A Systematic Literature Review on CMS-based Web Applications

Feliu Trias, Valeria de Castro, Marcos López-Sanz and Esperanza Marcos
Kybele Research Group, Rey Juan Carlos University, Tulipán S/N, 28933, Móstoles, Madrid, Spain.

Keywords: Web Engineering, Model Driven Engineering, Systematic Literature Review, Content Management System.

Abstract: CMS-based Web applications have increased their presence in the market because of the advantages they offer with regard to information management, thus offering a new landscape in Web application development. In this paper we analyze the current state-of-the-art on the existing approaches for the development of CMS-based Web applications. To do it, we have followed a Systematic Literature Review (SLR). As a result of this SLR we found four approaches in the literature focused on this issue. In this paper we focus our attention in studying the insights of each approach by following four criteria (Web Engineering views, MDA abstraction levels, modeling language and level of automation) and we discuss about the interest shown by the Web Engineering community about the CMS-based Web application development.

1 INTRODUCTION

In the last years, organizations have experienced the necessity for using powerful management tools to maintain their large Web applications and manage the vast amount of information generated (Boiko, 2001) (McKeever, 2003). Some problems that these companies frequently face during the management of their Web applications include bottlenecks for the webmaster role, inconsistencies in the webpage look-and-feel, complex and confusing navigation or access to unauthorized content. To address all these problems one of the most popular adopted solutions has been the use of Content Management Systems (CMS) as platforms for the development of Web applications (Vidgen et al., 2001). The CMS-based Web Applications allow users to collect, manage and publish content online in a comprehensive manner. Moreover, such systems help companies to stay organized so that their Web applications can grow and evolve quickly while maintaining high quality of execution (Boiko, 2001).

CMS-based Web applications offer advantages that differentiate them from traditional Web applications. Some of these advantages are (Vidgen et al., 2001): 1) *Dynamic creation of content*, content is created and added dynamically by non-technical users of the Web, without requiring the intervention of the webmaster; 2) S*eparation between content and design*, the page graphical design is stored in a template and the content is stored in a database; 3) *Different levels of access authority*, many CMS-based Web applications permit the definition of different levels of access rights and 4) *Functionality extension*, achieved through module addition that reduces development costs.

Due to all of this, a vast number of large organizations have decided to base their sophisticated Web applications on these platforms (Vidgen et al., 2001). Besides, in the last decade the number of available CMS platforms in the market has growth very rapidly meeting different domains, such as blogs, e-commerce or e-learning (Shreves, 2011).

Seeing the success experimented by CMS-based Web applications we posed the following research questions: **1) Is there any approach in the literature for the development of CMS-based Web applications?** In particular, we are interested in knowing which approaches exist and in analyzing their insights. To perform this analysis we study the Web views (Deshpande and Hansen, 2001) covered by each approach, the Model Driven Architecture (MDA) (Mellor et al., 2002) abstraction levels they consider, the modeling language they use and their level of automation. On the other hand, the other research question is **2) Which is the interest of the Web Engineering community in researching about CMS-based Web applications?** Concretely, we want to know which research groups or

Trias F., De Castro V., López-Sanz M. and Marcos E..
A Systematic Literature Review on CMS-based Web Applications.
DOI: 10.5220/0004433401320140
In *Proceedings of the 8th International Joint Conference on Software Technologies* (ICSOFT-EA-2013), pages 132-140
ISBN: 978-989-8565-68-6

organizations are leading the research in this area and which forums publish studies focused on this issue.

To answer these questions, we follow a Systematic Literature Review (SLR) process (Kitchenham et al., 2009). In this paper we present the results extracted from this SLR. To perform this SLR we follow a rigorous and reproducible method for searching, collecting, analyzing and evaluating the available relevant information to the subject of the investigation. For that aim, we base our SLR on the guidelines proposed by Kitchenham et al. (Kitchenham et al., 2009) and Biolchini et al. (Biolchini et al., 2005).

In this paper we present the primary studies of our research that allow us to answer our two research questions. Therefore, we present an analysis of each approach focused on the development of CMS-based Web applications by applying four criteria (Web Engineering views covered, MDA levels considered, modeling language used and level of automation). Besides, from the author information, the year of publication of the primary studies and the forum where they were published we can discuss the interest shown by the Web Engineering community in CMS-based Web applications.

