Mod-late: an innovative model for the construction of web templates for CMSs
G. Dimauro, P. Quintavalle, S. de Nichilo
Dipartimento di Informatica - Università di Bari, Italy

Abstract – In this paper we propose an innovative model called Mod-late, whose main aim is to automate the process of web template creation and customization of the overall structure of the web pages. Mod-late describes how to produce fully customized web templates by standardizing the construction tasks. These templates, both simple and complex, are created in order to be exported in packets which can be interpreted by the CMSs on the web or ready to be stored in a fully modifiable format. Our model was designed in open mode and allows full compatibility with next-generation CMSs and E-Learning platforms.

Key words: Content Management System, Web Template, E-learning

I. INTRODUCTION

In the latest years the Internet has constantly changed and it does not stop so that there is someone who talks already about the Web 3.0 and who tries to put the base for the web of the future, based on the semantic and the natural language processing [1]. Despite the growing use of CMSs their maturation has not reached a turning point and the cause must be sought in the gap between the skills required for their use and the knowledge of the average users. The area concerned is the Web 2.0 where the user is no longer just reader but a producer of networked information. “You can see the Web 2.0 as a set of principles and practices that tie together a veritable solar system of sites that demonstrate some or all of those principles, at a variable distance from that core” [2]. In this solar system several CMSs present the same problem: the only way to structurally interact with their web template is through the technical changes as well as with the E-Learning platforms. As it is known, the Content Management System is a tool designed to publish content to the Web without knowledge of HTML in a controlled environment that ensures a consistent look and feel across all the pages of a web site [3]. A CMS is not really a product or a technology. It is a catch-all term that covers a wide set of processes that will underpin the "Next Generation" large-scale web site [4].

The state of the art about the context of automated web templates creation for CMSs and E-Learning platforms is not so advanced. “Artisteer” is the only Web design automation product that instantly creates fantastic looking Website and Blog templates. It is a powerful software that makes the user a professional Web designer of Websites [5]. Starting from a basic theme, it is possible to create visually appealing templates even if they are not structurally very customizable.

II. AN INNOVATIVE MODEL

The model proposed in this work aims to automate the process of creating and editing web templates. This way the user can save time creating web sites using a CMS as well as creating and editing the E-Learning platforms view. Additionally the implementation of a software system based on our model, and its subsequent use, leads to standardize the processes and results in the creation of web templates, increasing the percentage of compliant templates. The distance between the web users and the CMS and E-Learning platforms is very important. These systems are common on the internet and in order to open them new doors the ideal solution would fit an application between the users and the structure of the web application they prefer to use. The purpose of Mod-late is to describe how to allow the users to build web templates and to export them towards the desired platform. Mod-late is fully compatible with such platforms: according to the guidelines dictated by the model is possible to produce web templates for every web application. The model describes a particular generator and operator of HTML, PHP and CSS which is expandable by adding logics in order to import / export web templates for the system that the user wants to look out. For a web programmer the model describes the techniques to quickly produce web portals. For a common user the module describes a powerful tool for creating web templates associated with the publishing tool that he prefers. The model is expandable by simply integrating the modules which describe how to import and export web templates.
III. MODEL DESIGN

The model is structured on a three-level architecture:
- The presentation level, designed to handle the interaction of the model with the outside world;
- The domain level, which includes a set of business rules;
- The data level, which deals with the continuing entity treated.

The term “entity” means a set of objects used by the model to build up web templates.

Three type of entity are described:
- An entity type which can contain any other entity (Container);
- An entity type which can horizontally divide the other entity types (H-Divisor);
- An entity type that divides vertically any entity (V-Divisor).

The domain level is divided into independent modules used for particular functions. Each module must have a low level of coupling with other modules and a high internal cohesion while respecting the principle of modularity. Modularity is required to allow the developer to easily implement a system based on the model. It substantially simplifies the design process by substantially simplifying the individual components to be built [6].

At the edge of domain level two managers must be placed to enhance the modularity between the levels of the proposed model.

These managers may be called “StructureManager” and “PresentationManager”.

The "StructureManager" puts in communication the data level with the domain level while the "PresentationManager" must have as its purpose the management of the information exchanges between the domain level and the presentation level.

The core modules of the domain level are the Hierarchy module and the Representation module.

The Hierarchy module describes how to create the hierarchy of entities of a web template. This module requires that the root of the hierarchy is a container entity. All entities should be included in the root container. Every incorporated entity must be adapted to its container entity; it can just keep distance from the limit of the container. Entities that are on the same level of the hierarchy cannot overlap.

The Representation module outlines how the entities of the model can be represented. The main aim is to enable the creation of web templates, so the geometric shape that could represent the entities of the model is the rectangle.

The location of the figure must be well defined on the Cartesian plane. There are no restrictions on the use of rectangles to represent entities.

The developer who implements the model may decide to adopt any kind of geometric figure to represent the model entities. The only condition required is that the chosen figure must be genuinely representative.

Instead the presentation level is not particularly descriptive: it establishes only that a good presentation must include all the necessary elements to display the hierarchy of entities. This view must respect the hierarchy built in the Hierarchy module and must additionally respect the type of representation chosen in Representation module.

IV. MODEL IMPLEMENTATION

The proposed model was applied in the development of an application of web templates creation.

This basic application is developed following the object-oriented paradigm and it works primarily by using the entities described in the data level of the proposed model and with the implementation of the described modules. This application is able to produce web templates and export them in complete interpretable formats.

