Open Source Content Management Systems: An Argumentative Approach

Dimitrios Michelinakis

A report submitted for the award of
MSc Electronic Business Management

The University of Warwick
Warwick Manufacturing Group

August, 2004

Abstract

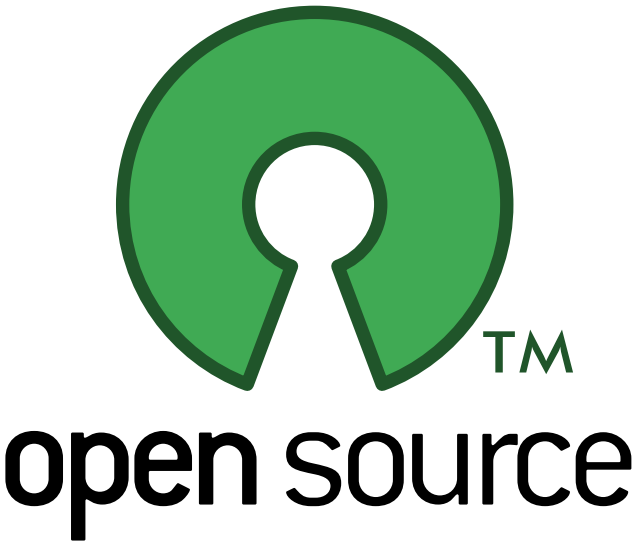
Businesses currently face the daily challenge of managing content efficiently. These businesses are being flooded with information from web Content Management Systems (CMS) that present an all-too-simple picture. Instead, content management systems should solve the problem of turning content into information and information into knowledge.

Content Management Systems are not just a product or a technology. CMS is defined as a generic term which refers to a wide range of processes that underpin the "next-generation" of medium to large-scale websites. Content management is a process which deals with the creation, storage, modification, retrieval and display of data or content.

This report evaluates seven open source CMS products. The comparison is based on eight categories as seen from a business perspective. These categories are; applications, data repository, deployment, integration, revision control, user interface, user management and workflow. Each category is scored from 0 to 10 points and the overall score is determined based on the average of all categories.

The comparison clearly shows how most CMS products require further development prior to being used within a commercial environment. The few CMS products which are ready for commercial deployment contain an inherent design flaw. This flaw refers to the inefficient management of large-scale user databases.

Businesses are currently seeking alternative methods to improve their services and Open Source Software (OSS) is one such method. This will require OSS authors to consider the implications of running their software within commercial environments and accommodate business requirements. A CMS product which follows these rules will be commercially sustainable.


TM
open source

Contents

List of Tables

List of Figures

Acknowledgements

This report is dedicated to my family, for their unwavering trust and support over the years. I would also like to thank my supervisor Peter Lever for his support and for being an all-round nice guy.

Declaration

I declare that this report represents my own work except where otherwise stated.

Chapter 1

Introduction

1.1 Background

The past decade has seen a rigorous change in the way we understand and use Information Technology within a business context. Advancements in the field of research and development has led to technologies such as; distributed computing, content management, data mining and processing, all of which fulfil a range of business needs. The move from localised computing platforms to distributed web technologies has been caused by, among other factors, the take-up of commodity computer and network components based on faster hardware and sophisticated software (Roure, Baker, Jennings & Shadbolt 2003).

Although the term "content management" is relatively new, content management practises have been around since the late 1980's and is derived from existing techniques. Even though the term itself wasn't established until the corporate business sector started using it, the field of content management was already being practised through other fields of management within businesses.

The timeframe between late 1995 and 1996 was the one off the most aggressive periods of growth in the internet. The enormous market potential was evident to everyone in the Information Technology sector. Companies scrambled to build websites in order to compete in the emerging market of the internet, which meant that they had to harness their existing resources and combine them with new internet technologies. As a result, the need for content management became evident (Nakano 2001).

Content management can be a great advantage for businesses, especially e-businesses. This process enhances e-business technologies and helps maintain an electronic presence over the internet while playing a key role as part of the solution to information and data overload. It is also the key to understanding information, thus adding substance and value to data (Boiko 2001).

1.2 About the project

The objective of this report is to analyse and compare a specific category of open source content management systems, within the context of small to medium businesses; this specific category is about web portals. The project aims to create a comprehensive comparison which deals with the specific requirements of small to medium businesses only. Thus, providing a clear understanding of the current trends within the commercial sector and the open source community.

This report involves a comparison between existing open source, web portal content management systems. The comparison is based on a set of business requirements which represent the needs of small to medium businesses, which aims to find an open source solution as an alternative to commercial solutions. Due to the nature of this subject, this report assumes that the reader has some understanding about Information Technology.

1.3 How this report is organised

This report involves the following six chapters.

Chapter 2 is about content and the content management, how it relates to Information Technology and why content is an asset to businesses.

Chapter 3 is about open source licenses, how they affect small to medium businesses and why they can be an advantage over commercial licensing.

Chapter 4 describes the different categories of content management systems and defines the structure of the comparison.

Chapter 5 compares seven web portals based on the rules defined in chapter 4 and summarises the results.

Chapter 6 is the conclusion which talks about the achievements of the comparison and future work.

Figure 1.1: Structure of this report

Chapter 2

Content Management

2.1 Content from a business perspective

Computers where initially created to perform time-consuming or complex mathematical computations and in many ways replace human labour. Boiko (2001) describes the computer model as follows: "If you can reduce a problem to a series of simple mechanical operations on numbers and logical entities (entities that are either true of false), it is amenable to solution by a computer".

At their lowest level, computers process data. The data processed by computers at a low level is not immediately readable or understandable, because it is made to be understood by the computer only. This data is used to perform a set of operations as described above.

The fact that computers are data-processing machines makes it hard to process content, which by definition is not just data. Technology has evolved over the years and computers are now required to perform computations on content while retaining their human meaning. Thus, they are able to offer

content such as; books, radio, TV, films and communications such as email (Boiko 2001).

Content, similar to data, is also perceived as information. Although, content retains human meaning which may be irrelevant to the computer. So when does raw data become content? Once data is given a usable form it also receives value. This value is based on its form and intended application, recognition and uniqueness. Thus, the data used to create information is not content because it has no value. Once that data is given value, it becomes content. For example, data stored in a corporate data centre may not be of immediate use or hold any particular value to the business. This is true for random or historical data, which may not seem valuable to the human reader. Once data mining techniques are used on the data, relevant and valuable content can be collected.

The most important characteristic of content is the human aspect. Content carries an identification or meaning which is understandable by humans, something intuitive that makes content impossible to process via a computer. Although data has no immediate meaning to humans, content does have meaning and conveys information. Information is conceived or understood by the humans that view the data. Thus, information is subject to connotation, context and interpretation, making it impossible to process as data (Boiko 2001).

2.2 Metadata - encapsulation of content

In order to process content while maintaining its human understanding and interpretation, a wide range of encapsulation methods have been developed. These methods try to encapsulate content and information along with the human aspect, in a way that is possible to process as data. The computer may not understand the finer meaning behind the content that it is processing, but

processing such content will produce the required results for the human user. Those results will contain an abstract meaning that can only be interpreted by a human user.

Defining data with information and making it into content is a process similar to the operations performed in every day situations. For example, searching for a book in a library or finding a movie in a video store. Both operations have the similarity of providing information about other information. A library, offers a computerised search engine that searches through categories of "author" and "title", while the video store may search for "actor" and "year of release". Therefore, a room full of books may be seen as a pile of data, while the same room with a categorised search engine may be seen as real content. The books become more than just data, because they have been given a description.

The method of content description is called metadata. Metadata is data about data, which defines the human aspect of content. Metadata first appeared on the web when the immense amount of data over the internet became impossible to process or to even understand. Some of the leading technologies and standards on metadata are seen in table 2.1.

Metadata technologies are themselves based on published internet standards. This method of creating a new standard based on another existing standard is very useful within businesses in order to make the exchange of content as smooth as possible. The leading standard technologies are eXtensible Markup Language (XML), which defines the Resource Description Framework (RDF) syntax as recommended by the W3C. Building on top of metadata and XML, are a number of advanced technologies and projects; peer-to-peer networking, the Semantic Web, Grid computing and Hypermedia.

Technology	Description
Data Documentation Initiative (DDI)	Standard suitable for social and behavioural sciences URL: http://www.icpsr.umich.edu/DDI/
Dublin Core	General purpose 15-element standard URL: http://www.dublincore.org/
Encoded Archival Description (EAD)	Standard for encoding archival finding aids URL: http://www.loc.gov/ead/
Federal Geographic Data Committee (FGDC)	Definitions for digital geospatial data URL: http://www.fgdc.gov/
Instructional Management Systems (IMS)	Definition of learning materials in learning systems URL: http://www.imsglobal.org
Metadata Encoding and Transmission Standard (METS)	Management & exchange of digital library objects URL: http://www.loc.gov/standards/mets/
ONline Information eXchange (ONIX)	International standard for book, serial and video products URL: http://www.editeur.org/onix.html
Sharable Content Object Reference Model (SCORM)	Reference model standard for learning objects URL: http://www.adlnet.org/
Text Encoding Initiative (TEI)	Physical and logical structure of textual material URL: http://www.tei-c.org/
Visual Resources Association (VRA)	Describes works of visual culture, including images URL: http://www.vraweb.org/vracore3.htm

Table 2.1: Metadata technologies and areas of application

2.2.1 Peer-to-Peer

Peer-to-Peer (P2P) technologies are hard to define, since they are widely used throughout the internet for different purposes. Though one clear characteristic makes P2P technologies stand out from any similar internet technologies; P2P technologies enable two systems to communicate with each other without any requirements for a server, thus enabling the client to offer server-based services (Michelinakis 2003).

P2P technologies have introduced the concept of a servent. The network is shared among systems, without any traditional servers. Instead, P2P clients

offer services previously offered by the traditional servers only. Communication between servents is based on open standard protocols which use content management for indexing and identification. P2P protocols also use metadata to transport humanly understandable meaning along with the data.

2.2.2 The Semantic Web

The Semantic Web is an improved version of current internet technologies. It provides a common framework that allows content to be shared and reused across different applications, hardware and organisations. The Semantic Web is defined by Berners-Lee, Hendler & Lassila (2001) as "an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation". Technologies such as RDF and XML are the driving force behind the Semantic Web. Following in their steps is the Semantic Grid which combines the Grid with Semantic technologies.

2.2.3 Grid Computing

Grid computing is seen as the "Grid problem", defined by Foster, Kesselman & Tuecke (2001) as "flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources". Once computers became interconnected and started communicating with each other, computers effectively became distributed systems.

Distributed systems have been around for a long time and there has been a lot on their design, implementation and deployment. On top of distributed technologies, researcher groups have implemented middleware, libraries and tools for wide-area distributed computing. In effect, they allow geographically distributed resources to act as a single powerful platform which supports parallel and distributed applications (Roure et al. 2003).

2.2.4 Hypermedia

Hypertext is text based documents linked with each other, as was the original web. Hypermedia is an extension of hypertext, which includes a combination of text, video, images, sound, plain text hyperlinks, miscellaneous multimedia content and other elements. Hypermedia tries to map the human way of thinking into documents, by allowing the user to make associations between different topics instead of browsing a single category at a time. During the late 1980's two major hypermedia categories had been recognised; Open Hypermedia Systems and Adaptive Hypermedia Systems.

Open Hypermedia Systems (OHS) separate links from documents, which allows the hyperstructure to be processed separately from the media and format it relates to. In order to combine different OH systems and help with their interoperability, the Open Hypermedia Protocol (OHP) was also developed, on top of existing OH systems (Bailey, Hall, Millard & Weal 2002). Commercial OH systems include; Microcosm (Fountain, Hall, Heath & Davis 1990), Chimera (Anderson, Taylor & Whitehead 1994) and DHM (Grønbæk & Trigg 1994).

Adaptive Hypermedia (AH) systems assist the user in knowledge acquisition. AH systems are based on existing technologies such as; Artificial Intelligence (AI), Intelligent Tutoring Systems (ITS) and User Modelling. AH systems include client-side adaptive systems that follow the user while browsing the web (Bailey et al. 2002). Some AH systems include; WebMate (Chen & Sycara 1998), Letizia (Lieberman 1995) and LiveInfo (Maglio & Farrell 2000).