The structure of this paper is as follows: Section 2 presents the primary studies as a result of our SLR; Section 3 analyzes the insights of each approach found from the primary studies, Section 4 presents the discussion based on the two research questions posed and Section 5 presents the conclusions of our review.

2 SYSTEMATIC LITERATURE REVIEW RESULTS

The first task of our SLR was to define the objective of our research by defining the two research questions posed in the introduction. In the second task we defined the Research Query (RQ) including the keywords related to the main goal of our research. In this case, our RQ was composed of the following terms classified in a) and b):

a) "Web Content Management" OR "Content Management System" OR CMS.
b) "Web Engineering" OR "Web application".

RQ= (a) AND (b)

After the definition of the RQ, we selected the digital libraries to launch the RQ and to obtain the initial set of studies. The seven selected digital libraries are: 1) ACM Digital Library [**ACM**], 2) IEEEXplore [**IEEEX**], 3) ISI Web of Knowledge [**ISI**], 4) Science Direct [**SD**], 5) SpringerLink [**SL**] and 6) Scopus [**SCP**].

In the first launch of RQ, we found 1,484 studies by using the widest scope allowed for each digital library. To select the studies which provide valuable evidence for our research, we filtered the 1,484 studies by applying inclusion criteria. Therefore, we considered only the studies containing the terms defined in a) at least in the title, abstract or keywords as well as containing the terms in b) in any part of the text. Also, we considered those studies whose abstract let us to conclude that the main purpose of the study was a methodological approach focused on the development of CMS-based Web applications.

Surprisingly, we obtained 74 relevant studies out of the 1,484 total studies. We determined that most of the studies belonged to other research areas different from Software Engineering or Web Engineering. In the case of Scopus, the 45% of the studies found was about other research areas such as mathematics or physics. Afterwards, we removed the duplicated studies.

Finally, to accurate the SLR we applied the exclusion criteria to obtain the primary studies of our research. Therefore, we read entirely each study and we rejected those which only used CMS-based Web applications for supporting or implementing a concrete case study (not proposing a methodological approach) as well as those which were focused on the management of content not designed to be available through the Web. As a result, we obtained the list of 15 primary studies.

Table 1: Primary Studies list.

ID	PrimaryStudy	Publication
APPR1	(Souer et al. 2011)	iiWAS
	(Luinenburg et al. 2008)	MDWE
	(Souer, Luinenburg, et al. 2008)	iiWAS
	(Souer, Honders, et al. 2008)	JDIM
	(Souer, Honders, et al. 2007)	ICDIM
	(Weerd et al. 2006)	IJSPIP
	(Souer, Weerd, et al. 2007)	IJWET
APPR2	(Souer et al. 2009)	ICWE
	(Souer & Kupers 2009)	MDWE
APPR3	(Saraiva & Silva 2010)	Inforum
	(Saraiva & Silva 2009b)	ICSEA
	(Saraiva & Silva 2009a)	ICSEA
	(Saraiva & Silva 2008)	SEAA
APPR4	(Vlaanderen et al. 2009)	ICEIS
	(Vlaanderen et al. 2008)	ICEIS

Table 1 contains the complete list of the primary studies extracted. This table groups the primary studies in approaches according into their commonalities in terms of authors and proposed ideas. Accordingly, we define four different approaches identified with an ID (APPR*Num*), as it is shown in the first column. It is worth noting that APPR1 and APPR2, in spite of belonging to the same research group, have been considered as different approaches because of their different nature.

3 RESULT ANALYSIS

In this section we analyze the insights of each approach found in this SLR defining four different criteria which allow us to compare them and extract the main conclusions. The set of criteria is explained below.

The **first criterion** is to analyze which views proposed by Web Engineering are covered by each approach: 1) *content view,* is the view used to represent the business and data objects; 2) *navigation view,* is the view which expresses the composition of the interface in terms of container and the navigation map of the Web application; 3) *process view*, is the view that defines how the application reacts to the events raised by the user's navigation and 4) *presentation view,* is the view for specifying the layout and the look & feel of the interface. Moreover, to complete this analysis we have considered two views belonging to the traditional Software Engineering: the *requirements view* which is the view that reflects the functionality that Web applications offer to users and the *implementation view* which expresses Web applications in terms of artifacts implemented by code.

