additional plug-ins
- No central content tracking

DRUPAL



Drupal is one of the 'big 3' on the open source CMS market (Water & Stone 2012). Generally, it is used to run community web sites. It comes with plenty of modules, and has the capacity to run virtually any type of web site. The recently released *Drupal 7* is a significant update in terms of usability and user uptake.

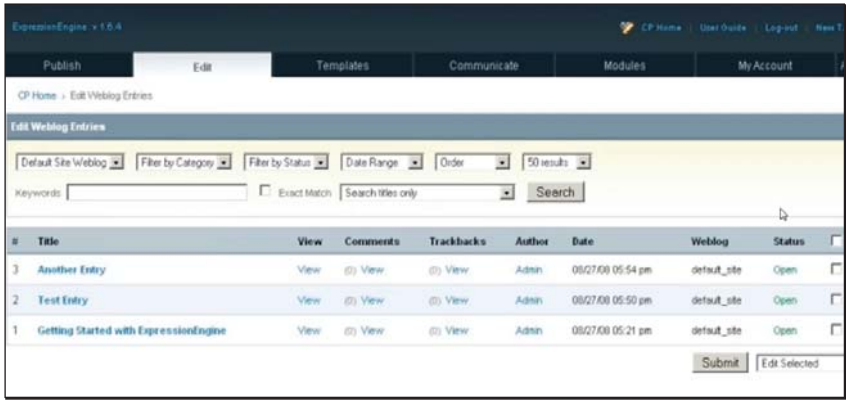
Advantages

- Because *Drupa*l is such a powerful system, it can run virtually any kind of web site.
- Locating modules and getting help is facilitated by a large community.
- It is a fast process to create and launch a well featured web site.
- Files, modules and themes, are easily kept up-to-date.
- The range of modules is very extensive, with literally thousands available.

Disadvantages

- *Drupal* has a relatively steep learning curve in terms of comprehending administrator setup and work flow, especially to the novice CMS user. With a multitude of different names and configurations it can be hard to know exactly where to start. Also, front-end themeing in this software really requires at least some programming knowledge and awareness.

EXPRESSIONENGINE



ExpressionEngine allows the user to publish an unlimited number of information streams using custom fields, and is thus notable for its flexibility. There are three versions available; personal, commercial, and the free core version. *ExpressionEngine* will provide a feature-rich web site that can include both dynamic content and static pages, without significant programming knowledge on the part of the user.

Advantages

- *ExpressionEngine* provides excellent support through web forums, even though the development community is not huge.
- Custom fields allow for an excellent level of control by the designer.
- The way in which the templates work makes building a custom theme relatively easy, again, even without prior programming knowledge.

Disadvantages

- The concertina-style arrangement of the menus, meaning the way in which they are often hidden within other menus, can be confusing.
- When first learning the system of weblogs/templates/categories it can get confusing and seem counter-intuitive.

JOOMLA!

Joomla! is one of the 'big 3' CMS's in the open source market (Water & Stone 2012). It has been around since 2005 and has been downloaded over 30 million times (Grevet, 2012). *Joomla!* web sites manage documents, photos and videos easily.

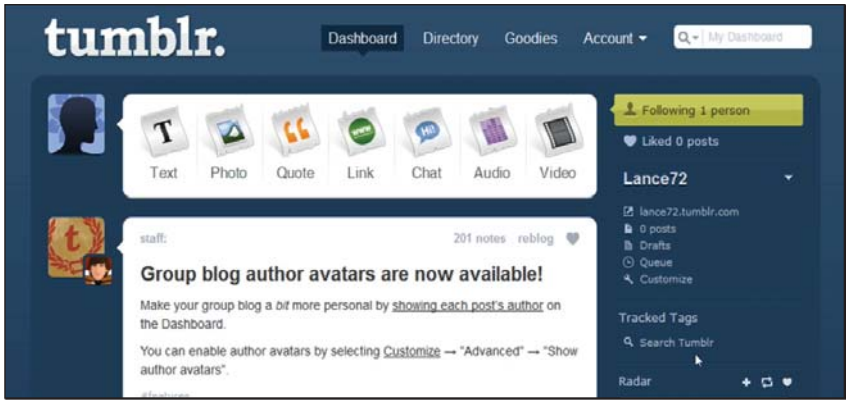
Advantages

- *Joomla!* will suit developers who are looking for a powerful CMS, with plenty of capacity for extension.
- Although it will suite those with some experience, this software is relatively user-friendly, and won't overwhelm the user on login.
- There are many useful third-party extensions available for the core, which is already high on features.
- Large community and online forums to provide support to first time users.

Disadvantages

- Theme development is not as easy as some other systems.
- Finding the right module can be difficult, due to the sheer number that are available.
- Some projects will require commercial modules which are costly for this system.

TUMBLR



Tumblr is a 'mini-blogging' tool, which offers an entry level where no technical ability is necessary. It offers rich media tools such as photos, video support, and feeds. The users Blog appears as a digital scrapbook. *Tumblr* comusers are able to post quickly from mobile platforms, and also enjoy the standard benefits of a blog, like varied page themes and rich-text formatting.

Advantages

- Site is fully hosted
- Ability to make custom themes very easy
- Great community feeling and sharing of information
- Completely free

Disadvantages

- The 'Chat' button can be confusing to first time users - this feature is not for live interaction, but is for formatting pasted text to look like a conversation.
- The capability for leaving comments is not as extensive as on other blogging services. Comments need to be done via the external provider, *Disqus*.

WORDPRESS

Wordpress is the other member of the open source market 'big 3'. (Water & Stone 2012). It started out as a blogging tool, but now has an impressive list of features which make it a fully fledged CMS. Users can apply page management features, upload media and manage all types of content.

Advantages

- Wordpress is ideal for blogging, and for the adventurous, it can go much further.
- Plug and play interface
- Easy operations
- Features include comments, links, media uploads, etc.
- Great for simple web sites; many free plug-ins.

Disadvantages

- This system is not ideal for really big web sites with more complicated kinds of content.
- Can be difficult to adapt to a non-blog CMS
- Here have ranked the workshop CMS options
- Level of complexity & Learning curve

A way forward for teaching web design

Open source software provides really exciting possibilities to the web generator - community developed software with powerful features and capabilities, that is free to use and easily installed. There may be a perception that open source CMS solutions can not be as powerful as software developed within the 'cathedral' realm, however this perception is incorrect. Hopefully, this study has demonstrated that open source software covers a broad range of capabilities effectively and successfully, and also that it allows the student or novice to get in and experiment with back-end development on their own, with no costs attached.

For businesses, one disadvantage of open source CMS's is that they are less structurally organised and bug-proof, so deadlines can blow out, and things can go wrong. For the student, however, this can be seen as a positive, since it is in the process of solving problems that they will learn the most about back-end development.

It is wise to remember that commercial CMS solutions have drawbacks also, notably the cost, which works against smaller businesses. By giving graduates the ability to engage with the open source realm, we are equipping them to occupy a niche in the market - catering to these smaller companies with lower budgets for web development.

Open source CMS, in my experience, requires a commitment of time on the part of the user, to really get to know the tool they are working with. It can be challenging, when it comes to product support, and product stability and security are not always assured. As stated previously, however, it can be of benefit to both teacher and student to be forced to interact with a development community, and to engage in the process of problem solving, rather than have a fool-proof solution straight out of the box.

In Figure 42 we see a graphic illustration demonstrating the options that open source CMSs make available to graphic design students studying web design. Students should be able to engage in programming at a level suitable to their preference and skill.

Study paths for designers and teachers

The introduction of a CMS into course curriculum can be done gradually and with little impact on the current syllabus. The following CMS choices guide is based on the research from my workshop activities. Progression is based on confidence with code and the intended work placement outcomes. CMS options suggested here are limited to those included in the study and the existing student learning path.

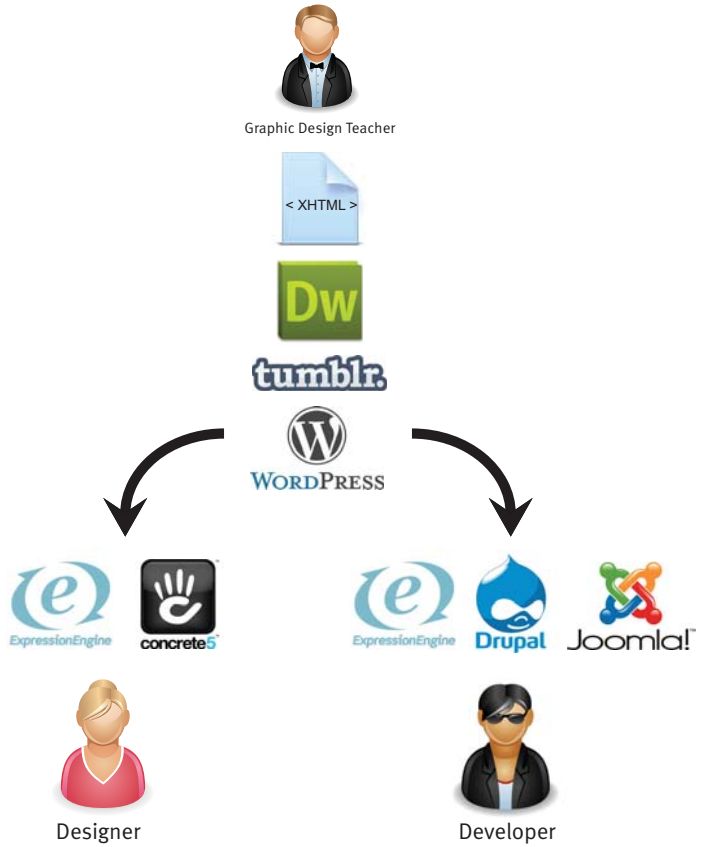


Figure 42 : Educational CMS directions - authors diagram

A PERSONAL APPROACH TO CMS SELECTION IN EDUCATION

The following pages contain fictitious examples of some of stereotypical users based on my literature review, workshop survey research data and my time teaching and working in web and graphic design for the last 16 years.

These scenarios have been derived upon reflection from my interviews, survey research, comments from the attendees at the workshops and my personal experience installing, learning and building web site in these CMS solutions for a variety of clients.

In the hands of an experienced developer most systems are capable of delivering a complete solution, however all have nuances that set them apart from each other.

It is important to note that there is no one solution that suits all situations and developer/ client needs. Some developers will move between a few different CMS solutions depending on the requirements of the project.

*

Steven — Graphic design teacher

Q I'm a 50 year old design teacher and my students want to know about dynamic web sites... What should I be teaching them and what should I get someone else to do?

A Dynamic web sites fall into three main areas and students could be introduced to each in consecutive weeks. As a design teacher you are very capable of introducing the students to the first two. Costs for all 3 areas are free or inexpensive.

1) Hosted CMS solutions occur where the site is hosted and set up on the providers internet space. The task is filling the site with content and choose a predefined theme. *Wordpress.org*, *Tumblr* and *Blogger* are excellent starting points. These solutions offer tools which require no existing web knowledge. There may be limitations in terms of flexibility/modification/expansion. These solutions have a learning curve that would take 30 minutes to briefly cover.

2) Self hosted CMS solutions allow the designer to download the CMS and install it on their web server. Once installed the user customizes and controls the CMS themselves, adding and changing functionality. Initially easy to install, problems may arise if your server configuration is different to the expected server environment.

If you have an error the only assistance is an online forum that may or may not be answered. *Wordpress.org*, *Drupal*, *Joomla!* and *ExpressionEngine* are very suitable here.

3) Complete code CMS solutions requires a student to have a good grasp of the basic web languages (or quickly develop one), XHTML, CSS, PHP or Ruby or .ASP and Javascript.

Students start out in *Dreamweaver* or similar software and learn the code as the software writes XHTML and CSS. The students then move towards more serious web development as they might learn a variety of scripting environments such as PHP, Ruby, .ASP, Java and MYSQL. These provide a great foundation however other languages are certainly possible. Code can be implemented into a self hosted CMS or a custom CMS can be written from scratch.

A specialist web programming teacher would be needed to assist the students in their endeavors. This direction could easily take 6 months+ to master before the student felt confident to work independently.

* Illustrations courtesy of www.psdblast.com

Sue — Web development teacher

Q In teaching PHP should we build our own CMS from scratch or seek an Open Source solution?

A PHP is the powerful, behind-the-scenes scripting language that allows the building of dynamics sites quickly. The traditional way PHP is taught is line by line, code snippet by code snippet starting with the classic "hello world" page. This can often be dry and detached activity for the students,one which yields limited results, particularly amongst those students with a more artistic focus. Some students have difficulty with the abstract nature of writing out code, and by the time a project has been worked through the students has lost interest.

