A DYNAMIC, COST-EFFECTIVE AND EFFICIENT INFORMATION PORTAL SOLUTION USING THE MAMBO CONTENT MANAGEMENT SYSTEM

A thesis presented to

the faculty of

the Fritz J. and Dolores H. Russ College of Engineering and Technology

of Ohio University

In partial fulfillment

of the requirements for the degree

Master of Science

Adnan B. A. Dakhwe

August 2006
This thesis entitled

A DYNAMIC, COST-EFFECTIVE AND EFFICIENT INFORMATION PORTAL SOLUTION USING THE MAMBO CONTENT MANAGEMENT SYSTEM

by

ADNAN B. A. DAKHWE

has been approved for

the School of Electrical Engineering and Computer Science

and the Russ College of Engineering and Technology by

Chang Liu

Assistant Professor of Electrical Engineering and Computer Science

Dennis Irwin

Dean, Russ College of Engineering and Technology
Abstract

DAKHWE, ADNAN B. A., M.S., August 2006. Computer Science

A DYNAMIC, COST-EFFECTIVE AND EFFICIENT INFORMATION PORTAL SOLUTION USING THE MAMBO CONTENT MANAGEMENT SYSTEM (89 pp)

Director of Thesis: Chang Liu

The world has progressed immensely in the last decade, and so have the common skills held by people. Nowadays most employees in an organization have skills to use the Internet and Microsoft Word. There has been an effort to harness these skills held by these people to develop web-applications by masking the technical details from them.

This work investigates the techniques for facilitating rapid development of an information portal and evaluates the effectiveness of the technique chosen to develop and maintain an information portal. Our chosen technique involves using the Mambo content management system to develop an information portal architecture rapidly and delegating the responsibility of content management i.e. to add, change and delete content to novice users (who have no web-programming skills), by masking the technical details from the novice users. We have tested our approach in the development of the Centers for Osteopathic Research and Education (CORE) information portal. We have performed initial quantitative analysis to evaluate the applicability of Mambo CMS to the CORE information portal, which is a basic mid-size information portal.

Approved:

Chang Liu

Assistant Professor of Electrical Engineering and Computer Science
Dedication

This thesis is dedicated to my parents.
Acknowledgements

I would like to thank my advisor, Dr. Chang Liu for his patience and support, without whom this thesis would not have been possible. I appreciate his feedback and guidance throughout my thesis.

I would like to thank my other committee members, Dr. Frank Drews and Dr. Jundong Liu for agreeing to serve on my committee. I would like to express sincere gratitude to Dr. Joy Matthews-López (external college representative – committee member) for giving me the opportunity to work on the CORE (Centers for Osteopathic Research and Education) information portal project. Additionally, I would like to thank her for her help with the statistical analyses involved in this thesis. I would like to thank Dr. Doug Mann for his support and guidance for the CORE information portal project. Special thanks to the CORE staff for their patience and support.

I would like to thank my mom and dad and my sisters Asma and Tazyeen for their continual affection, support, guidance and encouragement. I would like to thank my brother-in-law Saif for his support, encouragement and for having faith in me. I would like to thank my brother-in-law Fauwaad for his support and guidance. A special thanks to my friends and roommates at Ohio University. I would also like to thank my family in Canada and my relatives all over the world. Last but not the least; I would like to thank my beloved wife, Virginia (Nia) for her continual affection, support, patience and encouragement.
# Table of Contents

ABSTRACT .................................................................................................................. 3  
DEDICATION ............................................................................................................. 4  
ACKNOWLEDGEMENTS ............................................................................................ 5  
LIST OF TABLES ....................................................................................................... 9  
LIST OF FIGURES .................................................................................................. 10  
LIST OF ABBREVIATIONS ....................................................................................... 11  

CHAPTER 1: INTRODUCTION ............................................................................. 13  
1.1 Background ....................................................................................................... 13  
1.2 Related Work ................................................................................................... 15  

CHAPTER 2: CASE STUDY: THE CORE INFORMATION PORTAL (CIP)  
PROJECT ................................................................................................................... 20  

CHAPTER 3: OPEN SOURCE CONTENT MANAGEMENT SYSTEMS ............ 23  
3.1 Drupal ................................................................................................................ 23  
3.2 PHP-Nuke ......................................................................................................... 24  
3.3 Mambo ............................................................................................................... 25  
3.4 Plone ................................................................................................................... 25  
3.5 Decision Made: Mambo is Selected ................................................................. 26  

CHAPTER 4: MAMBO IN DETAIL .................................................................... 27  
4.1 Mambo’s Fundamental Concepts ................................................................. 27  
4.2 User Management and Access Control ....................................................... 28  
    4.2.1 Public Front-end ......................................................................................... 28
4.2.1.1 Registered Group ................................................................. 29
4.2.1.2 Author Group ............................................................................ 29
4.2.1.3 Editor Group ............................................................................. 29
4.2.1.4 Publisher Group ........................................................................ 29
4.2.2 Public Back-end ........................................................................... 29
4.2.2.1 Manager .................................................................................. 30
4.2.2.2 Administrator .......................................................................... 30
4.2.2.3 Super Administrator ................................................................. 30

4.3 Mambo Database Tables .................................................................... 31

4.4 Mambo Front-end .............................................................................. 32

4.5 Mambo Back-end .............................................................................. 34

4.6 Mambo Template Management .......................................................... 36

4.7 Mambo Add-ons ............................................................................... 36
4.7.1 Module ...................................................................................... 37
4.7.2 Component .................................................................................. 37
4.7.3 Mambot ...................................................................................... 37
4.7.4 More Information About Add-ons ................................................ 37

CHAPTER 5: THE CORE INFORMATION PORTAL (CIP) SYSTEM ........ 39
5.1 Front-end of the CIP ......................................................................... 40
5.2 Template of the CIP .......................................................................... 43
5.3 Back-end of the CIP .......................................................................... 46
5.4 The Database Tables of the CIP .......................................................... 48

CHAPTER 6: USER INTERFACES OF THE CORE INFORMATION PORTAL
(CIP) ........................................................................................................ 50
6.1 User Interface for the Visitors ............................................................ 50
6.2 User Interface for the Content-Contributors ...................................... 52
6.3 User Interface for the Administrators ................................................ 54

CHAPTER 7: EVALUATION .......................................................... 56
7.1 Evaluation of the CORE Information Portal (CIP) ................................ 56
7.1.1 Ease of Use ................................................................................. 56
7.1.1.1 Web-development competency of users .................................. 57
7.1.1.2 Summary of present technical competencies ....................... 60
7.1.1.3 Confirmation of non-user-friendliness of old website .............. 62
7.1.1.3.1 Content creation or updation ........................................... 62
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Tabular description of related work</td>
<td>17</td>
</tr>
<tr>
<td>4.1 List of Mambo database tables</td>
<td>32</td>
</tr>
<tr>
<td>5.1 Database tables of the CIP</td>
<td>48</td>
</tr>
<tr>
<td>7.1 Web-development competency measurement scale</td>
<td>57</td>
</tr>
<tr>
<td>7.2 Website development and website maintenance cross-tabulation</td>
<td>59</td>
</tr>
<tr>
<td>7.3 Internet use and Word knowledge cross-tabulation</td>
<td>60</td>
</tr>
<tr>
<td>7.4 Browser and Word knowledge cross-tabulation</td>
<td>61</td>
</tr>
<tr>
<td>7.5 Web-programming knowledge required to maintain content on CIP</td>
<td>64</td>
</tr>
<tr>
<td>7.6 Summary of user-friendliness of CIP interfaces</td>
<td>65</td>
</tr>
<tr>
<td>7.7 Table for similarity of the CIP online editor to Microsoft Word</td>
<td>67</td>
</tr>
<tr>
<td>7.8 Responses for overall content management experience</td>
<td>68</td>
</tr>
<tr>
<td>7.9 Responses for time to update content on CIP</td>
<td>69</td>
</tr>
<tr>
<td>7.10 Responses for use and learning experience of the content-contributor interface</td>
<td>70</td>
</tr>
<tr>
<td>7.11 Tabular comparison of our work with the related work</td>
<td>76</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 The CIP Front-end</td>
<td>41</td>
</tr>
<tr>
<td>5.2 The Front-end Template with Inline Positions</td>
<td>45</td>
</tr>
<tr>
<td>5.3 The Back-end of the CIP System</td>
<td>47</td>
</tr>
<tr>
<td>6.1 The User Interface for the Visitors of the CIP</td>
<td>51</td>
</tr>
<tr>
<td>6.2 The Front-end WYSIWYG Editor</td>
<td>53</td>
</tr>
<tr>
<td>6.3 The User Interface for the Administrators</td>
<td>55</td>
</tr>
<tr>
<td>7.1 Bar Graph of the Web-development Score by User ID</td>
<td>58</td>
</tr>
</tbody>
</table>
List of Abbreviations

AMD – Advanced Micro Devices
ASP – Active Server Pages
BSD – Berkeley Software Distribution
CIP – CORE Information Portal
CMS – Content Management System
CORE – Centers for Osteopathic Research and Education
CSS – Cascading Style Sheets
FTP – File Transfer Protocol
GB – Gigabyte
GNU – GNU’s Not Unix (recursive acronym)
GPL – General Public License
HTML – Hypertext Markup Language
IIS – Internet Information Service
IT – Information Technology
JSP – Java Server Pages
PHP – Hypertext Pre-processor (recursive acronym)
RAID – Redundant Array of Independent (or Inexpensive) Disks
RAM – Random Access Memory
RDE – Rapid Development Environment
RSS - Really Simple Syndication
URL – Universal Resource Locator
WYSIWYG – What You See Is What You Get

XML – eXtensible Markup Language
Chapter 1: Introduction

In this chapter we discuss the background and the related work for our approach. Section 1.1 talks about the background and gives an introduction to the topic, whereas section 1.2 talks about the related work.

1.1 Background

We are in the information age and in this age, in order to be efficient, it is vital for us to store and retrieve information in the most convenient way. Hosting a website is one of the ways in which information can be organized. A basic website can be developed using HTML (Hyper Text Markup Language) [1]. It can be enhanced by using the client side scripting tools such as JavaScript [2] or can be made feature-rich and powerful by using server side scripting languages such as PHP (Hypertext Pre-Processor) [3], ASP (Active Server Pages) [4] or JSP (Java Server Pages) [5].