The **second criterion** is related to the abstraction levels proposed by MDA: Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM). We are interested in knowing the MDA abstraction level of the models used in each approach.

The **third criterion** is related to the modeling language used to define the models. We consider interesting to know if the approach is supported by a standard modeling language such as UML, or otherwise the approach proposes a Domain Specific Language (DSL) (Van Deursen and Klint, 2002) for the definition of their models.

The **fourth criterion** is about the level of automation of the approach. It is of interest to check if the approach implements automatic model-to-model (M2M) transformations and model-to-text (M2T) transformations (Mellor et al., 2002). If the approach defines both transformations (M2M and M2T) we consider it with a high level of automation. If the approach defines some of the transformations (M2M or M2T) we consider it with a medium level of automation. Finally, if the approach does not consider any type of transformation we consider it with a low level of automation.

3.1 Web Engineering Method (WEM)

This approach is identified as APPR1 in Table 1 WEM is a method for the development of CMS-based Web applications. It covers from a correct requirement definition to the implementation. This method is defined following Situational Method Engineering (SME) approach (Ralyté et al., 2003) that consists in taking parts of other existing engineering methods and customizing a new one for a certain domain.

In this case, WEM is composed of parts from the Unified Process (UP) (Jacobson et al., 2000) and UML-based Web Engineering (UWE) (Kraus et al. 2007) methods. Even UWE is a model-driven method, WEM is not considered as such. WEM is composed of six tasks (*Acquisition, Orientation, Definition, Design, Realization and Implementation*).

Considering the first criterion we realized that the requirements view is addressed within the Acquisition, Orientation and part of the Definition task by means of *Use-case models* and *Feature lists*. The presentation and navigation views are addressed only by the Definition task. Both are modeled by the *Application model* took from UWE. The process view is addressed by the Design task which determines how the requirements are realized and a suitable architecture is created. Finally, the implementation view of the CMS-based Web application is addressed during the Realization and Implementation tasks.

The former is the responsible for creating the CMS-based Web application and integrating the graphical user interface design. The latter, is when the CMS-based Web application is generally deployed straight to production. On the other hand, the unique view not addressed is the content view. Figure 1 presents the tasks of WEM and the Web engineering views that they address.

Some of the models which WEM proposes are: *Domain model*, *Use-case model* and *Application model*. According to the second criteria we can say

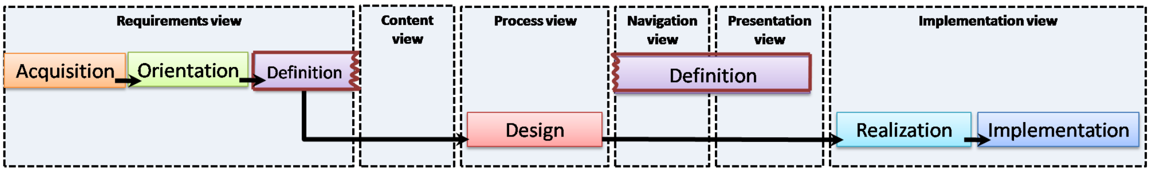

Figure 1: Matching APPR1 with Web Engineering views.

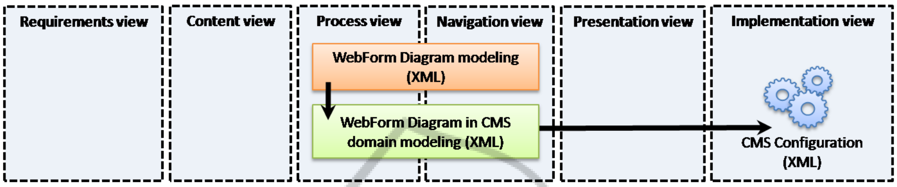

Figure 2: Matching APPR2 with Web Engineering views.

that the *Domain model* is the unique model that can be considered at CIM level. The rest of the models are mainly at PIM level because they are not related to any specific platform. On the other hand, WEM does not propose the use of any PSM model.

