**UNIVERSITY OF THE PHILIPPINES MANILA**
**COLLEGE OF ARTS AND SCIENCES**
**DEPARTMENT OF PHYSICAL SCIENCES AND MATHEMATICS**

# Movable Type Template Editor

A Special Problem in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science

Submitted by

Alvin P. Tabun
April 2004

**ACCEPTANCE SHEET**

The Special Problem entitled "Movable Type Template Editor" prepared and submitted by Alvin P. Tabun in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

_____
Prof. Ma. Sheila A. Magboo
Adviser

| EXAMINERS | APPROVED | DISAPPROVED |
|---|---|---|
| 1.  Prof. Gregorio B. Baes | _____ | _____ |
| 2.  Ms. Avegail D. Carpio | _____ | _____ |
| 3.  Mr. Richard Bryan L. Chua | _____ | _____ |
| 4.  Dr. Vincent Peter C. Magboo | _____ | _____ |
| 5.  Mr. Philip D. Zamora | _____ | _____ |

_____
DATE

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

Vincent Peter C. Magboo, M.D., M.S.
Unit Head
Mathematical and Computing Sciences Unit
Department of Physical Sciences and Mathematics

Harry L. Engle, M.S.
Chair
Department of Physical Sciences
and Mathematics

Marilou G. Nicolas, Ph.D.
Dean
College of Arts and Sciences

# ABSTRACT

Movable Type Template Editor is a web-based editor for Movable Type (MT) templates. It is an extension of the MT software. It simplifies the task of editing templates by providing special tags or codesets which can be used instead of blocks of MT code. Codesets are stored in a database and can be inserted to MT templates. Features for HTML editing, template preview and weblog rebuilding are also available.

**Keywords:** weblog, template

# TABLE OF CONTENTS

**I. INTRODUCTION**

**A. BACKGROUND OF THE STUDY**

Weblog, or blog, is a collection of web links and articles arranged chronologically. Each link is usually accompanied by a comment that highlights the interesting parts of the article, gives an expert interpretation, verify its accuracy, add additional information on the said article or related topics, or add an opinion or a differing viewpoint. Weblogs also allow its visitors to post their own comments on each entry.[1]

Weblogs act like filters, that are focused on topics that interests its owner, pre-surfing the web. They give a glimpse of the web for a person who doesn't have the time to surf the whole web and is useful for those who need to keep in touch with new information as it happens.[2]

The first weblogs were created by web designers and software developers, who were aware of the capabilities of the technology so only a few of them existed. With the availability of several free or inexpensive weblogging services in 1999, weblogging has become more accessible. Weblog tools automate and accelerate the weblog publishing process. Users provide a template which indicates the location of the posts. Using a form, users then provide the content to be published on the web. Using these systems requires little technical knowledge.[3]

Movable Type (MT) excels in flexibility and power. It is elegant and more customizable. MT has many features that can be used to design templates. It is free to use for non-commercial purposes.[4] It has a lot of useful features like: entry categorization, search and replace function, draft and publish entries, built-in comment system, template-based customization, email notification system, multiple archiving options, and import function.[5,6] The makers of MT decided to give its users more control of their weblogs but compromising its ease of use.

MT uses templates and Cascading Style Sheets (CSS) to control the appearance of the weblog. MT templates are HTML tags interspersed with MT tags, a markup used by MT which represents the entries in the database. These tags allow users to add content to the templates without using the actual information from the database. For instance, a <$MTBlogName$> tag in the template would mean the weblog's title.

## B. STATEMENT OF THE PROBLEM

Movable Type (MT) system gives its users more control over their weblogs by allowing more options to design their templates. However, to benefit from these features, weblog users must be familiar with the MT tags that correspond to these features and incorporate them into a HTML template. MT does not include a template editor so a weblog owner is left to edit his templates on his own.

This process of creating templates can be confusing especially to people who have little experience in web development. Because of this users may not take full advantage of MT features.

**C. OBJECTIVES**

The aim of the special problem is to extend the Movable Type (MT) software by simplifying the task of editing templates by providing a layer of abstraction through the use of code sets.

An code set is a collection of tags which performs a specific procedure grouped to form an entity, a representation of data in which the implementation details are hidden (abstracted). By combining functionally-related tags, it eliminates the need to inspect the interactions of individual tags but rather it would be viewed as a set. Descriptions of a code set is less technical because it focuses on the code set's purpose, contrast to the MT manual's tag descriptions which tells how MT interprets a specific mark up. Should there be a need to modify tags, code sets simplify editing by compartmentalizing the script into smaller, more manageable portions.

The system will have the following capabilities:

1. Add/Edit/Delete/Search/View codesets to the codeset database.

2. Export/Import codesets to/from an import file.

3. Preview template rendered by a browser.

4. View and color-code template source code.

5. Edit template.

    a. Insert codesets.

    b. Insert HTML tags.

    c. Apply styles from a style sheet

    d. Replace codesets in the template with MT code.

6. Rebuild the weblog.

**D. SIGNIFICANCE OF THE STUDY**

The system will simplify the integration of MT tags and HTML. It will be easier to customize MT templates because of the user friendly interface that will be provided by the editor. This would encourage more specialists to create and maintain their own weblogs and help internet researchers by providing quality sources of knowledge.

**E. SCOPE AND LIMITATIONS**

The system is specifically designed for MT weblog tool only. The proposed system will not alter the existing MT system. It will only extend it through the addition of a template editor. The system does not access the data entries stored by MT. The system renders the code between the Body tags excluding the Body tags themselves.

**F. ASSUMPTIONS**

1. The user has a weblog uploaded to the server.

2. The templates of the weblog are externally linked to a file with a .tmpl extension.

3. Templates which may contain codesets have a .cst extension.

4. A .cst file produces a .tmpl file with the same name. For example, sample.html.cst will produce sample.html.tmpl

5. All errors or codeset updates will be discussed at the MT support group.

**II. REVIEW OF RELATED LITERATURE**

Information is constantly and rapidly expanding. It is difficult to make decisions not because there aren't enough information but because there are too much information which still has to be sorted to get only the relevant ones. Weblogs provide a way to filter this information.[2]

One of the greatest impacts of the weblog phenomenon is in the field of journalism. Weblogs are now considered as an alternative to traditional media. News is delivered directly from the source in its raw and complete form. It has no restrictions on length or topic scope so every detail gathered by a reporter may be included. It doesn't undergo editorial processes which tends to dull the article. A reporter may also include his own opinion on his articles. Weblog browsers who have an alternate point of view may as well post their opinion via the weblog's commenting feature. This way a better analysis of the article can be given.[7]

Weblogs have no marketing constraints. They are not tailored for a certain demographic. Browsers go to the user's weblog if they find it interesting but it is not necessary for the user to change his manner of writing just to suit his visitors.[8]

Weblogs also have more integrity because weblog owners are not paid to publish information unlike traditional media. They are open to every idea not just the ones that are backed by big named companies. Weblogs therefore offer immediacy and conjecture at the expense of accuracy and reflective thought.[7]

Comments on links and articles allows weblog owners to interpret information based on his own understanding rather than just providing information. Comments are mostly in an irreverent and sarcastic tone challenging sources on their vested interests and expertise. By reading comments, browsers get to know weblog owners personally; his biases and interests, which helps evaluate the trustworthiness of the source.[1,9]

Weblogs create a new kind of environment where the public are not just fed information but participate in the discussion as well. It also builds a sense of community with other weblog browsers.[10]

Most weblog owners have full-time Internet jobs and have developed web researching skills enabling them to find quality and hard to find web links.[11]   Finding information with weblogs is faster and complete.

Figure 1 is an example of a weblog. Seb's Open Research is a collection of links and articles on interdisciplinary communication, knowledge sharing, and knowledge representation. It is located at http://radio.weblogs.com/0110772/. This resulted in the growth in the number of weblogs. It is now gaining respect as a valuable source of information. Companies like Terra Lycos, America Online, and Google are now commercializing this weblog phenomenon, using them as marketing tools.[12]