One of the most used CMS today is Joomla!. It is a CMS developed using PHP and open source published under “GPL v.2” license. It is at the forefront of one of the most dynamic trends in open source software: the growth of open source content management” [7].

Because of its characteristics "Joomla!" was used as an example for the development of the system that implements the model proposed in this paper.

V. SYSTEM DATA LEVEL

The main aim of the data layer is to manage the system entities. These entities result from the model description as follows:
- Container, entity used to hold every other entity;
- Band, which takes care of splitting horizontally the other entities;
- Sidebar, entity that creates vertical subdivisions in the entities.

During the design phase of the model-based system the entities have been further extended.

The band has been specialized into FreeBand and ModulesBand: the FreeBand can contain each type of entity while the ModulesBand may contain only a limited number of adjacent containers.
During the export phase inside of these containers the chosen Joomla! modules will be placed.
The Sidebar instead has been specialized in LeftSidebar and RightSidebar to simplify its positioning.
Every entity described by the model, within those entity extensions, was carried out in order to have all the properties to be represented on the web.
Among the properties it is important to underline: width, height, margin, padding, background image, background color and borders.

VI. SYSTEM DOMAIN LEVEL

The domain level is the business services environment which allows the presentation layer to communicate with the data layer.
In the development of the domain level two basic modules have been implemented: the Hierarchy module and Representation module which are used by the managers to communicate among themselves and with the presentation layer.
In order to implement the hierarchy in the relative module a tree data structure has been used.
With regard to the Representation module the developers have decided to use rectangular shapes for the representation of the entities.
During the analysis phase some needs have come up, needs that led to the creation of various modules in the cloud of the Utilities.
First, the need to translate the system in multiple languages has led to the creation of the Localization module through a series of parameter files (*.ini) communicates the text data translated to the presentation layer.
To manage application-level events and to configure the system two additional modules have been implemented: the Application module and the Configuration module.
The modules implemented inside the utilities cloud communicate with each other as much as they can communicate with adjacent modules (Hierarchy and Representation) only through the operators.

VII. SYSTEM PRESENTATION LEVEL

The presentation layer deals with the representation of the entities created by the hierarchy module using the Hierarchy Level Domain and the rules of representation given by the Representation of the same level form of architecture.
The presentation layer is completely independent from the domain level in order to respect the principle of modularity on which the model is based. This level is not divided into modules such as the domain level but has several interfaces that allow the management and system configuration and the creation of web templates.
The essence of the presentation layer is the preview screen that is useful to show the user what is actually being built.
In order to make the work more easy for the experienced users a property window has been placed. Such a window shows the properties of the entity the user selects in the preview so that it is possible to interact regularly with them without navigating menus.
It was finally introduced a view of the hierarchy created.
From this hierarchy it is also possible to insert or delete these entities while respecting the rules of the model.

![Figura 2 - Possible layout](image)

The division of web templates, as shown in Figure 3, in separate areas is useful for two reasons:
- first, the good structure of the web site or portal that you want to build;
- each area can also be set as an area for placement of the contents of the CMS.
The user can then decide on the exact locations where the content has to appear on its website, menus, forms and more. He becomes a master of the CMS to which exports as able to interact with the framework of the CMS itself.

VIII. TEST PHASE

The test phase was carried out concerning the creation of templates for web sites and portals built with Joomla!. To prepare these tests were collected from web various web templates already constructed and sample images of the web templates to be created from scratch. It was not adopted any policy for the collection of research material to make the results independent from the conditions that constrain the choice of web templates.
Several experiments have been performed to modify existing web template and creating new web templates. From these experiments the following results were obtained.
To edit a web template, it is needed for an advanced user from 10 to 30 hours, which should be added to the other if the user is willing to radically change the positions in which information is exposed.

![Figura 3 - Joomla! Template Edit Trend](image)
The creation of a web template from scratch however, requires knowledge of HTML, CSS, PHP and frameworks of "Joomla"; in this case about 20 hours to achieve the desired result are necessary. The necessary time for the creation of web templates in the experiments does not include the creation of graphics, but only the formatting of text, images and structure.

After collecting the data about the creation and the modification of web templates for Joomla! several experiments were carried out on the application which implements the model proposed in this work. These experiments are aimed at creating web templates and do not include the processing of graphics in order to make true comparisons with results in the creation of web templates from scratch, shown above. The experiments were also carried out on several people including an experienced user and an average user. The results of the experiments performed with the system based on the following model are very interesting.

### IX. Comments and Conclusion

The model brings a great innovation in the development of web sites and portals. Unlike the main concepts of "Artiseer" application, the center of the state of the concerned application domain, Mod-late makes significantly faster the creation and customization of the layout. The Mod-late innovation leaves behind the state of the art so that it can be implemented by a new class of applications, not existing yet.

It is important to underline that the time savings for the production of a basic layout in the experiments performed is interesting. A new structure can be created in few minutes and do not require long learning time as the operations to achieve the desired outcome are few and simple.

Allowing the creation of web templates very quickly and so easily it is possible to eliminate much of the barrier between the Web and average users who could be transformed into real protagonists of the design. This approach could mean an advancement of the use of CMSs as well as the use of E-Learning platforms and the reduction of the costs in a market where such systems abound.

Taking the study of CMSs and web templates to a more advanced stage it could be possible to make the model more precise giving more details about the description of the hierarchy creation. It could be eventually possible to look for commonalities between several systems in order to describe an additional part of the model dedicated to an abstract import / export policy for web templates.

The next step will be the implementation of a complete software system based on Mod-late that will result useful to test the power of the model and to give more precise results.

### X. References