The introduction of a CMS could help students grasp and contextualise the 'complete picture' and understand how PHP is just one component of a web site.

As web development has matured, sites have become larger and a greater need for efficiency has arisen. There is a school of thought that PHP should be taught using Object Orientated Programming (OOP) methods and Best Practices before doing any actual coding This idea whilst sound in principle, goes against the way every PHP programming course and text book is currently written. Common sense dictates that the students will be happiest if they can actually write some code and get to see it take shape at the front end, so that the results become their feedback. PHP offers the opportunity to quickly jump in and start coding an application as it is one of the easiest scripting applications to learn.

A CMS provides the student with a coding community to interact with, and help answer questions. In the case of PHP, this online community numbers in the thousands, providing a fantastic support network beyond the classroom. The student could begin by examining PHP plug-ins to popular open source CMS solutions, modifying and experimenting with their own desired goals.

Learning PHP within the context of a CMS would expand the students capabilities, leveraging off the existing platform and keeping their interest in the development scene.

Lara — Print designer

Q The web all seems too technical for me, all I want to do is display my portfolio. I am not really interested in learning how to make a web page. Is there an easy way?

A Yes it is certainly possible to have an online presence without any coding or programming knowledge. You could host sample images of your work on a blog site like *Tumblr* or *Wordpress.com*, however, there are a number of other sites which are dedicated to portfolio hosting.

Behance.net offers a complete service for uploading images, video, text and audio - and best of all, its free! A new Australian based service, *theloop.com.au* also provides online portfolio opportunities connecting you with local industry. Be aware that the downside of these services is that it is very easy for a prospective client to jump to another designer's portfolio.

Many artists host their work on a community site like *deviantart.com*. This service allows artists to post their work for free, in effect creating an online blog gallery, which requires no technical expertise. As an alternative to a dedicated portfolio site or a community based art site you might consider a free web site service such as *wix.com*. It offers free site hosting and setup. You are limited in terms of site design to their prescribed templates, and it is worth noting that *Wix* used *Flash* as its display technology to make an appealing professional site. The software uses a drop and drag interface and has good searching capabilities for your content. *Wix* has commercial sign-up options for more features starting from as low as $5 per month.

Another portfolio dedicated option is *The Secretary* an open source CMS made specifically for designers and artists.

Finally you may actually have built in free web site construction with your internet host. Many hosts operate software called *cPanel* and within this area various web site wizards enable you to setup a dynamic site with limited or no technical knowledge. Contact your hosting company for more information.

Mario — Web designer

Q The company I work for makes big web sites and has a team of developers who know 5 or 6 web languages. I am a front end designer making the designs in *PhotoShop* and laying out the initial page in XHMTL/ CSS. Is there a CMS for me?

A There is a multitude of options available for you - the only difficult part will be choosing one. One question to consider is how much code you want to learn. There are 3 directions you might consider. Scripting, template restyle and programming language specific.

The first is a scripting CMS where you use predefined 'scripts' to generate functionality. The scripts come from a manual. They are very easy to implement and 'human readable'. These scripts work in tandem with XHTML and CSS on top of the CMS programming platform such as PHP or Ruby. *ExpressionEngine* and *Movable Type* are good examples of this type of script based CMSs.

A late entry in the CMS market are predefined sites that allow you in re-skin but provide a strict guidelines in order not to break their CMS structure. *Tumblr* and *Twitter* are good examples of such CMS sites.

The second option is re-skinning a template and activating various plug-ins for extra functionality. You install the CMS and activate plug-ins with the tick of a box. To change the look of the template, you can choose one that has as similar layout and then change the CSS of the site to customise the look to your clients brief. CMS's in this category include *Wordpress*, *Joomla!* and *Drupal*. All three solutions have an extensive range of plug-ins so the real task is selecting the right plug-in to suit your needs. Problems occur when there is no plug-in to fit with your requirements, or you are unhappy with the plug-in in terms of functionality or design layout.

The final option is to jump right in and learn some PHP or Ruby coding. You might be pleasantly surprised at how quickly you can learn a script based programming language and there is video based training instruction, which is generally better suited to visual learners such as designers.

Ann — Web development student

Q I am half way through my web course. Is there a CMS I should focus on in order to gain better employment opportunities?

A Any CMS knowledge is regarded well by employers however they are likely to be already using *Wordpress* for many of their clients blog sites so it is a good starting point. Employers are currently highly interested in *Facebook* applications and custom *Twitter* pages as it is the fashion at present.

Your future employer may go down one of several different directions in coding development - Java, PHP, .net or Ruby are all common choices.

PHP is the easiest web programming environment and it also has active CMS communities in *Wordpress*, *Joomla!* and *Drupal*.

Ultimately your future employer may be using their own custom CMS, an open source CMS, a commercial or enterprise level CMS. There are literally thousands of CMS options out there and these systems can cost up to $100,000. It is worth bearing in mind that the development principles are the same for an open source CMS as a commercial CMS and employers are aware of this.

Fundamental to your success in CMS development is a good understanding of semantically correct code and object orientated coding practices. Employers are seeking this focus in graduate students as we see the slow integration of computer science into web development.

Julie — Design degree student

Q I love creating custom designs and hate the template graphics provided by CMS solutions. Is there an easy way I can make a site totally original and mine without relinquishing control to a developer?

A As a graphic design student you should have had an introduction to web development though XHTML, CSS and probably some *Adobe Dreamweaver*. Whilst it is not critical to be able to write code from scratch it is important to be able to read the markup of a page, so when you are creating pages in *Dreamweaver* always work in split screen mode so that you can get both code and design view.

The next step would be looking at a custom CMS like *ExpressionEngine* or *Moveable Type*. These solutions allow you to paste in your exact XHTML and then you add predefined scripts in order to acquire more functionality. These solutions run on the PHP and Perl platforms respectively but you don't need to worry about that. All the additional code is defined in the software manuals.

Another even simpler direction might be to use a CMS like *Perch*, *Concrete 5*, *Textpattern*, *Pixie*, *Get Simple* or *CMS Made Simple*. These solutions admittedly have limited functionality and in effect offer only a static web site with the ability for a client to update. On the positive side, however, these solutions are free and make a good starting point.

There are commercial offerings also, favoured by graphic designers for their ease of use and simplicity. For simple site creation I'd suggest *CushyCMS*, *LightCMS*, *SquareSpace* and *GoodBarry* amongst others. They offer you a fully featured CMS that is branded with your design and company logo.

Sal — Diploma design student

Q I am new to all this web stuff, do I have to learn coding to make a web site? Can I just use a front end program like *Dreamweaver* that writes all the code for me?

A There is certainly no problem with using a WYSIWYG (What You See Is What You Get) program like *Adobe Dreamweaver*. Historically, WYSIWYG's do not write clean efficient code, often using depreciated HTML tags. This is no longer true with Adobe Dreamweaver so it is a very good starting point.

It is important to be able to read the XHTML written by *Adobe Dreamweaver,* so work in split screen code/design view and you will be up and running in no time. Web development purists would prefer that you build you page by hand coding the entire site and this way you would certainly understand the code better and quicker. Adobe Dreamweaver offers 'rapid development' - a way of quickly building a site, whereas hand coding is certainly a slower process.

You may also have heard of Microsoft *FontPage* and *Adobe Page mill* or *Adobe GoLive*. These are older WYSIWYG editors and *Microsoft FrontPage* is particularly renowned for writing poor quality code, so watch out for this.

Another option to consider is called *RapidWeaver* which operates only on the Mac platform. Many applications now support HTML markup including *Microsoft Word* and Apple's *Pages* however these are not an integrated solution so I'd recommend staying with Adobe's *DreamWeaver*.

Having built your site in *Adobe Dreamweaver* you then make areas of it editable in *Adobe Contribute* which operates as your CMS. The software is designed to integrate and make *Adobe DreamWeaver* pages dynamic. The client then edits predefined areas of the site live on the internet in *Adobe Contribute*. This is a good cheap solution for clients with brochure type sites and it is very simple. This approach comes with some negatives too with reduced automated functionality and other advantages of a fully featured CMS.

Sarah — High school student

Q I like talking to my friends online, social networks are so cool. Are there other options for me to creatively have a presence online?

A Sarah, social networks are cool and a great introduction to the web and getting your voice out there. You are probably already aware of the main online social networking sites like *Facebook*, *Flickr*, *Twitter* and *MySpace*. These online communities allow you some control over how your site looks and the types of content you can display.

Probably the first choice might be to register with *Tumblr* or *Pintrest*. These sites allow sharing or 'tagging' of content and are totally community focused. Design theme can be customised, however the site structure is strictly controlled. These types of CMS solutions fall somewhere between a traditional blog and *Twitter* in terms of their capability and content direction. They can be called a 'micro-blog' and are very popular at the moment.

I suggest you look into making your own blog with a hosted service as it will allow you even further creativity and expression.

A service like *Wordpress.com*, *DrupalGradens.com* or Google's *Blogger.com* will allow you to create a site in a few minutes for free. These are 'hosted' solutions meaning these is no install or technical setup for you to do, and all content is located on the host's server. All interaction is conducted via your web browser. Whilst initially free, some of these providers offer advanced features for a small monthly fee.

If you are seeking total control over your site you could of course host one with your own internet service provider (ISP). There are thousands of CMS options out there, however, a few are specifically designed for social collaboration and interaction. *Wordpress* (used with *BuddyPress*), *Drupal*, *Joomla!*, *Dolphin 7*, *Pligg* and *Social Web CMS* are all good tools to investigate.

Alex — Postgraduate student

Q I just finished a travel degree and hope to move into tourism management. What would be the best CMS for me to learn to impress employers at a job interview?

A Without question you should look into *Wordpress* at *Wordpress.com*. It is a hosted solution, meaning they organise most of the technical requirements, and all you have to do is log-in and configure various settings.

The process for learning *Wordpress* is very easy and the site has linked video tutorials hosted on *Youtube.com* in addition there are numerous guides available at your local book shop.

A lot of businesses these days are implementing daily news and hosting blogs to promote their organisation. *Wordpress* is the most popular blogging platform and there is a good chance your employer is already implementing it in their communications.

Learning a CMS is a good idea as it is a transferable skill and whilst almost every CMS has a different 'back-end', the principles being implemented, such as tagging, content generation, uploading and editing, are pretty much the same between the different CMS's. This means that adapting to a new system should not be too difficult.

Depending on the type of business you are hoping to enter, you could create a weekly updated blog to demonstrate your passion and dedication to the industry, that would really impress employers!

Figure 43 provides a triangulation of the research results with a venn diagram. It is intended to help students find their ideal CMS, from those featured in the *WhichCMS?* Workshop. As the illustration shows each CMS occupies a unique place in the sector and each has its own particular advantages and disadvantages. Any individual user must decide which of the four considerations *(Ease of Use, Functionality, Dependability, Cost)* is most important.

This study recognises that the 'ideal' CMS will be different for every student. For a developer, *Functionality* might take precedence over *Ease of use*. A designer might have *Cost* as their primary consideration. Holistically there are many complex and varied decisions to be taken into account when selecting a CMS. One of the research conclusions of the *WhichCMS?* Workshop is that there is no ideal CMS suited to everyone as all participants have different abilities and their clients will have different requirements depending on their specific project.