Figure 1: Sample weblog - Seb's Open Research

There are many weblog tools available each having their own strengths. The most popular are Blogger, Movable Type (MT), and Radio Userland.[13] Blogger is the easiest to use. The weblog tool is hosted on Blogger's server; no installation is needed. Google has recently purchased Pyra Labs, the creator of Blogger, making the tool run on fast servers. Posts can be made from any computer and there is a modest selection of templates. The disadvantages of this tool is that the weblog is dependent on the reliability of its host, its features are basic, and although it comes free, blogger weblogs have advertisements included. A screenshot of the Blogger editing screen is shown in Figure 2.[14]
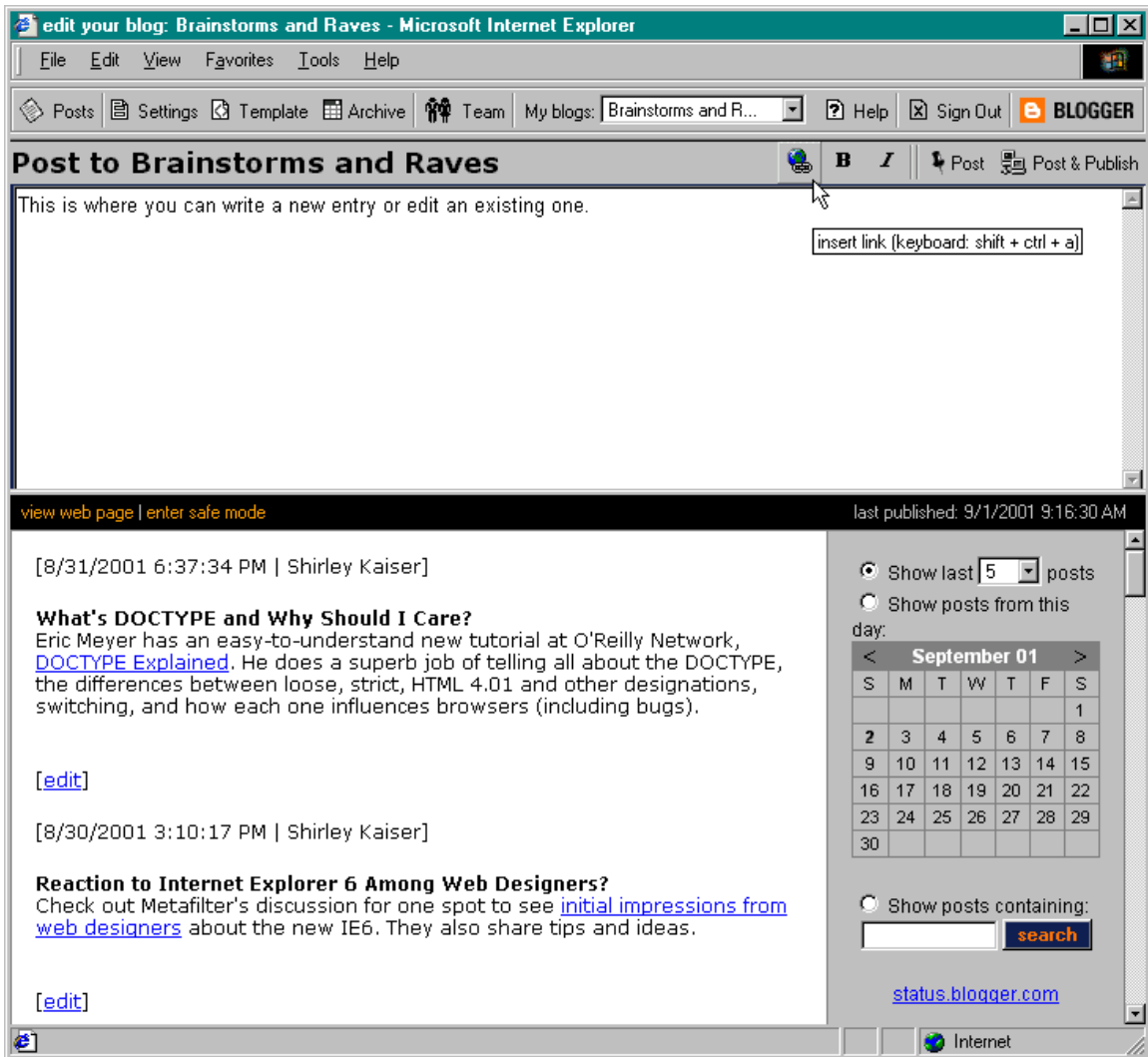


Figure 2: Blogger Entry Editing Screen

MT may be hosted on any server that can run CGI scripts. Posts may also be made from any computer. However, MT requires installation which is not that easy. MT also has an extensible, library driven code written in Perl which means that anyone knowledgeable on Perl may make bug fixes or make enhancements to the MT system. MT has an active community where users may ask questions and share information regarding the software. Figure 3 is a screenshot of MT entry editing screen. [15]
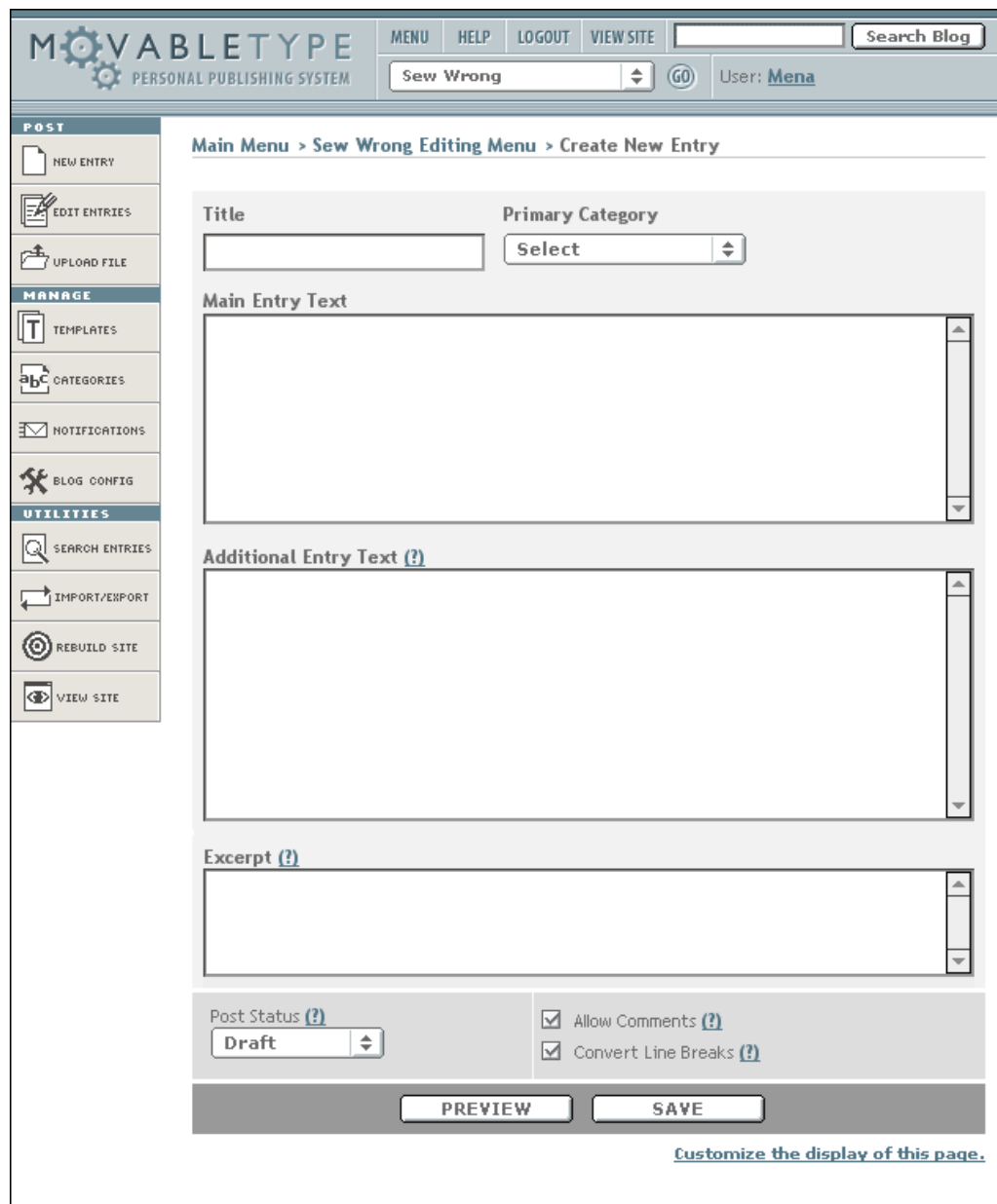


Figure 3: Movable Type Entry Editing Screen

Radio Userland tool resides on the user's desktop allowing offline editing of posts; only connecting online when the user is ready to update his weblog. The user may only posts from his desktop and is also required to pay for the usage of the software. Figure 4 is a screenshot of the Radio Userland entry editing screen.[16]
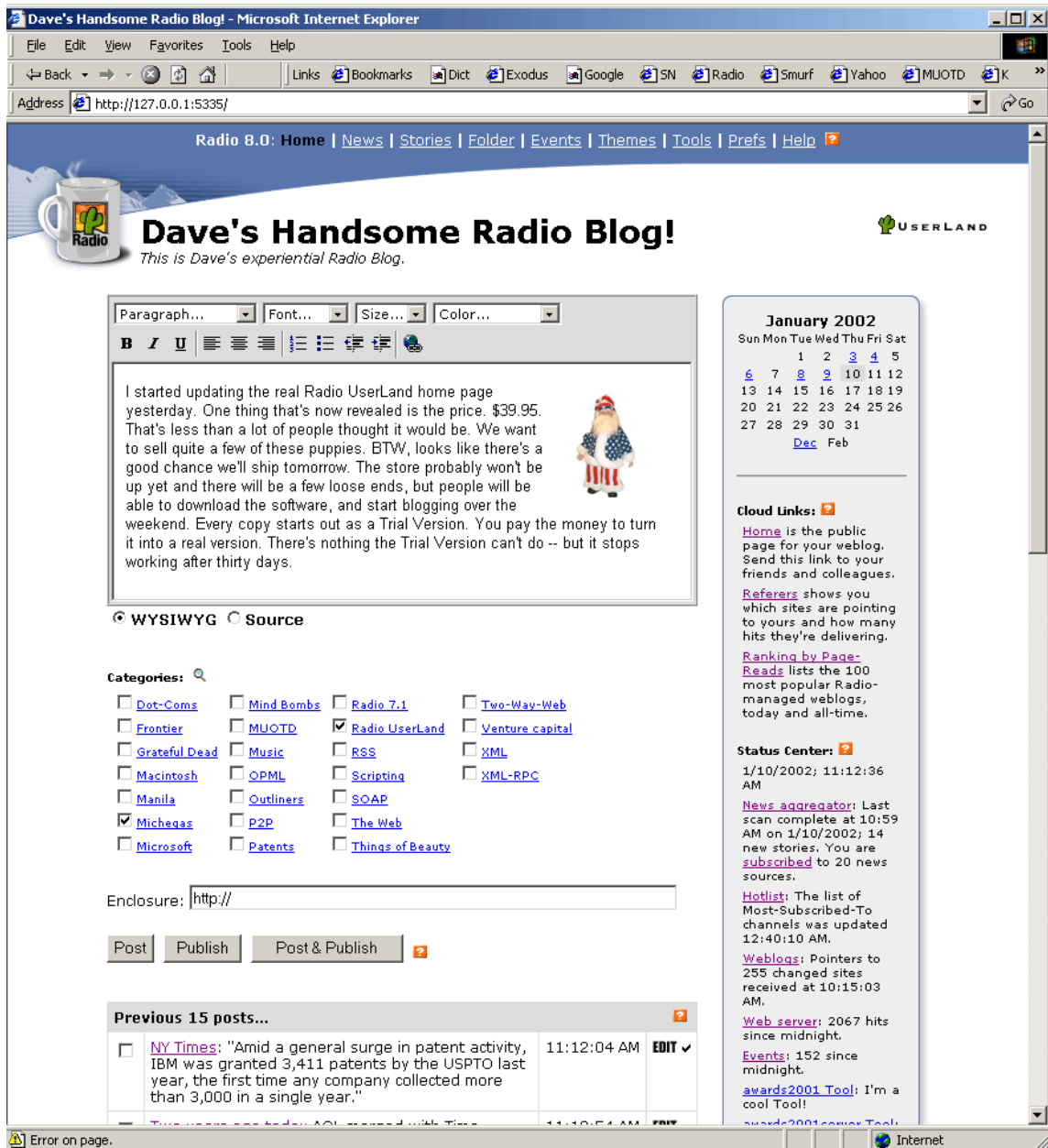