2.3 Content management & web assets

Content management refers to the principles and practises for the development, management, maintenance and deployment of content within a single

organisation or across multiple organisations. It combines rules, business and manufacturing processes as well as workflows. Content management also provides either centralised or decentralised access to webmasters and web developers, as specified within a business framework or requirements procedures (Nakano 2001).

Content management is a topic which covers a wide range of areas within a business. This report specifically covers the web aspect of a business, which deals with web-related content as well as representation of that content over the web. From a business perspective content is seen as asset to the business. Thus, content management is about managing web assets.

Web assets within a business extract the content and logic of operations from raw data. Content management over the web unifies previously separate efforts within the business. For example, marketing and product information where previously the province of the marketing department within the business, which produces assets such as price lists and brochures. Another department, like the Information Technology department, maintains supply-chain information and order lists. Content management of web assets brings the two departments together within a web-based integrated system (Nakano 2001).

As a result of consolidating web assets within an integrated system, the business is able to respond to a dynamic market, while providing fresh content and updated service offerings. Fresh content is a product of experimentation and iteration at all levels. Due to the fact that web technologies are fairly new within businesses, the possibilities for improvement are not clear at first.

Content management can be seen as a the sum of contributors working on it, these include; developers, artists, marketers and others. Within a business, they can be full-time employees, contractors, outside vendors and others (Nakano 2001). The following sections identify a set of “best practises” of asset management and web development.

2.3.1 Revision control

Revision control, also known as website versioning, falls under the Software Configuration Management (SCM) category of tools and plays an integral role in medium to large scale projects. It is vital to keep track of changes, while marking known milestones and working versions of the project. Revision control helps by giving the developer the chance to roll back changes to a working snapshot of the project. Everyday development usually affects a small number of items or assets within the website. Thus, when a problem surfaces, the developer can compare different versions of those parts or assets in order to understand the problem (Nakano 2001).

Revision control is not only helpful in medium to large scale projects. Individual developers can keep track of their own changes, mark their progress and maintain a reliable and documented approach to software or web development. In addition, many of the GUI versioning tools provide a visual comparison which helps the developer understand changes. Several revision control implementation tools exist, which are widely used in software engineering and web development, some of them are listed in table 2.2.

2.3.2 Concurrent changes management

Project completion skew occurs once the team has grown into a substantial number of developers, at which point they are all working on different parts of the project, possibly in small groups. These small groups usually work on diverse activities separated from each other or sometimes in conjunction. As a result, each group will be developing, integrating and testing their work separately, before committing their work into the complete project. These groups will also be working under different schedules. This implies that a group may be starting its work while another is getting ready to commit theirs (Nakano 2001).

Tool	License	URL*
Aegis	GPL	http://aegis.sourceforge.net/
Arch	GPL	http://gnuarch.org/
CVS	GPL	https://www.cvshome.org/
Monotone	GPL	http://www.venge.net/monotone/
OpenCM	GPL	http://www.opencm.org/
Vesta	LGPL	http://www.vestasys.org/
SVK	Perl artistic	http://svk.elixus.org/
Subversion	Apache-style	http://subversion.tigris.org/
BitKeeper	Commercial	http://www.bitkeeper.com/
Code Co-op	Commercial	http://www.relisoft.com/co_op/
Synergy	Commercial	http://www.telelogic.com/
Perforce	Commercial	http://www.perforce.com/
		*last access date: 20-4-2004

Table 2.2: Revision control tools

Software and web development of medium to large projects require a set of "best practises" or methodologies which minimise conflicts with the developers. Concurrent changes management are the methodologies which deal with these types of conflicts.

Concurrent changes management, another part of the SCM category of tools, is usually performed by the versioning tools described in section 2.3.1. Concurrent changes are very frequent within a group of developers. One or more developers may try to modify a particular asset which is being modified by another developer. Once both parties involved commit their changes, some modified parts may get overwritten or reverted back to their original state.

Revision control tools support the management of concurrent changes by keeping track of the developers and the assets they are working on. This method is often called "watching" and the developers involved are usually called "watchers". The tool itself will not allow multiple developers to work

on the same asset and will notify all developers involved about the conflict. The developers may still choose to modify the same asset once they know which particular parts they will be working on.

2.3.3 Deployment

Deployment in software engineering is usually the process of submitting the final product to the customer. It includes subprocesses like installation of the software, on-site testing, training and finally running the system in a live environment. Web development is the process of committing the new assets of the website from the development system into the production server, also known as a "live server". Deployment tools and infrastructure copy or move assets into the desired place within the production server, at the appropriate time without conflicts. This gives the developers the opportunity to revert the production server to its original state (Nakano 2001).

To avoid conflicts and misunderstandings between the developers, a clear methodology must be defined. This methodology first identifies a single person or a set of persons who are the only ones allowed to commit changes to the production server. Finally, the developers agree upon the assessment and approval of assets for submission. This process is usually written in a "release agreement" which is understood and followed by all the developers involved in the project.

2.3.4 Workflow

Workflow is a collaboration process used within businesses to develop and maintain business assets. Workflow processes are important in businesses where time is a significant factor during the development process as well as when patterns of interaction within processes are repeated frequently. Workflow improves productivity by optimising processes, minimising wait

time between successive steps and by automating the core business logic of the organisation (Nakano 2001).

Workflow in business processes has the ability to automate routing of information, review and approval of jobs. Another ability is to enforce a formal business process which can be used throughout the lifetime of the project and reused in other projects with slight modifications. Workflow is essential when projects span different departments within the business. A unified specification defined as a formal business process can improve the productivity of the business by minimising wasted time and communication problems.

Chapter 3

Open Source Software

3.1 Free software

This report deals with certain types of free software; open source content management systems. Therefore, it is very important to define the term free software, because the concept itself is ambiguous. A wide range of software is distributed as "free" because it does not cost anything to download or use. However the source code is not made available or the software is distributed with a restrictive license. Binary or source code distributions could be copyrighted and covered by a license agreement, which could hold a range of few to extreme restrictions, like a disclaimer of reliability.

> "Free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech," not as in "free beer".
>
> – Free Software Foundation (FSF 2004)

- The freedom to run the program, for any purpose.
- The freedom to study how the program works, and adapt it to your needs. Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbour.
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. Access to the source code is a precondition for this.

Figure 3.1: Requirements of free software

Restrictions on these 'free' software come with licenses which; prohibit its use or require a fee for commercial user, prohibit or limit redistribution, including redistributing modified versions. Some licenses also require redistribution of derived works to use the same license as the original product or even release the modified source code. A few licenses also discriminate against individuals or groups.

The term free software is widely used in the Information Technology industry. However, its ambiguity hampers communication due to arguments over whether a particular piece of software is 'free' or not (OSI 2004). Figure 3.1 lists the rules which define the term "free software" as published by the FSF (2004).

3.2 The Open Source Model

As seen in section 3.1, this report deals with free software and more specifically with open source software, also known as OSS, which deal with content management. Based on the clear definition of free software it is now possible to clearly define what open source software is, what it means for businesses and how open source software can be used effectively within a commercial environment.

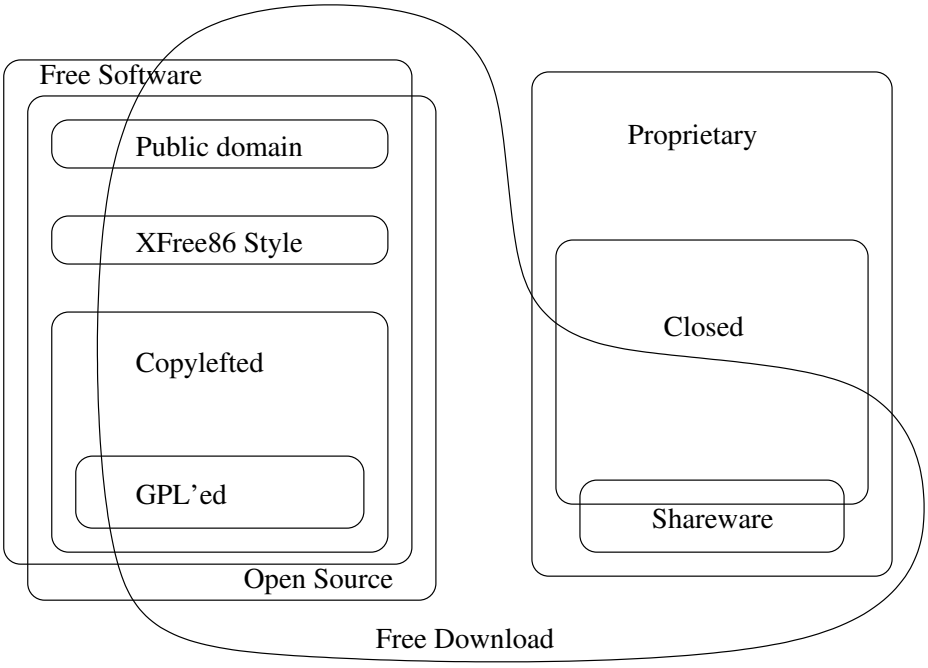


Figure 3.2: License categories

Although open source software, by definition, means the source code is publicly available, it also means the source code is distributed under a license, which falls under the criteria imposed by the OSI (2004). Once a piece of software is distributed with its source code and an OSI approved license, it is then accepted as open source software. Distribution of the source code is not an absolute requirement, it depends on the type of open source license.

Table 3.2 shows the different types of licenses and how they exist as "free" software (Chao-Kuei 2004).

Open source software is sometimes perceived as "public domain". This is a common misconception because public domain software is unlicensed. Open source software is copyrighted and comes with a license, whereas public domain software have their copyrights released by the author and distributed without a license.

Public domain software can be re-licensed by anyone, which removes it from the public domain, or re-branded under a different author (Perens 1997).

Although open source software is widely regarded as "free" for all uses and purposes, however some open source software is restrictive. There are a wide range of OSI approved licenses which may pose various restrictions on the source code. Still, OSI approved licenses are much more "open" than other 3rd party licenses due to the fact that OSI upholds strict guidelines for approving a license. Sections 3.2.1 to 3.2.10 are the ten basic criteria for OSI approved licenses. Followed by section 3.3 which gives examples of open source software and their licenses.

3.2.1 Free redistribution

"The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale" (OSI 2004).

By ensuring free redistribution, open source software is not hampered by short-term gains which would affect real long-term sales from customised versions of the software or contracted support and maintenance. Thus, a supplier may generate copies of the software and sell them or give them away without paying anyone for that privilege. As a result, many open source software can be bought on CD or DVD by paying for the cost of the medium only, since the supplier is not adding any extra costs.

3.2.2 Source code

"The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is

not distributed with source code, there must be a well-publicised means of obtaining the source code for no more than a reasonable reproduction cost-preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed" (OSI 2004).

To evolve and expand open source software, the source code must be available and in a modifiable state. The original or modified source code is then provided along with the software and any derived works, in order to ensure future repair or modifications.

3.2.3 Derived works

"The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software" (OSI 2004).

Future software updates and maintenance of the distributed source code, as seen in section 3.2.2, has no real use if the modified software cannot be distributed. Therefore, the ability to simply modify the source code is not enough to support independent peer review and rapid evolutionary selection. Instead, it should be possible to redistribute the modified software along with the modified source code.

Redistributed software can use the same license terms as the original software. Although this is not a requirement to do so but an option at the hands of the distributor. This requirement means; a license may not allow re-licensing or modification of its terms, or may allow re-licensing and sub-licensing of derived works.

An example of restricted licensing is the GNU General Public License (GPL) and an example of unrestricted licensing is the Massachusetts Institute of Technology (MIT) license, which is also known as the Expat license. The GPL license is listed in appendix D and the MIT license is listed under appendix F.

3.2.4 Integrity of the author's source code

"The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software" (OSI 2004).

Open source licenses allow distribution of the source code as seen in section 3.2.2. In some cases the author may not want modified versions of the software to be distributed as an original copy. Therefore, a license may restrict source code from been distributed in modified form, but allow derived works to include patch files which modify the original source code at compile time. Patch files are usually text files generated by "diff" and applied by "patch" utility commands.