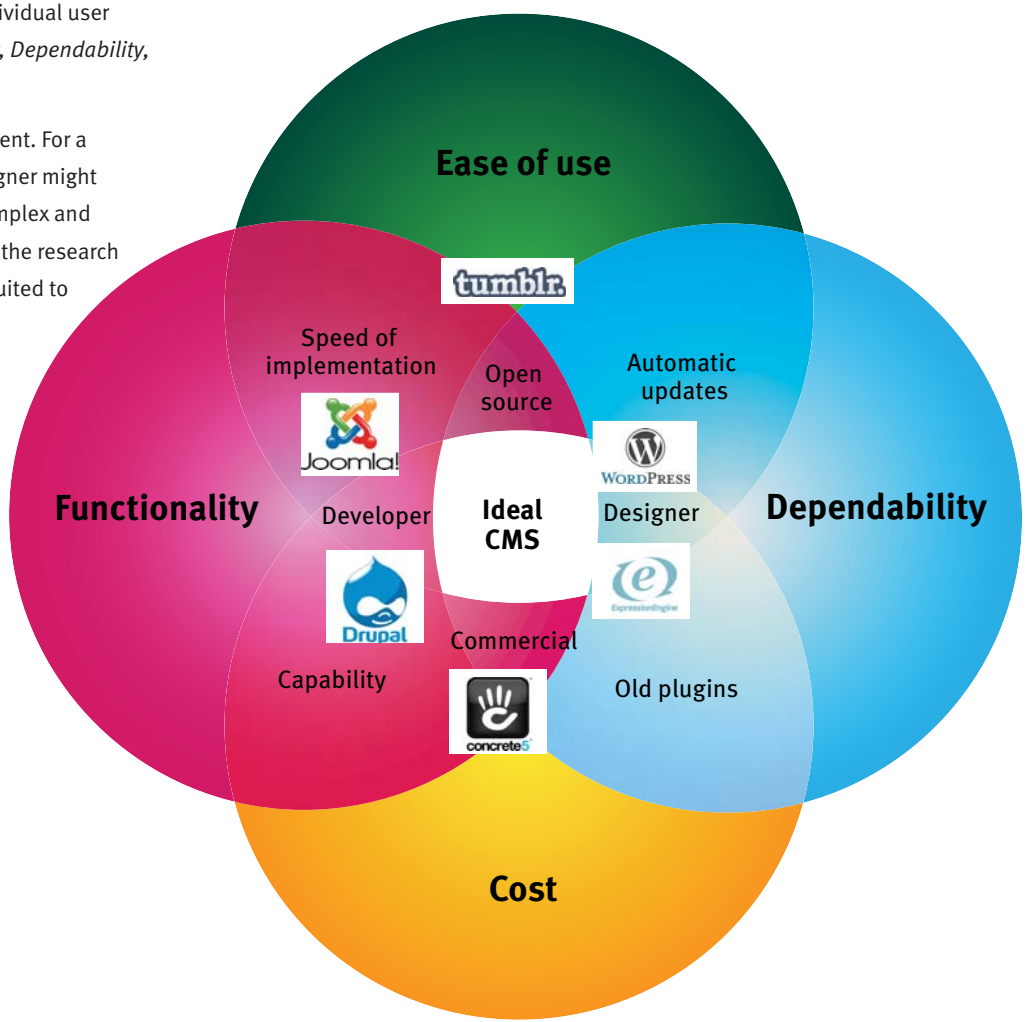


Figure 43: CMS directions - authors diagram

CMS MARKET OVERVIEW

As a result of my research, Figure 44 synthesizes the location of the 28 CMS solutions studied in the literature review.

This diagram Indicates where the designer is positioned in an array of available choices from the following pages. This, combined with the size of the community, helps to position the CMS in the marketplace. Position and indicated sizes are based on my research in the literature review.

Selecting a CMS is a difficult and complex task and it should be understood that there is no perfect solution. In consideration of this I would suggest the following criteria for selecting the an appropriate CMS. Selection is on a project by project basis because client requirements are subject to change.

Figure 44 demonstrates the increasing complexity and learning curve of various CMS solutions from achievable for the novice designer to complex for the accomplished developer.

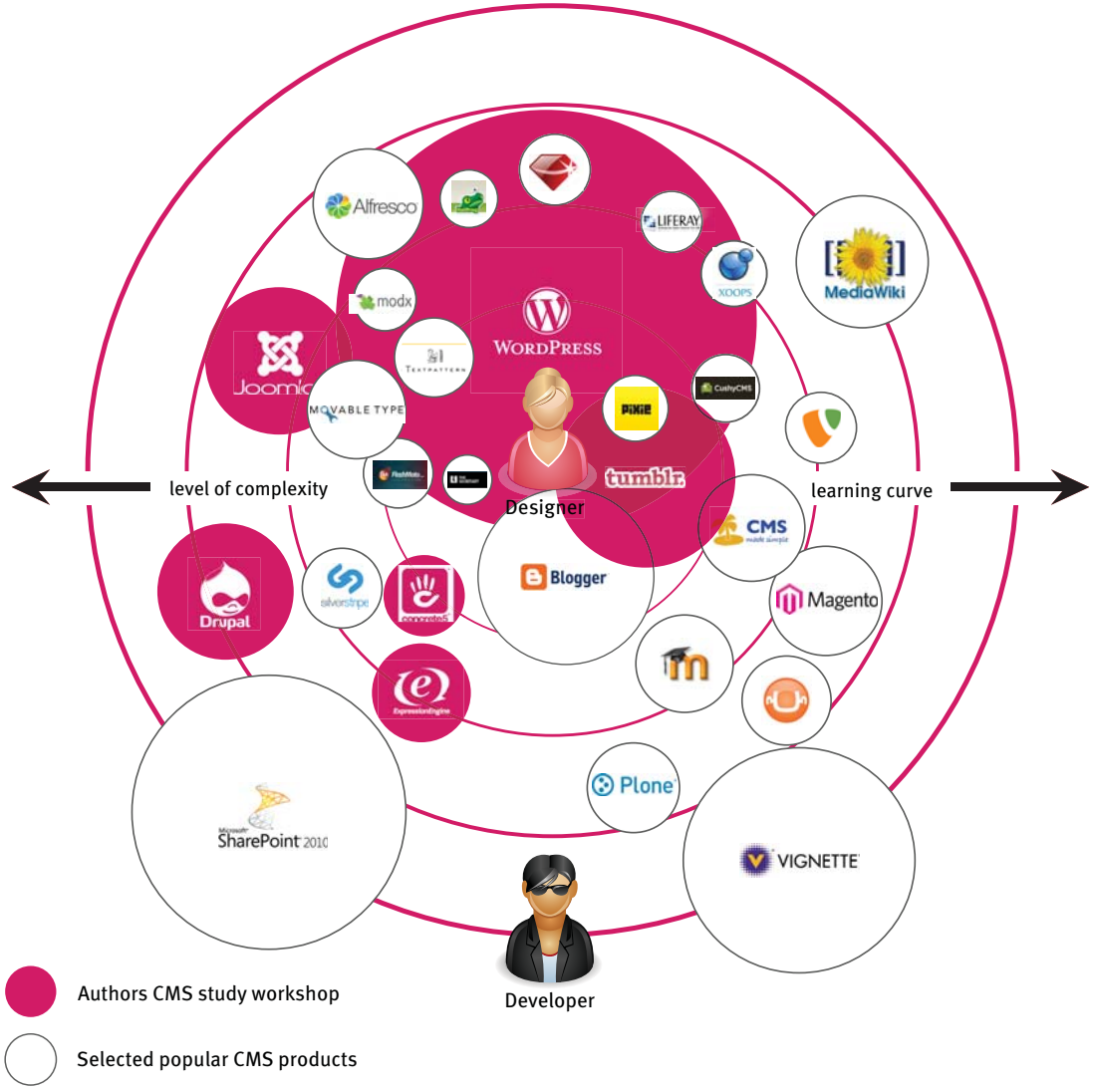


Figure 44: The CMS Market - authors diagram

Ten CMS selection recommendations

The following recommendations reduce and clarify this research into ten decision criteria to ask when selecting a CMS.

1. **The CMS should be focused on functionality**.
   If you are hosting a blog then use a CMS that does this well. Some CMS solutions will need a plug-in so make sure one is available; for example e-commerce plug-ins are not always available. Plug-ins range in cost from free to around $300+ dollars. When no plug-in will solve development requirements, custom code will need to be written at additional cost.

2. **The CMS needs an easily learned interface.**
   The main purpose of a CMS is to easily facilitate the updating of content. CMS selection must reflect the end-user who will be uploading the majority of web site content. This back-end can give end-users too much control over their content where they 'break' the web site with the incorrect use of HTML or too little control where they don't take ownership of content generation and renewal. Limited training should be provided by the developer until the end-user is competent in the CMS.

3. **The CMS regularly used by the developer is not the only CMS considered for implementation.**
   Often developers automatically implement what they know as a 'best fit' solution. This does not always fit with the clients needs or the requirements of the web site. Web 'generators' really need to learn a range of CMS options and choose from amongst more than one option, in order to effectively fulfill the needs of the client and the end-user.

4. **The CMS development time needs to be suitable to the available budget.**
   Clients and their web sites come in a variety of sizes and budgets. The size of the budget is usually proportional to web site requirements/capabilities. The client needs education in the choice of the CMS so that future grown can be factored in.

5. **The CMS should only install the main tool set.**
   Many CMS solutions will install tools that the web site will never need. For example why install e-commerce if the web site does not sell products online? Most CMS solutions offer plug-ins as ways of adding additional functionality. Unnecessary functionality can slow down a web site's performance, so the right toolset is critical.

6. **The CMS community and development should be well supported**
   There are thousands of CMS solutions and the web is a rapidly changing environment. CMS solutions can be maintained by one person or hundreds of developers. This will have an impact on implementing bug fixes and adding new features to the CMS software. Additionally, developers need technical assistance which is generally provided via books, online forums and the CMS's own help system.

7. **The CMS should allow for back-end customisation.**
   Clients have individual needs in terms of uploading content whether images, text or files. A customised back-end improves end-user experience and capability. Some customised features might include: versioning, limiting content uploaded, automatic thumbnail conversion, pre-formatting content, a WYSIWYG editor and author permissions.

8. **Does the CMS automatically have good Search Engine Optimisation (SEO).**
   Not all CMS's install with SEO as an option and it is a key component for a client to be found by a search engine.

9. **The designer/developer should have complete control over template appearance.**
   If the CMS is too difficult for the developer to alter the web site's design, customisation of appearance and functionality will be limited.

10. **The CMS should be easy to upgrade.**
    CMS developers often upgrade their software, fixing problems and flaws. Web site developers then need to update the CMS for example: security an area open sources is particularly prone to and very important to the client.

The future of CMS is promising. Both commercial and open source arenas provide an exciting opportunity as technology matures in this relatively new sector. Open source will continue to challenge commercial and enterprise CMS's and to represent real value for money. New technologies in 'cloud' hosting, HTML5/ CSS3, social media and the influence of the increasing web mobile market share will keep CMS developers updating and refining their products. The increased popularity of open source CMS's will bring commercial support and learning resources to the broader community.

APPENDIX 1 – 4

In this section:

Interview Data	**93**
Survey Instruments	**97**
Web site Screenshots	**112**
Bibliography	**113**

APPENDIX 1 – INTERVIEW DATA

Interview with Jeremy Epstein
PHP developer - Eskimo Digital

Q: Thanks for your time Jeremy, what's new on the Drupal scene?

There is a cool new installer for Drupal called Acquia Drupal, basically it is like XAMPP, basically a self hosing application that runs Drupal with ease. You should check it out.

As part of my CMS workshop the students undergo a series of CMS solutions. The first is Tumblr. Have you used Tumblr?

Yeh, yeh, we are running a company semi blog on Tumblr, it is really quick and easy, it is fairly limited and we can't use it for most of the work we do but it does have i ts place.

Q: Have you heard about Concrete 5?

No I have not heard about that CMS.

Q: What about *ExpressionEngine*?

Yeh, I have heard about really good things about *ExpressionEngine*.

I should say that although I call myself a Drupal developer more recently I have been looking at Django which is a framework in Python and not PHP. http://www.django-cms.org/ Django is probably not suitable for your students as it is not very easy to install and comes with limited functionality. You write templates specifically to fit you sites data.

Q: How do you feel when you are charging clients for a web site when the software is free?

The Drupal learning curve has always been an issue.

Q: Did you know that Wordpress currently has 240,000 sites built in Wordpress?

Yet I was just looking at the stats about 5 minutes ago Wordpress has 10 times the amount of sites in than that built in Drupal. But then again are most of them of just one dude writing about his cat and a Drupal site can have hundreds of thousands of people viewing it.

You should check out Acquia hosted Gardens. It is still in beta, you can set up a Drupal site much like a Tumblr site, it has an in-place theme editor and it is really cool.

Q: For me the complexities in Drupal complexities of how you set up a site. Have you done anything in Joomla!?

Yeh, after I got into Drupal I was roped into working in a Joomla! 1.0 site and is was just horrendous, particularly the templating and there is a lot of hardcoded CSS and HTML. Coming for a CMS where the motto is that all template work can be overwritten, it was just terrible. I hear 1.5 is better; but as far as I am aware Joomla! Is much less developer friendly than Drupal but out of the box it is better because It imposes a site structure on you. It does not have Drupal's taxonomy system.

Q: How do you find it when you starting an install in Drupal?

From my POV I already have plenty of tools when I have a starting install of Drupal. A long time ago when I started learning PHP by making my own half baked CMS like everyone does but systems like Drupal they give you a lot of functionality out of the box. It is just not practical for any sizable site to write your own CMS.

Q: What about the lack of support in Open Source software?