Figure 4: Radio Userland Entry Editing Screen

## III. THEORETICAL FRAMEWORK

### A. INFORMATION HIDING

David Parnas first introduced the concept of information hiding around 1972. He argued that the primary criteria for system modularization should concern the hiding of critical design decisions. He stressed hiding "difficult design decisions or design decisions which are likely to change." Hiding information in that manner isolates clients from requiring intimate knowledge of the design to use a module, and from the effects of changing those decisions. [17]

### B. CONTENT MANAGEMENT SYSTEMS (CMS)

Content Management is effectively the management of content by combining rules, process and/or workflows in such a way that centralized webmasters and decentralized web authors/editors can create, edit, manage and publish all the content of a web page in accordance with a given framework or requirements.[18]

The principle of all CMS is to allow non-technical personnel to add, edit, or delete content within a web site in as close to real-time as possible. Most content management systems offer the following capabilities[19]:

- Ability to set up rules to establish which areas of a web site are managed by which individuals or departments within an organization.
- Ability to work within any web-based environment: Internet (publicly available web site), intranet (web site available to employees only), or extranet (web site available to employees, customers, vendors, and partners).
- Scalability (i.e. add more web sites, web pages, or applications within the existing content management system with minimal or no computer programming knowledge).
- Adherence to existing technology standards.

Types of CMS

1. General CMS - used to store and subsequently find and retrieve large amounts of data

2. Framework - a set of tools for developing content management systems.

3. User Interface - front end for developing CMS.

4. News Portal - a specialized CMS that syndicates its headlines and stories into news feeds

5. Weblog - a single web page with dated chunks of information called posts. Posts are arranged in reverse chronological order so the most recent is always at the top. Each post has an anchor that allows a hyperlink to the post from anywhere on the web. This allows blog syndication.

6. Wiki - a collection of web pages with novel linking structures. Each link is the name of a page in the Wiki, with a special capital-letter plus medial-cap syntax that makes it a "WikiWord."

## C. WEBLOG TOOLS

Weblog tools are designed to build the weblog site, organize and archive posts to the weblog, and publish content without the user having to know any HTML, etc. These tools are template-driven and have a database backend that effectively separates management of layout and content respectively. MT is one such tool. [20]

## D. MOVABLE TYPE TEMPLATES

Each entry made by the user is stored in a database. The templates and CSS are uploaded to the server. The weblog must be rebuilt after adding new entries or modifying templates to keep it updated. In this process, MT interprets the templates then inserts the information found in the database.

Templates and CSS may be linked externally to a file to keep from manually downloading and uploading templates every time they are modified. Linking externally makes MT check if changes have been made to the external file; and if there are, it automatically copies the said file. MT templates contain HTML tags interspersed with MT tags. There are two basic kinds of MT tags: container tags and variable tags.

Container tags are of the form <foo> .. </foo>. They represent a loop or conditional. The code between them is executed when certain conditions are met. <MTEntries> is a  container tag that represents a list of entries from the weblog. The code inside the container tags are run for each entry in the database. Variable tags are of the form <foo> and may optionally be written as <$foo$>. They are placeholders to be replaced by some data. <MTEntryTitle> is an example which will be replaced by the title of the entry. Context is important in using MT tags. Some tags are meaningless when not used correctly. It works similar to HTML's table tags that a <TR> tag is meaningless outside of a <TABLE> tag. An <MTEntryTitle> will do nothing outside of a <MTEntries> tag.[6]

Figure 5 is a screenshot of a sample MT weblog using the default template. It is accompanied with the source code of a template fragment and its corresponding weblog counterpart to illustrate the process of building the weblog.



Figure 5: Sample weblog using MT's initial template with 'Recent Entries' section highlighted

**Source Code of the 'Recent Entries' section of the template**

```
<div class="sidetitle">
Recent Entries
</div>
<div class="side">
<MTEntries lastn="10">
<a href="<$MTEntryPermalink$>"><$MTEntryTitle$></a><br />
</MTEntries>
</div>
```

**Source Code of the 'Recent Entries' section of the weblog**

```
<div class="sidetitle">
Recent Entries
</div>
<div class="side">
<a href="http://localhost/newblog/archives/000008.html">Liberal Party endorses Arroyo for President</a><br
/>
<a href="http://localhost/newblog/archives/000009.html">Malabon supports drive vs graft &
corruption</a><br />
</div>
```

When the weblog is built, the MTEntries container tags tells the weblog tool to retrieve the last 10 entries in the database and perform the code between the tags for each entry retrieved. In this particular example, there are only two entries. The tool replaces the MTEntryPermalink and MTEntryTitle with the entries' archive link and title respectively.

## IV. DESIGN AND IMPLEMENTATION

## A. ENTITY RELATIONSHIP DIAGRAM

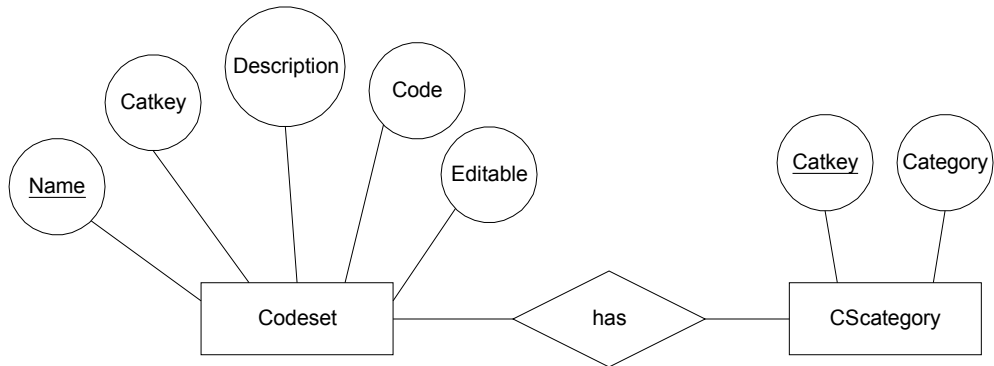Figure 6 show the entity relationship diagram (ERD) of the system.

Figure 6: Entity Relationship Diagram of Movable Type Template Editor

## B. DATABASE TABLES

The two database tables obtained from the ERD are shown in Figure 7. The arrow points to the fields referenced by an attribute. The "catkey" field in the "Codeset" table references the "catkey" field of the "CScategory" table.
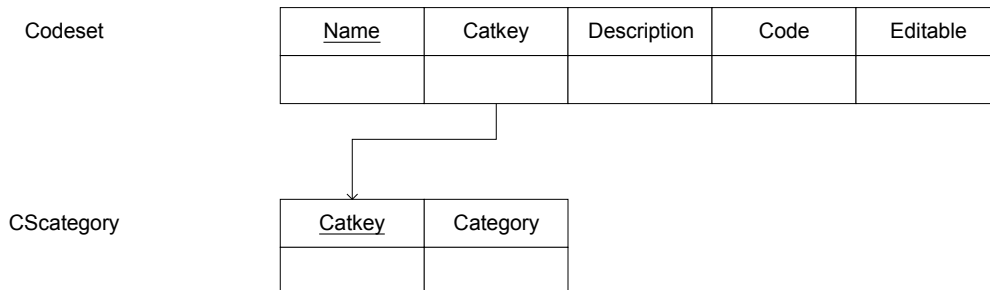
Codeset

| Name | Catkey | Description | Code | Editable |
|------|--------|-------------|------|----------|
|      |        |             |      |          |