Regarding the third criterion, we saw that the modeling language proposed by WEM is UML, since it is the modeling language used by UP and UWE.

Finally, according to the fourth criterion, we realized that WEM has a low level of automation because it does not consider transformations M2M or M2T to implement the Web application.

3.2 Model-driven Method based on a Webform Diagram

The ID of this approach is APPR2 in Table 1. It is a model-driven method that allows the automatic configuration of a CMS-based Web application from a *Business process model*.

This method starts with the definition of a Web form diagram (henceforth, WebForm Diagram) that is implemented in XML. Afterwards, this model is transformed automatically to another XML model that contains the concepts available within the CMS domain. Finally, from this last XML model is automatically generated the specific XML that configures a CMS-based Web application.

Regarding to the first criterion we can expose that this approach only addresses the navigation and process views. The navigation view is addressed because the WebForm Diagram considers concepts such as, forms, steps, form elements and pages, otherwise the process view is addressed by concepts, such as action and handler. The implementation view considers the automatic generation of the XML model that configures the CMS-based Web application. Figure 2 shows the correspondence between the APPR2 and the Web Engineering views.

Regarding to the second criterion we consider the WebForm Diagram as a PIM model since it captures navigation concepts which are not related to any platform. On the other hand, this approach does not consider models at CIM and PIM level.

The WebForm Diagram intends to be understandable and intuitive enough to be used by non-technical users such as business users. Thus, considering the third criterion, APPR2 proposes a specific modeling language in form of DSL. To define the elements of the DSL they also followed the SME approach. They took into consideration other Web application modeling languages such as WebML (Ceri et al., 2000), OOWS (Valverde et al., 2007) and OOHDM (Schwabe and Rossi, 1995).

According to the fourth criterion, APPR2 has a high level of automation because it defines M2M and M2T transformations that generate the configuration of the CMS-based Web application from the WebForm Diagram. The M2M transformation is implemented by using Java. Otherwise, the M2T transformation is implemented in EXtensible Stylesheet Language Transformation (XSLT) (Anon 2012).

3.3 Model-driven Method based on a Webform Diagram

This approach is identified as APPR3 in Table 1. It is a model-driven method for the development of CMS-based Web applications. This method is composed of three tasks: the first task is the definition of the Web-site Templates model and the

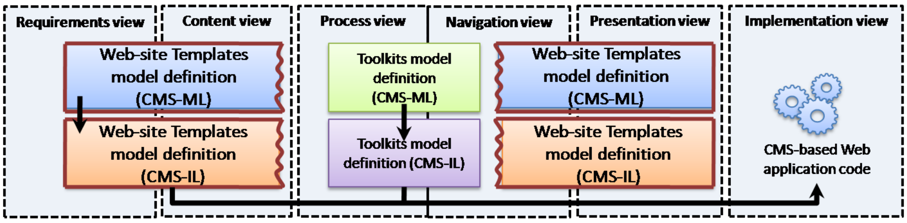
Requirements view
Content view
Process view
Navigation view
Presentation view
Implementation view
Web-site Templates model definition (CMS-ML)
Web-site Templates model definition (CMS-IL)
Toolkits model definition (CMS-ML)
Toolkits model definition (CMS-IL)
Web-site Templates model definition (CMS-ML)
Web-site Templates model definition (CMS-IL)
CMS-based Web application code

Figure 3: Matching APPR3 with Web Engineering view.

Toolkits model by using a modeling graphical language. The former reflects the structure, the content, the navigation and look and feel of the Web application. The latter allows the addition of task-oriented extensions and complex user interface elements to the elements of the Web-site Templates model. The second task is focused on the automatic definition of these models into a textual modeling language. Finally, the last task is to implement automatically the CMS-based Web application from these models.

Regarding the first criterion we can state that this method addresses the requirements, content, navigation and presentation views with the definition of Web-site Templates model.

On the other hand, with the definition of the Toolkits model, it addresses the process and navigation views. Finally, the implementation view is addressed by the automatic generation of code.

Considering the second criterion we can say that the Web-site Templates model and Toolkits model are considered at PIM level. Any CIM or PSM models are considered in this approach.