Well, for me it is not an issue but it is companies, who work in Open Source to offer commercial support, I mentioned Acquia earlier run by Dries Buytaert who founded Drupal offering enterprise support and setting up partnerships with companies like Microsoft and IBM. Digital Eskimo offers ongoing maintenance for its Drupal sites to clients and that ends up being the same as commercial CMS.

Q: Do you think it is a risk using open source?

Using 3rd party plug-ins and modules is a big issue in Drupal and in Open Source in general. There is no guarantee about the quality, the lifespan, whether it will be maintained, no support. In Joomla! And Wordpress is quite common to pay for modules but in Drupal it is not that common.

Q: Given that the software is free how can you charge for a Drupal site?

We charge for the service and not the product. If we were using a commercial product we would charge them for than so in effect they are getting the software for free.

Q: Is there a specific preference for Digital Eskimo to go with OpenSource?

Digital Eskimo has a number of sustainable and forward thinking goals company and OpenSource fits these goals.

Q: Do you modify existing plug-ins.

Basically I do not trust the quality of any given module and I assume that they are bad quality when I download them. It is a problem. When I find bugs I try and patch the bugs and submit them back to the Drupal community. Pretty much on every project we give back code to the community. When selecting a Drupal module you should look for: online documentation, how many bug reports there are, fixed vs fixed bugs, frequency of code changes, how active the developer is in responding to bugs. Usually it is worth keeping the modules code rather than writing your own, Drupal has a great API and is very flexible and ways to work around things you don't like.

Q: Are you committed to Drupal?

Basically we are committed to Drupal, Wordpress and Django are the three technologies we specialize in these days. However we are heading towards Django based on ease of use, maintenance, staff capabilities and has a good client administrative interface.

Q: Do you have clients come to you with preset ideas of what CMS you should use?

Yes, we have quite a few clients who have come to us asking us to build a Drupal site. If clients CMS expectations are really out of wack then we talk to them and try to assist them.

Q: What about the lack of support in Open Source software?

Well, for me it is not an issue but it is companies, who work in Open Source to offer support, I mentioned Acquia earlier and they have just started to offer this kind of support.

Q: How did you get started in web design and what were your qualifications?

I did a Bachelor of IT at UTS, I was doing HTML and CSS as a hobby, I taught myself PHP whilst an undergrad; I have never had any formal training in PHP and pretty soon after stumbled into Drupal and before I knew it I was a Drupal developer. After I finished at uni I have ended up here at Digital Eskimo. Right now I am rewriting my personal site in Django, it has been in Drupal for the last 5 years.

Q: What is a disadvantage of using Drupal?

One of the largest disadvantage's in Drupal is the upgrade path. Upgrading a Drupal site between major versions in a whole project in itself. Drupal has a tradition up upgrading the API between major versions. Once again there is the issue of contributed modules becoming obsolete and not maintained between major versions.

Q: How do you stay current with Drupal news?

I subscribe to a blog feed called Planet Drupal, visit Drupal.org and a member of various groups including a monthly Drupal meeting here in Sydney. I have been to 2 Drupal conferences in the US and New Zealand. I was also involved in a social innovation camp recently and actually built a site for refugees in Django.

Are you investigating other CMS options?

Not really, we just use the main 3 CMS options, Drupal, Djagio, Wordpress and also Tumblr. These cover all options of what we need.

Q: If you had to recommend a CMS to a student what you choose?

I would have to recommend Wordpress as the greatest entry point to the world of CMSs, templating and dynamic development and then Drupal after that. Wordpress has some serious limits but this upgrade path is very nice, auto install, auto update and this is largely because Wordpress core have a policy of not changing the API radically and supporting backwards compatibility, which Drupal has mainly ignored.

Q: Do you think Wordpress is limited by being a blog?

Yeah, it is its biggest strength and biggest limitation at the same time. Wordpress is incredibly successful because it s o simple and they have focused on the end user experience. They have not really put any resources towards a good API, developer experience and making it a flexible framework.

Q: Do you think learning a CMS has helped you become a better coder?

I often wonder Drupal had not been such a part of me learning of programming in general what would be different, I guess in some ways learning so much programming via a CMS has gotten me away from pure programming but it has taught me practical development. I could not have taught this by working on my own custom code, nothing compares with working on an OpenSource project with thousands of other people, absorbing he code and all the pain and lessons that have gone into the development of that code.

Q: Are you are committed to only staying in PHP?

I have at time felt scared that I am locked into PHP and Drupal and maybe I need to expand and have more options for development and my clients.

Q: One last question; now you have been a developer for a number of years what do you wished you had learnt?

I was not taught any of the stuff I use every day now. I was taught programming foundations, which I have found essential in helping me learn programming languages that I use these days. Also I had some classes about database theory, which is really helping me these days. I would really have liked some classes about design theory, typography and colour theory because you end up being a jack-of-all-trades in a web design company and it is frustrating.

Q: What do you think the future of CMS is?

It is all changing so fast, I think developers are not making the smartest choice right now to lock themselves into a CMS, I think everything we are using right now will be obsolete in 10 years and now maybe in 5.

Q: Any other thoughts?

I am very concerned about students learning PHP without proper programming foundations. It is far too easy to learn PHP without actually understanding important programming concepts, not necessarily OOP, underlining theories about data structures, efficient sorting and searching, what makes code perform well and secure. It is very easy for a student to become a script based coder as opposed to a programmer.

Interview with Adam Quark

Chief Technology Officer, - Mass Media

Q: What is your perspective on open source CMS vs. commercial software?

A: As a business, we support open source when it's appropriate, and as a business, often that means that we don't support it very much. This business itself is a dot-net house – we do do PHP development, but we generally work with dot-net, and, as a result...there's not very many good, dot-net, open source platforms out there. There's the likes of dot-net nuke, and site friendly which is a commercial licence but then open code, so they open it up a little bit, so it's only semi-open source licence. We actually use that quite a bit, because it's actually a good tool. In terms of going back to Joomla! and Drupal, and other purely open source PHP driven applications, we don't use it a hell of a lot, because commercially it's difficult to support, in that...

I've found in my research that unless you've got a 'crack team' of PHP guys, then supporting those actually becomes difficult in the long term. You can do the projects well, and you can do the projects reasonably cheaply, but your long term support and maintenance for ongoing relationships becomes more difficult, because I feel that PHP can be too easily updated at it's core. The environment you host it on gets updated and then we have things drop out and fall over. So you've got components of the system getting deprecated, where that just doesn't happen in Microsoft stuff. That's the core reason why we did go the Microsoft path. With the open source action then, I actually really back it strongly, I think it's a great...I think philosophically and as an approach I think it's really great because you actually get the chance to enjoy the development of the broader community, as opposed to just what you team can come up with, and that's a really good thing.

Our experience with that, and my development experience with that, I think, shows though, unfortunately a lot of things that happened out there, in the same way as, like, say the Android App Store has, you know, lots of 'apps' and lots of potential for 'apps', but lots of them were crap, um, is because when you have open source, you allow lots of people to learn, and then publish. So, you get this false sense of security that there's a hundred thousand 'plug-ins' for your CMS, but there's probably two hundred that are OK.

Q: There's nine thousand for Wordpress, and there's like, four survey modules, and none of them are any good.

That's right, and I mean, one of the killer ones for me for open source -when we usually have a budget for an open source 'app' CMS build, it's usually a small organisation and, what they want is, a forum and community organisation tools. Event planning tools, so, a 'what's on' calendar and event planner, and somewhere that people can comment on articles, and, your general basic site structure in CMS and a contact phone number. That sounds really obvious and really easy, and they sound really simple, and, you'll tick three or so of those boxes really quickly with any of the tools that you guys are looking at.

And then, you get to something like the event management tool and every one – every one of the open source CMS's I know certainly has this as a module, and they're terrible. And, in terms of then, the client will then

always say: “Cool, can I actually make it do this?”. And then, that’s the horrible question, because then you’re actually at this point where you go: “Well yeah, you can but now we can’t use the free module”. Or we have to start with the free module and then figure out what it does, and then code it up and then extend it, and then we’re back into the price, pretty much the same scope as something that you would have bought, correctly that wasn’t open source.

What we usually find is that by the time we actually do all the customisations to the open source platform, that they wanted, you’re actually at the point…you’re close to the cost of the non open source platform, alright? And, we can’t establish an SOA, with the people who are providing the platform, that ensures that our business needs are met, and that’s the challenge. So, we have though done, like Wordpress web sites, and stuff like that, for people, and quite nicely, and they’re good, and we like them, and they’re great, but, we know that they are fragile things, you know. And if Wordpress decides at any time to shut down their operation, ah, change the rules, ah…

Q: Wordpress doesn’t update the API, but if they update the API, you’re in…

Yeah, you’re in trouble. Even like I mean, it’s very difficult when you’re building your business around things that aren’t within your control, and open source is a bit of a free-for-all. It’s a great enabler for things, but it takes a consistent hand, and that’s why when you’ve got an in-house team whose managing things, it can be an awesome thing, because your not paying for these out-sourced people like us to do something for you. But for us to do business with you, we need to have a whole bunch of protections in place, whereas when you’re internal if you….Say internally, you go: “ We’re going to go with Joomla! or Drupal…”, and you build the whole network of your infrastructured sites, and then Drupal changes it’s rules to say: “You can’t do it this way…”, and that changes things for your business? Then you’re going to roll out the new one, and you haven’t… You still lose money on that translation, but you haven’t lost… and you’re not in a position to be sued by yourself you know what I mean… If we actually deliver something that has enough time and processes around it that then can be negated by a third party that has nothing to do with us that becomes a problem. So we’ve recently had to review something that had a similar issue with iPhone. The App store just changed the rules and you can’t generate Apps from a third party application, they did that to stuff Adobe around so you can’t say in CS5 save to iPhone option, you now can’t do that because they changed the rules. But that also means that there are several businesses out there that have their business build on bring your RSS feed to us, we’ll stick an interface design on it and then you can upload it as an App. Great cheap ways to publish content but if we had build a business on that then we would be out of business and we have had proposals out there to say that we can’t do that anymore so we would have to hand code that and it would triple the expense. My fear with open source is always that - I will have something in the market and they will change the rules or release the new version and I have a happy ongoing relationship with my client. If you are doing small local non-critical web sites I think they are great as a designer getting your teeth into it, they are great because you can for no outlay understand how this works and that’s really good. The hard transition I see for designers is to make sure you are aware of where the cut point for business where students at home on the weekend can build a web site cut it into HTML and have it released. They can’t understand why as a business I can’t meet the price point. The smart ones ask why not and want to understand the details of the operation and sometimes they will be angry or disenchanted and then I say I’m trying to protect the business and as a young guy with no liabilities if it falls over the client says “I going to sue you” the designer says that is about $3.50 it doesn’t matter but if I did that it would shut down the jobs of 50 people with a trailing debt that no so good and also when you scale up the business you scale up the risk. I could get wiped out for any mistake especially when you are doing a competition driven web site you open yourself up to liabilities from the site owners and everyone who entered the competition they could form a class action against you and it all becomes very complicated simply because the site became broken.

Putting us a site for a mates band is a very different proposition to putting up a site for EMI and saying please buy records here. They are very far away from each other in terms of the requirements. For me the selection of CMS’s comes down to functional and business requirements. Having a super solid reliable tool is critical to business. You need something that is totally controlled. This is the hard thing for designers and design students to understand the risk to your business and the clients business.

APPENDIX 2 – SURVEY INSTRUMENTS

The following anonymous surveys were hosted with the online service Survey Gizmo.

http://app.sgizmo.com.

WhichCMS – Designer survey 1/6/2010 Until 1/12/2010 - Compiled: 04/12/2010 via Survey Gizmo

Total Responses: **19**

Q1. How many years have you been working as a designer?

	COUNT	PERCENT %
10+	6	31.58%
1	2	10.53%
2	2	10.53%
4	2	10.53%
12	1	5.26%
15	1	5.26%
20	1	5.26%
5	1	5.26%
6	1	5.26%
7	1	5.26%
8	1	5.26%

Q2. What type of designer would you consider yourself?

	COUNT	PERCENT %
Web - frontend developer	15	78.95%
Web - interface	11	57.89%
Graphic	9	47.37%
Web - backend developer	7	36.84%
Identity	5	26.32%
Multidisciplinary	4	21.05%
Photographer	3	15.79%
Motion	2	10.53%
Other	2	10.53%

Q3. Were you taught any web technologies during your studies and if so what were they?