CScategory

| Catkey | Category |
|--------|----------|
|        |          |

Figure 7: Database Tables of Movable Type Template Editor

## C. CONTEXT DIAGRAM

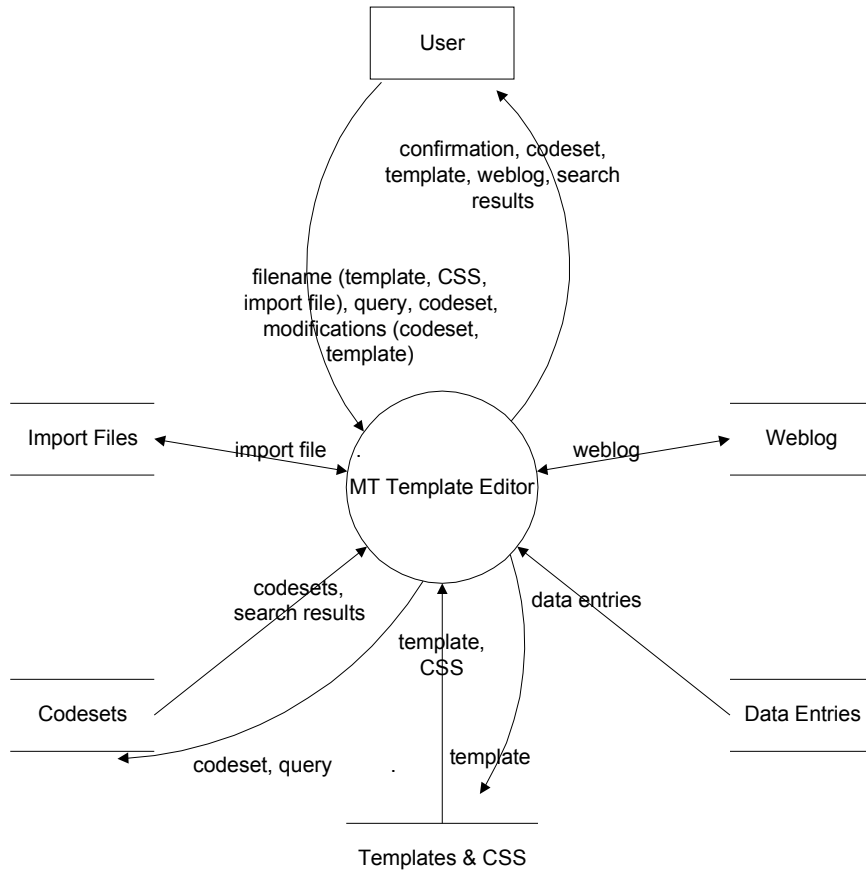The context diagram is shown in Figure 8. The user is the weblog owner.

Figure 8: Context Diagram of Movable Type Template Editor

## D. DATA FLOW DIAGRAM

The top level data flow diagram is shown in Figure 9. The user can manage the codesets and templates as shown in process 1 and 2 respectively. The user supplies information about the templates, codesets, style sheet, and import files. The processes searches and loads them and the user may make the necessary changes. Anytime, the user may rebuild and view the weblog as shown in process 3 and 4 respectively. Rebuilding the weblog applies the changes of a modified template and/or new data entries added to the database.

Figure 9: Top-level DFD of Movable Type Template Editor

Figure 10 shows the sub-explosion of process 1 Manage Codeset Database. Codesets may be added one by one, or in groups by importing codesets. Conversely, codesets may be shared by creating an import file through the export function. The user may perform searches on the database to retrieve codesets and modify them. Similarly, the sub-explosion of process 1.5 Configure Codeset is seen in Figure 11.
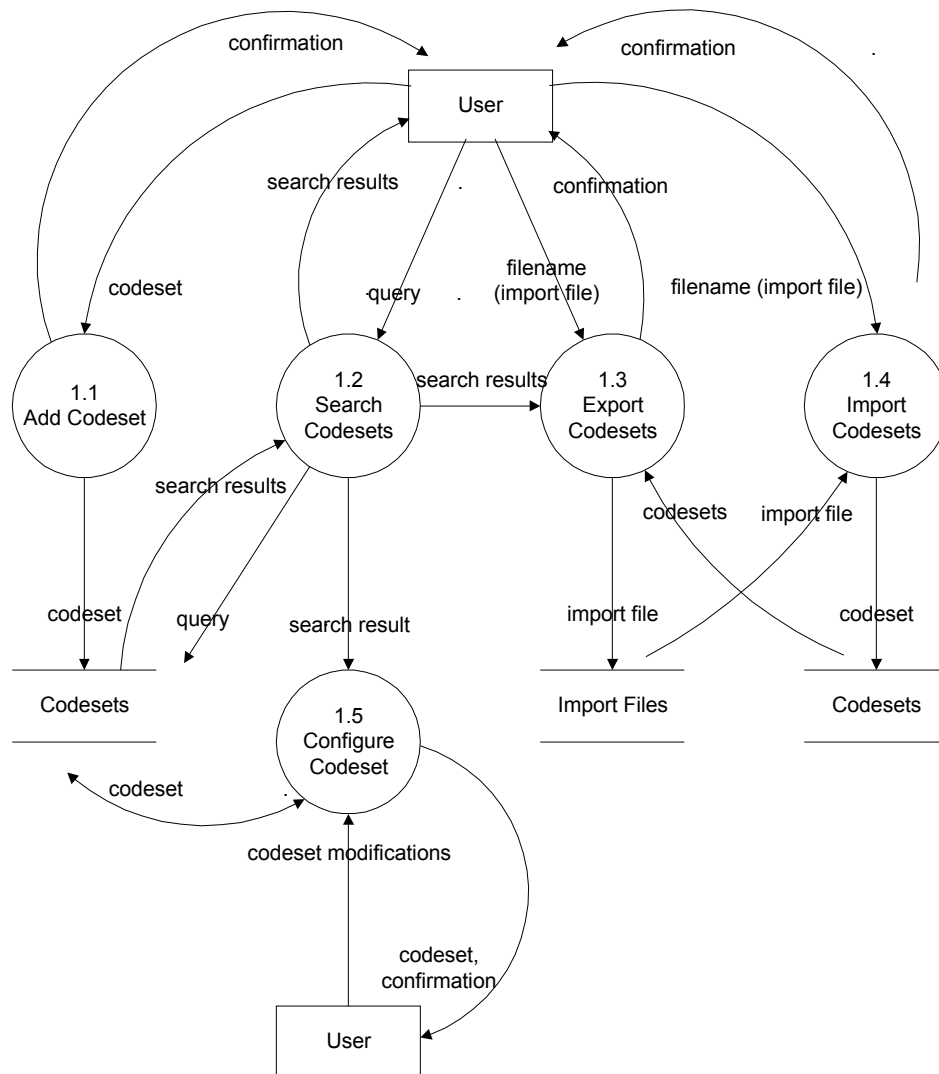


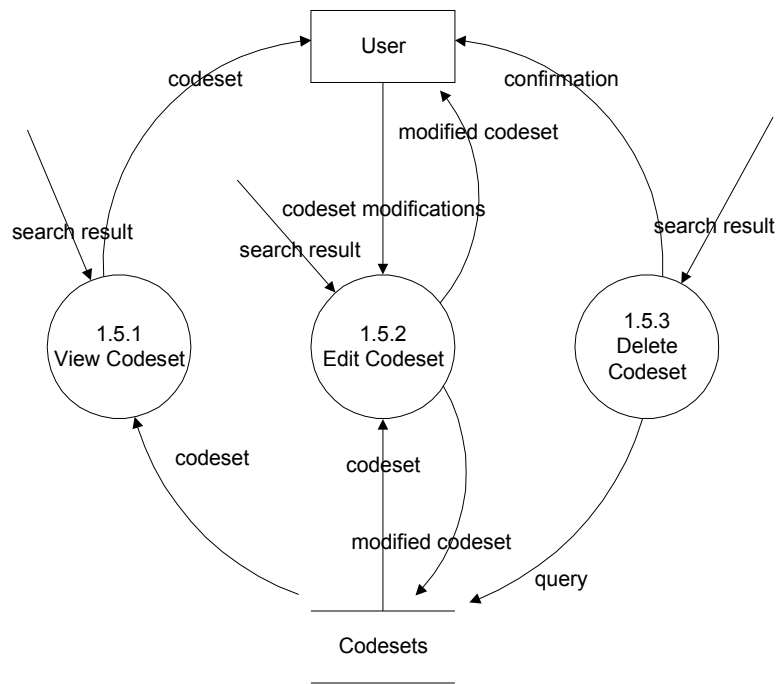Figure 10: Sub-explosion of Process 1 (Manage Codeset Database), Movable Type Template Editor

Figure 11: Sub-explosion of Process 1.5 (Configure Codeset), Movable Type Template Editor

