T100 – A Content Management System for PHP Web Applications Development

Dragos Badea*, Anton Duca**

** Electrical Engineering Department, Politehnica University of Bucharest, Bucharest, Romania, (Tel: +4076-659-2465; e-mail: dragos@webstarsltd.com)*
*** Electrical Engineering Department, Politehnica University of Bucharest, Bucharest, Romania, (e-mail: anton.duca@gmail.com)*

Abstract: This paper proposes T100[*] as a new Content Management System (CMS) dedicated to the PHP web applications developers. T100 can be used as an end product CMS but its main goal is to provide an application framework and tools in order to support and speed up the web applications development process. The proposed CMS has a router – dispather based structure wich allows developers to create application specific modules and integrate them with the CMS withouth changing the CMS core code. The paper also presents details about the structure, the functionality, and the features of T100, a comparison with the main PHP CMSs available on the market at the present time, and some of the most representative real world web applications wich use T100.

1. INTRODUCTION

A CMS is a software system which provides website authoring, collaboration and administration tools designed to allow users with little knowledge of web programming languages or markup languages to create and manage the site's content with relative ease [1][2]. A rich CMS provides the foundation for collaboration, offering users the ability to manage documents and output for multiple author editing and participation.

Most systems use a database to store content, metadata, or artifacts that might be needed by the system. Content is frequently, but not universally, stored as XML, to facilitate, reuse, and enable flexible presentation options.

A presentation layer displays the content to web-site visitors based on a set of templates. The templates are sometimes XSLT files.

Most systems use server side caching boosting performance. This works best when the CMS is not changed often but visits happen on a regular basis.

Administration is typically done through browser-based interfaces, but some systems require the use of a fat client.

Unlike web-site builders, a CMS allows non-technical users to make changes to a website with little training. A CMS typically requires an experienced coder to set up and add features, but is primarily a Web-site maintenance tool for non-technical administrators.

A CMS typically has the following characteristics: automated templates, scalable expansion, easily editable content, scalable feature sets, web standards upgrades, workflow management, collaboration, delegation, document management, content virtualization, content syndication, multilingual, versioning.

The main advantages of the CMSs are: low cost, easy customization, easy to use and work flow management.

Low cost. Some CMSs are free like Drupal [3], Joomla [4], and WordPress [5]. Others may be affordable based on size subscriptions. Although subscriptions can be expensive, overall the cost of not having to hire full developers can lower the total costs. Plus software can be bought based on need for many CMSs.

Easy customization. A universal layout is created, making pages have a similar theme and design without much code. Many CMS tools use a drag and drop AJAX system for their design modes. It makes it easy for beginner users to create custom front-ends.

Easy to use. Designed for non-technical people, the simplicity in design of the admin UI allows content managers and other users to update content without much training.

Work flow management. Being able to control how content is published, when it is published, and who publishes it. Some CMS systems allow administrators to set up rules for the workflow management.

The main disadvantages of the CMSs are: cost of implementation, cost of maintenance, storage volume and latency.

Cost of implementation. Larger scale implementations may require training, planning, and certifications. Certain CMSs may require hardware installations. Commitment to the

[*] T100 CMS is a registered trademark of Webstars LTD, UK (http://www.webstarsltd.com).

software is required on bigger investments. Commitment to training, developing and upkeep are all costs that will incur for enterprise systems.

Cost of maintenance. Maintaining CMSs may require license updates, upgrades, and hardware maintenance.

Storage Volume. Volume of files may be large in HTML based systems. A site that contains many files leaves itself open to errors. For example a client updating the site may create errors. Large amounts of files can cause issues with updating. Trying to find the right file may take time, and maybe hard to find.

Latency issues. Larger CMSs can experience latency if hardware infrastructure is not up-to-date. Also can cause latency issues if databases are not being utilized correctly. Large cache files can also be a problem. Caches have to be reloaded every time data is updated. Load balancing issues may cause issues with caching files. Refreshing caches often can cause issues with performance, and can defeat the purpose of caching in the first place.

Besides the mentioned disadvantages, another important problem is the building of the new application specific features in addition to those provided by CMS. When a web application is built using a CMS most of the time the integration of new features involves changes in the CMS core code.

In order to solve these kind of problems a new CMS with a router – dispatcher structure is proposed in this paper. The T100 CMS provides an application framework and tools in order to support and speed up the web applications development. The main advantage of the T100 CMS is the ability to integrate new application specific modules with the CMS core withouth changing the CMS core code.