	COUNT	PERCENT %
HTML	11	57.89%
CSS	6	31.58%
No	6	31.58%
Javascript	5	26.32%
PHP	4	21.05%
Visual Basic	3	15.79%
Actionscript 2 or 3	2	10.53%
No, but I would have like to have been taught one	2	10.53%
Other	2	10.53%
AJAX	1	5.26%
ASP	1	5.26%
Flex MX	1	5.26%
Java	1	5.26%
Perl	1	5.26%

Q4: If you have learned to code in a scripting language like ASP, PHP, Perl, Java or Ruby how would you rate your learning experience?

	COUNT	PERCENT %
OK	12	66.67%
Difficult	2	11.11%
Very Difficult	2	11.11%
Easy	1	5.56%
Very easy	1	5.56%

Q5: Have you heard of any of these?

CMS	Never heard of it	Heard of it	Have actually used it	Love it!
CMS Made Simple	52.6%	42.1%	5.3%	-
Concrete5	31.6%	42.1%	5.3%	21.1%
CushyCMS	47.4%	31.6%	5.3%	15.8%
Drupal	26.3%	36.8%	31.6%	5.3%
EzPublish	52.6%	15.8%	21.1%	10.5%
e-Blogger	42.1%	36.8%	21.1%	-
ExpressionEngine	21.1%	63.2%	5.3%	10.5%
Freestyler	63.2%	15.8%	15.8%	5.3%
Joomla!	10.5%	31.6%	47.4%	10.5%
Mambo	26.3%	36.8%	36.8%	-
MODx	47.4%	36.8%	15.8%	-
PHPNuke	31.6%	57.9%	10.5%	-
Pixie	52.6%	42.1%	5.3%	-
Plone	47.4%	47.4%	5.3%	-
Radiant	68.4%	31.6%	-	-
Railfrog	89.5%	10.5%	-	-
Secretary	68.4%	31.6%	-	-
Serendipity	73.7%	26.3%	-	-
SkyBlueCanvas	63.2%	26.3%	10.5%	-
Sitecore	84.2%	15.8%	-	-
SilverStripe	47.4%	31.6%	21.1%	-
SurrealCMS	78.9%	21.1%	-	-
Symphony	73.7%	21.1%	5.3%	-
Textpattern	52.6%	21.1%	10.5%	15.8%
Tumblr	42.1%	42.1%	5.3%	10.5%
Typo3	42.1%	31.6%	26.3%	-
Vignette	68.4%	15.8%	15.8%	-
Umbraco	92.3%	7.7%	-	-
Wordpress	8.3%	16.7%	41.7%	33.3%
XOOPS	66.7%	25.0%	8.3%	-
Custom made in-house CMS	25.0%	16.7%	41.7%	16.7%
Other	66.7%	16.7%	8.3%	8.3%

Q6: How did you hear about Content Management Systems (CMS) and what what you can do with them?

	COUNT	PERCENT %
Blog posting	8	42.11%
Friend	7	36.84%
Lecturer	7	36.84%
Other	5	26.32%
Magazine	3	15.79%
Web advertisement	3	15.79%
Book	2	10.53%
Never heard about one whilst a student	2	10.53%

Q7: Have you self taught yourself a CMS?

	COUNT	PERCENT %
Concrete5	2	10.53%
Drupal	2	10.53%
All that I've listed in previous question.	1	5.26%
Drupal, EE, Wordpress, Textpattern	1	5.26%
Drupal, Joomla!, Concrete5	1	5.26%
EE, Joomla!, WP, etc	1	5.26%
ExpressionEngine, Joomla!, Tumblr	1	5.26%
Joomla!	1	5.26%
Joomla!, Concrete5, Wordpress	1	5.26%
Joomla!, Drupal and of course C5	1	5.26%
Mambo, Joomla!, Drupal, Wordpress, GetSimple	1	5.26%
na	1	5.26%
sitezen	1	5.26%
Sportal - Wests Tigers NRL	1	5.26%
symfony, Wordpress	1	5.26%
Taught my employees using an internal CMS	1	5.26%
Wordpress	1	5.26%

Q8: If you have experienced a CMS what is your opinion on the modification of templates with concerns to aesthetic appeal?

1. -
2. At the end, it just HTML and CSS
3. Concrete5 is the simplest
4. Don't Understand
5. don't understand the question!
6. easy
7. easy to manipulate
8. Great
9. Great idea
10. use Joomla! and it's greats for templates. We create our own.
11. I'm not sure what you mean here
12. Largely irrelevant - in most situations, the design is custom-built from scratch then integrated later in the project.
13. Limited frontend ability
14. Much of it depends on the framework
15. na
16. Need to tailor to clent's branding
17. Some of them are two hard, some of them are easy, like concrete5
18. This isn't enough room to discuss these thoughts.

Q9: Has learning a CMS helped you improve your web coding skills?

1. Absolutely
2. better understanding of web elements
3. CSS
4. Customizations
5. helped wikth PHP and Javascript
6. i think more ahead
7. I've contributed where I've good, but most it is a lot of "They don't
8. know what they're doing!", then do it my self differently.
9. Improved PHP & XML
10. it has
11. makes php easier for me
12. motivates and gets you used to looking and modifying others code

Q10: Do you use an CMS to maintain your portfolio online and if so what software?

1. concrete5
2. Joomla!
3. Not applicapable
4. blogspot
5. c5
6. concrete5 and Joomla!
7. currently old custom coded - moving to EE with time!
8. Drupal, Wordpress, Menalto Gallery
9. ExpressionEngine
10. ironically I work so much I have little time to update my portfolio site
11. Just no. OK?
12. our company's proprietary cms
13. To lazy to make an online portfolio
14. Wordpress
15. Wordpress MUWhichCMS – Employer survey 1/6/2010 Until 1/12/2010 - Compiled: 04/12/2010 via Survey Gizmo

Total Responses: **19**

Q1: How many full-time and freelance employees do you have continually working on projects?

	COUNT	PERCENT %
1-5	10	52.63%
6-12	5	26.32%
13-20	1	5.26%
21-35	1	5.26%
35-59	1	5.26%
60+	1	5.26%

Q2: How many frontend & backend coders do you employ in the web/ mobile development team?

	COUNT	PERCENT %
2	5	26.32%
3	5	26.32%
1	2	10.53%
5	2	10.53%
8	2	10.53%
10+	1	5.26%
4	1	5.26%
6	1	5.26%

Q3: Which platforms do you currently use?

	Not implemented	Implemented years ago	Occasionally implemented	Heavily implemented
Actionscript 2	26.3%	31.6%	42.1%	-
Actionscript 3	10.5%	15.8%	57.9%	15.8%
ASP.net	47.4%	21.1%	15.8%	15.8%
Flex MX	78.9%	5.3%	15.8%	-
Java	68.4%	10.5%	15.8%	5.3%
Javascript	5.3%	10.5%	21.1%	63.2%
Perl	57.9%	15.8%	21.1%	5.3%
PHP	5.3%	5.3%	26.3%	63.2%
Python	84.2%	10.5%	5.3%	-
Ruby	73.7%	10.5%	5.3%	10.5%
Visual Studio	63.2%	21.1%	5.3%	10.5%
Other	63.2%	10.5%	10.5%	15.8%

Q4: Which are the more popular CMS solutions used in your organisation?

	Not implemented	Implemented years ago	Occasionally implemented	Heavily implemented
CMS Made Simple	94.4%	-	5.6%	-
Concrete5	94.4%	-	-	5.6%
CushyCMS	88.9%	-	11.1%	-
Drupal	66.7%	-	16.7%	16.7%
e-Blogger	100.0%	-	-	-
EzPublish	100.0%	-	-	-
ExpressionEngine	88.9%	-	5.6%	5.6%
Freestyler	94.4%	5.6%	-	-
Joomla!	52.6%	5.3%	31.6%	10.5%
Mambo	94.4%	5.6%	-	-
MODx	94.4%	-	-	5.6%
PHPNuke	100.0%	-	-	-
Pixie	100.0%	-	-	-
Plone	94.4%	-	5.6%	-
Radiant	88.9%	-	11.1%	-
Railfrog	100.0%	-	-	-
Secretary	100.0%	-	-	-
Serendipity	100.0%	-	-	-
SkyBlueCanvas	100.0%	-	-	-
Sitecore	94.4%	-	-	5.6%
SilverStripe	100.0%	-	-	-
SurrealCMS	94.4%	5.6%	-	-
Symphony	100.0%	-	-	-
Textpattern	100.0%	-	-	-
Tumblr	88.9%	5.6%	5.6%	-
Typo3	-	-	-	-
Vignette	94.4%	-	5.6%	-
Umbraco	88.9%	-	5.6%	5.6%
Wordpress	29.4%	5.9%	47.1%	17.6%
XOOPS	100.0%	-	-	-
Custom made in-house CMS	22.2%	11.1%	11.1%	55.6%
Other	70.6%	5.9%	-	23.5%

Q5: For commercial projects do you use open source software and why?

1. Allows better project outcomes for lesser investment and is often requested by our clients
2. Client flexibility
3. easier to add modules
4. extensibility
5. Fewer barriers to entry and commercial support for our CMS (Modx) will soon be available, anyway.
6. For e-commerce we use magento, quality commercial options are usually well beyond the budget our customers.
7. If we can support it
8. It reduces TCO for the client, and provides more choice than commercial software
9. Magento, it has a free version and great community support
10. Our in house CMS
11. To take advantage of the strong codebase and community creativity
12. We use open source almost exclusively because it means we can completely customise the solution for the client and the client can be safe knowing anyone would be able to make changes.
13. why not? It's often more stable than one might think
14. yeh, cheap
15. Hard sell, lack of support and confidence
16. Not part of our philisophy or core skill set
17. prefer to use our own solution as can create a complete customised solution
18. too many bugs and not SEO friendly
19. We use both so wasn't sure what to put.

Q6: What are the web design/ development skills you are currently seeking when hiring graduate students?

	COUNT	PERCENT %
HTML	17	89.47%
Javascript	17	89.47%
AJAX	13	68.42%
PHP	15	78.95%
CMS experience	11	57.89%
ActionScript 3	6	31.58%
Other	5	26.31%
Personal blog/ portfolio	5	26.31%
ASP	3	15.79%
Java	3	15.79%
Ruby	4	21.05%
ActionScript 2	2	10.53%
C#	1	5.26%
C++	1	5.26%
Perl	1	5.26%

Q7: Do you feel that educational institutions are teaching skills to match the demands of employers?

	COUNT	PERCENT %
No	9	47.34%
Yes	10	52.6%

OTHER VALUES: NO

1. dont really know...
2. Need real experience
3. Not even close.
4. Not sure about programmers, not in multimedia/creatives
5. they do not have any sense of SEO! most importantly, we do not need beautiful design but enquiry creating web desgin / development ! think about it
6. they dont teach CMS
7. they only touch the surface
8. Too much emphasis on tools instead of the underlying technology
9. Web industry is so broad it is very hard to teach every discipline and skillset - self-starters always do better because they learn as they go. Good education is always a benefit

OTHER VALUES: YES

1. In general
2. Most of the time
3. Mostly
4. no 100% sure
5. Our developer interns have always been sufficiently capable to do our work,
6. though this could also be from their own self-teaching.
7. Some does
8. Some, I am also a part time teacher in the TAFE system, and I feel the standard is quite good, however could do with some higher level programming, providing the resources.
9. The internet changes faster than institutions can adapt, but for the most part the gap in skills is the students responsibility.
10. they are doing a good job
11. yes and no. It would be better if students understood the practicalities of design and development in a business context.

Q8: How did you decide which CMS to implement?

1. After years of custom development including writing our own templating language in C, we tried Drupal on a community web site. We liked the extensibility and have adopted it for all our projects going forward.
2. Usability and easy to extend
3. Needed the most flexible solution, which handled templates and front-end code well.
4. easy to use admin..
5. Functionality required by client, cost effectiveness, timeline available
6. build and use our own...
7. Based on the clients needs and the needs really. If its an ecommerce web site, we would use a package that enables us to have more control over the site and its sales, no point reinventing the wheel. If it requires custom development tho, we use a Ruby based CMS that can be expanded easily .
8. Best practice and .NET based.
9. Features, support and availability of developers
10. Business requirements Design flexibility Ease of Use Support
11. Clients preferences
12. we use our in house developed cms software, but occasionally use open source software
13. It is solid
14. Ease of extensibility and how usable it is for clients to use.
15. Best suited to our and our client's multi-media needs
16. We choose from a wide range of CMS Systems, based on the client's needs.
17. Broadly, though, we use open source, easy to update systems that are supported by a strong community.
18. If a client requests a specific CMS, we will use that. Otherwise, we use our own internally developed CMS
19. Functionalty, stable roadmap, continuing support. We looked for a system that was very flexible for developers over being easy for novices to install.