Figure 12 shows the sub-explosion of process 2 Configure Template. After supplying the filename of the template, the user may view, edit, and save the template. The user may also view the template with a style sheet. The template editor allows the user to add HTML tags, insert codesets and apply styles to the template.
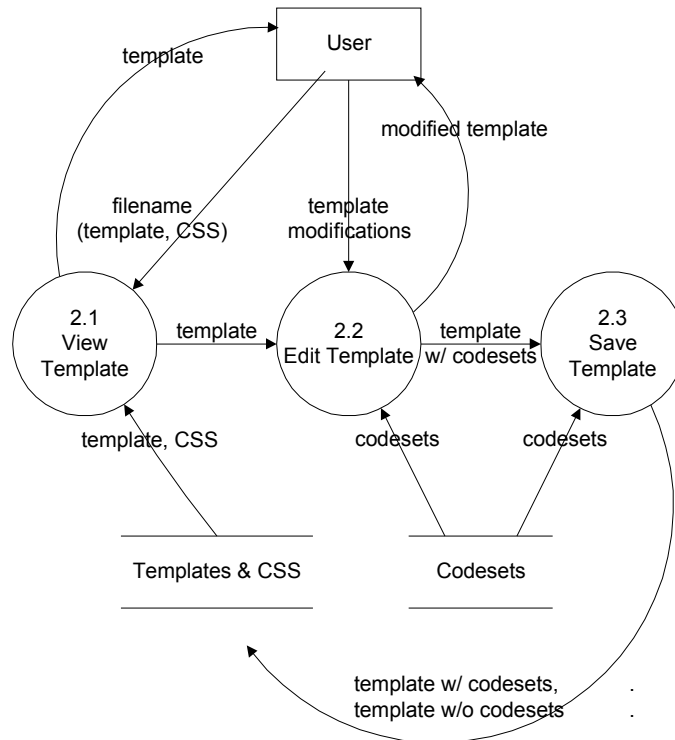


Figure 12: Sub-explosion of Process 2 (Configure Template), Movable Type Template Editor

The user may view the template in two modes: Normal and HTML mode. The Normal mode renders the template as how it would be viewed in a browser using a style sheet if provided. The HTML mode gives the source code of the template. The source code is color coded to distinguish MT tags, codesets, and HTML tags from each other. The template could be edited in either modes. The template is rendered using the MSHTML editor. It enables the editing of HTML content in a manner similar to text editing in word processors. This feature is provided by Internet Explorer 5.5 or higher. As of IE 6, MSHTML only renders code between the body tags excluding the body tags themselves (see Figure 13).
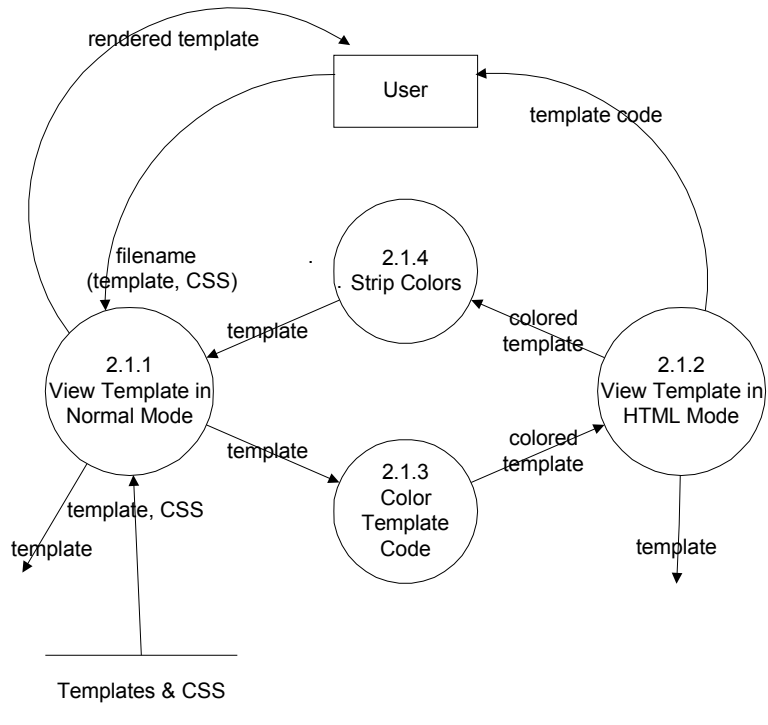


Figure 13: Sub-explosion of Process 2.1 (View Template), Movable Type Template Editor

When a template is saved, two files are produced. The system first saves the template as is, preserving the codesets within the template for further editing. Then, it replaces the codesets in the template with its corresponding MT tags which produces the actual template to be used by the weblog (see Figure 14).
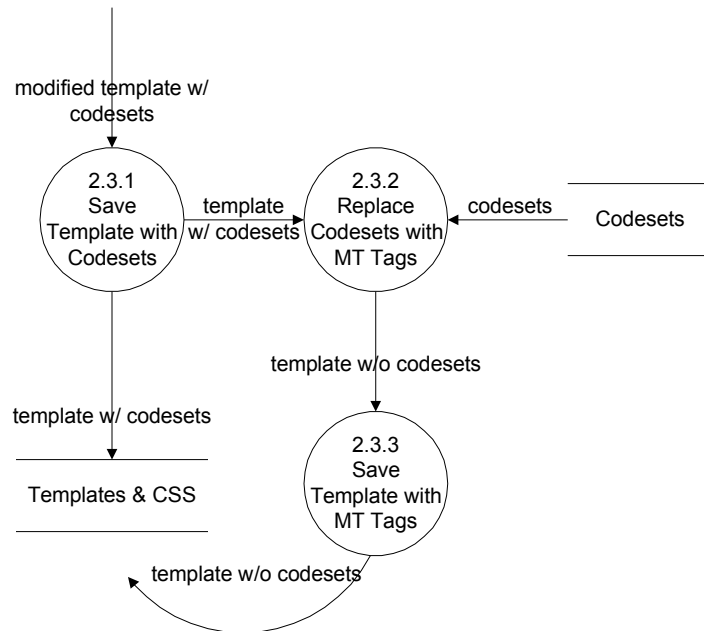
Figure 14: Sub-explosion of Process 2.3 (Save Template), Movable Type Template Editor

**E. DEFINITION OF TERMS**

**Weblog** - a chronological collection of web links with personal narrative, opinion, and comments

**Movable Type (MT)** -  a decentralized, web-based personal publishing system designed to ease maintenance of regularly updated news or journal sites, like weblogs.

**Cascading Style Sheets (CSS)** - a simple style sheet mechanism that allows authors and readers to attach style (e.g. fonts, colors and spacing) to HTML documents.

**Codeset**  - a collection of tags grouped to form an entity which performs a specific procedure.

**F. TECHNICAL ARCHITECTURE**

Requirements:

- Movable Type, at least version 2.6 recommended

- Perl, version 5.004_04 or higher

- PHP, version 4.2.3 or higher

- MySql database

- Internet Explorer version 5.5 or higher with Javascript enabled

- Microsoft Windows 98, 2000, or XP

## V. RESULTS

Figure 15 shows the interface of the template editor. The user may edit a template using the toolbar located above the editor box.
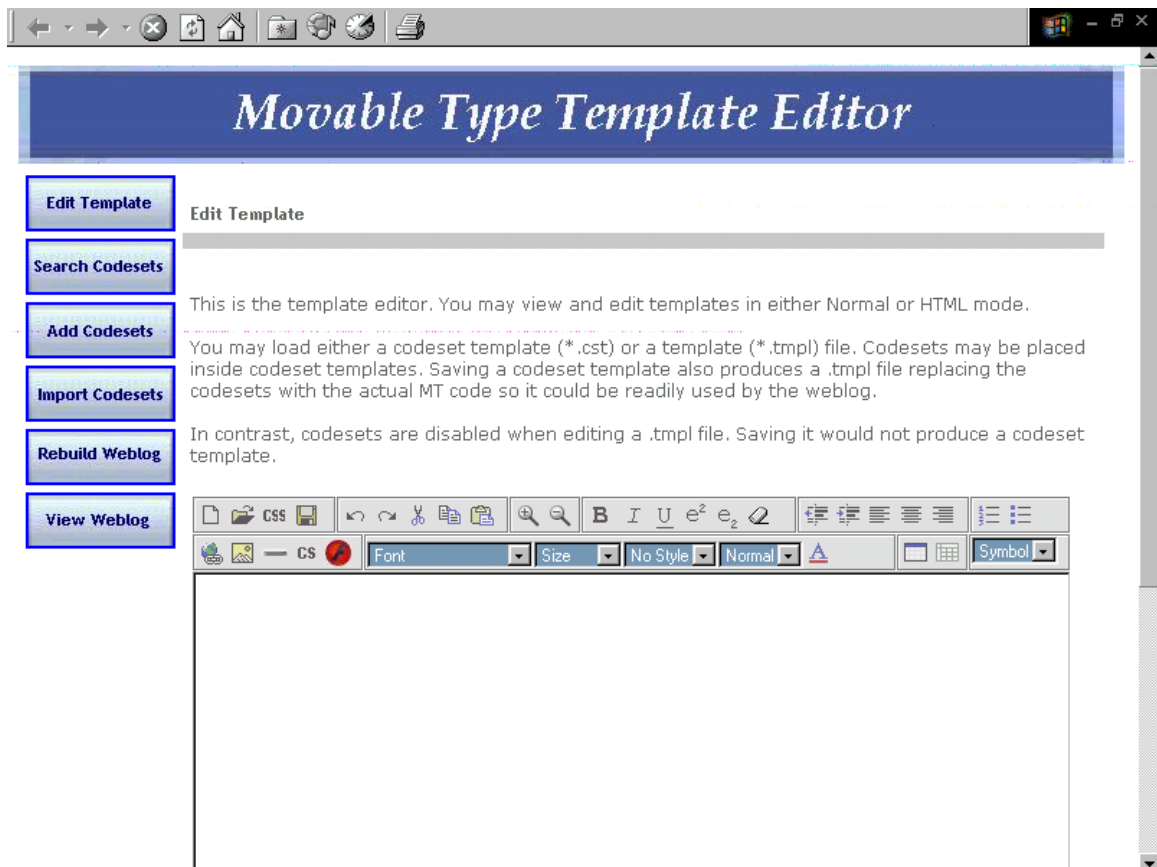