3.2.5 No discrimination against persons or groups

"The license must not discriminate against any person or group of persons" (OSI 2004).

Following in the steps of anti-discrimination laws, open source licenses do not enforce any discrimination against persons or groups of persons. Historically, the license provided by the Regents of the University of California, Berkeley,

would prohibit licensed software from been used by the police of South Africa. This restriction was based around the apartheid era, which at this moment no longer applies (Perens 1997). Open source licenses are prohibited from having discrimination restrictions, even commendable ones.

3.2.6 No discrimination against fields of endeavour

"The license must not restrict anyone from making use of the program in a specific field of endeavour. For example, it may not restrict the program from being used in a business, or from being used for genetic research" (OSI 2004).

Primarily, this clause does not allow open source licenses from preventing commercial uses of the licenses themselves or the software they protect. In addition, restrictions against fields of endeavour mean that software should be usable in an abortion clinic or by an anti-abortion organisation (Perens 1997).

3.2.7 Distribution of license

"The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties" (OSI 2004).

Open source licenses should not contain limitations or restrictions for closing the software by indirect means, like a non-disclosure agreement (NDA). Also, the license itself is considered automatic and no signature is required for its validity. Both parties are under the similar terms of Pacta Sunt Servanda, a basic principle of civil law and of international law.

3.2.8 License must not be specific to a product

"The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution" (OSI 2004).

An open source licensed software is not restricted to a particular Linux distribution or particular operating system. Software distributed with one distribution should remain free if moved to another distribution or operating system.

3.2.9 License must not restrict other software

"The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software" (OSI 2004).

To protect software distributors, open source software should not limit how the software is distributed. For example, a particular software which uses open source libraries does not inherit the license used by those libraries. In addition, the distribution of open source software may not be restricted from being distributed along with commercial software.

3.2.10 License must be technology-neutral

"No provision of the license may be predicated on any individual technology or style of interface" (OSI 2004).

This clause prevents licenses, which require an explicit gesture of assent in order to establish a contract between licensor and licensee, from using indirect means of limiting open source software. Especially, software distributed under "click-wrap" technologies must allow for re-distribution over non-web methods or distribution paths that do not support click-wrapping.

Click-wrap typically consists of a pop-up dialog with a license or a payment submission form. The user is required to indicate the acceptance of the terms and provisions before proceeding with any further downloads. Click-wrap agreements are usually presented in a dialog window with an "I Agree" button.

Non-GUI environments which do not support pop-up dialogs should not be restricted in any way. This provision should also prevent scams which request money for the distribution of free software.

3.3 Examples of Open Source Software

3.3.1 Apache

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards (Apache 2004). Apache is distributed under the Apache v2 license which is a permissive non-copyleft free software license with a few requirements that render it incompatible with the GPL license.

3.3.2 Compiere

Compiere is an ERP+CRM solution for Small-Medium Enterprises in the global marketplace covering all areas from customer management, supply chain and accounting (Compiere 2004). Compiere is distributed under the Compiere License which is based on the Mozilla Public License (MPL) Version 1.1 and the Netscape Public License (NPL). The MPL license is not GPL compatible though version 1.1 has a provision (section 13) that allows a program (or parts of it) to offer a choice of another license as well. If part of a program allows the GNU GPL as an alternate choice, or any other GPL-compatible license as an alternate choice, that part of the program has a GPL-compatible license.

3.3.3 FreeBSD

FreeBSD is an advanced operating system for x86 compatible, AMD64, Alpha, IA-64, PC-98 and UltraSPARC® architectures. It is derived from BSD, the version of UNIX® developed at the University of California, Berkeley (FreeBSD 2004). FreeBSD is distributed under the modified BSD license which is the original BSD license, modified by removal of the advertising clause. It is a simple, permissive non-copyleft free software license, compatible with the GPL license.

3.3.4 Linux

Linux is a clone of the operating system Unix, written from scratch by Linus Torvalds with assistance from a loosely-knit team of hackers across the Net. It aims towards POSIX and Single UNIX Specification compliance (Linux 2004). Linux is distributed under the GPL v2 license which is a free software license, listed in appendix D.

3.3.5 Mozilla

Mozilla is a project to continue Netscape Communicator as an open project. The project is maintained by employees of Netscape (now a division of AOL), RedHat, some other companies, as well as contributors from the community (MozillaFoundation 2004). Mozilla is distributed under the MPL v1.1 license.

3.3.6 Perl

Perl is a stable, cross platform programming language. It is used for mission critical projects in the public and private sectors (Perl 2004). Perl is distributed under dual-licenses, the Artistic and GPL licenses. The Artistic license is not a real open source license, though Perl is considered open source because it offers GPL as an alternative license.

3.3.7 PostgreSQL

PostgreSQL is an enhancement of the POSTGRES database management system, a next-generation DataBase Management System (DBMS) research prototype. While PostgreSQL retains the powerful data model and rich data types of POSTGRES, it replaces the PostQuel query language with an extended subset of SQL (PostgreSQL 2004). PostgreSQL is distributed under the modified BSD license.

3.4 The importance of licenses

As a commercial organisation, the sole purpose of a business is to generate profit. Open source software is advertised for its zero cost initial ownership

1. Public domain without any copyright (technically not a license).
2. Those that apply no restrictions on the distribution of derivative works (also known as GPL-incompatible).
3. Those that do apply such restrictions (also known as GPL-compatible).
4. Fully restricted with only a few rights under copyright unrestricted.

Figure 3.3: Types of open source licenses

due to its free nature. However, its commercial exploitation may be limited or even impossible due to a restricted license. It is important to understand the different open source licenses before choosing an open source solution for a commercial organisation.

Section 3.1 defines the term free software, while section 3.2 defines open source licenses. Based on those definitions, a clear and precise description can be given for the range of licenses used by content management systems.

In order to keep the subject within the topic of open source content management systems, this report will not go into the details behind copyright law or mention every single open source license. Instead, a simplified description will be given which covers all the major aspects of open source licenses.

Licenses can be categorised into four types, as seen in figure 3.3. Public domain software (first category) is free of all restrictions, since all rights under copyright having been granted to the general public.

Software which is GPL-incompatible (second category) retains its copyright but grants all rights under its copyright to the user and does not restrict the source code from being used by non-open source applications.

Software which is GPL-compatible (third category) retain their copyright, grant all rights under copyright to the user but apply at least one restriction; the redistribution of the software, whether modified or unmodified, must

be under the same license. Proprietary licensed software (fourth category) retain their copyright and only grant a few rights under copyright, usually only the rights to perform and display.

GPL-compatible licenses can be a major problem from a business perspective. GPL-compatible licenses require that all software that link against GPL-compatible licensed code must also be licensed under a GPL-compatible license. This requirement does not exist in the LGPL license. Running GPL-compatible licensed software in a commercial environment can be a problem.

From a business perspective, the safest type of license is a GPL-incompatible license. Public domain software has no copyright and its origin, if unknown, could be a problem from a security point of view; for example, the code could contain malicious content. GPL-compatible software may not be commercially exploitable due to the restriction on redistribution or even its use within a commercial product.

The business may not be able to re-license the software in order to offer it as a commercial product. Proprietary licensed software may be too expensive and unmodifiable since its source code may not be available. Even if it is, its restricted license makes it impossible to exploit commercially.

Open source software which is GPL-incompatible has the following properties; its source code can be modified, it can be included within a commercial product, it can be relicensed and its redistribution is unrestricted. If the business has no plans to redistribute the software and only use it "in-house" then GPL-compatible licensed software can be easily used without any restrictions. One example of an unrestrictive license is the BSD license, also known as the modified BSD license.

Chapter 4

Content Management Systems

4.1 CMS categories

Content Management Systems (CMS) are not just a product or a technology. CMS is a generic term which defines a wide range of processes which underpin the "next-generation" of medium to large-scale websites (Browning & Lowndes 2001). A content management process; creates, stores, modifies, retrieves and displays data, or content, as seen in chapter 2.

The applications of CMS cannot be clearly defined. Even though a CMS is range of processes and managed software, the boundaries of the CMS space are blurred. The area covered by CMS overlaps with a wide range of traditional software systems, as seen in figure 4.1. As a result of this overlap of functionality, an intranet groupware system or virtual learning system can easily be implemented via the same CMS (Browning & Lowndes 2001).

CMS have no single interface or implementation, they are effectively designed on the requirements of each business. The implementations of CMS differ from web based to integrated server-side applications. Popular web based

- Document management systems
- Knowledge management systems
- Enterprise application integration systems
- E-commerce solutions
- Web portals

Figure 4.1: CMS categories

implementations vary from PHP, Perl and Python. Integrated application server implementations use popular languages like Java 2 Enterprise Edition and C++. Figure 4.2 shows a visual interpretation of the structure of a typical CMS.

This report does not deal with the application or use of CMS, for example; document management or virtual learning. Instead, this report takes a comparative approach to web portals only, based on their functionality from a business perspective. Web portals are websites which act as a main "point of entry" for users. They offer a range of services, for example; news section, search engine and web catalogue. Web portals are CMS solutions which offer content over the web, thus they may seem limited in functionality over traditional applications. To the contrary, due to the pervasive nature of the internet, the web has become the preferred method for content delivery (Browning & Lowndes 2001).

4.2 Requirements & prerequisites

Although requirements on software packages vary between businesses, they still have certain common requirements. The objective of this report is to compare the widest possible selection of open source content management systems, which can be used by businesses. The most suitable CMS solutions

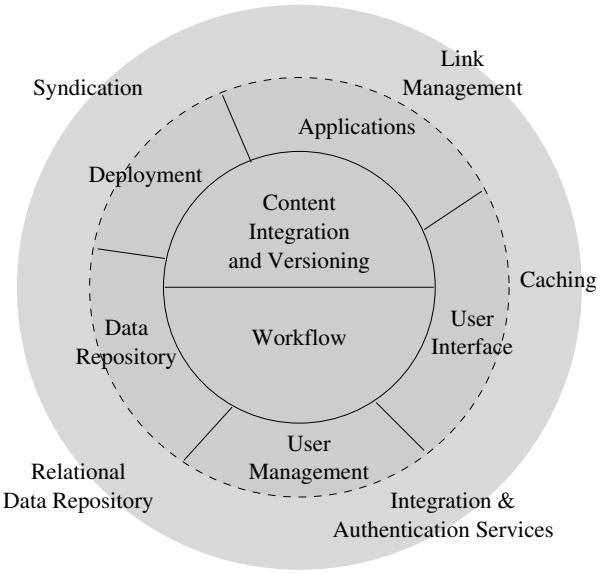


Figure 4.2: CMS feature onion

are selected based on a set of clearly defined requirements, all others have been rejected. Figure 4.3 lists these requirements.

All the systems compared within this report are required to be licensed by an OSI-approved open source license, as defined in chapter 3. Open source software is widely recognised for its standards compliance, which is vital for businesses. For example, creating a website which uses proprietary data structures will hinder future expansion to new systems or technologies due to incompatibilities. Open source software are more likely to follow standards like the W3C Extensible HyperText Markup Language (XHTML) or XML specifications while ensuring they can interact with each other. Commercial software tend to be incompatible with each other in order to keep the customer hooked to a specific technology or supplier.

Compatibility with the Apache HTTP server is vital. The Apache HTTP server is the most widely used web server on the internet. Netcraft (2004) reports that more than 67% of the websites on the internet are using Apache, with 4 million new hostnames growth in the first half of 2004. The Apache

1. Open source license
2. Compatible with the Apache HTTP server
3. Written in PHP, Perl or Python
4. Compatible with MySQL or PostgreSQL
5. Core technological features
6. In active development

Figure 4.3: CMS requirements

HTTP server is also an open source project of the Apache Software Foundation (ASF), which makes it ideal for businesses.

PHP, Perl and Python are some of the most popular scripting programing languages. PHP is a server-side scripting language, while Perl and Python are generic languages used for web development as other generic purposes. Other server-side scripting languages exist such as JSP, although PHP, Perl and Python are best suited for web development because they are open source, compatible with the Apache HTTP server and relatively lightweight in requirements. Application servers, like Java, will not be considered due to their nature; they can be system specific, have large resource requirements or high complexity compared to interpreted languages.