While developing a website, one has to have knowledge about web servers and web browsers. A web server (e.g. Apache [6]), presents information in the form of an HTML file to a web browser, such as Mozilla Firefox [7] or Microsoft Internet Explorer [8]. Web browsers are generally freely available to download and use. Additionally one needs to have an FTP (File Transfer Protocol) [9] client to transfer the content files (webpages) to the server space where the web-server resides. Hence developing a website from scratch requires one to undergo the tedious steps and one has to perform the mundane steps even if one has to make a small update. There are tools available to help to create content using WYSIWYG (What-You-See-Is-What-You-Get) editors such as Adobe Dreamweaver [10]. They are useful to rapidly develop webpages, although they
require that the programmer has knowledge of HTML and other web-programming languages. Even after using an editor like Dreamweaver, a programmer would still have to undergo the same series of steps to make the webpage go live on the internet, even for small changes or corrections. On the other hand there are online website builders such as Tripod [11] and Yahoo site builder [12], which allow a novice user to build a website. These packages are, however, limited in their applications and features.

We are looking into developing a specialized kind of website called an “Information Portal”. The word “portal” implies a website that acts as a gateway or entrance to various sites on the Internet. An information portal is a specialized website that is a gateway to all the resources and sources of information for a particular organization. One should not confuse an information portal with a “web portal”. A web portal is a website that offers personalized capabilities to a user, by providing a broad array of services from various sources. An example of a web portal would be www.yahoo.com, which provides services such as free email, chatroom and calendar. On the other hand, an information portal provides tremendous amounts of information relating to various departments of a specific organization. For example, a basic mid-size information portal may link 150-200 webpages related to an organization. If an organization just has a single webmaster or a website developer then it would be an uphill task for the webmaster/developer to create and maintain that information portal.

We need to look into ways of rapidly developing an information portal and also finding ways in which we can involve novice webmasters (people with no knowledge of HTML or any other web-programming language) to contribute toward the content
management on the information portal. This would relieve the lone webmaster of the burden of updating the huge information portal.

1.2 Related Work

Rapid advancement in technology in the last decade has caused advancement in the skills held by people as well. Nowadays, most of the employees in an organization have skills to use the Internet and word processor applications such as Microsoft Word [13]. There has been an effort to utilize these skills held by these employees (novice users or non-programmers) to develop web-applications by masking technical knowledge from them.

We have the idea of rapidly developing an information portal and then utilizing the resources of non-programmers to maintain the information portal. We have looked into the work of Jochen Rode (see Rode in Table 1.1) [14], who has worked on the idea of developing an environment which is conducive for non-programmers to develop and maintain web-applications. Rima Gerhard (see Gerhard in Table 1.1) [15] rebuilt a website using a content management system (CMS) [16] for the Master’s thesis, which was developed to be accessible for updates to be made by non-programmers. Since our case study was to redesign the CORE (Centers for Osteopathic Research and Education) website (into an information portal), we looked into the work of Simpson (see Simpson in Table 1.1) [17] and Dudek et al (see Dudek et al in Table 1.1) [18] who have discussed about their experiences with CMSs.

Jochen Rode [14] in the dissertation has concluded that non-programmers can also create and maintain web-applications if given the specific environment. We are
interested in such an application that would allow us to develop the information portal rapidly and then allow the novice users to contribute and update content. Dudek et al [18] discuss about a website redesign using a content management system, which made it easier to maintain the website in the long run. Rima Gerhard [15] has implemented a website redesign using a content management system in the Master’s thesis. In the thesis, the author (Rima) has implemented a departmental website using a content management system called Zope and then delegated the content management (content creation, deletion and updation) to the departmental staff (novice users).
<table>
<thead>
<tr>
<th>Author</th>
<th>Work</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rode</td>
<td>Jochen Rode “Web Application Development by Nonprogrammers: User-Centered Design of an End-User Web Development Tool”, Dissertation, Virginia Polytechnic Institute and State University, 2005 [14]</td>
<td>Jochen Rode’s work talks about end-user web-application development and focuses on shaping web programming technology and tools according to the end-user’s natural-mental models and expectations. The author’s research is directed toward casual webmasters without programming experience who are interested in developing web applications.</td>
</tr>
<tr>
<td>Dudek et al</td>
<td>D. T. Dudek, H. A. Wieczorek, “A Simple Web Content Management Tool as the Solution to a Web Site Redesign”, ACM SIGUCCS, September 2003 [18]</td>
<td>Here the authors have implemented a simple content management solution for the IT (Information Technology) web site of University of Buffalo. They had to develop the website without having all the project specifications, which implied that the site would undergo layout and navigational changes in the future. They implemented the site successfully by developing a CMS driven website which would incorporate future changes without compromising on the consistency of the website.</td>
</tr>
<tr>
<td>Gerhard</td>
<td>Rima Gerhard “Reimplementation of the Computer and Information Science and Engineering (CISE) Department Website Using a Content Management System”, Dissertation, University of Florida, 2003 [15]</td>
<td>Rima Gerhard’s work focuses on reimplementing a departmental website using a content management system. The author has used the Zope content management system and has developed a website confirming to the University of Florida’s Web Administration group and the College of Engineering. The resulting website is being maintained by people who lack fundamental knowledge of HTML or programming.</td>
</tr>
<tr>
<td>Simpson</td>
<td>Doug L. Simpson, “Content for One: Developing a Personal Content Management System”, ACM SIGUCCS, November 2005 [17]</td>
<td>Doug Simpson talks about the experience of developing a CMS from scratch for personal use and then utilizes a packaged CMS later. The author found the packaged CMS was a better choice, because enhancing the CMS the author developed for personal use would require more time than just using the packaged one. Additionally the packaged CMS offered a lot of features.</td>
</tr>
</tbody>
</table>
We started looking into a class of systems called ‘Content Management Systems (CMS)’ which allow organization of content by differentiating content, style and logic from each other. A special kind of CMS, the web-CMS is independent of the hardware and the operating system of the content-contributor’s computer. One just needs a computer with an internet connection and a web browser to create, delete and update content. There is no need to install any software. One can build a web-CMS from scratch, but it would take a lot of time to implement all the required features [17]. Since we are looking for a rapid development environment (RDE), we decided not to reinvent the wheel. Instead, we can use the available CMS (hereafter CMS refers to web-CMS) and customize it to the benefit of a particular project.

We found that there are commercial and open source [19] solutions. The commercial ones are very expensive to purchase and maintain, while the open source solutions are free to use. Commercial solutions come with technical support, while the open source ones generally don’t have official technical support. Open source solutions do, however, have online forums to help understand certain concepts and may also have some documentation. Open source solutions are not as accessible and intuitive as proprietary (commercial) solutions [20]. We found that the open source solutions’ key advantage was reduced dependence on software vendors [21]. Additionally open source solutions are easy to customize [21]. Since the open source solutions are cost-effective as they involve only the development cost (implementation hours) for the information portal, we thought of concentrating on the open source solutions.
To implement our idea of developing a dynamic, cost-effective and efficient information portal solution, we developed the Centers for Osteopathic Research and Education (CORE) information portal, as a case study. In next chapter we describe the case study project and in chapter three we discuss the open source CMSs we looked into. In chapter four we have described Mambo CMS in detail. Mambo CMS is the CMS which we chose out of the four popular open source CMSs. In chapter 5 we have described the CIP system and in chapter six we have discussed the CIP user interfaces. In chapter seven we have performed initial quantitative analysis to evaluate the applicability of Mambo CMS on a basic mid-size information portal, by evaluating the CIP project. Additionally in chapter 7, we have mentioned the contribution of our work in comparison with other related work.
Chapter 2: Case Study: The CORE Information Portal (CIP)

Project

The Centers for Osteopathic Research and Education (CORE) is an integrated statewide education consortium formed by the affiliations between Ohio University College of Osteopathic Medicine (OUCOM) [22] and fourteen teaching hospitals across the state of Ohio and three other colleges of osteopathic medicine in the United States of America. The list of hospitals and colleges is as follows.

List of Hospitals Affiliated to CORE:

- Akron City Hospital
- Cuyahoga Falls General Hospital
- Doctors Hospital, Columbus
- Doctors Hospital of Stark County
- Firelands Regional Medical Center
- Grandview Hospital and Medical Center
- MetroHealth Medical Center
- O’Bleness Memorial Hospital
- St. Elizabeth Health Center
- St. John West Shore Hospital
- St. Joseph Health Center
- St. Vincent Mercy Medical Health Center
• South Pointe Hospital
• Southern Ohio Medical Center

List of Medicine Colleges Affiliated to CORE:

• Ohio University College of Osteopathic Medicine (OUCOM)
• Kansas City University of Medicine and BioSciences (KCUMB/COM)
• Des Moines University College of Osteopathic Medicine (DMU/COM)
• T. Still University / Kirksville College of Osteopathic Medicine (ATSU/COM)

The old CORE website was built in 2004. After that, the website has not been updated frequently, because every update to the website requires profound knowledge of HTML or other advanced web-programming languages. To make even small changes to content, the webmaster has to undergo a series of steps. For example, the webmaster has to download that particular file, change the content, and then upload the file to make the change. This puts a lot of burden on the lone webmaster. Hence the old website quickly became outdated. The needs of the CORE increased with its advancement and the only way it could keep up was by transforming the website into a specialized website called an information portal.

The CORE had the following requirements for the information portal solution:

• Dynamic system: The information portal needs to be updated frequently. Hence the CORE information portal needs to be dynamic. The CORE required the system to be scalable to its needs in the future.
• Cost-effective: The CORE required the system to be cost-effective. CORE did not want to allocate a large budget to buy a special software solution or to hire more skilled developers to develop and maintain the system.

• Strict Timeline: The CORE wanted at least the site-structure (all of the pages and their links and menu driven architecture without the content) to be complete in sixty days. It implies rapid development of the information portal architecture.

• User friendly: CORE required the content management to be delegated to its staff, but the staff members were novice users with no knowledge of HTML or any other web-programming language. Hence it required the system to be user-friendly for the staff to perform content management.

• Easy-learning curve: The staff members (content contributors) of the CORE are novice computer users, i.e. most of them familiar with Microsoft Word [13] and the use of internet for email application. Easy learning curve would mean a short training period.

• Efficient: Finally, it required the system to be up and running so that all the resources spent in the process produce a successful result. Since a web-CMS is platform independent, there won’t be any need to install any program on the content-contributor’s computer, hence it would improve the efficiency.