2. The PHP CMSs

For PHP there is a significant number of CMSs some free and others available based on size subscription. The most widely used PHP CMSs are Drupal [3], Joomla [4], and WordPress [5].

Drupal is a free and open source CMS written in PHP and distributed under the GNU General Public License. It is used as a back-end system for websites ranging from personal blogs to corporate, political, and government sites. It is also used for knowledge management and business collaboration. The standard release of Drupal, known as Drupal core, contains basic features common to CMSs. These include user account registration and maintenance, menu management, RSS-feeds, page layout customization, and system administration. The Drupal core installation can be used as a brochureware website, a single or multi-user blog, an Internet forum, or a community website providing for user-generated content.

Joomla is a free and open source CMS for publishing content on the web and intranets. Joomla is written in PHP, uses object-oriented programming techniques and software design patterns, stores data in a MySQL database, and includes features such as page caching, RSS feeds, printable versions of pages, news flashes, blogs, polls, search, and support for language internationalization.

	Drupal	Joomla	Wordpress	T100
Framework for development	no	yes	no	yes
Object Oriented Programming	no	yes	no	yes
Access Control List	yes	yes	weak	yes
Multilanguage content	yes	yes	addon	yes
URLs Search Engine friendly	yes	yes	yes	yes
Resize image	addon	yes	weak	yes
System for templates	weak	yes	no	yes
Developer role	no	no	no	yes
Content Publishing Programming	addon	yes	addon	yes
Content revision	addon	addon	yes	yes
Multilanguage interface	yes	yes	yes	yes
Database tables abstracting	no	no	no	yes
Custom emails	yes	no	no	yes
Per page layout	yes	yes	no	yes
Layout blocks	no	yes	yes	yes

Table 1. Features comparasion between PHP CMSs

WordPress is an open source CMS, often used as a blog publishing application, powered by PHP and MySQL. It has many features including a plug-in architecture and a template system.

A comparision between the three PHP CMSs is shown in table 1. All CMSs have a component in common (manageable pages) but very few find the balance between expanding the pages module to facilitate the development and the implementation of modules independently of the pages module. In between are a variety of problems, such as SE-friendly URLs (difficult to link to particular modules), the order of HTTP request handling, the access rights for developers and the lack of an application framework. Though some CMSs (such as Joomla) have an application framework the integration of new features involves changes in the CMS core code.

The T100 CMS aims to ease the development process through the implementation of key features missing in most of the PHP CMSs: application framework, developer access rights, link of particular independent modules with SE-friendly URLs.

Besides the specific features for the development process, the T100 will also offer features such as bootstrapping, access control lists, encryption, database access, filters, email, web services, unit testing, multilanguage, PDF generation, image processing.

3. PROBLEM DESCRIPTION

3.1 Assumptions

Let us consider a CMS that allows editing and adding pages from the administration section. The CMS frontend menu contains a contact page and the pages accessible from the frontend menu can be ordered from the administration section.

3.2 Requirements

The developer has to build a web application for which the contact page should contain, in addition to the CMS contact page, a contact form. The contact form allows the user to contact the site owner through email and in the same time user specific information is stored in a database.

3.3 Solutions

The problem stated above can be approached in two ways:

- contact page is created as a CMS page and in the frontend is implemented the following condition: if the selected page is the contact page the contact form is added and the data processing is made.
- a separate controller is created; the CMS contact page content is handled by the controller; the controller also processes the contact form

First approach has the following disadvantages:

- if the number of pages with specific application content is increasing more changes have to be made
- if specificity is complex code changes will be more difficult

The second solution is closer to the standard development, but brings problems of integration with the CMS:

- the content of the page has to be handled *manually* by the controller
- the contact link from the frontend menu has to be changed and point to the controller
- because the pages accessible from the frontend menu can be ordered the link positioning is difficult

In both cases changes to the CMS core code have to be made. Because the changes are application specific the changed CMS core can not be used for different projects.

4. The T100 CMS STRUCTURE

4.1 The T100 CMS structure

The T100 CMS structure consists of 4 main blocks (figure 1): a router, a dispatcher, the CMS core and a MVC action block.

The router is based on a Zend Framework [6] implementation. The router processes the HTTP request and based on the URL rewrite rules decides whether the request should be sent to the dispatcher or a MVC action has to be invoked. By default, if no rule is matched, the request is sent to the dispatcher.

The dispatcher analyzes the request from the router and searches a CMS page that meets the criteria received. If the page is found the CMS module executes all the page characteristic actions (builds menu and possible submenus, meta tags, layout, etc.).