Figure 15: Interface for Edit Template, Movable Type Template Editor

Figure 16 shows a template viewed using Normal View. The user may toggle between Normal View and HTML View using the tabs below the editor box or by using the context menu. On the other hand, Figure 17 shows the same template in Figure 16 but in HTML view.
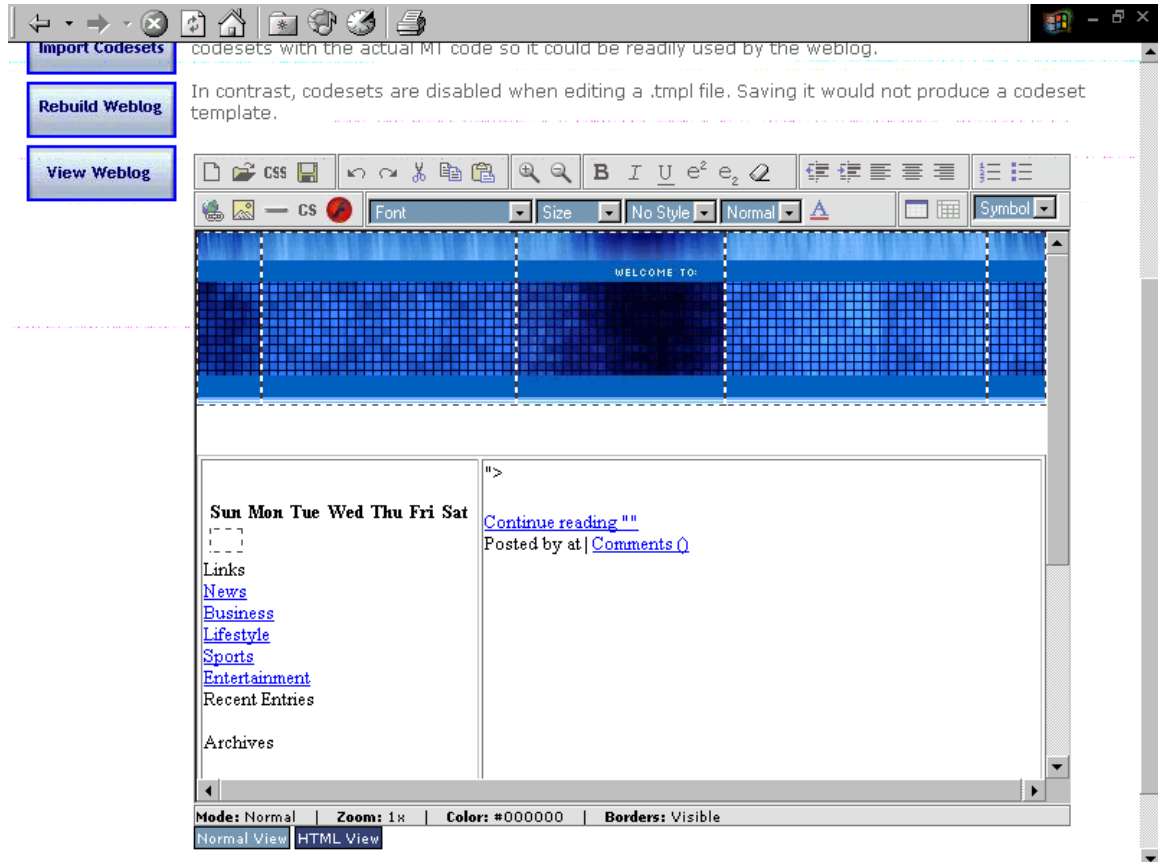


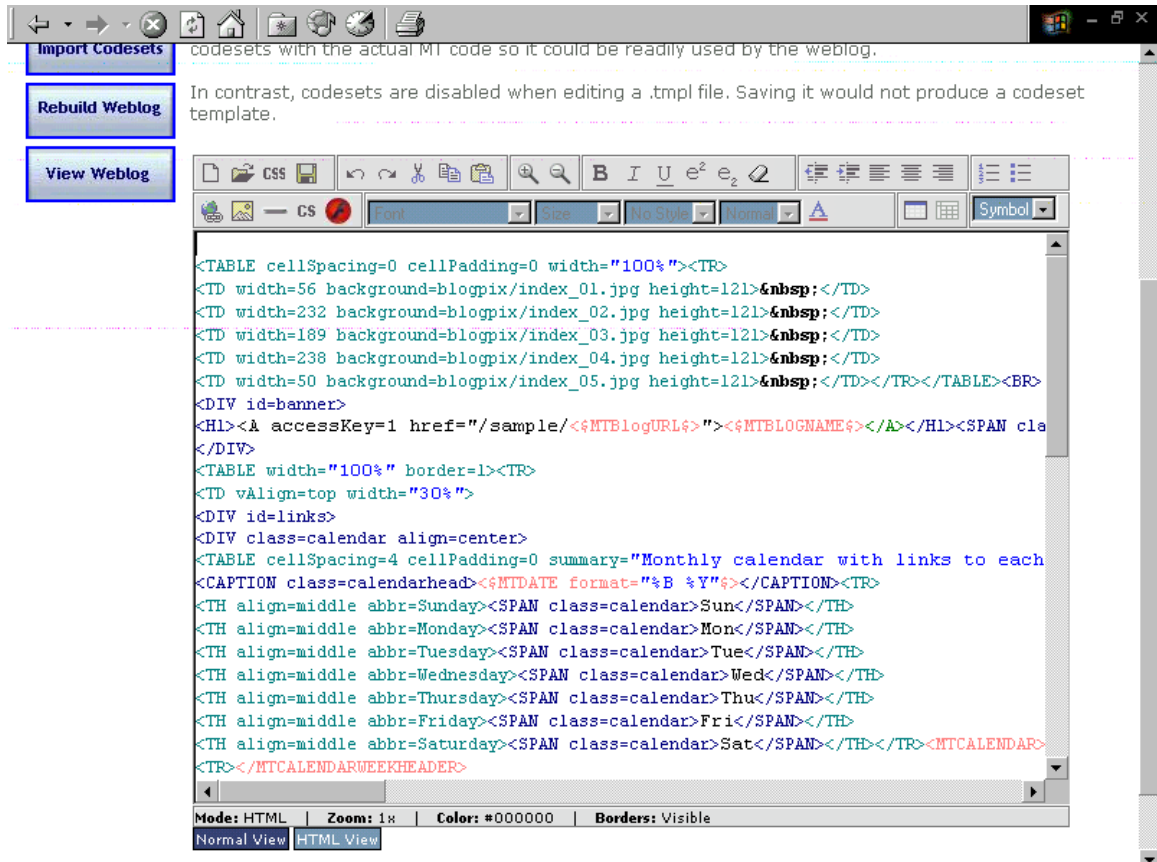Figure 16: Edit Template in Normal View, Movable Type Template Editor

codesets with the actual MT code so it could be readily used by the weblog.

In contrast, codesets are disabled when editing a .tmpl file. Saving it would not produce a codeset template.

```
<TABLE cellSpacing=0 cellPadding=0 width="100%"><TR>
<TD width=56 background=blogpix/index_01.jpg height=121> </TD>
<TD width=232 background=blogpix/index_02.jpg height=121> </TD>
<TD width=189 background=blogpix/index_03.jpg height=121> </TD>
<TD width=238 background=blogpix/index_04.jpg height=121> </TD>
<TD width=50 background=blogpix/index_05.jpg height=121> </TD></TR></TABLE><BR>
<DIV id=banner>
<H1><A accessKey=1 href="/sample/<$MTBlogURL$>"><$MTBLOGNAME$></A></H1><SPAN cla
</DIV>
<TABLE width="100%" border=1><TR>
<TD vAlign=top width="30%">
<DIV id=links>
<DIV class=calendar align=center>
<TABLE cellSpacing=4 cellPadding=0 summary="Monthly calendar with links to each
<CAPTION class=calendarhead><$MTDATE format="%B %Y"$></CAPTION><TR>
<TH align=middle abbr=Sunday><SPAN class=calendar>Sun</SPAN></TH>
<TH align=middle abbr=Monday><SPAN class=calendar>Mon</SPAN></TH>
<TH align=middle abbr=Tuesday><SPAN class=calendar>Tue</SPAN></TH>
<TH align=middle abbr=Wednesday><SPAN class=calendar>Wed</SPAN></TH>
<TH align=middle abbr=Thursday><SPAN class=calendar>Thu</SPAN></TH>
<TH align=middle abbr=Friday><SPAN class=calendar>Fri</SPAN></TH>
<TH align=middle abbr=Saturday><SPAN class=calendar>Sat</SPAN></TH></TR><MTCALENDAR>
<TR></MTCALENDARWEEKHEADER>
```