According to the third criterion, this method proposes two modeling languages to define their models: the CMS Modeling Language (CMS-ML) and the CMS Intermediate Language (CMS-IL).

CMS-ML is used in the first task in order to define graphically the models. It is a graphical modeling language that allows the definition of Web applications in a simple and efficient way. This language is also defined by using a DSL.

Although, APPR3 does not consider the SME approach to define their models, it is worth noting that the CMS-ML language is based on already existing languages such as WebML or UML.

On the other hand, the CMS-IL language is used on the second task. It is a textual language that provides a mechanism, independent to any particular CMS implementation, which can be used by technical stakeholders to address low-level computation aspects, and deploy a Web application model in any CMS platform.

According to the fourth criterion, this approach has a high level of automation because it defines M2M transformations that generate automatically the CMS-IL models from the CMS-ML models and M2T transformations that implement automatically the CMS-based Web application from the CMS-IL models. Even they propose the M2M and M2T transformations do not mention the language to implement these transformations. Figure 3 presents the matching between APPR3 and the Web Engineering views.

3.4 Adaptation of OOWS within the CMS Domain

This approach is presented in Table 1 with the ID APPR4. It is focused on the adaptation of the model-driven Web Engineering method OOWS (Valverde et al. 2007) to the development of CMS-based Web applications. OOWS's has been extended in order to address the modeling of these Web applications by including new tasks. This method has followed the SME approach to define these new tasks.

OOWS process is presented in Figure 4. For space restrictions, we will not explain the entire process; we will just focus on the required tasks for the development of CMS-based Web applications presented in red within Figure 4 which are: *the Define legacy classes task, the Identify stub services task, the Session information specification task, the Dynamic user information task and the Detail navigational map task.*

The *Define legacy classes task, the Identify stub services task and the Session information specification task* are introduced to complete the existing *Object model* with the necessary information for modeling CMS-based Web applications. Otherwise, the *Dynamic user information task* is introduced to complete the existing *User model* and the *Detail navigational map task* has been extended to cope with the CMS features.

Regarding the first criterion we can say that OOWS method addresses all the views proposed by the Web Engineering, as we can see in Figure 4. It is interesting to note that the models which have been adapted to the CMS domain address the content and

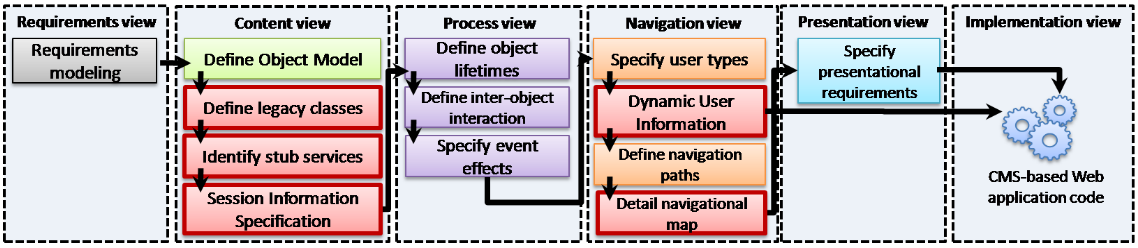

Figure 4: Matching APPR4 with Web Engineering views.

navigation views.

According to the second criterion, we can say that OOWS consider models at PIM and PSM levels. Concretely, the *Object model* and the *User model* are PIM models. Otherwise, we have not identified any CIM model.

Considering the third criterion, we can state that this approach does not define any specific modeling language to model CMS-based Web applications. They propose the use of standard modeling languages such as UML or BPMN for the definition of the models.

According to the fourth criterion, OOWS has a high level of the automation since it defines M2M transformations and the automatic generation of the code by M2T transformations. M2M transformations such as the transformation between the *Functional model* and the *Navigational model* are implemented by XSLT. As for the implementation of M2T they use Xpand language (OpenArchitectureWare, 2009).

4 DISCUSSION

After presenting and analyzing the approaches found in the literature we discuss and conclude in this section answering the two research questions leading this research.

Table 2: Summary of the SLR results.