The dispatcher then checks if the page is attached to a forwarding action. If not, the CMS selected page is rendered.

If the page has a forwarding action attached then the designated MVC action is invoked. The MVC action block can decide the next view to be rendered. By default the initially selected CMS page is displayed.

The forwarding is one of the most important features that the T100 CMS has to offer. Any page can have besides the basic functionality an attached MVC action. Using the MVC action the developer can implement any application specific functionality without altering the existing CMS core code.

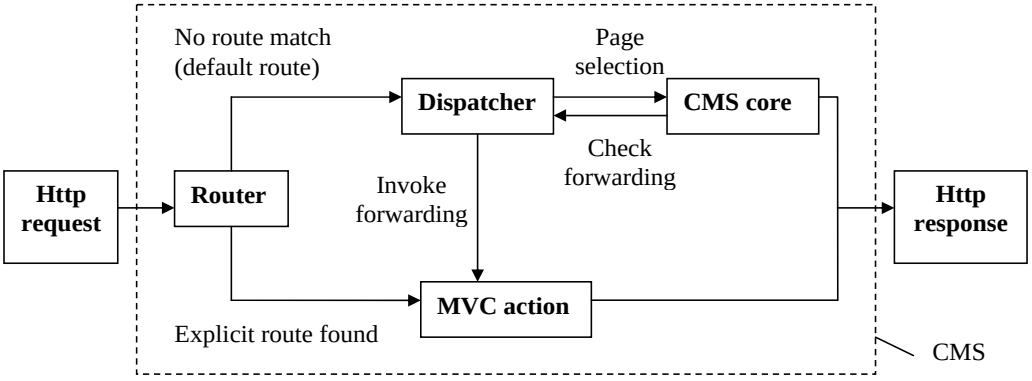

Fig. 1. The T100 CMS structure and HTTP request processing

In the production environment, the forwarding can be enabled or disabled (per page) through the administration panel. Activation or deactivation can be done only by a special administrator role called *developer*.

4.2 The T100 CMS implementation

The T100 CMS contains two different libraries: a general library (a framework) and an application specific library.

The T100 general library has two components the Zend framework (as a base layer) and the WS library (as a high level layer). The WS library is a library built specifically and only existing for the T100 CMS. The WS library extends the functionality of the Zend framework and provides features like encryption, database access, email, web services, multilanguage and image processing. In the same time WS library contains a series of abstract objects, collections, controllers, models, helpers, blocks and views. For Models are available the facilities: database interaction, data retrieval, collection mapping, persistence, paging, observers, etc.

The application specific library is implemented as a structure based on modules (figure 2). A module can contain one or more of each of the following entities: controller, model, helper, block, view. The three main modules are *admin*, *CMS* and *default*.

The *admin* module handles only the application backend. The *admin* module has its own type of controller that automates many tasks in a typical administration section (lists, forms, error messages etc.).

The module has three main sections: *ACL* (Access Control List), *CMS* and *settings*. The *ACL* section deals with management of access lists. This is in turn separated into three subsections actions, groups and administration accounts. The *CMS* section contains subsections that allow editing of all components related to content management: pages, layouts, emails, blocks. The *settings* section contains features which do not fit in the other sections. The section allows the configuration of the following parameters: maintenance page, content management help section, details about the Google Analytics service.

The *admin* module contains a number of helpers meant to ease the construction of the user interaction for administration purposes: Acl (used in assigning rights to users and groups), Crop (used to resize images in a visual way), Error (used to visually build error messages), Help (to display helpful content for the current page), List (implements certain facilities for list pages), Snapabug (for service reporting bugs).