The CORE information portal had a low-level security requirement. The implementation of the CORE information portal is discussed in detail in chapter 5 and the user interfaces of the CORE information portal are discussed in chapter 6.
Chapter 3: Open Source Content Management Systems

We looked for an apt open source CMS for the CORE information portal project. When we started looking for an open source CMS, there were a few options available and then later, more and more were available. Due to lack of time, it was not advisable to look into all the CMSs. Hence we picked the ones which were easily available to download, easy to install and work with. It was important that the system was easy to install and configure, so that we could start building the structure of the information portal quickly. Along with rapid development of the information portal, we also wanted it to be robust and scalable. We selected four open source CMSs; this selection was also influenced by our most important requirement of a user-friendly content management system. We looked into the following CMSs:

- Drupal
- PHP-Nuke
- Mambo
- Plone

3.1 Drupal

Drupal [23] is a software which allows the user to easily publish, manage and organize content on a website. Drupal is an open source software and it is licensed under the GNU (GNU’s Not Unix) General Public License (GPL) [24]. It is maintained and developed by thousands of users and developers. Drupal supports multiple platforms, including Unix [25], Linux [26], BSD (Berkeley Software Distribution) [27], Solaris
Drupal provides the user authentication and provides the template feature to separate style from content, and hence allows one to control the look and feel of the website. It allows web administration and hence the entire administration can be done using a web browser. Hence it doesn’t require any additional software to be installed on the computer. We tested and evaluated the Drupal version 4.5.4.

3.2 PHP-Nuke

PHP-Nuke [32] is an PHP based content management system. It is specially designed to be used in Intranets and the World Wide Web. It is a database driven system and utilizes MySQL [33] as the database. It needs Apache web server and PHP version 4.x. PHP-Nuke can support the following operating systems:

- Linux
- Any Unix flavor
- OS/2 [34]
- Any Windows flavor
- MacOS
- FreeBSD [35]

But the system has been highly tested only on Linux. It gives the administrator total control of the website and it supports other registered users. We tested and evaluated the PHP-Nuke Version 7.7 for the CORE information portal project.
3.3 Mambo

Mambo [36] is an open-source web-content management system. It is one of the most successful content management systems in the world [37]. It was originally a commercial system introduced in 2001 by the Australian company Miro [38], which is based in Melbourne. Miro introduced it as an open-source system in 2002 to test it and also to increase the number of users. Miro split its product Mambo into a commercial version called “Mambo CMS” and open-source version called “Mambo Open Source”, “MOS”, or simply “Mambo”. Mambo is available under the GNU General Public License (GPL). In August 2005, the Mambo Foundation [39] was formed as a supporting organization for the Mambo Open Source Project. Mambo is continuously being enhanced by an ever-growing number of users and developers.

Mambo is an PHP and MySQL based CMS. It works best with the Apache webserver and the Linux operating system. It has a user-friendly interface [40]. Mambo can be used for a range of applications such as simple and small personal website or a large organizational website or an information portal. We tested and evaluated the Mambo Version 4.5.1a stable.

3.4 Plone

Plone [41] is built using an object oriented application server called Zope [42]. Plone and Zope are driven by the Python [43] language. Plone is platform independent as it can work on a vast array of platforms, including Linux, Windows, Mac OS X, Solaris and BSD. It supports more than 50 language translations, and hence is multilingual. The Plone foundation was formed to protect it in the year 2004. Plone is an open source
software licensed under the GNU General Public License. Plone also has many add-on products that add new features to it. We tested and evaluated Plone Version 2.0.5.

3.5 Decision Made: Mambo is Selected

After all the speculation we zeroed it on Mambo. Out of the four CMSs we reviewed, Mambo and Plone stood-out; because Mambo and Plone both have a better template management system and they both provide more add-ons. Additionally, we found that Mambo had a better user-friendly interface. Plone uses structured text for its online editor, which is not as user-friendly as the online editor which Mambo uses. Mambo’s online editor is also similar to the widely used Microsoft Word interface.

Additionally, Mambo has one of the most active forums [44]. The forum had answers to most of the preliminary questions about installation and use, and also to some of the advanced questions for customizing the software to one’s own project. Since it has a large user base, it has a plethora of add-ons, which are essential to extend the features of Mambo. Hence, Mambo appeared to be the most suitable choice for the CORE information portal project. Mambo is discussed in detail in the next chapter.
Chapter 4: Mambo in Detail

Mambo is an PHP driven CMS which utilizes the MySQL database. To install Mambo we must have the LAMP (Linux, Apache, MySQL and PHP environment) [45] environment. There are no special requirements on Apache or MySQL. The PHP version needs to be higher than 4.1.2 and it needs to be compiled with support for MySQL, XML (eXtensible Markup Language) [46] and ZLib [47]. The function of the ZLib library is to make it possible for PHP to read file packages that are compressed with the ZIP procedure.

On a Windows environment, one can use the XAMPP [48] project. XAMPP is a complete development environment with all the ingredients necessary for Mambo, i.e. PHP, MySQL and Apache. It also has support for Perl [49].

Since we had to rapidly develop our CORE information portal we bought an economical Linux based server space which already supports all the open source products we required, i.e. PHP, Apache and MySQL.

4.1 Mambo’s Fundamental Concepts

The Mambo CMS has a front-end and back-end. The front-end is the information portal interface, which is what the visitors and the logged in content-contributors see. The back-end contains the administration layer of the information portal for the portal administrator. The content in Mambo is organized using the following hierarchy, i.e. section, category, content-item (item) [50]. A section is a container that holds one or more categories. A category is a container that holds one or more items. Items are articles
that make up the website content. Also there are other ways in which content can be represented, such as static content, which is an item without a section or a category. Content-contributors can perform content management by logging into the front-end or the back-end, depending on their access rights. The content-contributor needs to undergo user authentication to access the front-end WYSIWYG editor or the back-end area.

4.2 User Management and Access Control

The Mambo CMS has two main hierarchies for user groups [51]:

- **Public Front-end**: Access to the front-end; i.e. users who can log in to the portal or website and view the designated sections and pages

- **Public Back-end**: Access to the Back-end Administration

The users belonging to Public Back-end can access the front-end as well.

4.2.1 Public Front-end

This group consists of four types of users:

- Registered
- Author
- Editor
- Publisher

The user with the lowest access rights is the “Registered” and the one with the highest is the “Publisher”. The rights provided to a parent group like the “Registered” are inherited by the child group such as the “Author”. This is generally the case unless the rights are specifically denied by the “Super Administrator” (the group with the highest rights among Public Front-end and Public Back-end).
4.2.1.1 Registered Group

This group allows the user to login to the front-end. Once the user has logged then additional information (sections and pages) may be available.

4.2.1.2 Author Group

These users have access to submit (post) new content and edit their own content items/pages by logging into the front-end. These users can post content by using the link in the User Menu when they have logged in.

4.2.1.3 Editor Group

These users have access to edit and post any content by logging into the front-end.

4.2.1.4 Publisher Group

This group has access to post, edit and publish any content by logging into the front-end.

4.2.2 Public Back-end

This group consists of three types of users:

- Manager
- Administrator
- Super Administrator

The user with the lowest access rights is the “Manager” and the one with the highest is the “Super Administrator”. The rights provided to a parent group like the “Manager” are inherited by the child group such as the “Administrator”. This is generally the case unless the rights are specifically denied by the “Super Administrator” (the group with the highest rights among Public Front-end and Public Back-end).
4.2.2.1 Manager

This group has access to content creation and other system information. The Manager can access the following back-end functions:

- **Menu**: Edit Existing Menus, Including creating Menu Items. All related functions
- **Content**: Section Manager, Category Manager, Content Managers, Static Content Manager, Frontpage Manager and Archives Manager. All related functions.
- **Site**: Preview, Media Manager.
- **Help (general)**, including System Info.

4.2.2.2 Administrator

This group has access to most of the administrative functions. The Administrator can access all the functions accessed by the Manager plus all the following functions:

- **Components, Modules, Mambots**: Edit, Create, Install / Uninstall. No access to the Mass mail component.
- **Site**: User Manager (create or edit or delete any type of user except Super Administrator), Statistics, Trash Manager.
- **Menu**: Menu Manager.

4.2.2.3 Super Administrator

This group has access to all the administrative functions. The Super Administrator can access all the functions accessed by the Manager and the Administrator, plus all the following functions:

- **Components**: Mass Mail.
• System: Global Checkin, Clear Cache (If cache has been enabled in Global Configuration).

• Site: Template Manager, Global Configuration, Language Manager.

• Messages: Configuration, Inbox.

4.3 Mambo Database Tables

The Mambo content management system comes with thirty-four pre-configured database tables. We found a neat representation of them by Brian Diaz at www.howtomambo.com. But later this website got redirected to another website. Anyways one can find the database Scheme of Mambo at the following URL (Universal Resource Locator):

http://www.mamboportal.com/component/option,com_remository/Itemid,46/func,fileinfo/parent,folder/filecatid,1388/ [52]

We have summarized the information about the Mambo database tables in this section. Table 4.1 gives the list of the tables present in Mambo. Tables #1, #2 and #3 store data about the banners. Table #4 stores information about the contacts. Tables #5, #6 and #7 are related to content details. Tables #8 and #9 are about items and search. Tables #10, #11 and #12 store information about the add-ons: Mambots, Modules and Components respectively. Tables #13, #14 and #15 are related to the poll feature of Mambo. Tables #16 and #17 are for menus. Tables #18, #19, #20 and #21 are for access control. Tables #22 and #23 are for messages. Tables #24, #25 and #26 are for users and groups. Tables #27 and #28 are for sections and categories. Tables #29 and #30 are for
32

templates. Tables #31 and #32 are for newsfeeds and sessions respectively. Tables #33 and #34 are for statistics and weblinks respectively.

Table 4.1 List of Mambo database tables

<table>
<thead>
<tr>
<th>Table #</th>
<th>Table Name</th>
<th>Table #</th>
<th>Table Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mos_banner</td>
<td>18</td>
<td>mos_core_acl_aro_group</td>
</tr>
<tr>
<td>2</td>
<td>mos_bannerclient</td>
<td>19</td>
<td>mos_core_acl_aro_sections</td>
</tr>
<tr>
<td>3</td>
<td>mos_bannerfinish</td>
<td>20</td>
<td>mos_core_acl_groups_aro_map</td>
</tr>
<tr>
<td>4</td>
<td>mos_contact_details</td>
<td>21</td>
<td>mos_core_acl_aro</td>
</tr>
<tr>
<td>5</td>
<td>mos_content</td>
<td>22</td>
<td>mos_messages</td>
</tr>
<tr>
<td>6</td>
<td>mos_content_frontpage</td>
<td>23</td>
<td>mos_messages_cfg</td>
</tr>
<tr>
<td>7</td>
<td>mos_content_rating</td>
<td>24</td>
<td>mos_users</td>
</tr>
<tr>
<td>8</td>
<td>mos_core_log_items</td>
<td>25</td>
<td>mos_usertypes</td>
</tr>
<tr>
<td>9</td>
<td>mos_core_log_searches</td>
<td>26</td>
<td>mos_groups</td>
</tr>
<tr>
<td>10</td>
<td>mos_mambots</td>
<td>27</td>
<td>mos_sections</td>
</tr>
<tr>
<td>11</td>
<td>mos_components</td>
<td>28</td>
<td>mos_categories</td>
</tr>
<tr>
<td>12</td>
<td>mos_modules</td>
<td>29</td>
<td>mos_template_positions</td>
</tr>
<tr>
<td>13</td>
<td>mos_polls</td>
<td>30</td>
<td>mos_templates_menu</td>
</tr>
<tr>
<td>14</td>
<td>mos_poll_date</td>
<td>31</td>
<td>mos_newsfeeds</td>
</tr>
<tr>
<td>15</td>
<td>mos_poll_menu</td>
<td>32</td>
<td>mos_session</td>
</tr>
<tr>
<td>16</td>
<td>mos_menu</td>
<td>33</td>
<td>mos_stats_agents</td>
</tr>
<tr>
<td>17</td>
<td>mos_modules_menu</td>
<td>34</td>
<td>mos_weblinks</td>
</tr>
</tbody>
</table>

4.4 Mambo Front-end

The Mambo front-end [37] is the actual appearance of the website or the information portal. When you install the Mambo software, then the sample page you get for the Mambo website has various different kinds of modules. A module in Mambo is an PHP script, which gives a specific functionality to a specific area. A module is designed to handle the display duties of the front-end. These modules correspond to the various features of the website. The menus of the website are due to the “Main Menu” module. The polling function is achieved due to the “Polls” module, and also the RSS (Really
Simple Syndication) [53] feed function is achieved using the “Syndicate” module. Each module on the sample has its own grey bar. This grey bar contains the module name as the title. Under the bar we find the modules’ content.