	Views	MDA	Language	M2M	M2T
APPR1	R, PR, N, P, I	CIM PIM	UML	No	No
APPR2	PR, N	PIM	WebForm diagram	Yes	Yes
APPR3	R, C, PR, N, P, I	PIM	CMS-ML CMS-IL	Yes	Yes
APPR4	R, C, PR, N, P, I	PIM PSM	UML BPMN	Yes	Yes

Table 2 summarizes the results of the SLR. As for the column *views* we have used the initial of each Web Engineering view: (R)equirements, (C)ontent, (PR)ocess, (N)avigation, (P)resentation and (I)mplementation.

Is there any approach in the literature for the development of CMS-based Web applications?

With our SLR we found four approaches in the literature. In the following, we discuss the results regarding the four criteria defined.

As for the Web Engineering views we can say that all the approaches address at least one of these views. According to Table 2, the two most addressed views are navigation and process views and the least addressed one is content view. There are two approaches taking into account all the views: APPR3 and APPR4. Otherwise, APPR2 is the approach considering less views, only navigation and process views.

Regarding the use of models we say that all the approaches use models as important artifacts for their development process, but not all of them follow MDA principles (use of models + automatic transformations). For instance, APPR1 (WEM) is not considered as a MDA-based method, even being based on UWE and defining modeling tasks, because it does not consider automatic M2M or M2T transformations. Otherwise, APPR2, APPR3 and APPR4 are considered as MDA-based approaches.

As for the MDA abstraction level of the models and according to Table 2, we can say that most of the approaches consider models at PIM level. Just APPR1 bet for the use of models at CIM level and APPR4 consider models at PSM level. It is interesting to say that most of them try to generate automatically code from PIM models instead of PSM models. Only APPR4 consider PSM models to define M2T transformations.

Considering the modeling languages we can say that most of them use generic languages which are not specific to the CMS domain. Thereby, two of the approaches consider standard languages such as UML. As for new modeling languages proposed by the approaches we can consider three: the WebForm proposed by APPR2, and CMS-ML and CMS-IL proposed by APPR3. Except from CMS-IL, the other two are graphical languages. The three languages allow the modeling of navigation and process aspects, but WebForm's concepts are more generic (useful for modeling any kind of Web application) and not as focused on the CMS domain. Otherwise, CMS-ML and CMS-IL are more specific within the CMS domain. We consider that it would be interesting to provide a thorough CMS modeling language.

According to their level of automation, all the methods considered as MDA-based approaches take into account M2M and M2T transformations to automate their development processes.

It is worth noting that none of the four approaches have been defined from scratch. All of them are based on existing Web Engineering methods. Concretely, three of the approaches follow the SME approach to define their process.

Which is the interest of the Web Engineering community in researching about CMS-based Web applications?

We consider that the Web Engineering community has not dedicated much effort in the research about the CMS-based Web applications. It has surprised us considering the many advantages that they offer to companies and considering (how it is demonstrated by the APPR4) that the existing model-driven Web Engineering methods do not address thoroughly the development of this kind of Web applications. Just OOWS has shown interest in the adaptation of its process in the CMS domain.

Even so, we can find some active research groups such as the group in the Utrecht University and the group in the Technical University of Valencia.

Regarding the forums where the primary studies were published, we conclude that they are recent forums (from 2006 to 2011) but they are not high-level such as CORE-A or JCR levels, but they are mainly CORE-C.

Finally, we would like to comment that all the approaches found are methods for the traditional top-down development and we have not found any approach centered in the migration or modernization. Hence, we consider an interesting gap to cover in future works.

5 CONCLUSIONS

The interest shown by organizations in using CMS-based Web applications in the last years has aroused the necessity of analyzing the state-of-the-art on existing approaches for the development of this kind of Web applications. To do it, we have followed a SLR. As a result of this SLR we found four approaches. In this paper we studied the insights of each approach by following four criteria (Web Engineering views, MDA abstraction levels, modeling language and level of automation) and we discussed the interest shown by the Web Engineering community about the CMS-based Web application development.

After performing this SLR we conclude that the number of approaches focused in this issue is scarce despite the features of CMS-based Web applications and the advantages that offer to organizations. Most of these approaches are MDA-based. We state it is because of the excellent results obtained by MDA during the last years.