Database compatibility is vital for the business due to the importance of data storage. MySQL and PostgreSQL are two open source databases which are widely used in the Information Technology industry. Both support a wide range of features which make them suitable for all kinds of businesses, from small to large scale systems, especially since they are used in web hosting environments.

A core feature is the implementation of technologies which enable the system to perform certain processes. A CMS must have enough core features in order to perform the basic business needs. Even though the type of features can be

subjective and vary between businesses, this report will enforce the requirements for at least three; modular code, abstraction layer and customisable interface.

Modular code refers to the ability of the code to accept new features and modifications. This means modules or plug-ins can be developed in-house and used by the CMS without extra effort from the developers. The abstraction layer is the separation between the content and the design of the generated HyperText Markup Language (HTML) code, which allows the administrator to modify content without any knowledge of HTML. Finally, a customisable interface means themes can be developed to match the overall image of the business.

Active development is one of the most important features from a business perspective. A software package that looses its support and development would require the business to either scrap the package and move to another one or continue development in-house. Open source projects in active development also evolve much faster, due to the continued and fluid improvement from the community. Active development is deemed when a project has been modified in any way within the past three months; May, June and July 2004.

4.3 Rejected systems

Many CMS solutions did not meet the requirements defined in section 4.2. Their failure to meet the requirements does not mean these particular CMS solutions offer lower functionality, quality or support. Their rejection means that they are not suitable for small to medium businesses from a commercial perspective. Thus, the rejected systems can still be of value in certain situations.

The first requirement which splits all CMS in two is the requirement for an OSI-approved open source license as defined in chapter 3. Commercial solu-

tions also fail to meet requirements on web server and database compatibility. For example, Oracle Portals requires an Oracle iAS application server and an Oracle database, even though it runs under Apache.

Open source solutions which fail to meet requirements are usually due to their limited flexibility. Table B.1 from the appendix, shows the CMS solutions which are licensed under an open source license but failed to meet other requirements. For example, Plone is based on the Zope application framework, it runs on top of the Zope web server and the Zope build-in database (ZODB). Even though there is a customised interface to run Zope-based implementations under Apache and communicate with a third party relational database, it fails to meet expectations on compatibility.

One exception is PHP-Nuke. Even though it meets all the requirements, it has been rejected on the basis of its dubious future. The author has stated that the distribution license may change at any time, as a result of websites removing the copyright notice. On top of the obvious threat to the GPL license, the author requests payment for the latest release, which has to be payed for every new release. This hostile attitude towards the users makes it doubtful that businesses would want to consider it as an option.

4.4 Structure of the comparison

Each business has its own set of requirements for a CMS solution, which depend on various parameters such as; the size of the business, field of operation, type of managed data and target customers. It is highly unlikely that a single product will have all the required functionality. As a result, this report tries to identify potential products which are scalable and expandable. As a base for the comparison, eight areas of functionality will be analysed based on Browning & Lowndes (2001). The eight areas of functionality are listed in figure 4.4.

- Applications
- Data repository
- Deployment
- Integration
- Revision control
- User interface
- User management
- Workflow

Figure 4.4: Areas of functionality

Applications are about general functionality which compliment the entire CMS solution. Availability of the development API allows developers to expand the existing functionality and add custom processes per business requirements. Marketing and advertising features allow the website to display banner advertising or offer opt-in permission marketing forms. Localisation and multi-language support for all documents and processes. Time-based event functions, like scheduling. Site-wide searching engine which allows transparent searches over different content. Finally, e-commerce functionality which allows the system to perform online transactions.

Data repository is about flexibility in content storage. Information is an asset to every business, proper management of the data storage can be an advantage. A CMS solution may use multiple storage methods, including a Relational DataBase Management System (RDBMS) and Network File System (NFS) or other file system based storage. Apart from the storage medium, it is also important to use a standards compliant storage format such as XML. In particular, XML provides transformation services and content validation along with the split between content, format and business logic.

Deployment can be one of the most important features from a business per-

spective. Medium to large scale systems use multiple servers for fault tolerance and improved availability. A CMS product's ability to scale through multiple servers gives the extra advantage for reliability. Replication is also an issue, the flow of updates that go from testing into production should be able to replicate reliably and with roll back support, as discussed in section 2.3.3. Finally, multiple output formats can increase the target audience, for example; by providing mobile phone access via SMS or WAP.

Integration deals with the every day management of the system. Metadata management via content classification systems which enable arbitrary data to become useful information assets for the business. Information can be used along with 3rd party web applications, such as log analysers and spam filters. Data conversion, allows users to publish or submit data in different format from the one used to publish their data, for example PDF to HTML conversion.

Integration is also about compliance with the internet standards published by the W3C such as; HTML 4.01 and XHTML 1.0/1.1. Based on those standards are the requirements for compliance with the Disability Discrimination Act 1995 chapter 50 (HMSO 1995) which came into effect in the United Kingdom. This Act enforces new rules for websites to create content which is accessible by disabled people. Standards compliance means that a CMS product must be able to generate code that is compliant with the Web Accessibility Initiative guidelines.

Revision control, as seen in section 2.3.1, is about management of changes, while keeping track of known milestones and working versions of the entire web site development and content. Revision control allows users to keep track of changes, while protecting them against overlapping changes by other users. Roll back functionality gives the user a chance to return to a known working copy of data, which also makes it easy to compare changes over time.

User interface is not just about the client "visual" interface. The user interface is a collection of interface features which help the user or administrator

to effectively manage the system. Interface tools enhance the control of processes, some of these tools are; HTML forms, WYSIWYG content editor and document linking. The user interface should provide the choice between high and low level editing, either edit the code directly, or provide a suitable interface which generates the required code.

User management is about access and control of the system. The system should allow for 3rd party authentication, such as; SQL database, LDAP, NIS/YP, PAM. In addition, the interface should provide adequate user management control, for example; system-wide user modification.

Workflow is a collaboration process for the development and maintenance of business assets which involve steps such as; varied information types, cross-departmental staff and functions based on a submit/review/approve steps. As seen in section 2.3.4, workflow is important to clearly define processes which perform specific functions, with various dependencies between them. These functions automate routing of information, review and finally approve changes.

4.5 Scoring

Each CMS solution is marked for each category with a score, ranging from 0 to 10. A score of 0 means this particular category is not covered by the product or has no real effect from a business perspective. A score of 10 means this particular category is fully covered, the product has an excellent implementation which covers current technologies as well as the possibility for future expansion. Any scoring in between shows how accurately a product meets the category criteria.

An overall score is produced by averaging all the scores. The results are displayed in a table which shows a simple visual representation of the scoring for each category and each product, each point scored is represented by a tick

mark "√". Figures 4.5 and 4.6 list the eight categories of functionality, the associated main features and their score.

All categories are divided into two or more sections. Categories with two sections receive more weight than categories which contain more sections because the ten points of scoring are only divided in half. Weighting highlights categories which are more important than others.

For example, each section in the applications category scores for 1.6 points, which gives this category a higher chance to score more points, while the sections in the revision control category score 5 points each, which makes it easier for a CMS product to receive zero points for the entire category.

The categories data repository, deployment, revision control, user management and workflow are considered the most important from a business perspective. Integration comes next and is considered more important from a technological perspective. The applications category comes last in weighting since it is considered the least important because most businesses have the capacity to create their own set of applications and features.

Applications score 1.6 per item

- API
- Marketing/advertising
- Localisation/multi-language
- Time-based events, scheduling
- Site-wide searching
- E-commerce

Data repository score 5 per item

- Multiple storage methods
- XML storage

Deployment score 5 per item

- Scale, fault tolerance and replication
- Multiple output formats

Integration score 3.3 per item

- Metadata content classification
- Data conversion
- Standards compliance

Figure 4.5: Scoring analysis (categories 1 to 4)

Revision control score 5 per item

- Versioning
- Roll-back

User interface score 2.5 per item

- HTML forms
- WYSIWYG editor
- Document linking
- Different levels of editing

User management score 5 per item

- Ease of management
- 3rd party authentication

Workflow score 5 per item

- Routing of information
- Submit/review/approve steps

Figure 4.6: Scoring analysis (categories 5 to 8)

Chapter 5

CMS comparison

In order to form a suitable comparison of CMS solutions, selection criteria were defined in chapter 4. Based on those criteria, the list of CMS products as seen in the appendix B.1 has narrowed down to seven products, listed in table 5.1, along with their distribution license. The dominance of the GPL license is purely coincidental.

CMS	**License**	**URL***
Drupal	GPL	http://drupal.org
Mambo Open Source	GPL	http://www.mamboserver.com
phpWebSite	LGPL	http://phpwebsite.appstate.edu
PostNuke	GPL	http://www.postnuke.com
Typo3	GPL	http://typo3.org
Xaraya	GPL	http://www.xaraya.com
Xoops	GPL	http://www.xoops.org
		*last access date: 20-4-2004

Table 5.1: Content Management Systems

5.1 Drupal

Drupal	
Applications	√√√√√√√√
Data repository	√√√√√
Deployment	√√√√√
Integration	√√√√√√√
Revision control	√√√√√√√√√√
User interface	√√√√√√√√√√
User management	√√√√
Workflow	√√√√√
Overall score	6.7/10

Table 5.2: CMS: Drupal

Drupal is a structured CMS which manages many different types of content, including weblogs, discussion-based forums and project collaboration. Installation is simple and smooth, the administrator is required to follow the steps described in the INSTALL.txt; create and populate the database, modify the configuration file includes/conf.php and connect to the site URL to complete the process. Figure C.1 shows a sample screenshot.

In the applications category, Drupal offers most of the required functionality. A comprehensive API based on a modular approach to the Drupal implementation gives administrators the ability to develop their own functionality. Drupal includes banner management, localisation and multi-language support via a GUI or 3rd party tools like gettext, site-wide content searching and e-commerce support with shopping cart and transaction payments. It does not offer any time-based event functions, like scheduling (score 8/10).

Drupal offers complete database independence due to its database abstraction layer. Communication with any SQL relational database is possible

through the abstraction layer, for example with MySQL and PostgreSQL. Unfortunately, data storage is table-based with data being stored based on their type, so there is no XML storage (score 5/10).

In the deployment category, Drupal offers extended data in RDF and RSS formats. Although, it offers no fault-tolerant features for replication, beyond those offered by the database itself (score 5/10).

Drupal integrates very well with content. Add-on modules allow Drupal to generate PDF documents from node data, import list data in CSV format and parse XML data generated from an external source. Drupal adheres with W3C standards and generates XHTML 1.0 Strict compliant pages. Metadata content classification is not available in Drupal, thus content can not be classified this way, or imported from a metadata-compatible source (score 6.6/10).

Revision control is properly implemented in Drupal. Version control allows Drupal administrators to track changes, like who made the change, what date/time and what was changed. Roll-back capability complements version controls and allows content to roll-back to an earlier revision (score 10/10).

The user interface of Drupal offers proper abstraction from the content. HTML forms are supported by an add-on module while a WYSIWYG editor offers low level as well as high level content modifications. In addition, add-on modules allow of bbcodes and other visual enhancements. Document linking is provided in the form of context linking and "permalinks" (score 10/10).

User management is performed by the administrator user interface, which offers no features for medium to large scale user databases. However, Drupal does offer user searching and user grouping in the form of "roles". Support for 3rd party authentication mechanisms is limited to sources like other web sites or an LDAP server (score 4/10).

Workflow is not immediately evident. Content goes through a submit-review-approve process, for example during article and comment submission. Unfortunately, there is no routing of information process (score 5/10).

5.2 Mambo Open Source

Mambo Open Source	
Applications	√√√√√√√√
Data repository	√√
Deployment	√√√√√
Integration	√√√√√√√
Revision control	
User interface	√√√√√√√√√√
User management	√√√√
Workflow	√√√√√√√√√√
Overall score	5.7/10

Table 5.3: CMS: Mambo Open Source

Mambo Open Source is a community based open source project which delivers a general purpose web application framework. Installation of MamboOS is clearly the best among all the other CMS products discussed in this report; once the administrator has an SQL database ready and visit the site URL, the installation script takes over and completes the installation process. The entire checking of files, populating the SQL tables and performing tests is done without any input or direct access to files. Figure C.2 shows a sample screenshot.