The important advantage with modules is that a module can be displayed on as many pages as one wants. Also it implies we can switch off a module, i.e. don’t display it on a specific page as well or only make the module available on a specific page.

A Mambo page layout is determined by the template and it is divided into various positions or place holders. A position is basically a table cell with a particular name. A template is a set of files which determines the look and feel of the page and is laid on top of the content like a screen. A position can contain any number of modules and these modules can be switched on or off. The classical positions of a template are left, right and top. Also there are positions without specifications such as user1, user2, user3, user4, etc. The location of positions is regulated by the specific template. We will discuss templates later.

Besides the visible elements, there are also the invisible elements in Mambo. They are:

- **Mambots**: Mambots performs important task such as inserting pictures, paginating the user interface, calculating ratings, etc.

- **Components**: Components are necessary for the display of content. It is responsible for the functionality and the special settings in the administration interface. Components can easily be controlled by the administrator. Polling, Banners and RSS feeds are examples of components which can be controlled by
the administrator. These are also the examples in which modules and components interact. A component cannot be present by its own. Also if a mambot is associated with a module and a component, then a trio can emerge.

You may find information about modules, components and mambots in the section 4.7 (Mambo Add-ons). The content-contributors can change content by logging in the front-end. When the content-contributors log in, then they get access to the particular page and then they can change content using the online WYSIWYG editor.

4.5 Mambo Back-end

The Mambo back-end has a different URL than the front-end. It is the administration section of Mambo. Once the administrators log in, then they get access to all the administrative features of Mambo. Some of the important features are present on the control panel. Also all the features are accessible through the menus at the top of the page.

- Global Configuration: Using the global configuration section one can control settings that apply to the entire website. The global configuration section controls Mambo’s operational settings by updating the “configuration.php” file. This includes switching the entire website offline, selecting a particular WYSIWYG editor if you installed a few of them or switching to no WYSIWYG editor, the title text in the browser window, etc. Also the configuration section gives information and access to change the language and location time of the website. It also gives access to various general features that apply to the content. It gives
information about the database and the server and also access to the frontpage metadata and the statistics and other important functions.

- **Menu Manager:** It is the centralized place to edit and manipulate the various menus of the system. There are four menus: Main menu, User Menu, Top Menu and Other menu. All these menus can be allocated to the required position on the template. One can assign and change menu items for all the menus.

- **Section Manager:** It manages all the sections on the website. Here one can create a new section, delete a section or edit a section.

- **Category Manager:** It manages all the categories on the website. Here one can create a new category, delete a category or edit a category.

- **All Content items:** It consists of all the content items. It is useful to manage all the content items. Here one can create a new item, delete an item or edit an item.

- **Static Content Manager:** It manages all the static content on the website. Here one can create a new static content item, delete a static content item or edit a static content item.

- **Components, Modules and Mambots (CMM):** To install and uninstall CMM and also to configure them.

- **System:** The system menu selection gives the system information. It also gives the option of Global Checkin, which allows a super-administrator to check in all the items.
4.6 Mambo Template Management

A template is a set of files that determines the look of the page or the entire site or of various sections of the site. The template separates the look and feel of the website from the content. It takes care of the style of the website. The template governs the colors, fonts and alignment of the content.

The main file is “index.php“, which consists of the HTML code which controls the layout elements such as tables and logos, and it also inserts the PHP code for all the dynamic elements. The other important file is the style sheet or the Cascading Style Sheets (CSS) [54] file which determines fonts, colors, and borders for each element of the website.

With the Mambo template manager one can change the default template of the entire website or one can apply different templates to different sections of the website. This can be done by logging into the back-end of the portal (specialized website) or a website and then accessing the template manager. Here one can install a new site template or uninstall an old site template. One can assign templates to the font-end of the portal or the sections in the portal. Additionally one can assign a new template to the back-end (the administration section of the portal). One can also install a new administration template using the template manager.

4.7 Mambo Add-ons

An add-on enables to extend the functionality of the Mambo system. There are three types of add-ons:

- Module (visible on the front-end)
4.7.1 Module

A module is an PHP script, which gives a specific functionality to a specific area in the front-end. A module corresponds to the various features of the portal such as Menus and Polls.

4.7.2 Component

A component is an application that adds functionality and special settings in the administration interface. An example of a component is the “DOCMan” [55] document manager and the Events Calendar.

4.7.3 Mambot

A mambot is a small, task-oriented function that intercepts content before it is displayed on the front-end. A mambot performs important task such as inserting pictures and paginating the user interface. The “MOS Image” mambot helps in inserting images using the online editor in the CIP.

4.7.4 More Information About Add-ons

An add-on can be a module, or a component along with a module, or a module with a mambot or all the three together, i.e. a trio of a component, module and a mambot. An example of an add-on is a forum or a news letter maintenance system. This feature of being able to extend Mambo’s features by using add-ons affirms the scalability of Mambo and also makes it more robust and powerful. One can have a look at the various add-ons available for Mambo at the add-on websites [56] [57]. At one of the sites, i.e.
mamboxchange.com [56] there are more than thousand add-ons. They range from a calendar component called “ExtCalendar CSS Events Calendar” to an E-commerce component called “VirtueMart”.
Chapter 5: The CORE Information Portal (CIP) System

The CORE information portal (CIP) system has been developed using the Mambo content management system. It utilizes the Mambo version 4.5.1a stable. The database we are using is MySQL and we are using a web-hosting service which gives us the LAMP (Linux, Apache, MySQL and PHP) environment, which is favorable for this solution.

The web-hosting service uses Dual AMD (Advanced Micro Devices) [58] Opteron 244 1.8 GHz processor, 2 GB RAM memory, two 250 GB hard drive; backup of RAID (Redundant Array of Independent or Inexpensive Disks) 1, and the Debian [59] Linux operating system. Using a web-hosting service really helped because we got a lot of features with it already installed, such as the mail server. We have used “phpMyAdmin” [60] to handle the administration of the MySQL database over the web. “phpMyAdmin” is a tool written in PHP.

The CIP would fall into the category of a basic mid-size information portal as it has more than 170 webpages and the content management on the portal is performed by non-technical users (novice users). These novice users have no web-programming skills. The content contributors (novice users) have skills to use the Internet and Microsoft Word.

The CIP has a front-end and a back-end. The URL for the information portal is the same as the URL for the content-contributors to log in and change content. The back-end is located at a different URL.
5.1 Front-end of the CIP

The front-end of the portal has two characteristic menus (see Figure 5.1). One is the horizontal menu which is the main menu and it includes all the sections in the system, and the other menu is the vertical menu which has a few links associated with it.
Figure 5.1 The CIP Front-end
The portal has the following sections:

- **About us**
  It has five sub-menu items in it. There are four static items and there is one category along with four content-items.

- **Education**
  It is one of the biggest sections of the portal presently. It consists of two static items and three categories. The whole section has more than 70 pages.

- **Members**
  This section has two categories. The first category Hospitals at present has fourteen items and the second category Colleges has four items. This section is basically dedicated to the affiliations of the CORE.

- **Research**
  This section has five static pages at present. All of these pages have a lot of content in them and it is frequently updated.

The other sections of the portal are the Resources, Application, News and Contact Us.

The other important parts of the website are the document repository and search facility. The document repository has been implemented with the help of a powerful add-on called “DOCMan”. DOCMan is a document manager which enables one to organize files by designating them to various categories. This basically helps to organize all the file downloads in one place in an organized way. It also has the user group management option, which allows us to assign restriction to a particular document, by assigning it to a specific group of people. This means for example if the minutes of a meeting should be
visible only to the members of a particular committee, then one can assign a group to all the members of the committee, and then they would be the only group of people who can access those minutes.

Additionally there is an events calendar on the frontpage of the portal. In this calendar one can enlist all the events such as conferences and meetings. A content-contributor (staff member) can log in from the front-end and add an event. It also has a plethora of options such as the day it should be visible and the time it should be visible. The events calendar is an add-on which has been installed in the Mambo system.

The extreme right side of the page has a “Random Image” add-on which changes image every time we refresh the homepage. The front-end has a neat and professional look because of the template. We will discuss the template in the following section 5.2.

5.2 Template of the CIP

The template of the front-end of the system has in all six positions (named table cells including left, right, top, user1, user2, and user5). The template layout is basically governed by the “index.php” file and the colors and fonts are governed by the CSS file. We have utilized a three column structure to enhance the features and look of the portal. The main content of the page is always in the center of the page, as we can see in the Figure 5.2.

The positions have been utilized by assigning components and modules to them, such as the “Events Calendar” module has been assigned to the user1 position and the “Random Image” module has been assigned to the right position. This also enhances the
flexibility of the portal, as we can change the positions of modules whenever we want, by logging in to the administrative back-end.
Figure 5.2 The Front-end Template with Inline Positions
5.3 Back-end of the CIP

The back-end of the portal (see Figure 5.3) has been protected by dual authentication mechanism. This has been done to enhance the security of the backend. The back-end section of the portal, which is basically an interface to the engine of the entire system, has all the administrative features.

In the back-end administrative section we can manage the following:

- menus of the front-end
- all the content items
- sections
- categories
- static items
- components
- modules
- mambots
- Template manager
- User Manager
- Site statistics
Figure 5.3 The Back-end of the CIP System
5.4 The Database Tables of the CIP

The CIP has 49 database tables (see Table 5.1). We have added 15 more tables to the initial 34 database tables, which Mambo initially had.