Mode: HTML | Zoom: 1x | Color: #000000 | Borders: Visible

Normal View  HTML View

Figure 17: Edit Template in HTML View, Movable Type Template Editor

The figure shown in Figure 18 is the interface for searching codesets. The user may search for codesets using any of the parameters. Figure 19 shows the interface of the search results. The user may view, delete or export codesets.
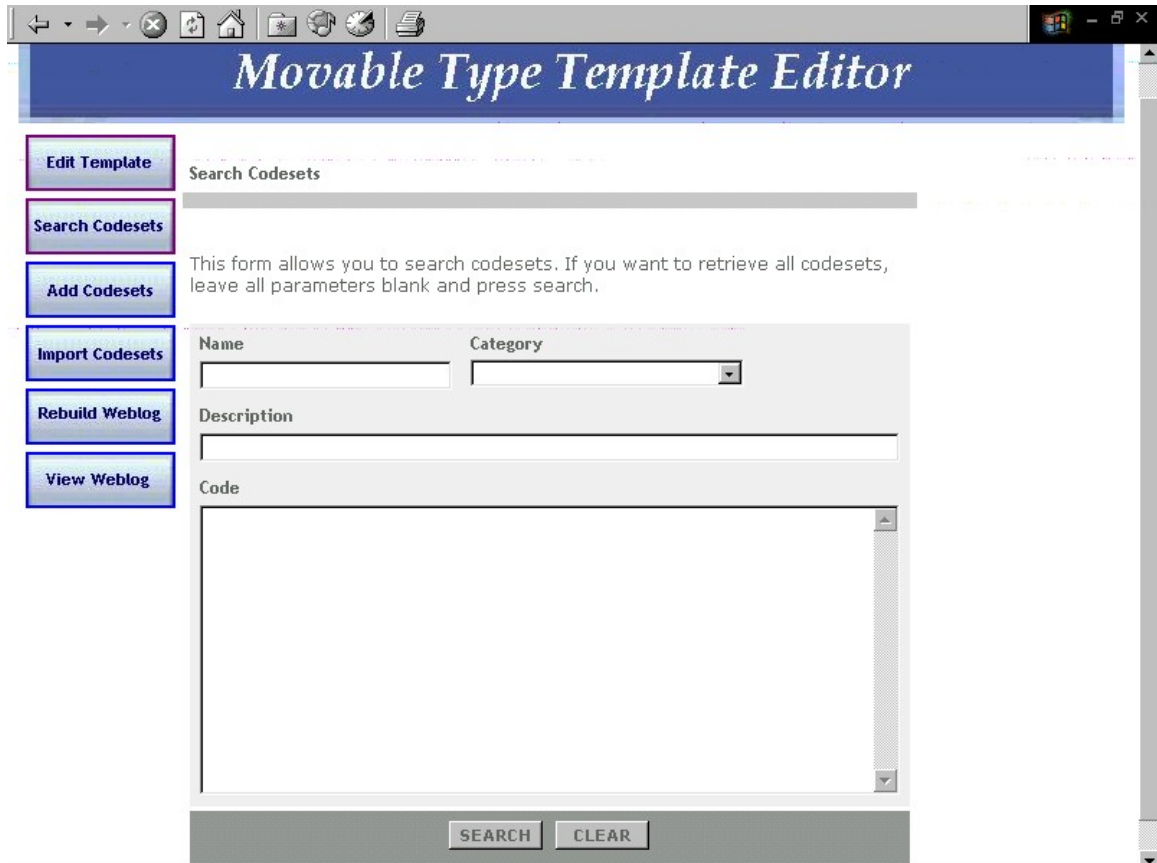


Figure 18: Search Codesets, Movable Type Template Editor

Figure 19: Search Results, Movable Type Template Editor

The interface for editing codesets is seen in Figure 20. The user may edit, delete or export a codeset after adding it or viewing it from a previous query. The interface for adding codesets is seen in Figure 21.
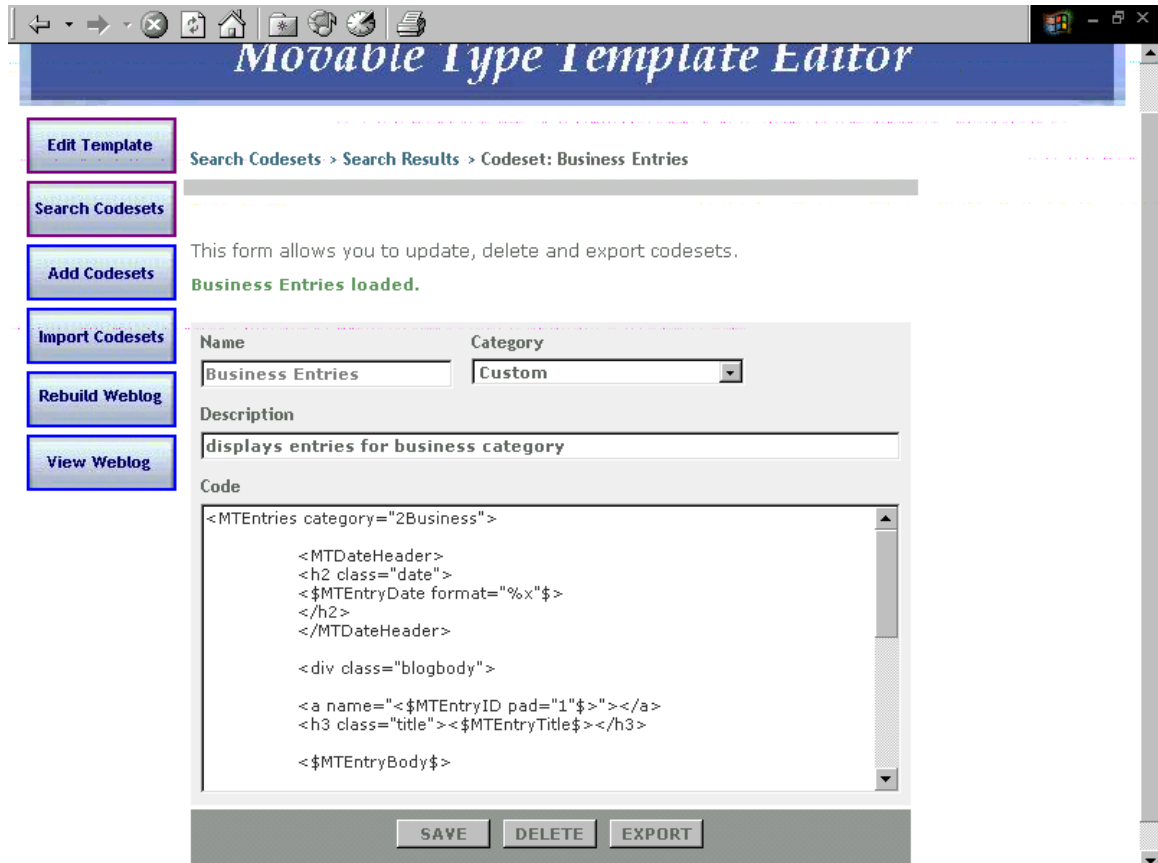


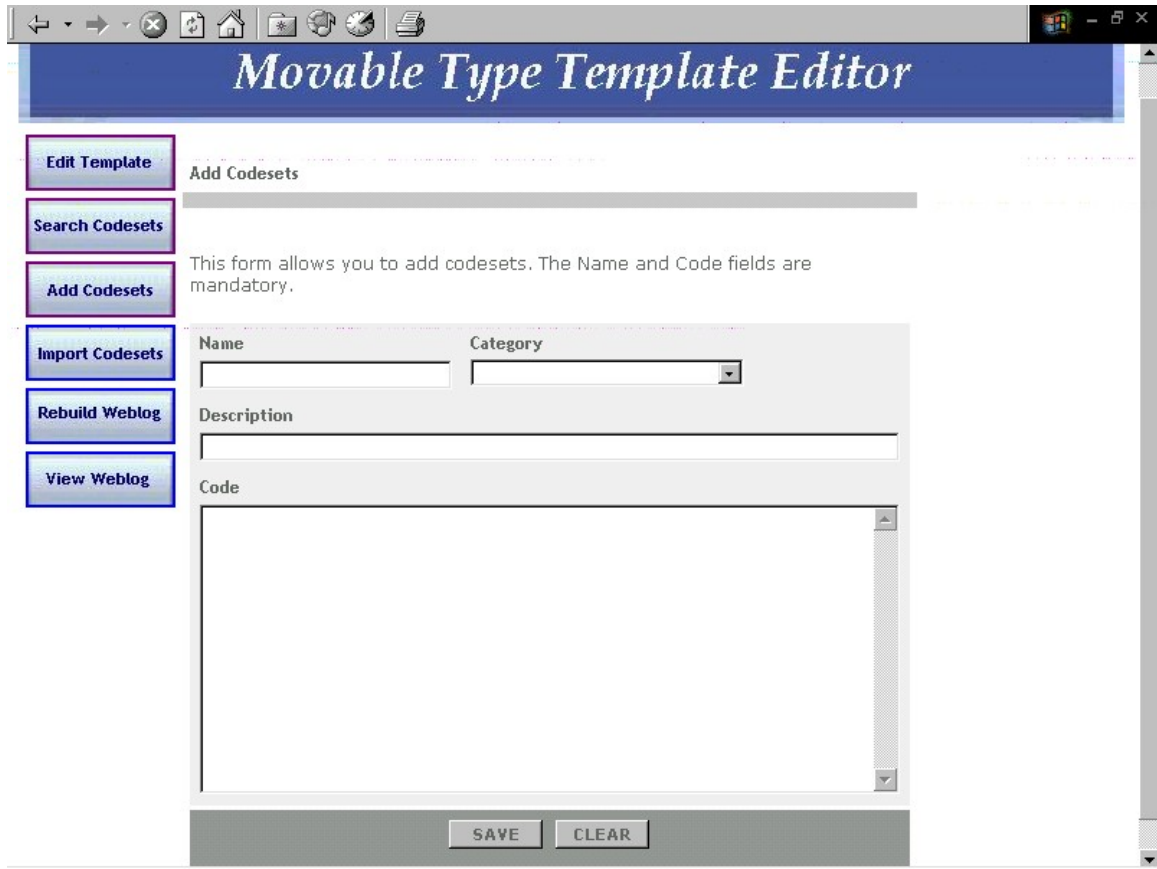Figure 20: Edit Codeset, Movable Type Template Editor