In the applications category, MamboOS offers a complete API set for developers, including proper documentation and functional reference. Advertising

is covered by the advertising management which includes banners. Localisation is covered by multi-language support, as well as time offset and country locales. Time-based events are not inherently supported, although some content such as news articles, have proper start/end scheduled publishing. Site-wide search is supported for all modules. E-commerce is not supported as a core feature, although a 3rd party add-on can offer limited e-commerce functionality (score 8/10).

MamboOS only supports MySQL as a database back-end as of version 4.5, while the 4.6 version which was still under development during the writing of this report has support for more. XML storage is not supported and data is table-based, stored based on their type (score 2/10).

In the deployment category, MamboOS supports the RSS syndication format as a core feature, while XML data can be imported with 3rd party modules. Although, there is no fault-tolerant features for replication, beyond those offered by the database itself (score 5/10).

MamboOS offers most features for integration; basic metadata description and keywords but without any particular classification. Data conversion is supported in HTML, plain text and PDF formats, in particular MamboOS offers the ezPDF and FPDF modules which deal with document conversion. Unfortunately, there is no XHTML or HTML standards compliance (score 6.6/10).

Revision control is not available in MamboOS, there is no versioning of documents, multiple author support or roll-back of changes (score 0/10).

The user interface is structured based on a modular approach. HTML forms are supported and automatically generated, WYSIWYG editor allows low level as well as high level modification of content. Document linking is supported via "related links" which links documents based on their metadata (score 10/10).

phpWebSite	
Applications	√√√√√√√√√√
Data repository	√√√√√
Deployment	√√√√√
Integration	√√√√
Revision control	
User interface	√√√√√
User management	√√√√
Workflow	√√√√√
Overall score	5.5/10

Table 5.4: CMS: phpWebSite

User management is performed by the administrator user interface, which offers no features for medium to large scale user databases. A user search engine and user grouping in the form of "groups" is available. Support for 3rd party authentication mechanisms is limited to LDAP servers (score 4/10).

Workflow is closely integrated in MamboOS. Content approval is supported, a registered user can submit content, while the administrator console allows for content to be reviewed and approved. Routing of information is supported based on user groups, while registered users can submit content, different administrators can review it (score 10/10).

5.3 phpWebSite

PhpWebSite is an open source project sponsored by the Appalachian State University, USA. At first glance, it may seem like the target audience is universities, but closer inspection reveals a well defined general purpose framework, suitable for most business needs. Installation is fairly straight forward,

although it requires the administrator to do some manual work with permissions. Once the /setup/ page is loaded, all the options are available for modification and the website is running without any further work. Figure C.3 shows a sample screenshot.

In the applications category, phpWebSite offers a functional API which helps developers implement custom modules or modify existing features. Advertising is possible via banner managed "blocks", as well as Google sponsored advertising via an extra module. Multi-language support is available through the language module, while there are plans to move to GNU gettext Portable Object (PO) format in a future release. The PO standard is a text file format which helps programmers and translators at producing, updating and using translation files. Site-wide searching is supported for all modules which implement the search feature. Time-based events are supported in the announcement and calendar modules, though they are not available site-wide. E-commerce is also supported by a 3rd party module which offers shopping cart and payment features (score 10/10).

Data repository is covered by the pear DB module which is a database abstraction layer and supports all the popular databases, including MySQL and PostgreSQL. XML storage is not supported in any way, data is table-based as per database (score 5/10).

In the deployment category, phpWebSite supports the RSS syndication format via a 3rd party module. Although there are no fault-tolerant features for replication beyond those offered by the database itself (score 5/10).

PhpWebSite does not integrate very well with content. Metadata content classification is not supported and content can not be allocated a metadata description. Data conversions are not supported either, except when plain text is converted to HTML by the form editor. PhpWebSite is fully compliant with the W3C XHTML 1.0 transitional standard (score 4/10).

Revision control is not available in phpWebSite, there is no versioning of

documents, multiple author support or roll-back of changes (score 0/10).

The user interface offers automated HTML forms via the "Form Generator", as well as management for the content submitted via the forms. PhpWebSite does not offer a WYSIWYG editor, but allows the user to insert low level code which will render when viewing the content. Different levels of author editing are not supported, although document linking is supported via the "Link Manager" (score 5/10).

User management is performed by the "Users Administration" module. User management is static, which offers no features for medium to large scale user databases. However it offers grouping and per-user or per-group permissions via granulated administration. Authentication is either internal or via an external PHP function, LDAP or other authentication mechanisms are not directly supported (score 5/10).

Workflow is supported by the core functions of phpWebSite. Data submission supports a submit-review-approve step process, including a new user registration approval process. Routing of information is available via submissions that require approval and comment threads, which can be associated with arbitrary published content (score 10/10).

5.4 PostNuke

PostNuke is an open development CMS which started as a fork from PHP-Nuke, but managed to include many new enhancements and improvements over the original system. Installation of PostNuke is one of the quickest and easiest among the other CMS products, with the exception of Mambo Open Source. After unpacking the distribution archive, the administrator loads the /install.php which follows a step-by-step process. Figure C.4 shows a sample screenshot.

PostNuke	
Applications	√√√√√√√√
Data repository	√√
Deployment	√√√√√
Integration	√√√
Revision control	
User interface	√√√
User management	√√√√
Workflow	√√√√√
Overall score	3.7/10

Table 5.5: CMS: PostNuke

In the application category, PostNuke shows a unique independence from 3rd party applications. A set of API is provided for developers. Localisation per country is not supported, although multiple languages are supported. Site-wide searching is fully supported by all the different modules and "blocks" within PostNuke. Time-based events and scheduling is provided in a limited number of modules, for example; "stories" can have a scheduled publication date. E-commerce is not available as a core module and some 3rd party modules are distributed with a commercial license (score 8/10).

Data repository is lacking in features. MySQL is the only supported database due to some MySQL-specific function queries. XML storage is not supported (score 2/10).

In the deployment category, PostNuke offers RSS syndication as well as XML feeds which download weather information. There is no fault-tolerant features for replication, beyond those offered by the database itself (score 5/10)

PostNuke has no specific content integration. Data conversion is supported via the xPDF module, which generates PDF documents from news stories.

Standards compliance is not unified throughout the generated pages, some pages may generate XHTML 1.1 compliant code while others do not. Metadata content classification is not supported (score 3.3/10).

Revision control is not supported in PostNuke, there is no versioning of documents, multiple author support or roll-back of changes (score 0/10).

The user interface supports HTML content in submission forms, without any WYSIWYG features. Document linking is not supported, neither is multilevel editing (score 2.5/10).

User management is done over the web interface of PostNuke, which lacks features to support a large user base. External authentication is supported via custom modules, though LDAP seems to have some problems authenticating users (score 4/10).

Workflow is not directly supported by PostNuke. A submit and review process is supported in some modules, as well as the administrator publish/unpublish option. Routing of information is not supported (score 5/10).

5.5 Typo3

Typo3 is an open source CMS which aims for an enterprise-level functionality. Typo3 is not a system that is easily understand, the learning curve is very steep and requires a considerable more work to getting running that another other CMS. Figure C.5 shows a sample screenshot.

In the applications category, Typo3 succeeds in being one of the most feature-complete packages. Developers have a complete API. Advertising is controlled via banner management. Multiple languages are supported. Time-based events and scheduling is supported for all articles. Site-wide searching is supported via an “indexed engine”, which can also search through external

Typo3	
Applications	√√√√√√√√√√
Data repository	√√√√√√√
Deployment	√√√√√
Integration	√√√√√√√
Revision control	√√√√√√√√√√
User interface	√√√√√√√√√√
User management	√√
Workflow	√√√√√√√√√√
Overall score	7.6/10

Table 5.6: CMS: Typo3

media. E-commerce is fully supported via extensible modules (score 10/10).

Data repository lacks database independence. Typo3 only supports MySQL as a database back-end. XML storage is supported vi the XML data exchange (score 7/10).

In the deployment category, Typo3 offers complete data independence with support for; PDA formats, WML for mobile phones, XML data exchange, SGML for printing. Although, there is no fault tolerance or scalability (score 5/10).

Typo3 integrates well with content. Data conversion is supported with PDF as well as XML formats. Metadata categorisation is supported via the extensible modules. Unfortunately, Typo3 does not generate standard compliant code (score 6.6).

Revision control is fully supported by Typo3. All content is marked with a version and all changes are recorded as revision history. Roll-back is supported via unlimited undo levels (score 10/10).

The user interface supports HTML forms via a form generator. Typo3 offers a complete WYSIWYG editor, based on an RTE interface. Document linking is supported via the internal link management. Different levels of editing is supported via the extended user management which allows user groups to perform authoring tasks (score 10/10).

User management is performed via the web user interface, which is relatively complex to use for large user lists. Unfortunately, Typo3 does not support LDAP authentication. Although, a custom module may offer external authentication (score 2/10).

Workflow is fully supported via the "workflow engine" which offers approval of content and grouping of authors, editors and reviewers. Routing of information is provided on top of the workflow integration. In addition, Typo3 offers a staging system which splits the live webserver from the production back-end (score 10/10).

5.6 Xaraya

Xaraya is an open source application framework, which started as a fork of PostNuke and PHP-Nuke. The installation if Xaraya is web-based and only requires manually setting the permissions of the writable directories. Figure C.6 shows a sample screenshot.

In the application category, Xaraya offers a complete API set for developers. Multiple languages are supported by the core interface. Time-based events and scheduling is supported via the scheduler module. Site-wide searching is fully supported for all extensible modules. Unfortunately, there is no e-commerce support or any advertising management (score 6.4/10)

Data repository is offered via MySQL and PostgreSQL support only, without any extensible abstraction for other databases. XML storage is not supported

Xaraya	
Applications	√√√√√√
Data repository	√√√√√
Deployment	√√√√√
Integration	√√√√√√√
Revision control	
User interface	√√√√√
User management	√√√√√
Workflow	√√√√√√√√√√
Overall score	5.3/10

Table 5.7: CMS: Xaraya

and there are no 3rd party supplementary extensions (score 5/10).

In the deployment category, Xaraya offers full RSS syndication, as well as XML-RPC communication. Though, there is no scalability or fault tolerance beyond what the database offers (score 5/10).

Xaraya offers close integration with content. Metadata support is offered for plain documents as well as generated content. Xaraya generates XHTML 1.0 Strict standard compliant code, although the generation of the code depends on the current theme. Data conversion is not supported (score 6.6/10).

Revision control is not supported in Xaraya, there is no versioning of documents, multiple author support or roll-back of changes. There are plans to integrate BitKeeper in a future version (score 0/10).

The user interface offers complete document linking as well as different levels of editing via the administration web interface. Unfortunately, there is no support for HTML form management or a WYSIWYG editor. A Java-based editor is been planned for a future version (score 5/10).

User management is performed via the "roles" interface. It offers advanced features such as mass-recall, mass-purge and system-wide lock. Unfortunately, the interface does not accommodate for a large user base. External authentication is supported via an LDAP extension module (score 5/10).

Xaraya offers complete workflow integration, which is based on the Galaxia Workflow Engine. It offers definition of processes, monitoring and activity options. Routing of information is supported via the workflow module, which integrates with document submission modules (score 10/10).

5.7 Xoops

Xoops	
Applications	√√√√√√√√
Data repository	√√√√√
Deployment	√√√√√
Integration	√√√
Revision control	
User interface	√√√
User management	√√√√
Workflow	√√√√√
Overall score	4.1/10

Table 5.8: CMS: Xoops

Xoops is a dynamic object-oriented open source CMS. Xoops stands for eXtensible Object Oriented Portal System. Installation is fairly simple, because a web interface is used to perform the necessary steps required once file permissions have been set. Figure C.7 shows a sample screenshot.

In the applications category, Xoops offers a complete development API. Ban-

ner advertising is supported via a core module. Multi-language support also offers multi-byte compatibility, for languages like Japanese, Simplified and Traditional Chinese and others. Time-based scheduling is supported via scheduled publication and expiration of content. Site-wide search is supported for all modules. E-commerce is not directly supported, though there is work on some extended modules (score 8/10).