Table 5.1 Database tables of the CIP

<table>
<thead>
<tr>
<th>Table #</th>
<th>Table Name</th>
<th>Table #</th>
<th>Table Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mos_banner</td>
<td>26</td>
<td>mos_poll_menu</td>
</tr>
<tr>
<td>2</td>
<td>Mos_bannerclient</td>
<td>27</td>
<td>mos_polls</td>
</tr>
<tr>
<td>3</td>
<td>Mos_bannerfinish</td>
<td>28</td>
<td>mos_sections</td>
</tr>
<tr>
<td>4</td>
<td>mos_categories</td>
<td>29</td>
<td>mos_session</td>
</tr>
<tr>
<td>5</td>
<td>mos_components</td>
<td>30</td>
<td>mos_stats_agents</td>
</tr>
<tr>
<td>6</td>
<td>mos_contact_details</td>
<td>31</td>
<td>mos_template_positions</td>
</tr>
<tr>
<td>7</td>
<td>mos_content</td>
<td>32</td>
<td>mos_templates_menu</td>
</tr>
<tr>
<td>8</td>
<td>mos_content_frontpage</td>
<td>33</td>
<td>mos_users</td>
</tr>
<tr>
<td>9</td>
<td>mos_content_rating</td>
<td>34</td>
<td>mos_usertypes</td>
</tr>
<tr>
<td>10</td>
<td>mos_core_acl_aro</td>
<td>35</td>
<td>mos_weblinks</td>
</tr>
<tr>
<td>11</td>
<td>mos_core_acl_aro_groups</td>
<td>36</td>
<td>mos_docman</td>
</tr>
<tr>
<td>12</td>
<td>mos_core_acl_aro_sections</td>
<td>37</td>
<td>mos_docman_groups</td>
</tr>
<tr>
<td>13</td>
<td>mos_core_acl_groups_map</td>
<td>38</td>
<td>mos_docman_history</td>
</tr>
<tr>
<td>14</td>
<td>mos_core_log_items</td>
<td>39</td>
<td>mos_docman_licenses</td>
</tr>
<tr>
<td>15</td>
<td>mos_core_log_searches</td>
<td>40</td>
<td>mos_docman_log</td>
</tr>
<tr>
<td>16</td>
<td>mos_groups</td>
<td>41</td>
<td>mos_events</td>
</tr>
<tr>
<td>17</td>
<td>mos_mambots</td>
<td>42</td>
<td>mos_events_categories</td>
</tr>
<tr>
<td>18</td>
<td>mos_menu</td>
<td>43</td>
<td>mos_gracess</td>
</tr>
<tr>
<td>19</td>
<td>mos_messages</td>
<td>44</td>
<td>mos_gracess_groupmenu</td>
</tr>
<tr>
<td>20</td>
<td>mos_messages_cfg</td>
<td>45</td>
<td>mos_gracess_usergroup</td>
</tr>
<tr>
<td>21</td>
<td>mos_modules</td>
<td>46</td>
<td>mos lxmenu</td>
</tr>
<tr>
<td>22</td>
<td>mos_modules_menu</td>
<td>47</td>
<td>mos lxmenu_config</td>
</tr>
<tr>
<td>23</td>
<td>mos_newsfeeds</td>
<td>48</td>
<td>mos lxmenu_main</td>
</tr>
<tr>
<td>24</td>
<td>mos_poll_data</td>
<td>49</td>
<td>mos lxmenu_sub</td>
</tr>
<tr>
<td>25</td>
<td>mos_poll_date</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tables #1 to #35 are the same as the ones described in section 4.3, except table #25 which is for Poll date. Tables #36, #37, #38, #39 and #40 are for the add-on “DOCMan”, which is a document management system. Tables #41 and #42 are for the
add-on “Events Calendar”. Tables #43, #44 and #45 are for user and group access. Tables #46, #47, #48 and #49 are for the add-on “Lxmenu”, which is a dynamic menu management system.
Chapter 6: User Interfaces of the CORE Information Portal (CIP)

In this section we will talk about the user interface for the visitors; i.e. front-end, the user interface for the content-contributors; i.e. the front-end WYSIWYG editor, and the user interface for the administrators, i.e. the back-end.

6.1 User Interface for the Visitors

The front-end is the URL which a visitor types in the browser to access the CIP (see Figure 6.1). It has a pleasant, neat and professional look because of the templating system. We have taken special care that we do not clutter the frontpage. This is really important to not to confuse the visitor.

The visitors to this site are a diverse group of people, i.e. medical students, researchers, faculty, staff members, and administrators. The user interface of the front-end is customized to service all these different kinds of audience. We also have a search option to enable the visitors to search for the content they require. The easy and non-cluttered homepage is the area of web development which is important, as the visitors play the major role in the success of information portals.
Figure 6.1 The User Interface for the Visitors of the CIP
6.2 User Interface for the Content-Contributors

The user interface for the content-contributors is the front-end WYSIWYG editor (see Figure 6.2). The front-end WYSIWYG editor has the features similar to Microsoft Word, which most of the content-contributors (users), i.e. staff members are used to. The editor is available online on the CORE Information portal, and it does not require installing any additional software on the user’s computer. The user interface is user-friendly and easy to learn. User-friendly interface for the CIP was an important requirement of the project.

Since the editor is similar to using Microsoft Word processor software to make documents, the users don’t need to have any knowledge of HTML to use it. Also the results are effective as can be observed on the CORE portal.
Figure 6.2 The Font-end WYSIWYG Editor
6.3 User Interface for the Administrators

The user interface for the administrator (see Figure 6.3) is the back-end of the CORE information portal. This back-end administrative interface is also a graphical user interface like the front-end and hence it is easy to use and learn. It lays out some of the prominent features of the back-end in the control panel. Additionally all the features are available by accessing the menus at the top.

All the functions that can be performed from the front-end can also be performed from the back-end. In addition to those functions, the back-end can perform many more functions, such as creating new content and installing a new module.
Figure 6.3 The User Interface for the Administrators
Chapter 7: Evaluation

Here in section 7.1 we have evaluated the CORE Information Portal (CIP), to test our approach. In section 7.2 we have compared our approach to the approaches of the other related works.

7.1 Evaluation of the CORE Information Portal (CIP)

In this section we have evaluated the CIP to test our approach and mentioned our results. We have evaluated the CIP on the following attributes:

- **Ease of Use** – The usability characteristic and learning experience of the content-contributors and administrators of the CIP (section 7.1.1, page 56)
- **Rapid Development** – The speed and time taken for the development of the portal architecture and the overall development of the portal (section 7.1.2, page 70)
- **Total Cost of Ownership** – The total cost to develop and maintain the CIP (section 7.1.3, page 71)
- **Efficiency** – Efficiency with regards to a web-application developer. The effectiveness of the CIP in comparison with the old CORE website (section 7.1.4, page 72)
- **Dynamism** - The extensible nature of CIP due to Mambo (section 7.1.5, page 74)

### 7.1.1 Ease of Use

We conducted two surveys with the users (content-contributors and administrators) of the CIP. The first survey was conducted before the training workshop and the other was conducted after the workshop. Twelve users participated in both the
surveys. The purpose of the first survey is to determine the technical and web-development competency of the users. Additionally the survey has a few open-ended questions regarding the old CORE website. The purpose of the second survey is to evaluate the ease of use and user-friendliness of the new CORE website (CIP). It has a few open-ended questions regarding the CIP. The following subsections summarize the results of the two surveys. The subsections 7.1.2.1 to 7.1.2.3 correspond to survey 1. The subsections 7.1.2.4 to 7.1.2.6 correspond to survey 2.

7.1.1.1 Web-development competency of users

The questions 7 to 14 evaluate the web-development competency of the users (survey 1 is attached in Appendix A.2). Here we collected all the responses by all the users and then we analyzed the data using SPSS [61]. We have used a score to evaluate the web development competency of the users. Each of these questions has five options. The option name and corresponding score is given in Table 7.1.

Table 7.1 Web-development competency measurement scale

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Poor</td>
<td>1</td>
</tr>
<tr>
<td>Poor</td>
<td>2</td>
</tr>
<tr>
<td>Fair</td>
<td>3</td>
</tr>
<tr>
<td>Good</td>
<td>4</td>
</tr>
<tr>
<td>Excellent</td>
<td>5</td>
</tr>
</tbody>
</table>

We added up the scores for each user. The lowest score a user could get is ‘8’, whereas the highest score a user could get is ‘40’. For a user to have minimum level of web-development competency, the user needs to achieve a score of at least 25. This has
been highlighted by the green line in the Figure 7.1. Also the lower threshold of ‘8’ has been highlighted by the blue line. We can see from the graph that none of the users have touched or crossed the green line (score 25). Hence we can conclude that 100% of the users have “Poor” or “Very Poor” web-development competency. The highest score was just 20. This conveys that none of the users have any substantial web-development knowledge.

![Bar Graph of Web-development Competency Scores by User ID](image.png)

**Figure 7.1 Bar graph of Web-development Competency Score by User ID**

Table 7.2 gives us interesting results. 91.7 % users (i.e. 11 out of 12 users) have never developed a website before. Only 33.3 % users (i.e. 4 out of 12 users) have
maintained a website before. Of all the users who said “No” to website development i.e. 11 users, 72.7% users (i.e. 8 users) said “No” to website maintenance. Of all the users who said “Yes” to website maintenance (4 users), only 25% of them, i.e. only one user said “Yes” to website development. We can conclude that the users (content-contributors) have little or no knowledge of website development or website maintenance.

Table 7.2 Website development and website maintenance cross-tabulation

<table>
<thead>
<tr>
<th>Website Development</th>
<th>Website Maintenance</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>% within Website Development</td>
<td>72.7%</td>
<td>27.3%</td>
</tr>
<tr>
<td>% within Website Maintenance</td>
<td>100.0%</td>
<td>75.0%</td>
</tr>
<tr>
<td>% of Total</td>
<td>66.7%</td>
<td>25.0%</td>
</tr>
<tr>
<td>Yes</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>% within Website Development</td>
<td>0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>% within Website Maintenance</td>
<td>0%</td>
<td>25.0%</td>
</tr>
<tr>
<td>% of Total</td>
<td>0%</td>
<td>8.3%</td>
</tr>
<tr>
<td>Total</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>% within Website Development</td>
<td>66.7%</td>
<td>33.3%</td>
</tr>
<tr>
<td>% within Website Maintenance</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>% of Total</td>
<td>66.7%</td>
<td>33.3%</td>
</tr>
</tbody>
</table>
7.1.1.2 Summary of present technical competencies

We found in the responses of the first survey that majority of the users have used the internet and Microsoft Word. Hence we thought of cross-tabulating the responses for the questions regarding internet, browser and Microsoft Word.