Q9: How does your organisation stay on top of rapidly changing technology in the CMS market?

1. We participate actively in the Drupal development community.
2. uhm, by using the internet?
3. We try to, but there's a lot going on and it's hard to keep up with all the different CMS's and then testing them.
4. open source and constant developement keeps me updated..
5. By keeping inline with advancement in technology through seminars, workshops and training, by employing people who's expert in their chosen area of expertise.
6. we follow trends to implement new features into out own package
7. Its actually not that hard. Clients needs keep changing, and force you to implement new technologies. Staff are encouraged to read online news / blogs everyday and to experiment with new technologies whenever possible.
8. Dedicated IT team monitor
9. Always researching the latest and we provide alternatives by using specialists
10. Research online Employees stay in touch Trending developments
11. We probably don't
12. we research and add function to our cms as require by client or suggest additional functions to clients and improve
13. reading blogs
14. We look at other CMS's when the web design community start raving about them.
15. With a LOT of hard work - we've selected one CMS, as it became inefficient for us to support multiple for clients who don't see why they should pay for CMS upgrades which often only mean back-end improvements.
16. We are always updating our knowledge.
17. From our own perspective, the "rapidly changing technology" is really just a proliferation of new software products that achieve basically the same result. We focus on our own CMS, making sure it's as good as it can be, defect free and secure. The core developers for our CMS live and breathe software, and spend at least an our a day reading and researching about web technologies. We stay abreast of the latest changes in web development technology in this way.
18. We're connected to our CMS's developers via Facebook and Twitter. We also extensively read blogs and twitter feeds to stay on top of general industry trends and news about CMS options.
19. user friendly CMS / SEO

Q10: Any further thoughts on what web design/ development students are being taught in educational institutions.

1. Learn to use version control software.
2. All web design/development courses are pretty basic (even the ones called advanced). They think OOP is useless, never heard about SQL injections etc.
3. They're being taught tools, not skills. Teaching students to problem solve, research, handle client and communicate are just as important as anything else.
4. there should b a CMS knowledge base
5. They should've been taught other skills related to being a designer/coder/programmer, other than just design skills. For example, time management, handling clients, self discipline, integrity, etc.
6. more practical experience is required. you cant learn from text books or online tutorials... a good quality programmer or designer need to have field experience and the only way you can get this is out of the classroom... Most of our freelances have never studied and we have found them to be leagues ahead of those that have been educated.
7. Basically, web students need to learn a range of technologies, and not be forced into any one language.
8. Should implement a 1 year on the job training as part of the course, paid in part by the Govt and as a paid "apprentice" position within the agency.
9. Technology is always changing. Teaching HTML/CSS/Javascript/AJAX is a minimum then they learn the rest on the job.
10. more business skills and lessons on the practicalities and realities of dealing with clients
11. Would like to see more students have knowledge in creating spec/data forms for internal use.
12. keep up with changes in the industry
13. none.
14. I think that marketing or user experience/information architecture needs to be taught along with the standard course modules.
15. They should ALL be encouraged to do internships in real working environments, solving real problems, learning what its really like servicing clients. If they are freelancing themselves out directly to end clients, they must be taught why NOT to underprice - as it reduces the value of the skill set a properly structured business MUST charge to stay sustainable. So if they work out their contractor rate to an agency is $30ph, any direct client's MINIMUM rate should really be $120ph. Obviously, the rates are always linked to their knowledge and how much they can output in that hour.
16. no.
17. Too much emphasis seems to be placed on how to use products such as Dreamweaver, or Photoshop (fine for web designers), but web developers should be taught the fundamentals of web development, eg: how to hand code a web site, the prototypical inheritance model of JavaScript v class-based
18. inheritance etc.

WhichCMS – Employer survey 1/6/2010 Until 1/12/2010 - Compiled: 04/12/2010 via Survey Gizmo

Total Responses: **3**

Q1: Which institution do you teach at?

	COUNT	PERCENT %
Qantm, SAE	1	33.33%
Qantm	1	33.33%
UTS	1	33.33%

Q2: What is the degree/diploma title of the course that you lecture in?

	COUNT	PERCENT %
Degree Creative Media	1	33.33%
Masters in IT	1	33.33%
Multimedia, Audio, Film	1	33.33%

Q3: Do you design/ develop web sites outside the teaching context?

	COUNT	PERCENT %
Yes	2	66.67%
No	1	33.3%

Q4: Which computer languages are you proficient in?

	N/A	Beginner	Hobbyist	Professional
Actionscript 2	50.0%	-	-	50.0%
Actionscript 3	-	-	-	100.0%
AJAX	-	-	100.0%	-
ASP	100.0%	-	-	-
C++	50.0%	-	50.0%	-
CSS	-	-	-	100.0%
Flex MX	50.0%	-	-	50.0%
Java	66.7%	-	-	33.3%
Javascript	50.0%	-	50.0%	-
Perl	50.0%	50.0%	-	-
PHP	50.0%	-	-	50.0%
Python	100.0%	-	-	-
Ruby	100.0%	-	-	-
Other	33.3%	-	-	66.7%

Q5: Do you feel that your institution is teaching coding skills to match the demands of employers?

	COUNT	PERCENT %
Yes	2	66.67%
No	1	33.33%

Q6: Do you teach web scripting/ programming languages in your course?

	COUNT	PERCENT %
CSS	2	100.00%
HTML	2	100.00%
Kinda, I teach web design - Dreamweaver	2	100.00%
Actionscript 2	1 5	0.00%
Actionscript 3	1	50.00%
C++	1	50.00%
Database design	1	50.00%
Flex MX	1	50.00%
Javascript	1	50.00%
Objective C, XHTML...	1	50.00%
PHP	1	50.00%
Ruby	1	50.00%
SQL	1	50.00%

Q7: How current is your knowledge in the Content Management System (CMS) environment?

	COUNT	PERCENT %
I am interested in CMS software but not sure where to start	2	66.67%
I am not interested	1	33.33%

Q8: Do you think teaching CMS/ blogging software, including template building would improve students web coding skills?

	COUNT	PERCENT %
Yes	2	66.67%
Not sure	1	33.33%

Q9: Within your institution is there flexibility in your course to introduce the teaching of CMS/ blogging software?

	COUNT	PERCENT %
Yes	2	66.67%
Not sure but probably yes	1	33.33%

Q10: Optional email address for the results of this survey when published.

Withheld for privacy reasons.

WhichCMS – Workshop First Choice survey 1/6/2010 Until 1/12/2010 - Compiled: 04/12/2010 via Survey Gizmo

Total Responses: **9**

Q1: What is your registration e-mail address?

Withheld for privacy reasons.

Q2: Which CMS did you first select?

	COUNT	PERCENT %
Concrete5	2	22.22
Joomla!	2	22.22%
Tumblr	2	22.22%
Wordpress	2	22.22%
ExpressionEngine	1	11.11%
Drupal	0	0%

Q3: Why did you select this CMS?

	COUNT	PERCENT %
Wanted to experiment	6	66.67%
Ease of use	5	55.56%
Looked easy	5	55.56%
Wanted to see if I could do something like make a portfolio site	5	55.56%
I hear it is popular	3	33.33%
No coding	3	33.33%
User interface	3	33.33%
Wanted to test out various themes/ plugins	3	33.33%
Seemed logical	2	22.22%
Allows more control	1	11.11%
I wanted to create a simple, quick way to make a portfolio for my friend and I also liked that you an be followed by others or following othecrs as I would use deviantart...etc. for selling her images and tumblr i would use for people who are not into commerce so I can show her stuff to different kinds of people.	1	11.11%
It seemed like a good introduction into the use of CMS	1	11.11%
Looked hard	1	11.11%
Lots of coding	1	11.11%

Q4: Does this CMS change way you look at web site development?

1. -
2. Although I would like more creatuve freedom
3. at its basic level it is essentially the same as the 'easier' ones like Wordpress which i am familiar with
4. definately, but i am already aware of the differences between static and dynamic web sites and the benefits of dynamic over static web site.
5. Gives me more options
6. It's easier than I thought after struggeling with WP, EE, Joomla!. So easy to use. Will use it in future for my friends supporting them without getting payed.
7. Looks not that easy but really useful
8. provided some good logical ideas about layout of content
9. Yes adds more apprecation

Q5: In order list all list your interest in the CMS software demonstrated.

Tumblr	4.8
Joomla!	2.7
ExpressionEngine	4.2
Wordpress	2.3
Drupal	3.8
Concrete5	3.2

Q6: Did your experience of the CMS match your expectations after the demonstration?

1. 100% and more... Video tutorial are looking easy but when you do them at the end it's more difficult. For Tumblr. It's easy as the tutorial looks like.
2. Absolutely
3. easy to create
4. i found it a lot more difficult, but this was only due to the template customisation that was required straight away. I think if i was able to play with the site first to get a basic sense of how it works, then stepping up to customising templates etc might have seemed more approachable.
5. it did. 1 11.11%
6. it was not at all difficult to get going
7. know more Wordpress and how to use it
8. so so.
9. Yes

Q7: Would you be encouraged to improve your coding skills if you learned this software completely?

1. 100% I'm using EE at the moment for my web site and learned a lot of coding through this. The same for WP for my Portfolio
2. Definitely
3. I'm happen to learn more, because I'm the person keen to learn more new system.
4. it does. i am not completely satisfied with the limitation of any software, and would add additional scripting to make the best of comcrete5 framwork.
5. its like learning another language
6. probably
7. to help give more functionality
8. Yes
9. yes for greater conrol and creativity

WhichCMS – Workshop Post workshop survey 1/6/2010 Until 1/12/2010 - Compiled: 04/12/2010 via Survey Gizmo

Total Responses: **9**

Q1: What is your registration email address?

Withheld for privacy reasons.

Q2: Did the workshop give you a good introduction content management systems (CMS)?

1. also into open source options
2. good understanding
3. Met all my expectations, even more
4. quite a lot of information
5. very good
6. Yes
7. Yes and i will use it in the future
8. Yes. I would like to learn and use everytime another CMS.

Q3: Was the duration of the workshop suitable?

1. yes
2. ... but I was a bit slow in actions
3. but could have been made shorter
4. but it could have been condensed a bit - particularly the overview of the different cms's as that was all covered in the tutorials anyway
5. suitable
6. the time was suitable
7. Yes.

Q4: Was the level of skill required to complete the tutorials suitable?

1. yes
2. Yes
3. basic knowledge only required so not itmidating - but the workshop
4. could be pitched more in this way
5. I have chosen Tublr and concrete5 as I haven't used them before and was interested to know them. Today's tutorial made it easy to follow it up without having any background in these 2 CMS's
6. nothing was too complicated
7. Some basic skills only
8. the video tutorials explained clearly for every step
9. tutorials were fairly easy

Q5: Were you clearly instructed as to the required tasks?

1. yes
2. Yes
3. easy to understand
4. I knew what was needed to do with the help of the video tutorials
5. I were
6. More than that.
7. tutorials were great
8. Used the intro file

Q6: Do you feel you understand the differences between the CMS solutions presented in the workshop?

1. yes
2. Yes
3. at a basic level they all seem the same to me - with a couple that required a more nerdy outlook and more time spent to achieve the desired result, im still unclear about the best options for me
4. i do
5. It was mentioned during the intro into each CMS by Lance and as well the tutorials where a great help to get it freshened it up and into details.
6. most of them yes, will deep into later
7. some can be more complicated than others
8. The workshop gave me quite lot of more information to understand the differences of each CMS.

Q7: Could a CMS influence your desire to code in a web scripting language like HTML, CSS, PHP or Ruby?

	COUNT	PERCENT %
yes	7	87.5%
no	1	12.5%

Q8: Will you investigate any of these CMS solutions further?

	No	Maybe	Yes
Tumblr	-	25.0%	75.0%
ExpressionEngine	25.0%	37.5%	37.5%
Wordpress	-	-	100.0%
Concrete5	12.5%	25.0%	62.5%
Drupal	25.0%	37.5%	37.5%
Joomla!	12.5%	12.5%	75.0%

Q9: Please enter a comment about the CMS/s you actually installed and completed the supplied video tutorials.

1. Easy easy easy
2. easy to follow
3. easy to use
4. Easy to use.
5. found it easy to work with
6. Good control, well organised, coding knowledge
7. great - that it also included an insight into downloads - templates etc
8. probably could have just done a very quick demo of managing content and then would have been great to see some examples of expanding your site with plugins
9. simlar to Wordpress
10. That would be my next blog for another friend who wants to be more interactive with her customers
11. The supplied video tutorials are really helpful. I could manage the Wordpress throught the workshop.
12. Very Very Easy, with limited design options

Q10: Following this workshop there will be one more survey in 3 months time. Completing the survey is completely optional and anonymous. Do you consent to be emailed the survey in 3 months time?