Data repository is limited to MySQL only, without any database abstraction. XML storage is not supported, although there are XML based modules. For example, the "Integrity Checker" which uses XML to verify the integrity of files (score 5/10).

In the deployment category, Xoops offers multiple output formats such as RSS syndication. Unfortunately, there is no support for scalability or fault tolerance beyond the replication offered by the database (score 5/10).

Xoops integrates well with content via the extensible modules. Metadata are supported via XHTML meta tags. Xoops has no support for data conversion and does not compliant with HTML or XHTML standards (score 3.3/10).

Revision control is not supported in Xoops, there is no versioning of documents, multiple author support or roll-back of changes (score 0/10).

The user interface offers simple editing, without HTML forms or a WYSIWYG interface. Document linking is supported via the "Links" module. Multiple levels editing is not supported (score 2.5/10).

User management is performed via the web interface, which is not suitable for large scale user databases. There are no extra features to run mass checks. External authentication is supported via LDAP (score 4/10).

Workflow is integrated within the core modules. A submit, review and approve process is used within document-related modules. Routing of information is not clearly defined (score 5/10).

CMS	Average score
Typo3	7.6
Drupal	6.7
Mambo Open Source	5.7
phpWebSite	5.5
Xaraya	5.3
Xoops	4.1
PostNuke	3.7

Table 5.9: CMS evaluation results

5.8 Results of the comparison

This report compared seven open source CMS products. These products were; Drupal, Mambo Open Source, phpWebSite, PostNuke, Typo3, Xaraya and Xoops. The comparison was based on eight categories as seen from a business perspective. These categories were; applications, data repository, deployment, integration, revision control, user interface, user management and workflow. Each category was scored from 0 to 10 points and the overall score was determined based on the average of all categories. Table 5.9 shows the average score achieved by each CMS product.

The objective of this report was not to select a winning product. Rather, the aim of the comparison was to show the suitability of each product from a business perspective, based on a set of business requirements. As seen in table 5.9, Typo3 and Drupal received the highest scores. Both Typo3 and Drupal achieved their scoring due to their implementation of features which were more suitable for small to medium businesses. Typo3 and Drupal showed a clear advantage over other products in Revision Control and User Interface categories. In addition, Typo3 received full score on the Workflow category.

Both Typo3 and Drupal are distributed under the GPL license. Therefore,

Category	Average score
Applications	8.3
Workflow	7.8
User interface	6.4
Integration	5.2
Deployment	5.0
Data repository	4.4
User management	4.0
Revision control	2.8

Table 5.10: Category evaluation results

this will require businesses to carefully consider the implications of using GPL licensed software, as explained in chapter 3. Commercial redistribution of these products or any subsequent derivatives is impossible. Instead, businesses will have to focus on paid services when distributing GPL licensed software.

CMS products that received the lowest scores where PostNuke and Xoops. PostNuke failed to meet expectations and did not meet the requirements. The most notable failure was the absence of Revision Control and database compatibility. Similar products like PHP-Nuke and derivative forks also failed in the same categories.

The only exception was Xaraya. Although this product is a fork from PostNuke and PHP-Nuke, it met more business requirements in comparison to PostNuke. The most important difference between Xaraya and PostNuke was the Workflow integration within Xaraya which exceeded expectations.

In addition to the results based on each CMS product, it is also important to look at the per-category average scores which represent another aspect of this comparison. The per-category average results show that some categories

were lacking throughout the comparison. Table 5.10 shows the average scores for each category.

The Revision Control category failed to meet the business requirements of most CMS products. More specifically, this category can be a major decision factor for businesses that deal with large amounts of content as well as large development teams. Although, User Management was the second lowest scored category, it is important because businesses require proper user management for both intranet and general public users. Without proper control and extensible features like data mining, businesses will not seriously consider open source CMS solutions.

Chapter 6

Conclusion

6.1 Summary of this report

The objective of this report was to analyse and compare the category of open source, web portal Content Management Systems within the context of small to medium businesses. This objective was achieved by introducing the meaning of content within businesses, emphasising the importance of open source licenses and providing a comparison of seven CMS products.

Chapter 2 dealt with content management and defined the terms; content, content management and content management systems. It presented the following five areas that use content management extensively; metadata, peer-to-peer, the Semantic Web, grid computing and hypermedia. It provided examples of successful open source products. This chapter also illustrated the importance of information to businesses and indicated significant areas where effective management of information is required. CMS products offer a solution to this problem by providing a new layer of management on top of data.

- Content management is an integral part of every business.
- Open source licenses can be an advantage as well as a liability.
- A single CMS product may cover multiple content categories.
- CMS features fall under eight main categories.
- No single product meets all business requirements.
- PostNuke and PHP-Nuke are unsuitable from commercial use.
- Open source CMS products focus on application features and not integration.
- Revision Control and User Management are the weakest categories.
- Typo3 and Drupal are mature products that can be used within a commercial environment.

Figure 6.1: Conclusions of this report

Chapter 3 discussed open source licenses and provided a clear definition for the terms free and open source software. It described significant aspects of OSI-approved licenses. This chapter also illustrated the importance of open source licenses within a commercial environment as well as the relation between free and commercial licenses.

Chapter 4 defined the basis of the comparison and showed the overlap between different CMS categories. It outlined the requirements and prerequisites for CMS products, which formed the selection criteria for products used during the comparison. This chapter also lists the categories used for the comparison and their respective scores.

Chapter 5 compared the seven CMS products. Each product was analysed based on the defined categories as stated in chapter 4. A total averaged score was calculated based on the individual scores for each category. The final section provides an analysis of the results.

6.2 Content management

The most important characteristic of content is the human aspect. Content carries an identification or meaning which is understandable by humans, something intuitive that makes content impossible to process via a computer. Information is conceived or understood by the humans that view the data. Thus, information is subject to connotation, context and interpretation, making it impossible to process as raw data by the computer. Metadata is the technology that makes it possible to process content while maintaining the human understanding and interpretation. CMS systems with metadata encapsulation techniques are a significant advantage to businesses that use data mining technologies.

Content management refers to the principles and practises for the development, management, maintenance and deployment of content within a single organisation or across multiple organisations. Content management is a topic which covers a wide range of areas within a business. This report specifically covered the web aspect of a business, which dealt with web-related content as well representation of that content over the web.

Content management over the web unifies previously separate efforts within the business. For example, marketing and product information were previously the province of the marketing department within the business, which produces assets such as price lists and brochures. Another internal department such as the IT department, maintains supply-chain information and order lists.

Content management of web assets can bring the two departments together using a web-based integrated system. As a result of consolidating web assets within an integrated system, the business is able to respond to a dynamic market, while providing fresh content and updated service offerings. This fresh content is a product of experimentation and iteration at all levels.

Furthermore, content management can be seen as a the sum of contributors working on it, these include; developers, artists, marketers and others. Within a business, they can be full-time employees, contractors, outside vendors and others. This report identified four 'key' processes, also known as "best practises" of asset management and web development. These processes are; revision control, concurrent changes management, deployment and workflow, which were used as the basis of the comparison requirements.

6.3 Open source software

The term free software is widely used in the IT industry. However, its ambiguity hampers communication due to arguments over whether a particular piece of software is 'free' or not. Free software is defined as the freedom to run, copy, distribute, study, change and improve the software or source code.

A wide range of software is distributed as "free" because it does not cost anything to download or use. However the source code is not made available or the software is distributed with a restrictive license. Binary or source code distributions could be copyrighted and covered by a license agreement, which could hold a range of few to extreme restrictions, like a disclaimer of reliability.

Open source software is free software that comes with a license that offers specific liberties and limitations. There are two main categories of open source licenses, GPL-incompatible and GPL-compatible. Software which is GPL-incompatible retains its copyright but grants all rights under its copyright to the user and does not restrict the source code from being used by non-open source applications. Software which is GPL-compatible retain their copyright, grant all rights under copyright to the user but apply at least one restriction; the redistribution of the software, whether modified or unmodified, must be under the same license.

GPL-compatible licenses can be a major problem from a business perspective. GPL-compatible licenses require that all software that link against GPL-compatible licensed code must also be licensed under a GPL-compatible license. This requirement does not exist in the LGPL license. Running GPL-compatible licensed software in a commercial environment can be a problem.

From a business perspective, the safest type of license is a GPL-incompatible license. For example, the BSD license allows its source code to be modified, it can be included within a commercial product and its redistribution is unrestricted. There are no commercial limitations for linking or running under a commercial product. Thus, existing commercial applications will be able to link to BSD licensed software.

6.4 CMS categories

Content Management Systems (CMS) are not just a product or a technology. CMS is a generic term which defines a wide range of processes which underpin the "next-generation" of medium to large-scale websites. A content management process; creates, stores, modifies, retrieves and displays data, or content. Thus, a CMS product must be able to perform as many of those processes as possible.

The applications of CMS cannot be clearly defined. Even though a CMS is range of processes and managed software, the boundaries of the CMS space are blurred. The area covered by CMS overlaps with a wide range of traditional software systems such as; document management systems, knowledge management systems, enterprise application integration systems, e-commerce solutions and web portals. As a result of this overlap of functionality, an intranet groupware system or virtual learning system can easily be implemented via the same CMS.

This report did not deal with the application or use of CMS. Instead, this

report applied a comparative approach to web portals only, based on their functionality from a business perspective. Web portals are websites which act as a main "point of entry" for users. They offer a range of services, for example; news section, search engine and web catalogue.

6.5 CMS features

Each business has its own set of requirements for a CMS solution, which depend on various parameters such as; the size of the business, field of operation, type of managed data and target customers. It is highly unlikely that a single product will have all the required functionality. As a result, this report identified seven potential products by using eight different areas of functionality for the base of their comparison. These eight areas are; applications, data repository, deployment, integration, revision control, user interface, user management and workflow.

The applications category was used because it adds general functionality which compliments the entire CMS solution. The data repository category was added due to the requirements for flexibility in content storage. The deployment category was used based on business requirements for scalability, fault tolerance and replication. The integration category was important due to the metadata analysis and standards compliance.

The revision control category was used due to the importance of asset control, modification and roll-back of changes. The user interface category was based on tools that enhance the control of processes. The user management category was based on the access and control of the system, for authentication as well as user data management. The workflow category was used for the development and maintenance of business assets.

The weight for each category was based on their importance from a business perspective. Categories which are integral for a small to medium business

are divided into two sections, while those of less importance are divided into three or more sections.

6.6 No winning product

The comparison of the seven CMS products shows how there are no winning products. None of the CMS products meet all business requirements as set by this report. The CMS products are come closest to this goal are considered the most powerful and likely to perform within a commercial environment.

6.7 Nuke products are unsuitable

PostNuke and PHP-Nuke are the most unsuitable products for commercial use. PostNuke offers a wide range of application features, mostly contributions from the open source community but has no real integration, database flexibility or any revision control features.

PHP-Nuke has similar limitations but was not included in the comparison due to licensing issues and its dubious future. The author of PHP-Nuke has stated that the distribution license may change at any time, as a result of websites removing the copyright notice and requests payment for the latest release, which has to be payed for every new release.

6.8 Focus is on application features

This report has demonstrated how the current trend of CMS products is to emphasise on application features instead of underlying system integration. The application category has achieved the highest average scoring of 8.3,

among all CMS products. This is an indication of open source products being lead by non-commercial requirements and personal development needs. This lack of commercial influence is evident in most CMS products but not all. Typo3 is clearly influenced by a commercial organisation which targets small to medium businesses.

6.9 The weakest categories

Revision control and user management are the two weakest categories. Revision control achieved a 2.8 average among all the CMS products while user management achieved a 4.0 average. These results show how most CMS products do not take business requirements into consideration. Without proper revision control has a high probability that it will be rejected by most businesses.

Although user management exists in all the CMS products, the supplied functionality does not consider large user databases. Thus, most businesses would find it very time consuming or even impossible to perform operations such as; data mining based on user information and user collection and control based on certain criteria.

6.10 Typo3 & Drupal

The objective of this report was not to select a winning product. Rather, the aim of the comparison was to show the suitability of each product from a business perspective, based on a set of business requirements. Typo3 and Drupal received the highest scores. Typo3 achieved an average score of 7.6, while Drupal achieved an average of 6.7.