**Table 7.3 Internet use and Word knowledge cross-tabulation**

<table>
<thead>
<tr>
<th>Internet Use</th>
<th>Word Knowledge</th>
<th>Fair</th>
<th>Good</th>
<th>Excellent</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Count</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>% within Internet Use</td>
<td>8.3%</td>
<td>33.3%</td>
<td>58.3%</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>% within Word Knowledge</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>% of Total</td>
<td>8.3%</td>
<td>33.3%</td>
<td>58.3%</td>
<td>100.0%</td>
</tr>
<tr>
<td>Total</td>
<td>Count</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>% within Internet Use</td>
<td>8.3%</td>
<td>33.3%</td>
<td>58.3%</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>% within Word Knowledge</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>% of Total</td>
<td>8.3%</td>
<td>33.3%</td>
<td>58.3%</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

We found that more than 91.6% of the users (i.e. 11 out of 12 users) had either “Good” knowledge or “Excellent” knowledge of Word (see Table 7.3). Additionally 100% users have used the internet.
Table 7.4 Browser and Word knowledge cross-tabulation

<table>
<thead>
<tr>
<th>Browser</th>
<th>Internet Explorer</th>
<th>Count</th>
<th>Fair</th>
<th>Good</th>
<th>Excellent</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>% within</td>
<td></td>
<td></td>
<td>10.0%</td>
<td>40.0%</td>
<td>50.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>Browser</td>
<td></td>
<td></td>
<td>100.0%</td>
<td>100.0%</td>
<td>71.4%</td>
<td>83.3%</td>
</tr>
<tr>
<td>% within</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>Word Knowledge</td>
<td></td>
<td></td>
<td>8.3%</td>
<td>33.3%</td>
<td>41.7%</td>
<td>83.3%</td>
</tr>
<tr>
<td>% of Total</td>
<td></td>
<td></td>
<td>8.3%</td>
<td>33.3%</td>
<td>41.7%</td>
<td>83.3%</td>
</tr>
<tr>
<td>Mozilla</td>
<td>Firefox</td>
<td></td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>% within</td>
<td></td>
<td></td>
<td>.0%</td>
<td>.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>Browser</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>% within</td>
<td></td>
<td></td>
<td>.0%</td>
<td>.0%</td>
<td>28.6%</td>
<td>16.7%</td>
</tr>
<tr>
<td>Word Knowledge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16.7%</td>
<td>16.7%</td>
</tr>
<tr>
<td>% of Total</td>
<td></td>
<td></td>
<td>.0%</td>
<td>.0%</td>
<td>16.7%</td>
<td>16.7%</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>% within</td>
<td></td>
<td></td>
<td>8.3%</td>
<td>33.3%</td>
<td>58.3%</td>
<td>100.0%</td>
</tr>
<tr>
<td>Browser</td>
<td></td>
<td></td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>% within</td>
<td></td>
<td></td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
</tbody>
</table>
| Word Knowledge|                   |       | 8.3% | 33.3%| 58.3%     | 100.0%

83.3% of the users (i.e. 10 users) use Internet Explorer to browse the internet (see Table 7.4). Only 16.7% of the users (2 users) use Mozilla Firefox to browse the internet. Of all the users who use Internet Explorer i.e. 10 users, 90% users (9 users) said “Good” or “Excellent” about their Word knowledge. All the users who use Firefox i.e. 2 users, 100% users have excellent knowledge of Word. Additionally of all the users who said “Excellent” about their Word knowledge, i.e. 7 users, 71.4% users (5 users) use Internet Explorer.
Hence we can conclude that Internet Explorer is the most popular browser and most users know Microsoft Word.

7.1.1.3 Confirmation of non-user-friendliness of old website

The old CORE website was made using HTML, ASP and with Microsoft Access [62] as the database. The web-application developer then had implemented a CMS for masking the users from the technical details of website development and maintenance. Questions 15, 16 and 17 in survey 1 were dedicated to this aspect of the old CORE website.

7.1.1.3.1 Content creation or updation

Updation of content was basically done by a webmaster as one of the respondents said,

Material to be uploaded was difficult to create or obtain. Once in hand, it was necessary to use at least one if not two intermediary people to post the material to the site. Often, it was posted incorrectly or not at all. Changes to the design were not possible because the prior developer had created a closed page which was ‘locked’ for design changes. Addition of ‘buttons’ etc was not feasible and changes to the graphic (graphical layout) were impossible.

One of the users said,

Done (website maintenance done) by support staff. Administration has many documents and lists of contact information that are simply uploaded to the website.

One user who had maintained content on the old CORE website responded as follows,
On the old CORE web site, I would have to cut and paste text in individual paragraph place holders. In order to add a paragraph, I would have to copy and paste each existing paragraph into a new holder and then create new text in the vacated slot; you could only add paragraphs on to the end of your text stream.

Additionally one more user said,

To update the ‘old’ CORE website, I would contact a programmer and ask for certain changes to be made. This was the only means I had to making such changes.

7.1.1.3.2 Time to update or create content

The CMS implementation of the old CORE website wasn’t user friendly. It took time for updating the website in general. One of the users said,

Creation of the material was not necessarily an issue except for production time. Updates of material were difficult to research and collate, and were time intensive because of prompt calls required. Once in hand, it required uploading by someone who had web experience or at least understood the present site. Often weeks or months would lapse between data collection and uploading.

Another user said,

This depended on how busy the programmer was when I contacted him/her regarding the necessary changes.

One of the users who had maintained content on the old CORE website said,

Updating the Old CORE site took quite a bit of time for some items, but other items could be updated relatively quickly.
7.1.1.3.3 Convenience to update or create content

One of the users responded,

Very inconvenient (to update or create content). Staff avoided it because of complexity. Content could be created in Microsoft products; however there was not sufficient knowledge among users’ staff to upload the material.

One of the users (technical user) who had worked with the maintenance of the old CORE website said,

I would not say that it was convenient, but it wasn’t really difficult. It just took time to cut and paste, and I could not always count on formatting to survive the process.

7.1.1.4 Maintenance of content on new website (CIP)

In survey 2, questions 3 to 8 were regarding the requirement of web-programming knowledge to maintain content on the CIP. We have tabulated the results below:

Table 7.5 Web-programming knowledge required for maintaining content on CIP

<table>
<thead>
<tr>
<th>Web Development Tools/Languages</th>
<th>Need to Know</th>
<th>No Need to Know</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>Microsoft FrontPage [63] or Dreamweaver</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>PHP</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>MySQL</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>JavaScript</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>CSS</td>
<td>0</td>
<td>12</td>
</tr>
</tbody>
</table>

100% of the users (i.e. 12 out of 12 users) said ‘No’ to the requirement, i.e. they do not require knowing any web-programming knowledge to maintain or create content on the CIP (see Table 7.5).
7.1.1.5 Summary of user-friendliness of the CIP interface (back-end / front-end)

Here we have taken into account the experience of the users with the two interfaces, i.e. the front-end and the back-end. Additionally we take into account how they felt about the front-end interface in comparison to Microsoft Word, since 100% of the users are familiar with Word.

7.1.1.5.1 Evaluation of user-friendliness of the CIP interfaces

In survey 2, responses to questions 1, 9 and 10 were recorded to answer this aspect. We performed descriptive statistical analysis on the data obtained for the above questions. The result of the analysis is shown below:

**Table 7.6 Summary of user-friendliness of CIP interfaces**

<table>
<thead>
<tr>
<th></th>
<th>Number of User Responses</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Content Management on CIP</strong></td>
<td>12</td>
<td>(Very Easy) 1</td>
<td>(OK) 3</td>
<td>2.00</td>
<td>.739</td>
</tr>
<tr>
<td><strong>Content-Contributor Interface on CIP</strong></td>
<td>12</td>
<td>(Very User Friendly) 1</td>
<td>(OK) 3</td>
<td>1.92</td>
<td>.669</td>
</tr>
<tr>
<td><strong>Admin Interface on CIP</strong></td>
<td>12</td>
<td>(Very User Friendly) 1</td>
<td>(OK) 3</td>
<td>2.42</td>
<td>.669</td>
</tr>
</tbody>
</table>

The first record in the Table 7.6, i.e. “Content Management CIP” is about how did the user find changing or creating content on using the Mambo CMS. It has five options: Very Easy (1), Easy (2), OK (3), Difficult (4) and Very Difficult (5). Since the maximum
is 3, none of the users answered “Difficult” or “Very Difficult”. The mean is at perfect 2.00, which implies the users found it “Easy” to change or create content on the CIP. Also the standard deviation is a low 0.739.

The second record in Table 7.6, i.e. “Content–Contributor interface on CIP” is about how the users found the Content-Contributor interface. It has five options: Very User Friendly (1), User Friendly (2), OK (3), Least User Friendly (4) and Not User Friendly (5). Since the maximum is 3, 100% users answered either between options 1 to 3. The mean is 1.92, which implies “User Friendly” Front-end interface. Also the standard deviation is a low number (0.669).

The third record in Table 7.6, i.e. “Admin interface on CIP” is about how the user found the administrator interface. It has 6 options, the first five options are similar to the second record and the sixth option is “Not Applicable”, as not all users have to learn about the Admin interface. 100% of the users answered between options 1 to 3. Since this survey was performed just after the first training session, the users were not much adapted with the admin interface as the first training session didn’t cover much of the admin interface.

Hence we got a higher mean of 2.42, which is still closer to “User Friendly” than “OK”. But if you take into account the standard deviation than it might become “OK” as well.

7.1.1.5.2 Similarity of the CIP online editor to Microsoft Word

This aspect of the survey was depicted by the question 2 of survey 2. We recorded all the responses and then tabulated them. We took the responses for “Very Similar” and
“Similar” together and clubbed them as “Similar” and we did the same for “Different” and “Very Different” and clubbed them into “Not Similar”.

Table 7.7 Table for similarity of the CIP online editor to Microsoft Word

<table>
<thead>
<tr>
<th>Similar</th>
<th>Not Similar</th>
<th>No Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Ten out of 12 users said that the CIP online editor was “Very Similar” or “Similar” to Microsoft Word (see Table 7.7). Hence we can conclude that the CIP online editor interface is similar to the Microsoft Word interface.

7.1.1.6 Summary of ease of use / ease of learning of new system

In survey 2, questions 11, 12 and 13 are the open-ended questions which are regarding the ease of learning and ease of use of the system. We have discussed the overall experience, the time required update content and the learning and use experience.

7.1.1.6.1 Overall experience of content management on the CIP

We have recorded a table of all the quotes we received in the second survey for question 11 (see Table 7.8). The general overview is that changing and creating content was easy and it wasn’t as hard it seemed initially.