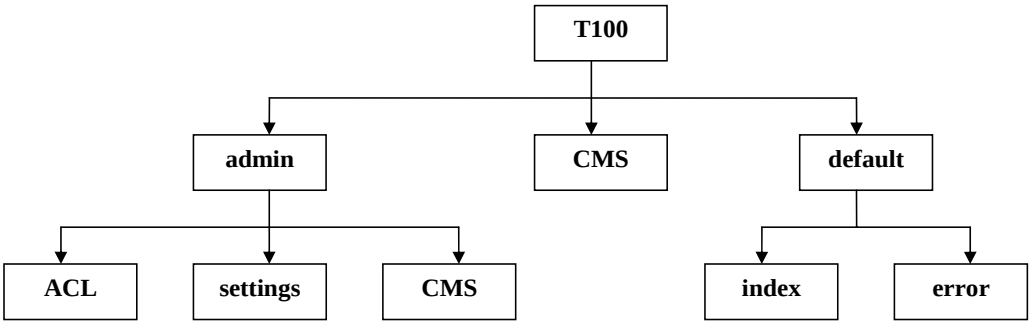

Fig. 2. The T100 CMS implementation

The *CMS* module is responsible for 90% of your application frontend. Although implementation is simple enough, almost all of the presentation of the application is found in this module. Among the models included in this module we can enumerate BlockModel, LayoutModel and PageModel. The BlockModel is the model that deals with processing and obtaining information for objects of type block in applications with manageable blocks per page or per layout. The Layout Model is a model that provides information about the structure and the layout of the text for a CMS web page (each page can have its own layout). The PageModel is a model that contains all information about a CMS page. The content of a page is stored in revisions, the return to a previous version can be made from the administration panel.

The *default* module contains a series of tasks (too simple to have their own module) and general helpers. The module has a controller for error handling and a controller for the main home page of the application (in case this is not a CMS page). Some of the helpers that can be found in this module are Date, File, Form, Google, Pdf and Text.

5. APPLICATIONS

The T100 CMS is currently used by a significant number of web applications. Some of the most representative real world web applications wich use the T100 CMS, are Shakespeare's Globe – www.shakespearesglobe.com, Royal Anyversary Trust – www.royalanniversarytrust.org.uk, Hanson Wade – www.hansonwade.com, YaRooms – www.yarooms.com, SipStatus Communication – www.sipstatus.com, Programme Recruiment – www.programme-recruitment.com, and Webstars LTD – www.webstarsltd.com.

Shakespeare's Globe is one of the most famous theatre in the world, where Shakespeare's plays were first shown to the public. The Shakespeare's Globe application is a good example of a CMS without many specific enhancements powering a fairly complex website.

Royal Anyversary Trust is a presentation website for the educational event organized by the Queen of the United Kingdom.

The implementation of the CMS is pretty standard and it uses layouts and forwarding for the special pages, like the news section. Also they make use of the media component shipped with the CMS which allows the user to aggregate images and videos in galleries. The app uses a video conversion system that converts standard video formats (avi, mpeg etc.) in flash (flv) format for web use.

Hanson Wade is one of the biggest event organizers in the world. The Hanson Wade application is a corporate website, as well as a tool for creating and managing event sites and community sites which reside on their own domains.

The main core of the application is the CMS on top of which there are implemented a lot of features using the application framework shipped with the CMS. The clients can create virtually unlimited websites with their own structure and predefined, special types of pages. This implementation makes intensive use of layouts, custom block types and metadata.

YaRooms is a web app that lets users manage meeting rooms, bookings and resources. The presentation website of the application is a standard CMS which makes intensive use of layouts and forwarding for the pages that contain web forms.

SipStatus Communication is an application for a VOIP services retailer and consultant with focus in Italy, UK, Spain and France. The web application uses multi-language support in the CMS, and also the URLs of the pages are translated versions of the main language page URL.

Programme Recruiment is a company specialized in recruiting project managers. The Programme Recruiment application is a presentation website, as well as a powerful online tool to evaluate and keep record of their job candidates.

The implementation features an online tests module as well as membership and payment modules.

Webstars LTD is a presentation website of the company holding the legal rights of the T100 CMS. The application consists of a standard CMS with enhancements for the the portfolio and blog accomplished with the use of forwarding.

6. CONCLUSIONS

The paper proposes T100 as a new CMS, with a router – dispatcher structure, dedicated to the PHP web application developers. The T100 CMS provides an application framework and tools in order to support and speed up the web applications development. The main advantage of the T100 CMS is the ability to integrate new application-specific modules with the CMS core withouth changing the CMS core code.

The paper presented details about the structure, the functionality, and the features of T100 and information about implementation. A comparision with three popular PHP CMS was made. Also some of the most representative real world applications wich use T100 CMS were shortly descibed.

REFERENCES

Ethier, K. and S. Abel (2007). Introduction to Structured Content Management with XML. CMS Watch. http://www.managingenterprisecontent.com.

Johnston, M. (2007). What is CMS?. CMS Critic. http://cmscritic.com/what-is-a-cms

Buytaert, D (2001). Drupal Content Management System. http://drupal.org/

The Joomla Project Team (2005). Joomla Content Management System. http://joomla.org/

Boren, R et al. (2003). Wordpress Content Management System. http://wordpress.org/

Zend Technologies (2005). Zend framework. http://framework.zend.com/