No	Yes
1	7

WhichCMS – Workshop 3 months after survey 1/11/2010 Until 1/12/2010 - Compiled: 04/12/2010 via Survey Gizmo

Total Responses: **3**

Q1: What is your registration email address?

Withheld for privacy reasons.

Q2: Have you visited the WhichCMS blog or the links from that site since the workshop?

1. i have not visited any of these
2. Regulary
3. Yes a few times for reference

Q3: How have you found the technical challenges of setting up a site, hosting, plugins, ftp, installation, database setup etc.

1. But easier starting from scratch than using a CMS 1 33.33%
2. No problems at all 1 33.33%
3. Tumblr very easy and Joomla! took a bit but wasn't to bad

Q4: Have you read any CMS reviews either online or in print format?

1. have read lots of articles particularly on a list apart, but these were largely about css not cms
2. No
3. Online through blogs and tutorial sites

Q5: How would you rate any experience since the workshop in a particular CMS?

	Not installed	Very Difficult	OK	Very Easy
Tumblr	33.3%	-	-	66.7%
Wordpress	33.3%	-	66.7%	-
Concrete5	66.7%	-	-	33.3%
ExpressionEngine	66.7%	-	33.3%	-
Joomla!	33.3%	-	66.7%	-
Drupal	66.7%	33.3%	-	-
I tried another CMS	66.7%	-	-	33.3%

Q6: Has learning about CMS solutions sparked you interest in coding web design/ development and if so what have you done lately?

1. But I have recently completed my own web site
2. I created for my friend a tumblr site as this was the easiest way and done in 5 minutes. It was actually voted at the international webby awards with a price.
3. regisnlp.com; eastsideink-bondi.com; sportmindandbeyond.com

Q7: Do you have any online projects as a direct result of the WhichCMS workshop?

1. elefteria.tumblr.com
2. have completed an online project, nothing to do with cms though
3. No, None.

Q8: Did you find the WhichCMS workshop useful in informing your choice of CMS?

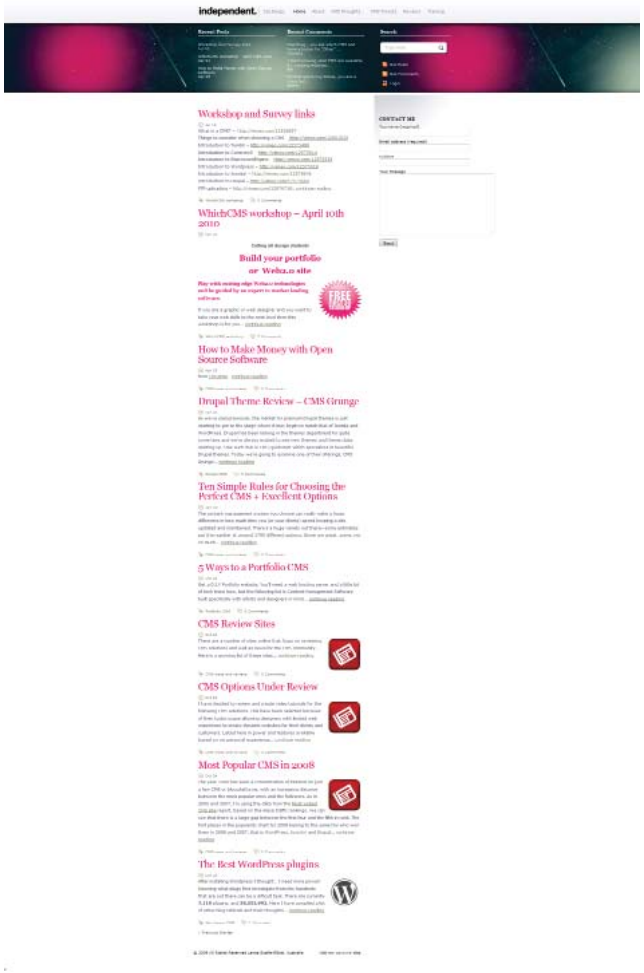
1. 100%. I started to love CMS.
2. I can now choose what CMS will be most efficient for each project

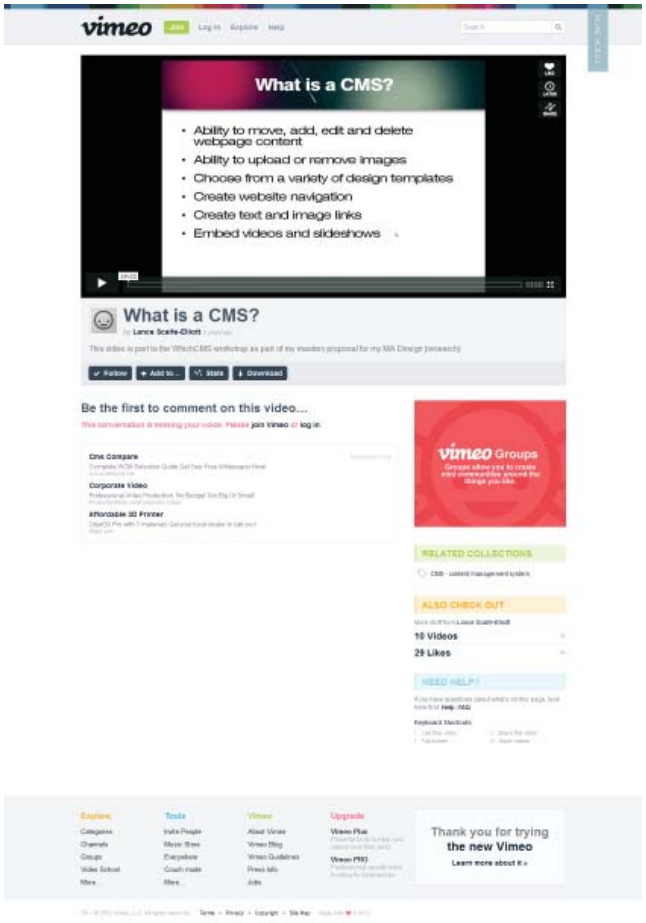
i thought it was an interesting idea, but it did not work for me Appendix 3 – Workshop description

APPENDIX 3 – WEB SITE SCREENSHOTS

There is a web site accompanying this thesis at:

http://www.independentdesign.com.au/whichcms created with Wordpress.





APPENDIX 4 – BIBLIOGRAPHY

Content Management System Software Review. Content Management System Software Review. n.d. http://cms-software-review.toptenreviews.com/ (accessed Janurary 23, 2010).

2010, Weblog software. Weblog software. 2010. http://en.wikipedia.org/wiki/Weblog_software (accessed March 12, 2010).

Admin. Holistic Web site Development and Promotion. Edited by http://www.web sitemagazine.com/content/blogs/posts/pages/holistic-web site-development-and-promotion.aspx. 2010. http://www.web sitemagazine.com/content/blogs/posts/pages/holistic-web site-development-and-promotion.aspx (accessed June 10, 2010).

ajwagner777. Joomla! vs Xoops vs Typo3 vs Drupal. 2007. http://drupal.org/node/184455 (accessed June 6, 2010).

Alexia. Alexia. 2010. http://www.alexa.com (accessed June 6, 2010).

Baker, W. Web Design Versus Web Development. 2010. http://articles.webdesigners123.com/web_design_versus_web_development.php (accessed June 6, 2010).

BlueOrg. Generalist vs Specialist. 2006. http://forums.whirlpool.net.au/forum-replies-archive.cfm/638470.html (accessed June 6, 2010).

Boag, P. 10 Things To Consider When Choosing The Perfect CMS. March 5, 2009. http://www.smashingmagazine.com/2009/03/05/10-things-to-consider-when-choosing-the-perfect-cms/ (accessed 2 25, 2010).

Boag, Paul. Web site Owner's Manual. Grennwich: Manning Publications Co., 2010.

Boiko, B. Content Management Bible. Indiana: Wiley Publishing, 2005.

Bruinsma, M. "Design Interactive Education." In The Education of a Graphic Designer, 175-180. New York: Allworth Press, 1998.

Budd, A. Blog Design Solutions. Berkley: Apress, 2006.

BuiltWith. BuiltWith Internet Technology Usage Statistics. 2010. http://trends.builtwith.com/ (accessed June 6, 2010).

Buytaert, D. Gartner puts Drupal in visionaries quadrant. 2009. http://buytaert.net/gartner-puts-drupal-in-visionaries-quadrant (accessed June 6, 2010).

Byrne, T. Looking beyond the magic quadrant to find the nitty-gritty. 2009. http://www.cmswatch.com/Blog/1660-Assessing-WCM-vendors (accessed June 7, 2010).

Choo, G. Make a better CMS product decision with help from these resources. 2002. http://articles.techrepublic.com.com/5100-10878_11-1051946.html (accessed June 6, 2010).

Chris, B. Should developers be specialists or generalists? 2008. http://stackoverflow.com/questions/17903/should-developers-be-specialists-or-generalists (accessed May 22, 2010).

Cluver, M. Concrete5, Content Management in the Web 3.0 World? 2009. http://www.piranhamethod.com/2009/05/concrete5-content-management-in-the-web-30-world/ (accessed September 5, 2009).

CMS compare : Concrete5, Wordpress, Silverstripe 2008. CMS compare : Concrete5, Wordpress, Silverstripe. 2008. http://www.webmasterbulletin.net/2008/09/cms-compare-concrete5-wordpress-silverstripe/236 (accessed July 2, 2009).

Cohen, J. The Unusually Useful Web Book. Indiana: New Riders, 2003.

Cooper, H. M., Hedges, L. V. & Valentine, J. C. (2009), The Handbook of Research Synthesis and Meta-Analysis, 2nd ed., Sage Publications, New York

Cushy CMS vs Surreal CMS - which is the best Lightweight free CMS, 2010. Cushy CMS vs Surreal CMS - which is the best Lightweight free CMS. 2009. http://handy-web-hints.blogspot.com/2009/10/cushy-cms-vs-surreal-cms-which-is-best.html (accessed December 12, 2009).

D, Houle. Why can't we be friends? Joomla! versus Drupal. 2007. http://www.communicopia.com/blog/Joomla!-versus-drupal (accessed April 23, 2010).

Douglass, R., Little, M,. and Smith J. Building Online Communities with Drupal, phpBB and Wordpress. New York: Apress, 2006.

Dynamic web page, 2010. Dynamic web page. 2010. http://en.wikipedia.org/wiki/Dynamic_web_page (accessed April 15, 2010).

Emmers, G, interview by L Scaife-Elliott. "Course coordinator- Graphic Design." What do you teach? Sydney, NSW, (June 10, 2010).

Epstein, J, interview by L Scaife-Elliott. "PHP developer." Open Source CMS development. Sydney, NSW, (May 12, 2010).

ExpressionEngine Review, 2009. ExpressionEngine Review. 2009. http://cms-software-review.toptenreviews.com/expression-engine-review.html (accessed February 16, 2010).

Fowler, C. My Job Went to India. n/a: The Pragmatic Programmers, 2005.

Frameworks. Frameworks Overview. 2010. http://trends.builtwith.com/framework (accessed June 6, 2010).

Gabriel-Petit, P. Specialists Versus Generalists: A False Dichotomy? 2009. http://www.uxmatters.com/mt/archives/2009/02/specialists-versus-generalists-a-false-dichotomy.php (accessed May 9, 2010).

Gartner. "Magic Quadrant for Web Content Management." http://www.gartner.com. 2009. http://www.gartner.com (accessed June 6, 2010).

Goto, K. Web ReDesign 2.0: Workflow that Works. New York: New Riders Publishing, 2004.

Gottlieb, S. CMS Selection - Death to the Features Matrix. 2009. http://www.cmswire.com/cms/web-cms/cms-selection-death-to-the-features-matrix-004211.php (accessed June 3, 2010).

—. How to choose an open-source CMS. 2006. http://www.zdnet.com.au/how-to-choose-an-open-source-cms-139234675.htm (accessed June 3, 2010).

Graham, P. Web 2.0. 2005. http://www.paulgraham.com/web20.html (accessed June 6, 2010).

Greenfield, M. Elevate Web Design at the University Level . 2009. http://www.markgr.com/elevate-web-design-at-the-university-level/ (accessed March 28, 2010).

—. Elevate Web Design at the University Level. 2009. http://cuwebd.ning.com/forum/topics/elevate-web-design-at-the (accessed March 12, 2010).