Table 7.8 Responses for the overall content management experience

<table>
<thead>
<tr>
<th>Respondent</th>
<th>Quote</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mambo is clearly a better user interface for page editing than “CommonSpot” [64], OU’s (Ohio University’s) CMS. However, it's important to understand some points about Mambo's data storage model</td>
</tr>
</tbody>
</table>
before attempting to do very much. For example, you need to know that images have to be stored in the image bank; they can't just be pasted on a page (unfortunately). Once you understand how Mambo is structured, the user interface is pretty good.

2 I was very anxious at first, but once I started using the program I found that it was set up in a somewhat elementary manner making it much more user friendly.

3 Extremely easy

4 I initially had to try and bring over content from the Old CORE site; the learning curve was not as steep as I had thought it might be. Once I felt comfortable that I could import text, I did so. I was able to import data on 12 hospitals in a relatively short period of time. I later worked at editing some of the research office pages and was able to jump right in and change the text easily.

5 Much more enjoyable and much less stressful….

6 Very easy to use; very few problem (like being ‘timed out’).

7 Extremely easy to use.

8 I have been through one training session which was easy and appeared quite understandable. I have not returned to the content “add” site because of password issues. I have reviewed the content extensively to monitor what has been posted (or not) by those responsible for the various sections. Overall the updating and content management seems straightforward. The limiting factor seems to be the time required or the accountability factor for those charged with updating a given area.

9 Fairly easy – uploading documents could be made easier, perhaps?

### 7.1.1.6.2 Time to update content on CIP

We recorded the responses for the time required to update some piece of content.

The general overview is that it doesn’t take much time after you are aware of how to update the content (see Table 7.9).

### Table 7.9 Responses for time to update content on CIP

<table>
<thead>
<tr>
<th>Respondent</th>
<th>Quote</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No longer than it would take to cut and paste, or type new content into MS Word. Saving could be slow sometimes.</td>
</tr>
<tr>
<td>2</td>
<td>Five to ten minutes</td>
</tr>
</tbody>
</table>
Once I was at the proper point on the page, adding text only took as long as the typing. Then saving/publishing was just a couple of clicks with the mouse.

A matter of minutes……

Very little time was needed…just the amount necessary to type the changes/additions

Less than 3 minutes from opening the browser.

10-15 minutes after becoming familiar with the template and admin interface.

Very little time – streamlining formatting was more difficult (in other words, making all sections look like content previously uploaded)

### 7.1.1.6.3 Use and learning experience of the content-contributor interface

We have recorded the responses for question 13 of survey 2. The general consensus is that it is fairly easy to learn and use the content-contributor interface (see Table 7.10).
Table 7.10 Responses for use and learning experience of the content-contributor interface

<table>
<thead>
<tr>
<th>Respondent</th>
<th>Quote</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I didn't spend enough time to become an expert, but I think after becoming familiar with it, a person could stay competent if they used the interface at least a couple of times a week. Daily use would make it very comfortable in a short time.</td>
</tr>
<tr>
<td>2</td>
<td>Yes!</td>
</tr>
<tr>
<td>3</td>
<td>Very easy</td>
</tr>
<tr>
<td>4</td>
<td>Yes.</td>
</tr>
<tr>
<td>5</td>
<td>Very easy.....</td>
</tr>
<tr>
<td>6</td>
<td>Yes, mostly easy.</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>Yes</td>
</tr>
</tbody>
</table>

7.1.2 Rapid Development

Rapid development is one of the important aspects of our approach. The CIP was implemented with the Mambo CMS. Due to time constraints it was important to complete the information portal architecture in time. We were able to complete the entire CIP architecture within sixty days. We started with the requirements stage, which was a little time-intensive, but then the analysis, design and implementation of the portal were done in a fast–paced manner. Once the entire architecture was up and running we started to focus on the training aspect of the portal.

We organized one training session for the content-contributors and then we were astonished by the results. The results were compelling, as 60% of the content on the portal was complete just after the first training session. This points out to the ease of learning of the CIP online editor.
Hence not only the CIP architecture was developed rapidly, but also the content management was done quickly, which in all resulted in the overall rapid development of the portal.

### 7.1.3 Total Cost of Ownership

If the CORE had opted for a commercial off the shelf CMS like “Microsoft CMS” [65] or “PaperThin Inc’s CommonSpot” [64], the CORE would have to spend money on the purchase of the software and then on the yearly license, maintenance and technical support. These costs generally range from $15,000 - $150,000 [66]. Additionally apart from these costs, CORE would still have to pay the IT staff for the efficient utilization of the system.

On the other hand by using the Mambo CMS (refer chapter 3 to find why we chose Mambo over other open source CMSs), CORE has to pay for only one IT employee (web-application developer) and then they can use the system year after year, as there is no purchase or licensing cost involved in using the Mambo CMS. Also having just one web-application developer is enough to maintain the entire Mambo driven CIP. This aspect of requiring only one web-application developer reduces the cost of developing and maintaining the portal considerably. Otherwise for an information portal of more than 150 pages, an organization needs to have an on staff web-development team, which would at least have four to five web-application developers. Rapid development (see section 7.1.2) of the CIP also helped in reducing the cost of ownership of the CIP.
For the CIP, we are training the current CORE employees to use the system. The employees are going through training sessions of varying length depending on their roles for maintaining the CIP. Hence we are utilizing the available resources of the CORE. The result is that the CORE has to spend little as compared to developing and maintaining the portal using a commercial CMS.

7.1.4 Efficiency

First, we will talk about efficiency with regards to a technical user (web-application developer). During the development phase, it was easy to add or delete Mambo Modules (see section 4.7.1) or Components (see section 4.7.2) or Mambots (see section 4.7.3) due to easy-to-use back-end interface. One can easily install or uninstall a Mambo add-on for testing it, and hence the portal was easily extensible. This helped us as we were installing and testing Mambo add-ons. Easy extensibility feature helped to reduce the development time appreciably. Additionally, in the maintenance phase we were able to add or delete a Mambo add-on easily without affecting the portal architecture. Since, Mambo has a modular architecture it is not cumbersome to make changes to the portal architecture even when the entire portal is up and running. Hence CIP is efficient with regards to a web-application developer due to the following:

- Rapid Development (see section 7.1.2) saves time and money
- Easily extensible system saves time
- Easy to maintain system saves time and money

Now we will talk about efficiency with regards to a non-technical user (content-contributor). Initially with the old CORE website, if the CORE employee had to make a
change to the CORE website, then the employee had to get in touch with an IT staff member to make a change. Even when the change was as small as a spelling error, the employee couldn’t do it independently. Hence the IT staff was burdened with a lot of requests, which turned into a lot of frustration and delay. This resulted into a bottleneck.

One of the respondents in the survey said the following about the old CORE website,

To update the ‘old’ CORE website, I would contact a programmer and ask for certain changes to be made. This was the only means I had to making such changes.

Another respondent said this about the old CORE website,

Material to be uploaded was difficult to create or obtain. Once in hand, it was necessary to use at least one if not two intermediary people to post the material to the site. Often, it was posted incorrectly or not at all. Changes to the design were not possible because the prior developer had created a closed page which was ‘locked’ for design changes. Addition of ‘buttons’ etc was not feasible and changes to the graphic were impossible.

Now with the CIP, the CORE employees can make changes to the portal on their own. The CIP is being maintained by the staff members of the CORE; hence the CIP has led to the efficient utilization of the staff resources. It has also led to the empowerment of the CORE employees as the employees now can make changes to their respective section on their own. One of the respondents of the survey said the following about the CIP,

Much more enjoyable and much less stressful (compared to the old CORE website).
Another respondent said this about the CIP,

I initially had to try and bring over content from the Old CORE site; the learning curve was not as steep as I had thought it might be. Once I felt comfortable that I could import text, I did so. I was able to import data on 12 hospitals in a relatively short period of time. I later worked at editing some of the research office pages and was able to jump right in and change the text easily.

The CIP has aided in the streamlined management of the portal, due to the distribution of various roles (see section 4.2) to the CORE employees. Additionally the CIP has led to improved workflow and accountability as the employees with their respective roles are responsible for the content management of their respective sections. The entire CIP has a consistent look and feel. The portal now looks more professional and appealing. Hence the CIP has proved to be an efficient system in comparison with the earlier system (the old CORE website)

7.1.5 Dynamism

The CIP system is extensible as it can be extended by installing add-ons such as Mambo Component (see section 4.7.2), Mambo Modules (see section 4.7.1), or Mambo Mambots (see section 4.7.3). The Original Mambo system (Mambo version 4.5.1a stable) came with just ten built-in modules. We have extended Mambo system to develop the CIP. We have added features such as “DOCMan” document management system to the CIP. DOCMan is a trio of a component, module and a mambot. It is a gigantic document repository which hosts all the documents of the entire CORE and segregates the documents into various folders without any hassle.
The portal is being updated frequently and the links between pages are all working. The pages are being updated in a timely fashion. All these were the requirements laid out by CORE for the CIP. With the CIP, not only a technical user can update and change content, but also a novice user can update and change content. Hence, this points out to the dynamic nature of the system, wherein everybody contributes to the change and progress of the portal.

The other dynamic nature of the CIP is that it targets to a variety of audience and still keeps it simple for the administrator to manage and maintain the CIP. For example, the CIP has secure password-protected areas (confidential documents) which can be accessed by only some CORE personnel. This and many other features point to the dynamic nature of the CIP.