They achieved their scoring due to their implementation of features which

were more suitable for small to medium businesses. Both CMS products showed a clear advantage over other products in Revision Control and User Interface categories. In addition, Typo3 received full score on the Workflow category. Both Typo3 and Drupal are considered mature products since they meet almost all the business requirements.

6.11 Future work

The comparison of this report clearly identified a lack of commercial influence in the current open source CMS products. There is a potential for improvement in the revision control and user management categories. Although current CMS products can enhance their functionality by implementing revision control processes, the user management category can be improved by an external application which offers desktop control over the user database.

Appendix A

Typography

In order to demonstrate how open source software are efficient, reliable and productive, this report has been produced entirely using open source software. Table A.1 lists the most important software tools.

	Tool
Typesetting system	LaTeX 2_ε
Formatting engine	TeX(Web2C v7.4.5) v3.14159
Spellchecker	aspell v0.50.3
Generation of figures	Xfig v3.2.4
IDE environment	Kile v1.6.3
Operating system	Fedora Core 2 Linux

Table A.1: Production tools of this report

The body of this report uses the Computer Modern font, which is distributed by LaTeX 2_ε in metafont and PostScript Type1 format. Some small parts of this report also use the European Computer Modern and TeX IPA (TIPA) fonts. The PDF edition of this report was produced by PDFTeX.

Appendix B

Rejected systems

CMS	URL*
Absolut Engine	http://www.absolutengine.com
AWF	http://www.liquidbytes.net
Back-End CMS	http://www.back-end.org
BolinOS	http://www.bolinos.ch
Bricolage	http://www.bricolage.cc
Cofax	http://www.cofax.org
CPG-Nuke	http://cpgnuke.com
DotNetNuke	http://www.dotnetnuke.com
eZ publish	http://ez.no
Geeklog	http://www.geeklog.net
GYO	http://growyourown.babel.com.au
Lenya	http://cocoon.apache.org/lenya
Magnolia	http://www.obinary.com
MDPro	http://www.maxdev.com

CMS	**URL***
MetaDot	http://www.metadot.com
MMBase	http://www.mmbase.org
NPDS	http://www.npds.org
Open CMS	http://www.opencms.org
Opus	http://opus.cx
Pathos	http://www.pathoscms.org
Phase	http://www.phasecms.org
PHP Nuke	http://phpnuke.org
phpwcms	http://www.phpwcms.de
Plone	http://plone.org
RedHat CCM	http://www.redhat.com/software/rha/cms
Sitellite CMS	http://sitellite.org
Tiki CMS/Groupware	http://tikiwiki.org
WebGUI	http://www.plainblack.com/webgui
WebMake	http://webmake.taint.org
XHP	http://xhp.sourceforge.net
ZeusCMS	http://sourceforge.net/projects/zeuscms
	*last access date: 20-4-2004

Table B.1: Rejected systems

Appendix C

Screenshots

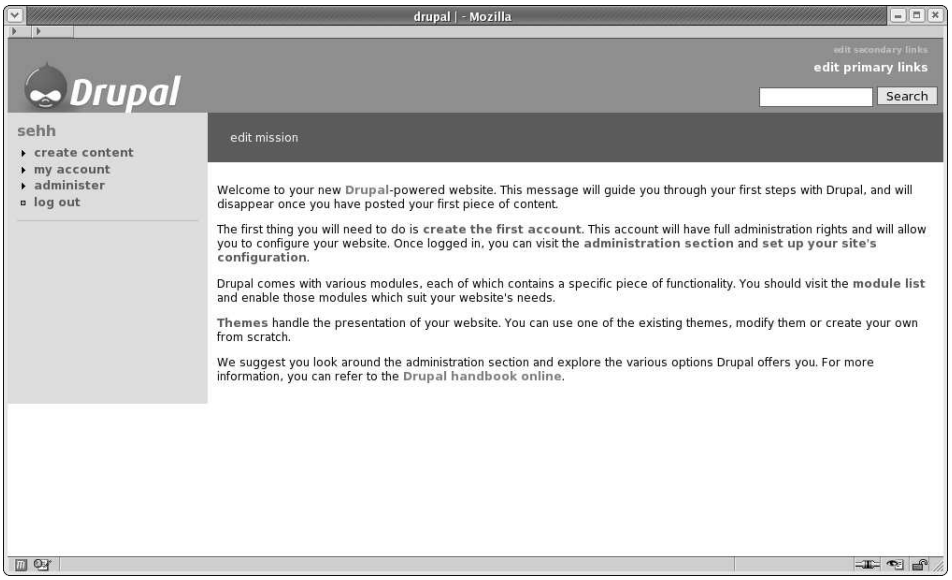


Figure C.1: CMS: Drupal

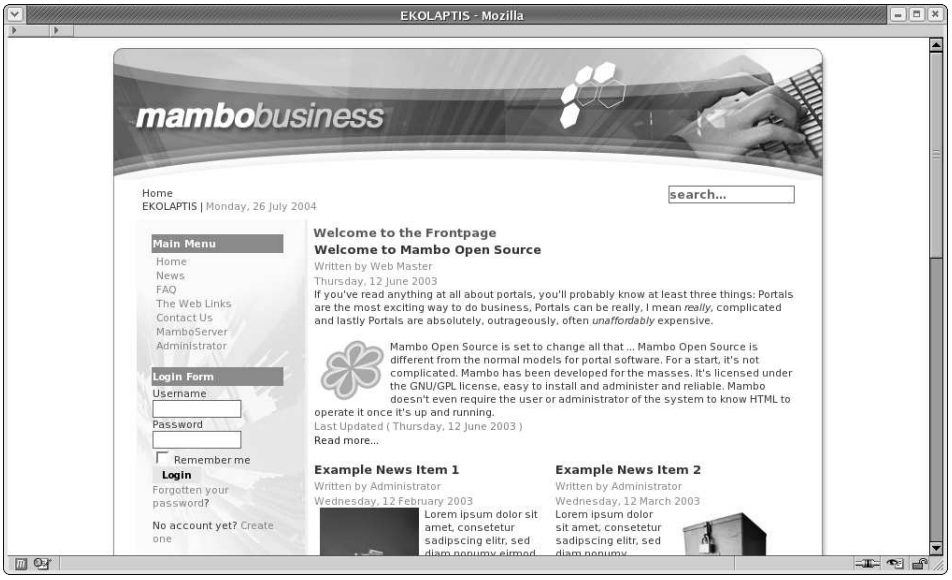


Figure C.2: CMS: Mambo Open Source

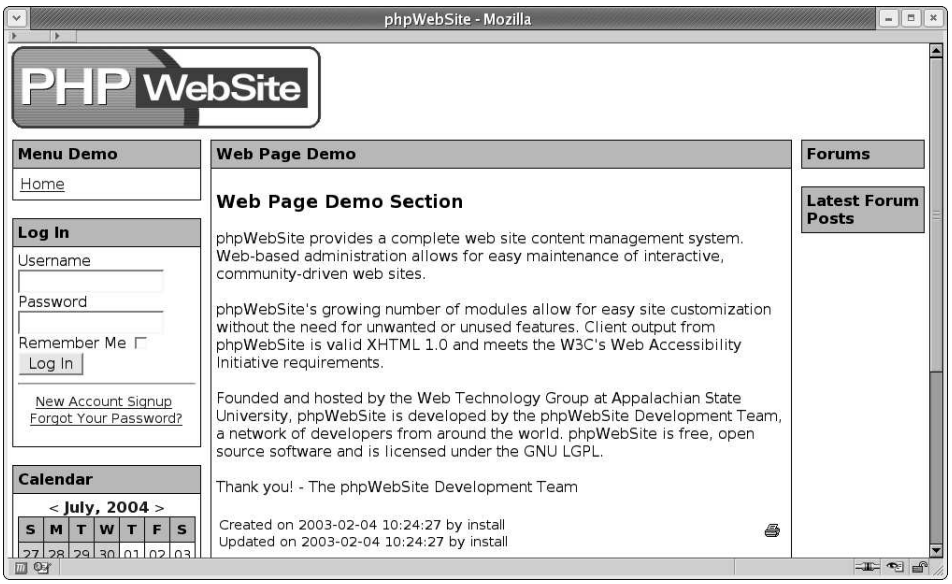


Figure C.3: CMS: phpWebSite

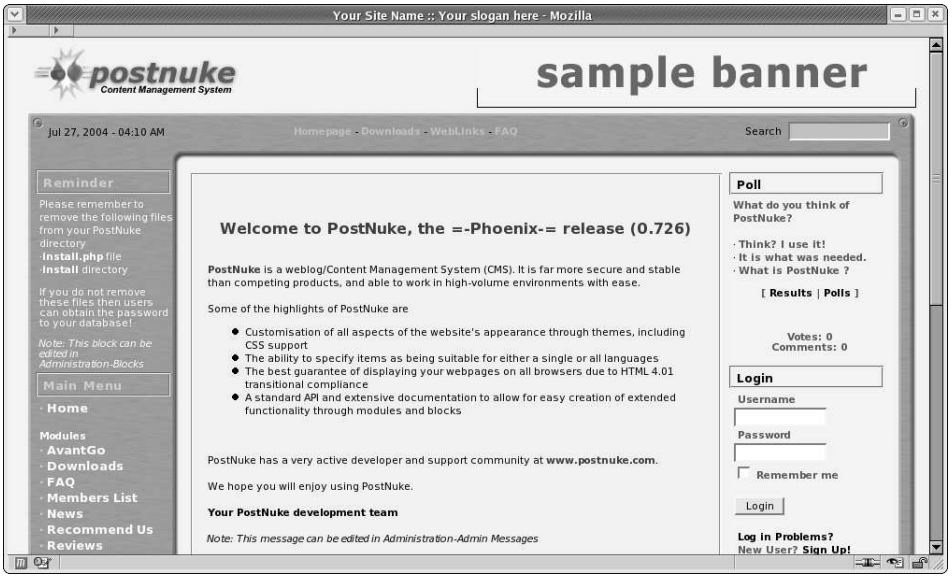


Figure C.4: CMS: PostNuke

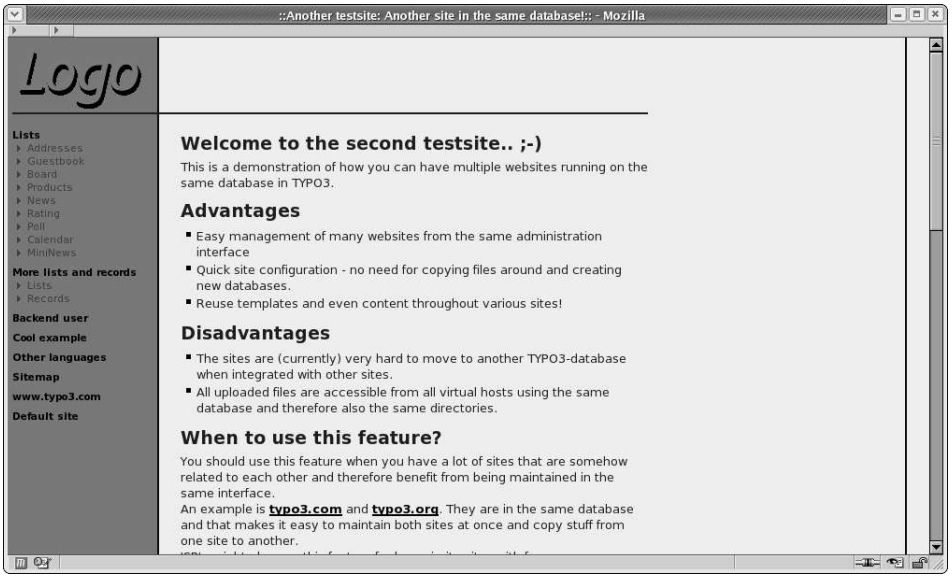


Figure C.5: CMS: Typo3

Figure C.6: CMS: Xaraya

Figure C.7: CMS: Xoops

Appendix D

GNU General Public License

Version 2, June 1991

59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software

Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's

free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

Terms and Conditions For Copying, Distribution and Modification

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   (c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

   Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise

the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   (a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   (b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   (c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that

is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions

are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

No Warranty

11. Because the program is licensed free of charge, there is no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide the program "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.