Gregory, A. http://www.sitepoint.com/blogs/2009/04/02/specialist-vs-generalist-who-wins/. 2009. http://www.sitepoint.com/blogs/2009/04/02/specialist-vs-generalist-who-wins/ (accessed June 6, 2010).

Gube, J. 10 Promising Content Management Systems. November 6, 2008. http://sixrevisions.com/web-applications/10-promising-content-management-systems/ (accessed 5 25, 2009).

Guseva, I. OmniUpdate Reports Strong 2009 with CMS for Higher Education. 2010. http://www.cmswire.com/cms/web-cms/omniupdate-reports-strong-2009-with-cms-for-higher-education-006551.php (accessed June 2, 2010).

Heller, S & Talarico L. Design School Confidential. Massachusetts: Rockport Publishers, 2009.

Heller, S. The Education of a Graphic Designer. New York: Allworth Press, 1998.

—. What this Country Needs is a Good Five-Year Design Program. 2004. http://www.aiga.org/content.cfm/what-this-country-needs-is-a-good-five-year-design-program (accessed June 5, 2010).

Hills, A, interview by L Scaife-Elliott. "CEO, MassMedia." A Business Prespective on CMS. Sydney, NSW, (May 20, 2010).

Houle, D. Why can't we be friends? Joomla! versus Drupal. 2007. http://www.communicopia.com/blog/Joomla!-versus-drupal (accessed March 20, 2010).

Hoyt, D. Wordpress as a CMS: Making Your Content Unbreakable . 2008. http://www.darrenhoyt.com/2008/10/21/wordpress-as-a-cms-making-your-content-

unbreakable/ (accessed May 27, 2010).

Grevet, A. Leadership Highlights from March 2012. http://magazine.joomla.org/issues/Issue-Apr-2012/item/736-Leadership-Highlights-from-March-2012 (accessed June 9, 2012)

Is Open Source Right for You?, 2007. Is Open Source Right for You? . 2007. http://www.trainingmag.com/msg/content_display/publications/e3i5adce9be1b1efb206c4d540a4194c9d3 (accessed September 22, 2009).

IVO. programming vs scripting. 2003. http://www.computing.net/answers/programming/programming-vs-scripting/5262.html (accessed June 6, 2010).

Jasper, E. Were We Raised to be Generalists? 2010. http://www.emilyjasper.com/gen-y/were-we-raised-to-be-generalists/ (accessed June 5, 2010).

Jensen-Inman, L. Elevate Web Design at the University Level. Janurary 20, 2009. http://www.alistapart.com/articles/elevate-web-design-at-the-university-level/ (accessed 4 23, 2010).

Johnston, M. CMSShowdown: Ultimate Showdown of Content Management Systems. 16 March 2009. http://www.cmscritic.com/cmsshowdown-ultimate-showdown-of-content-management-systems/ (accessed Janurary 26, 2010).

Kofahl, M. Selecting a Web CMS. Edited by M Smith. Sydney: CeBit, 2010.

LeBlanc, J. concrete5 – first impressions. 2009. http://www.designvsdevelop.com/concrete5-first-impressions/ (accessed November 23, 2009).

Lewis, B. Generalists vs specialists in the job market. 2006. http://www.infoworld.com/t/career-advice/generalists-vs-specialists-in-job-market-932 (accessed June 6, 2010).

Mang, Z. 5 Reasons Not to Use Wordpress As a CMS. August 27, 2006. http://bloghelper.is-there.net/5-reasons-not-to-use-wordpress-as-a-cms/ (accessed December 13, 2009).

Maruna, F. concrete5 vs. drupal . 2009. http://www.concrete5.org/community/forums/chat/concrete5_vs_drupal/ (accessed July 6, 2009).

Mischook, K. Scripting vs. programming: is there a difference? 2005. http://www.killersites.com/blog/2005/scripting-vs-programming-is-there-a-difference/ (accessed May 24, 2010).

Mischook, S. The Wordpress Web Design Strategy. 2010. http://www.killersites.com/blog/category/blogs-cms/ (accessed June 9, 2010).

Mosher, B. The Cost of Enterprise Content Management. 2008. http://www.cmswire.com/cms/enterprise-cms/the-cost-of-enterprise-content-management-003676.php (accessed May 28, 2010).

Muffatto, M. Open Source - A Multidisciplinary Approach. London: Imperial College Press, 2006.

My top 10 essential Wordpress Plug-ins, 2009. My top 10 essential Wordpress Plug-ins. 2009. http://handy-web-hints.blogspot.com/2009/11/my-top-10-essential-wordpress-plug-ins.html (accessed December 17, 2009).

Neto, M. Wordpress vs. ExpressionEngine: Apples and Oranges? 2010. http://www.webdesignerdepot.com/2010/03/wordpress-vs-expressionengine-apples-and-oranges/ (accessed April 23, 2010).

Niederst Robbins, J. Learning Web Design. Sebastopol: O'Reilly Media, 2007.

Niles, N. I'm a web designer I think? 2010. http://intuitivedesigns.net/blog/post/im-a-web-designer-i-think (accessed May 26, 2010).

Oblinger, J. and Oblinger, D. Educating the Net Generation. New York: Educause, 2005.

Opera, Learn to build a better Web with. Learn to build a better Web with Opera. 2010. http://www.opera.com/company/education/ (accessed June 6, 2010).

O'Reilly, T. What Is Web 2.0. 2005. http://oreilly.com/ (accessed May 16, 2010).

PacktPublishing. Wordpress Wins the 2009 Overall Best Open Source CMS Award. 2009. http://www.packtpub.com/news/wordpress-wins-2009-overall-best-open-source-cms-award (accessed June 6, 2010).

Peacock, M. Innovation in Education: Introducting iTunes University. 2007. http://www.cmswire.com/cms/web-content/innovation-in-education-introducting-itunes-

university-001926.php (accessed June 21, 2009).

—. Managing Web Content in Higher Education Needs Strategy, Resources. 2009. http://www.cmswire.com/cms/web-content/managing-web-content-in-higher-education-needs-strategy-resources-005274.php (accessed December 8, 2009).

—. TEDxNYED Examines Media and Technology in Education. 2010. http://www.cmswire.com/cms/web-publishing/tedxnyed-examines-media-and-technology-in-education-006884.php (accessed June 3, 2010).

Porter, J. Designing For The Social Web. Berkeley: New Riders, 2008.

Powlak, K, interview by L scaife-Elliott. "Mulitmedia lecturer." What do you teach? Sydney, NSW, (June 1, 2010).

Pressman, R. S and Lowe, D. Web engineering : a practitioner's approach. Pressman & Associates, Inc., 2009.

Raymond, E. The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionar. Sebastopol: O'Reilly & Associates, Inc, 2001.

Riley, J. InterACT With Web Standards. New York: New Riders, 2009.

Ruby, B. Drupal and Joomla! comparison. 2006. http://cmsreport.com/content/2006/12/drupal-and-Joomla!-comparison (accessed November 1, 2008).

Sagecircle. Vendor complains in a very public blog post about Gartner's Data Integration Magic Quadrant. 2008. http://sagecircle.wordpress.com/2008/12/29/vendor-complains-in-a-very-public-blog-post-about-gartners-data-integration-magic-quadrant/ (accessed June 6, 2010).

Schawbel, D. In 2009 Become a Generalist AND a Specialist to Keep Your Job. 2009. http://personalbrandingblog.wordpress.com/2009/01/02/in-2009-become-a-generalist-and-a-specialist-to-keep-your-job/ (accessed June 5, 2010).

Seth, A. Drupal vs. Joomla!: a frank comparison from an IBM consultant. 2009. http://www.topnotchthemes.com/blog/090224/drupal-vs-Joomla!-frank-comparison-ibm-consultant (accessed March 12, 2010).

Sharma, P. Core Characteristics of Web 2.0 Services. 2008. http://www.techpluto.com/web-20-services/ (accessed May 9, 2010).

—. Core Characteristics of Web 2.0 Services. 2008. http://www.techpluto.com/web-20-services/ (accessed June 6, 2010).

Shepherd, R. Wordpress 3.0 for Web Designers. 2010. http://carsonified.com/blog/design/wordpress-3-0-for-web-designers/ (accessed June 13, 2010).

Showdown, CMS. CMS Showdown. 2009. http://www.cmsshowdown.com (accessed June 6, 2010).

Spencer, L. Do You Need to Specialize to Succeed? 2009. http://freelancefolder.com/freelance-specialist-vs-generalist/ (accessed March 26, 2010).

Spool, J. Ideal UX Team Makeup: Specialists, Generalists, or Compartmentalists. 2008. http://www.uie.com/articles/ideal_UX_team/ (accessed March 15, 2010).

Stansberry, G. 10 Weblog Engines Reviewed. August 29, 2008. http://www.smashingmagazine.com/2008/08/29/10-weblog-engines-reviewed/ (accessed 3 12, 2009).

Stiffler-Dean, T. Wordpress vs Joomla! vs Drupal. 2009. http://www.goodwebpractices.com/other/wordpress-vs-Joomla!-vs-drupal.html (accessed March 11, 2010).

Suh, P., Ellis, J and Thiemecke D. Content Management Systems. Peer Information, 2002.

Ten Simple Rules for Choosing the Perfect CMS + Excellent Options, 2009. Ten Simple Rules for Choosing the Perfect CMS + Excellent Options. 2009. http://www.noupe.com/php/choosing-cms-tips.html (accessed March 27, 2010).

The History and Evolution of Social Media, 2009. The History and Evolution of Social Media. 2009. http://www.webdesignerdepot.com/2009/10/the-history-and-evolution-of-social-media/ (accessed April 12, 2010).

Think Tank. Research Specialists vs. Research Generalists. n.d. http://www.forethought.com.au/thinktank/news/research-specialists-vs-research-generalists (accessed June 6, 2010).

Top 5 Content Management Systems for the Complete Beginner, 2009. Top 5 Content Management Systems for the Complete Beginner. 2009. http://www.cmscritic.com/top-5-content-management-systems-for-the-complete-beginner/ (accessed July 16, 2009).

Tucker, L. Web Design Education Sucks. 2009. http://boagworld.com/design/web-design-education-sucks (accessed November 2, 2009).

Urquart, C. (2013), Grounded Theory for Qualitative Research: A Practical Guide, Sage Publications, New Delhi

W3C. W3C Semantic Web. 2010. http://www.w3.org/2001/sw/SW-FAQ (accessed May 16, 2010).

Walter, A. Brighter Horizons for Web Education. Janurary 20, 2009. http://www.alistapart.com/articles/brighter-horizons-for-web-education (accessed 4 18, 2010).

Water & Stone. Open Source CMS Market Share. CMS Wire, 2009.

—. "Open Source CMS Market Share." 2009. (accessed 2009).

Webopedia. Content Management Systems. 2010. http://www.webopedia.com/TERM/c/content_management_system.html (accessed June 7, 2010).

web site. http://en.wikipedia.org/wiki/Web site. Web site. http://en.wikipedia.org/wiki/Web site (accessed May 8, 2010).

Weitzman, L. Using open source software to design, develop, and deploy a collaborative Web site. 2006. http://www.ibm.com/developerworks/ibm/library/i-osource1/index.html?ca=drs- (accessed November 28, 2009).

What blog software reigns supreme?, 2009. What blog software reigns supreme? 2009. http://www.cmscritic.com/what-blog-software-reigns-supreme/ (accessed March 11, 2010).

White, M. The Content Management Handbook. London: Facet Publishing, 2005.

Wikipedia. Magic Quadrant. 2010. http://en.wikipedia.org/wiki/Magic_Quadran (accessed June 6, 2010).

—. System Configuration. 2010. http://en.wikipedia.org/wiki/System_Configuration (accessed June 6, 2010).

—. Web 2.0. 2010. http://en.wikipedia.org/wiki/Web_2.0 (accessed June 6, 2010).

Wong, K. Real-World Learning. n.d. http://www.edtechmag.com/higher/june-july-2010/real-world-learning.html (accessed June 10, 2010).

Wordpress. About. 2010. http://wordpress.org/about/ (accessed June 6, 2010).

Wordpress CMS Founder, Matt Mullenweg Interview, 2010. Wordpress CMS Founder, Matt Mullenweg Interview. 2009. http://www.cmscritic.com/wordpress-cms-founder-matt-mullenweg-interview/ (accessed Janurary 22, 2010).

Wordpress. Stastics. 2010. http://en.wordpress.com/stats/ (accessed June 6, 2010).

Worthington, K. 18 Essential Wordpress Plugins. 2010. http://www.kevinworthington.com/18-essential-wordpress-plugins/ (accessed June 6, 2010).