Figure 21: Add Codesets, Movable Type Template Editor

Figure 22 shows the interface for handling duplicates when importing codesets. The user is asked to rename, ignore or overwrite codesets to avoid duplicates in the database. After all duplicates are removed, the user is directed a page similar to Figure 19 which shows all codesets that have been imported.
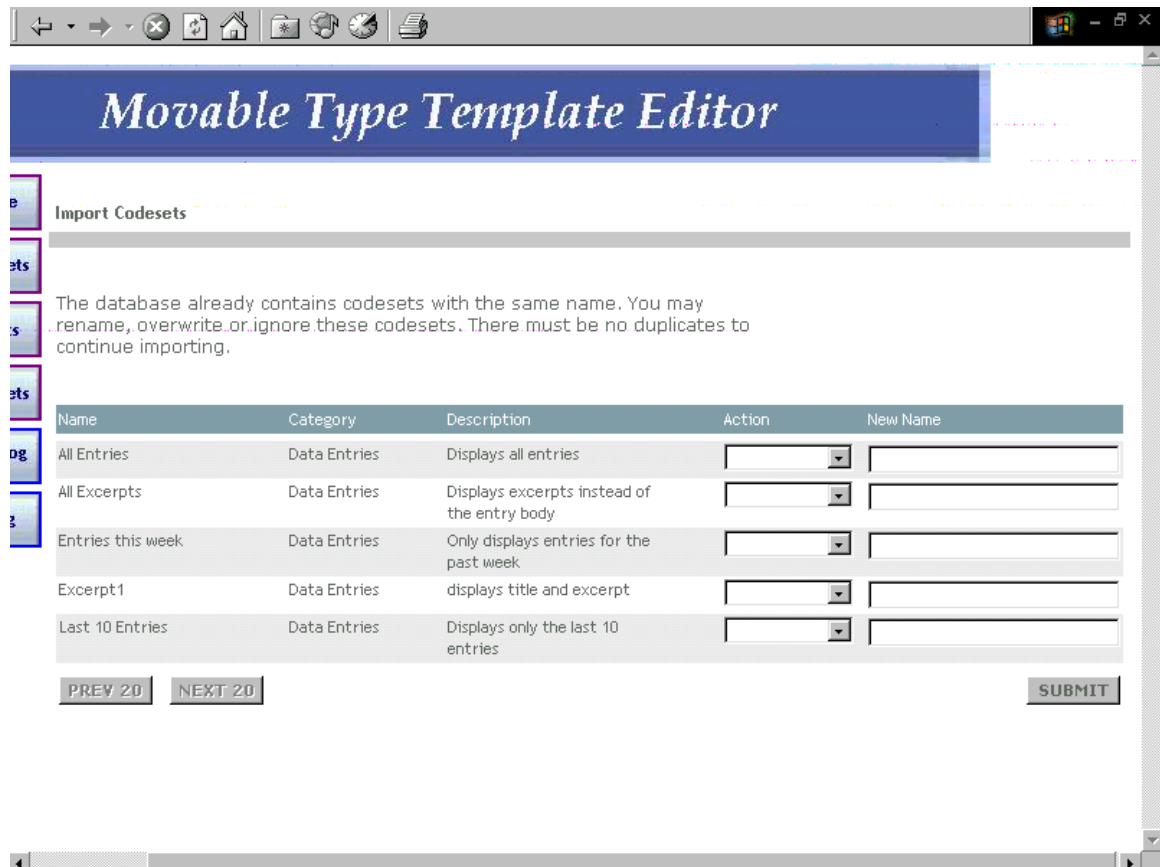


Figure 22: Import Codesets when there are duplicates, Movable Type Template Editor

When the user rebuilds the weblog, a pop-up window shown in Figure 23 emerges and performs the operation. The user may then view the updated weblog.



Figure 23: Pop-up for Rebuild Weblog, Movable Type Template Editor

**VI. DISCUSSION**

Movable Type Template Editor uses codesets to aid in template editing. Codesets are tags used in place of MT code. MT Template editor has a database to store and manage these codesets. It is easier to edit MT templates with codesets. A codeset tag performs more complex operations than MT or HTML tags. Its implementation details are hidden from the user. The source code of templates using codesets are also shorter and cleaner than regular templates. However, a user who does not have technical knowledge on MT markup are limited to the codesets stored in the database. To compensate, the codeset database is made extensible with the import and export functions for codeset sharing. Savvy users may share codesets with other people.

The system allows editing of MT templates either through its source code (HTML mode) or via its browser rendered form (Normal mode). The user may add HTML, MT and codeset markup. The user may also apply CSS styles to the template. It provides a graphical and user friendly interface for template editing. Due to the limitation of MSHTML, the engine that renders the template, only the code within the Body tags can be viewed and modified.

The user may rebuild the weblog through the MT Template Editor to apply the changes made to the template. The user does not need to use Movable Type weblog tool to rebuild or view the weblog.

**VII. CONCLUSION**


      The Movable Type Template Editor is able to view and edit MT templates. It is able to store codesets in a database and incorporate them with the templates. It is also able to rebuild and view the updated weblog.

**VIII. RECOMMENDATIONS**

The template editor could be improved by using images as a preview for codesets so that the user may have an idea of how the weblog will look like while editing the template without having to rebuild it to see the changes.

The limitation of the editor is that it can only edit the code within the body tags exclusive. This limitation is based on the underlying MSHTML program provided by Microsoft. If future versions of MSHTML has removed this limitation, the editor can be improved by providing functions that edit other parts of the template as well.

## IX. BIBLIOGRAPHY

[1] http://www.rebeccablood.net/essays/weblog_history.html

[2] http://www.extension.iastate.edu/mt/dcoates/extras/weblogs2002_part_one.html

[3] http://radio.weblogs.com/0110772/stories/2002/10/03/

personalKnowledgePublishingAndItsUsesInResearch.html

[4] http://www.smh.com.au/articles/2003/06/13/1055220765389.html

[5] http://www.extension.iastate.edu/mt/dcoates/extras/weblogs2002_part_two.html

[6] http://www.movabletype.org

[7] http://www.ojr.org/ojr/lasica/1019165278.php

[8] http://www.ojr.org/ojr/workplace/1017958873.php

[9] http://newhome.weblogs.com/historyOfWeblogs

[10] http://slashdot.org/features/99/05/13/1832251.shtml

[11] http://www.camworld.com/archives/001177.html

[12] http://www.cnn.com/2003/TECH/internet/03/10/bloggers.ap/index.html

[13] http://www.techtv.com/screensavers/answerstips/story/0,24330,3423506,00.html

[14] http://wmf.editthispage.com/discuss/msgReader$7629?mode=day

[15] http://hotwired.lycos.com/webmonkey/02/18/index3a.html?tw=authoring

[16] http://jasonnolan.net/archives/000813.html.

[17] http://www.javaworld.com/javaworld/jw-05-2001/jw-0518-encapsulation.html

[18] http://www.contentmanager.eu.com/history.htm

[19] http://www.terradoncommunications.com/about/cms.asp

[20] http://www.poynter.org/dg.lts/id.6626/content.content_view.htm

**X. APPENDIX (SOURCE CODE)**

1. temped.php
2. codesetadd.php
3. codesetsearch.php
4. codesetresult.php
5. codesetload.php
6. codesetimport.php
7. config.php
8. editor.css
9. source/ewe.js
10. source/popup.js
11. source/button.js
12. source/ewe_langEN.xml
13. source/style.css
14. source/style_syntax_highlite.css
15. popup/codeset.php
16. popup/editCS.php
17. popup/flash.php
18. popup/hyperlink.php
19. popup/image.php
20. popup/colorchooser.html
21. popup/editA.html
22. popup/editIMG.html
23. popup/editTable.html
24. popup/editTD.html
25. popup/hline.html
26. popup/table.html
27. popup/colorchooser.js
28. popup/popup.css
29. popup/colorchooser.css

**XI. ACKNOWLEDGEMENTS**

This special problem would not have been completed without the people who kept me motivated to finish this project. Sometimes to the point of nagging me. (Thank you).

To my family, for constantly reminding me that an INC is not an excuse to defer my SP for a year. You can sleep well now that we have all graduated, finally.

To Darwin, JP, Editha, Naid, Kai, and other fellow catch-uppers, it's very comforting that I'm not alone in this ordeal fighting both boredom and lack of allowance. Thanks for being there.

To Bobodolu, the biboest block in school, thanks for making up for my laziness and tardiness in college. 3 down four to go! You'd better finish your SP's too!

To Mam Sheila Magboo, my adviser, for being so supportive. You helped me a lot, from documentation writing to my SP defense. I wish you were my adviser from the start.

To my Pentium MMX computer, you're old but you managed to complete a SP. You may finally retire now.

To 4th year compsci students I just met during our SP defense. I was comfortable with your company. I'm glad to have known you before we part ways.

To my blockmates, the Noisy Bunch and Mascians. When's our next gimik?

To my dog Brunie who passed away, thanks for being my personal pet.