7.2 Comparison of Our Work with the Related Work

We discussed the related work with a tabular representation in section 1.2 (see Table 1.1). Here we have compiled a tabular representation of the contribution of our work in comparison with the related work (see Table 7.11).
Table 7.11 Tabular comparison of our work with the related work

<table>
<thead>
<tr>
<th>Author</th>
<th>Work</th>
<th>Comparison with our work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rode</td>
<td>Jochen Rode “Web Application Development by Nonprogrammers: User-Centered Design of an End-User Web Development Tool”, Dissertation, Virginia Polytechnic Institute and State University, 2005 [14]</td>
<td>Rode’s work is targeted at the end-users who are casual webmasters, i.e. they are users without programming experience, but have the interest in developing web-applications. Our work is directed toward the maintenance of a web-application, specifically an information portal by users who have no HTML or programming skills. 91.7% of the users of the CORE information portal have never developed a website before (Table 7.2).</td>
</tr>
<tr>
<td>Dudek et al</td>
<td>D. T. Dudek, H. A. Wieczorek, “A Simple Web Content Management Tool as the Solution to a Web Site Redesign”, ACM SIGUCCS, September 2003 [18]</td>
<td>The authors have developed a simple content management solution for designing the IT department website. Our work is directed toward rapid development of an information portal. The proof of concept is the CORE information portal, in which the portal architecture was developed within sixty days and then the content management was delegated to the CORE staff (novice users).</td>
</tr>
<tr>
<td>Gerhard</td>
<td>Rima Gerhard “Reimplementation of the Computer and Information Science and Engineering (CISE) Department Website Using a Content Management System”, Dissertation, University of Florida, 2003 [15]</td>
<td>Gerhard’s work is directed toward reimplementing the CISE department website using the Zope content management system. Our work is directed toward rapid development of an information portal, which we have demonstrated by the case study of the CORE information portal. The content contributor interface and administrator interface of our system are user friendly (Table 7.6). In Gerhard’s work, one can add new areas to a website only with a good understanding of Zope. That implies one requires good technical knowledge to add a new web-page or an area. This is not the case in the CORE information portal, since we are using Mambo.</td>
</tr>
<tr>
<td>Simpson</td>
<td>Doug L. Simpson, “Content for One: Developing a Personal Content Management System”, ACM SIGUCCS, November 2005 [17]</td>
<td>Simpson has utilized a packaged CMS for developing a personal website. Our work is directed toward utilizing the resources of novice users, who have no web-programming knowledge (Figure 7.1) for the maintenance of the information portal. We have demonstrated it with the development of the CORE information portal.</td>
</tr>
</tbody>
</table>
Chapter 8: Conclusion and Future Work

In this section we have summarized why Mambo is better than manually developing an information portal (not using a CMS or developing your own CMS), a commercial CMS and other open source CMSs for a basic mid-size information portal, i.e. an information portal with more than 150 webpages. We have demonstrated that Mambo is better by using it to develop the case study: the Centers for Osteopathic Research and Education (CORE) information portal project (see Chapter 3). The CORE information portal (CIP) system (see Chapter 5) has more than 170 webpages and 49 associated database tables. The CIP system is a gateway of resources and sources of information for the CORE, which is an education consortium of four osteopathic medicine colleges in the United States of America and fourteen teaching hospitals across the state of Ohio.

The major contribution of this work is the evaluation of the four major open source CMSs (Drupal, PHP-Nuke, Mambo and Plone) and the initial quantitative evaluation of the applicability of Mambo CMS to a basic mid-size information portal (the CIP system which we developed) with a low-level security requirement.

8.1 Mambo Compared to Manually Developing an Information Portal

Mambo provides dynamism, because it is easily extensible by adding add-ons such as Mambo Modules (see 4.7.1), Mambo components (see 4.7.2) and Mambo Mambots (see 4.7.3). The dynamic nature of Mambo saves time and effort. If we try to
develop an information portal manually it takes much more effort and time. Even if we try to implement our own CMS, it would be like reinventing the wheel, which is waste of time, effort and money. Hence, using Mambo one can develop a dynamic information portal solution.

8.2 Mambo Compared to a Commercial CMS

Mambo is more cost-effective as compared to using a commercial CMS such as “Microsoft CMS” [65] or “PaperThin Inc’s CommonSpot” [64], as there is cost associated with the purchase and licensing of commercial CMS (see 7.1.3). Additionally only one web-application developer is required to develop the portal using Mambo. This reduces staffing costs, which is demonstrated in our case study, the CIP project. Since Mambo aids in rapid development of the portal, it saves time and effort and hence money. Hence Mambo scores over commercial CMSs because of low total cost of ownership.

8.3 Mambo Compared to Other Open Source CMSs

Mambo is more user-friendly than the other open source CMSs i.e. PHP-Nuke, Drupal, and Plone (which we have mentioned in chapter 3). The user-friendliness and ease of use of Mambo are confirmed by the responses of the maintainers (content-contributors and administrators) of the CIP project (see 7.1.1.5 and 7.1.1.6). Hence using Mambo makes it efficient to develop an information portal compared to other open source CMSs.

Hence we can conclude that Mambo aids in the development of a dynamic, cost-effective and efficient information portal solution.
8.4 Future Work

We have evaluated our approach on our case study, the CORE Information Portal (CIP) project, which is a basic mid-size information portal. It would be interesting to test the applicability of our approach on a large-size information portal, one which has more than 1000 webpages.
References

[1] HTML. URL: http://www.w3.org/MarkUp/.


[26] Linux. URL: http://www.linux.org/.


[34] OS/2. URL: http://www-306.ibm.com/software/os/warp/.


[41] Plone Official Homepage. URL: http://plone.org/.


[54] CSS. URL: http://www.w3.org/Style/CSS/.


Appendix A CORE Surveys

A.1 IRB Approval

The title of our research was slightly modified later. Hence we have an original IRB approval and a revised IRB approval. We have attached both IRB approvals.

A.1.1 Original IRB Approval

![Image of the original IRB approval]

Office of Research Compliance
Research and Technology
Center 117
Athens 04305-1517
E: 740.593.3686
F: 740.593.8598
www.ohiou.edu/research

A determination has been made that the following research study is exempt from IRB review because it involves:

Category 2: research involving the use of educational tests, survey procedures, interview procedures or observation of public behavior

Project Title: A Dynamic Cost-Effective and Efficient Information Portal Solution using the Membio Content Management System for the Centers for Osteopathic Research and Education (CORE)

Project Director: Adnan B. A. Dakhwe

Department: Electrical Engineering and Computer Science

Advisor: Chang Liu

Rebecca Cale, Associate Director, Research Compliance
Institutional Review Board

Date: 4/3/06

The approval remains in effect provided the study is conducted exactly as described in your application for review. Any additions or modifications to the project must be approved by the IRB (as an amendment) prior to implementation.
A.1.2 Revised IRB Approval

The amendment, detailed below, and submitted for the following research study has been approved by the Institutional Review Board at Ohio University. Approval date of this amendment does not affect the expiration date of the original approval.

Amendment: Title Change

Project: A Dynamic, Cost-effective and Efficient Information Portal Solution using the Mambo Content Management System

Project Director: Adnan B. A. Dakhwe

Advisor: Chang Liu

Department: Electrical Engineering and Computer Science

Rebecca G. Cale
Institutional Review Board

Date: 4/18/06
A.2 Survey 1

Survey 1 for the CORE Information Portal project:
www.ohiocoreonline.org

This survey is for the educational research purpose (for the Master’s thesis). Participation is voluntary and the participant must be 18 years or older. There is no risk of participation. Data from this survey will help to evaluate the CORE information portal (website) project and also to improve the website training. Completion and return of the survey implies consent to use the data for the CORE website development and educational research purposes. Estimated time for the completion of the survey is 10 minutes. Please put a cross (x) across the alternative for your answer.

1. Have you used the internet?
   ______ Yes ______ No

2. Which web-browser do you use the most?
   ____ Internet Explorer ____ Mozilla Firefox ____ Safari
   ____ Netscape ____ Other: Please specify__________

3. Have you designed or developed a website or a webpage before?
   ______ Yes ______ No

   If yes, how would you rate your experience?
   ____ Very Easy ____ Easy ____ OK ____ Difficult ____ Very Difficult

4. Have you maintained a website or a webpage before?
   ______ Yes ______ No

   If yes, how would you rate your experience?
   ____ Very Easy ____ Easy ____ OK ____ Difficult ____ Very Difficult

5. How would you rate your knowledge about web-development?
   ____ Very Poor ____ Poor ____ Fair ____ Good ____ Excellent

6. How would you rate your knowledge about HTML?
   ____ Very Poor ____ Poor ____ Fair ____ Good ____ Excellent

7. How would you rate your knowledge about Microsoft FrontPage?
   ____ Very Poor ____ Poor ____ Fair ____ Good ____ Excellent

8. How would you rate your knowledge about Macromedia Dreamweaver?
   ____ Very Poor ____ Poor ____ Fair ____ Good ____ Excellent
9. How would you rate your knowledge about PHP (Hypertext Pre-Processor)?

   ___ Very Poor    ___ Poor    ___ Fair    ___ Good    ___ Excellent

10. How would you rate your knowledge about MySQL?

    ___ Very Poor    ___ Poor    ___ Fair    ___ Good    ___ Excellent

11. How would you rate your knowledge about JavaScript?

    ___ Very Poor    ___ Poor    ___ Fair    ___ Good    ___ Excellent

12. How would you rate your knowledge about Cascading Style Sheets (CSS)?

    ___ Very Poor    ___ Poor    ___ Fair    ___ Good    ___ Excellent

13. How would you rate your knowledge about Microsoft Word?

    ___ Very Poor    ___ Poor    ___ Fair    ___ Good    ___ Excellent

14. Please convey how would you go about updating or creating content on the old CORE website? (If Applicable)

15. Please convey how long did it take you to update or create content on the old CORE website? (If Applicable)

16. Was it convenient to update or create content on the old CORE website? (If Applicable)
A.3 Survey 2

Survey 2 for the CORE information portal project:  
www.ohiocoreonline.org

This survey is for the educational research purpose (for the Master’s thesis). Participation is voluntary and the participant must be 18 years or older. There is no risk of participation. Data from this survey will help to evaluate the CORE information portal (website) project and also to improve the website training. Completion and return of the survey implies consent to use the data for the CORE website development and educational research purposes. Estimated time for the completion of the survey is 10 minutes. Please put a cross (x) across the alternative for your answer.

1. How did you find changing or creating content using Mambo content management system on www.ohiocoreonline.org?

   ____ Very Easy    ____ Easy    ____ OK    ____ Difficult    ____ Very Difficult

2. How would you compare the online editor (content-contributor interface) at www.ohiocoreonline.org to Microsoft Word?

   ____ Very similar    ____ Similar    ____ Different    ____ Very Different    ____ No Comment

3. Did you need to know HTML to change or create content in your section on www.ohiocoreonline.org?

   ______ Yes    ______ No

4. Did you need to know Microsoft FrontPage or Macromedia Dreamweaver to change or create content in your section on www.ohiocoreonline.org?

   ______ Yes    ______ No

5. Did you need to know PHP to change or create content in your section on www.ohiocoreonline.org?

   ______ Yes    ______ No

6. Did you need to know MySQL to change or create content in your section on www.ohiocoreonline.org?

   ______ Yes    ______ No

7. Did you need to know JavaScript to change or create content in your section on www.ohiocoreonline.org?

   ______ Yes    ______ No

8. Did you need to know CSS to change or create content in your section on www.ohiocoreonline.org?

   ______ Yes    ______ No
9. How did you find the Content-Contributor interface (online editor)?
   _____ Very User Friendly   _____User Friendly   _____OK   _____Least User Friendly
   _____Not User Friendly

10. How did you find the Administrator interface (back-end interface)?
    _____Very User Friendly  _____User Friendly  _____OK   _____Least User Friendly
     _____Not User Friendly   _____Not Applicable

11. Please describe your experience of updating and creating content on the CORE information portal.

12. How much time did it take you to add a paragraph or a sentence of content to the respective section?

13. Was it easy to learn and use the content-contributor interface?