12. In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

End of Terms and Conditions

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> one line to give the program's name and a brief idea of what it does.
> Copyright (C) yyyy name of author

> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> Gnomovision version 69, Copyright (C) yyyy name of author
> Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
> This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary.

Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program
> 'Gnomovision' (which makes passes at compilers) written by James Hacker.
>
> signature of Ty Coon, 1 April 1989
> Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Appendix E

BSD License

> The following is a BSD license template. To generate your own license, change the values of OWNER, ORGANIZATION and YEAR from their original values as given here, and substitute your own.
>
> Note: The advertising clause in the license appearing on BSD Unix files was officially rescinded by the Director of the Office of Technology Licensing of the University of California on July 22 1999. He states that clause 3 is "hereby deleted in its entirety."
>
> Note the new BSD license is thus equivalent to the MIT License, except for the no-endorsement final clause.

<OWNER> = Regents of the University of California
<ORGANIZATION> = University of California, Berkeley
<YEAR> = 1998

In the original BSD license, both occurrences of the phrase "COPYRIGHT HOLDERS AND CONTRIBUTORS" in the disclaimer read "REGENTS

AND CONTRIBUTORS".

Here is the license template:

Copyright ©<YEAR>, <OWNER>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTH-

ERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Appendix F

MIT License

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Glossary

Apartheid	An official policy, which has now ended, of racial segregation and white supremacy enforced by the South African government. "Apartheid" means "apartness", separation of the people according to their race, page 21
API	Application Programmers Interface, page 34
ASF	Apache Software Foundation, page 31
CMS	Content Management Systems, page 28
Computer Modern	The Computer Modern font family is a large collection of text, display and mathematical fonts in a range of styles, based on Monotype Modern 8A. It was created using the metafont font design system, which was developed as part of the TEX Project at Stanford University, 1978–1988 (Knuth 1986*a*), page 67
Content management	Principles and practises for the development, management, maintenance and deployment of content within a single organisation or across multiple organisations, page 9
CSV	Comma Separated Values, page 42
CVS	Concurrent Versions System, page 12
DBMS	DataBase Management System, page 25

FSF	Free Software Foundation, page 16
GNU	GNU's Not Unix, page 16
GPL	General Public License, page 11
Grid Computing	Flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources (Foster et al. 2001), page 6
GUI	Graphical User Interface, page 11
HTML	HyperText Markup Language, page 32
Hypermedia	A computer-based information retrieval system that enables a user to gain or provide access to texts, audio and video recordings, photographs, and computer graphics related to a particular subject, page 6
IDE	Integrated Development Environment, page 67
J2EE	Java 2 Platform, Enterprise Edition, page 29
JSP	Java Server Pages, page 31
LDAP	Lightweight Directory Access Protocol, page 36
LGPL	Lesser General Public License, page 11
Metadata	Data about data. Definitional data that provides information about or documentation of other data, page 6
Metafont	A system for the design of raster-based alphabets. Metafont supports the creation of entire families of fonts from a set of dimensional parameters and outline descriptions. Metafont is part of the TEX Project at Stanford University, 1978–1988. Use of metafont is described by Knuth (1986*b*), page 67

MIT	Massachusetts Institute of Technology, page 20
MPL	Mozilla Public License, page 24
NDA	Non-Disclosure Agreement, page 21
NFS	Network File System, page 34
NIS/YP	Network Information Service/Yellow Pages, page 36
NPL	Netscape Public License, page 24
OSI	Open Source Initiative, page 16
OSS	Open Source Software, page 16
P2P	Peer to peer networking. peer [piə] n. an equal in rank, merit or quality, page 7
Pacta Sunt Servanda	Pacts must be Respected, basic principle of civil and international law, page 21
PAM	Pluggable Authentication Modules, based on the DCE RFC, page 36
PDF	Portable Document Format, page 35
Perl	Cross platform interpreted programming language, page 29
PHP	PHP Hypertext Preprocessor, server-side scripting language, page 29
PO	Portable Object, multi-language format, page 46
Python	Interpreted, interactive, object-oriented programming language, page 29
RDBMS	Relational DataBase Management System, page 34
RDF	Resource Description Framework, a metadata standard, page 6

Release Agreement	A binding agreement which acts as a social contract between development and production groups, page 13
RSS	Rich Site Summary, multipurpose extensible metadata description and syndication format, page 42
RTE	Rich Text Editor, page 51
SCM	Software Configuration Management, page 11
SGML	Standard Generalised Markup Language, page 50
SMS	Short Message Service, page 35
SQL	Structured Query Language, page 36
The Semantic Web	The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in co-operation (Berners-Lee et al. 2001), page 6
URL	Uniform Resource Locator, page 6
W3C	World Wide Web Consortium, page 6
WAP	Wireless Application Protocol, page 35
WML	Wireless Markup Language, page 50
WYSIWYG	What You See Is What You Get, page 36
XHTML	Extensible HyperText Markup Language, page 30
XML	eXtensible Markup Language, page 6

References

Anderson, K., Taylor, R. & Whitehead, E. (1994), Chimera: Hypertext for heterogeneous software environments, *in* 'Proceedings of the ACM European conference on Hypermedia technology', pp. 94–197.

Apache (2004), The apache http server project. Last access: 20-4-2004.
URL: *http://httpd.apache.org*

Bailey, C., Hall, W., Millard, D. E. & Weal, M. J. (2002), Towards open adaptive hypermedia, *in* P. De Bra, P. Brusilovsky & R. Conejo, eds, 'Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems', Springer-Verlag, pp. 36–46.

Berners-Lee, T., Hendler, J. & Lassila, O. (2001), 'The semantic web', *Scientific American* .

Boiko, B. (2001), *Content Management Bible*, John Wiley & Sons Inc.

Browning, P. & Lowndes, M. (2001), JISC techwatch report: Content management systems, Technical Report TSW 01-02, Joint Information Systems Committee.

Chao-Kuei (2004), Categories of free and non-free software. Free Software Foundation, GNU Operating System, last access: 20-4-2004.
URL: *http://www.gnu.org/philosophy/categories.html*

Chen, L. & Sycara, K. (1998), Webmate: a personal agent for browsing

and searching, *in* 'Proceedings of the second international conference on Autonomous agents', ACM Press, pp. 132–139.

Compiere (2004), Smart open source erp software solutions with integrated crm software solutions. Last access: 20-4-2004.
URL: *http://www.compiere.org*

Foster, I., Kesselman, C. & Tuecke, S. (2001), The anatomy of the grid: Enabling scalable virtual organizations, Technical report, International Journal of Supercomputer Applications and High Performance Computing.

Fountain, A., Hall, W., Heath, I. & Davis, H. (1990), Microcosm: An open model for hypermedia with dynamic linking, *in* A. Rizk, N. Streitz & J. Andre, eds, 'Hypertext: Concepts, Systems and Applications', Proceedings of ECHT'90, Cambridge University Press, pp. 298–311.

FreeBSD (2004), The freebsd project. Last access: 20-4-2004.
URL: *http://www.freebsd.org*

FSF (2004), The free software definition. Free Software Foundation, GNU Operating System, last access: 20-4-2004.
URL: *http://www.fsf.org/philosophy/free-sw.html*

Grønbæk, K. & Trigg, R. H. (1994), 'Design issues for a dexter-based hypermedia system', *Communications of the ACM* **37**(2), 40–49.

HMSO (1995), 'Disability Discrimination Act 1995 Elizabeth II. Chapter 50', The Stationery Office Limited, United Kingdom.

Knuth, D. E. (1986*a*), *Computer Modern Typefaces*, Vol. E of *Computers and Typesetting*, Addison-Wesley.

Knuth, D. E. (1986*b*), *The METAFONT book*, Vol. C of *Computers and Typesetting*, Addison-Wesley.

Lieberman, H. (1995), Letizia: An agent that assists web browsing, *in* 'Proceedings of the International Joint Conference on Artificial Intelligence'.

Linux (2004), The linux kernel archives. Last access: 20-4-2004.
URL: *http://www.kernel.org*

Maglio, P. & Farrell, S. (2000), Liveinfo: Adapting web experience by customization and annotation, *in* 'Proceedings of the 1st International Conference on Adaptive Hypermedia and Adaptive Web-based Systems'.

Michelinakis, D. (2003), SotonOne project, Master's thesis, University of Southampton, Department of Electronics and Computer Science.

MozillaFoundation (2004), The mozilla browser. Last access: 20-4-2004.
URL: *http://www.mozilla.org*

Nakano, R. (2001), *Web Content Management A Collaborative Approach*, Addison-Wesley.

Netcraft (2004), Web server survey. 2004 Web Server Survey, last access: 20-4-2004.
URL: *http://news.netcraft.com/archives/web_server_survey.html*

OSI (2004), The open source definition. Open Source Initiative, last access: 20-4-2004.
URL: *http://www.opensource.org*

Perens, B. (1997), The open source definition. Treasurer, Open Source Initiative, last access: 20-4-2004.
URL: *http://perens.com/Articles/OSD.html*

Perl (2004), The perl language. Last access: 20-4-2004.
URL: *http://www.perl.org*

PostgreSQL (2004), Postgresql database. Last access: 20-4-2004.
URL: *http://www.postgresql.org*

Roure, de, D., Baker, M., Jennings, N. & Shadbolt, N. (2003), The evolution of the grid, Technical Report 6871, University of Southampton.

Bibliography

AberdeenGroup (2001), Content management: At the center of e-business, Technical report, Aberdeen Group Inc. An Executive White Paper.

Bachler, M., Shum, S. B., Chen-Burger, J., Dalton, J., Roure, D. D., Eisenstadt, M., Komzak, J., Michaelides, D., Page, K., Potter, S., Shadbolt, N. & Tate, A. (2004), Collaborative tools in the semantic grid, *in* D. Michaelides & L. Moreau, eds, 'proceedings of GGF11 Semantic Grid Applications Workshop'.

Berman, F., Fox, G. & Hey, T., eds (2003), *Grid Computing – Making the Global Infrastructure a Reality*, Wiley, chapter 42.

bit10 (2003), Case study: The university of warwick enterprise content management system, Technical report, The University of Warwick.

Brand, A., Daly, F. & Meyers, B. (2003), Metadata demystified: A guide for publishers, Technical report, National Information Standards Organization. The Sheridan Press & NISO Press.

Collins, H. (2003), *Enterprise Knowledge Portals: Next Generation Portal Solutions for Dynamic Information Access, Better Decision Making and Maximum Results*, Amacom.

Dovey, M. (2001), JISC techwatch report: Java portals, Technical Report TSW 01-03, Joint Information Systems Committee.

McDonald, D. (2003), JISC techwatch report: Web services technologies, Technical Report TSW 03-04, Joint Information Systems Committee.

Nonaka, I. (1994), 'A dynamic theory of organizational knowledge creation', *Organization Science* **5**(1), 14–37.

Pedersen, J. S. K. (2003), A study of open source content management, Technical report, IT University of Copenhagen.

Raymond, E. (1999), *The Cathedral & the Bazaar*, O'Reilly & Associates.

Robertson, J. (2002), 'How to evaluate a content management system', *KM Column* .

Robertson, J. (2004), 'Open-source content management systems', *KM Column* .

Tanaka, H. & Tanaka, K. (2004), Continuous web: A new image-based hypermedia and scape-oriented browsing, *in* 'proceedings of The 13th International World Wide Web Conference (WWW2004)'.

Teeman, B. (2004), 'Mambo open source 4.5, real management', *Linux Magazine* **39**.

Wersin, D. (2001), Evaluation and comparison of content management systems, Technical report, University of Rostock. Bachelor Thesis.

Wills, G., Woukeu, A., Carr, L. & Hall, W. (2003), The need for deeper design in a semantic web, *in* 'proceedings of HTSW2003 - First International Workshop on Hypermedia and the Semantic Web'.

Zhou, J., Dialani, V., Roure, D. D. & Hall, W. (2003), A semantic search algorithm for peer-to-peer open hypermedia systems, *in* 'The First Workshop on Semantics in Peer-to-Peer and Grid Computing', pp. 43–54.