As for Web Engineering views criterion we conclude that the cornerstone of the development of this Web applications is on modeling navigation and process views.

In regard to the abstraction level criterion we say that these approaches define mainly models at PIM level and generate the code from this level.

Regarding the modeling language criterion, we consider that, even there are some specific modeling languages focused in the CMS domain, it would be interesting to provide another thorough CMS modeling language.

Finally, considering the level of automation criterion we can say that all MDA-based approaches take into account M2M and M2T transformations to automate their development processes.

The two most active research groups in the issue are the group in the Utrecht University and the group in the Technical University of Valencia. Otherwise, the primary studies found are published in recent forums (from 2006 to 2011) but they are not high-level, mainly CORE-C.

Finally, we conclude that all the approaches found are methods for the traditional top-down development. Hence, we consider the definition of a method for migrating CMS-based Web applications as an interesting gap to cover.

ACKNOWLEDGEMENTS

This research has been partially funded by the Project MASAI (TIN-2011-22617) from the Spanish Ministry of Science and Innovation.

REFERENCE

Anon, 2012. W3C. eXtensible Stylesheet Language Transformation (XSLT) v2.0. Available at: www.w3.org/Style/XSL/ (Accessed April 30, 2012).

Biolchini, J. et al., 2005. Systematic Review in Software Engineering. Engineering, (May).

Boiko, B., 2001. Understanding Content Management. Bulletin of the American Society for Information Science and Technology, 28(1), pp.8–13.

Ceri, S., Fraternali, P. & Bongio, A., 2000. Web Modeling Language (WebML): A modeling language for designing web sites. Computer Networks, 33, pp.137–157.

Deshpande, Y. & Hansen, S., 2001. Web engineering: creating a discipline among disciplines. IEEE Multimedia, 8(2), pp.82–87.

Van Deursen, A. & Klint, P., 2002. Domain-Specific Language Design Requires Feature Descriptions. Journal of Computing and Information Technology, 10(1), pp.1–17.

Jacobson, I., Booch, G. & Rumbaugh, J., 2000. The Unified Software Development Process,

Kitchenham, B. et al., 2009. Systematic literature reviews in software engineering – A systematic literature review. Information and Software Technology, 51(1), pp.7–15.

Kraus, A., Knapp, A. & Koch, N., 2007. Model-Driven Generation of Web Applications in UWE. Model-Driven Web Engineering, 261, pp.23–38.

Luinenburg, L. et al., 2008. Designing Web Content Management Systems Using the Method Association Approach. In 4th International Workshop on Model-Driven Web Engineering. pp. 106–120.

McKeever, S., 2003. Understanding Web content management systems: evolution, lifecycle and market. Industrial Management & Data Systems, 103(9), pp.686–692.

Mellor, S. J. et al., 2002. Model-Driven Architecture. In J.-M. B. and Z. B. (Eds.), ed. Advances in Object-Oriented Information Systems. Springer-Verlag Berlin Heidelberg 2002, pp. 233–239.

OpenArchitectureWare, 2009. Xpand. Available at: http://www.eclipse.org/modeling/m2t/?project=xpand.

Ralyté, J., Deneckère, R. & Rolland, C., 2003. Towards a generic model for situational method engineering. In J. Eder & M. Missikoff, eds. CAiSe 2003. Heidelberg: Springer, p. 1029.

Saraiva, J. D. S. & Silva, A.R. Da, 2009a. CMS-based Web-Application Development Using Model-Driven Languages. In 4th International Conference on Software Engineering Advances. Porto, Portugal: IEEE Computer Society, pp. 21–26.

Saraiva, J. D. S. & Silva, A. R. Da, 2009b. Development of CMS-Based Web-Applications Using a Model-Driven Approach. In 4th International Conference on Software Engineering Advances. Ieee, pp. 500–505.

Saraiva, J. D. S. & Silva, A. R. Da, 2008. The WebComfort Framework: An Extensible Platform for the Development of Web Applications. In 34th Euromicro Conference on Software Engineering and Advanced Applications, ed. 34th Euromicro Conference Software Engineering and Advanced Applications. pp. 19–26.

Saraiva, J. D. S. & Silva, A. R. Da, 2010. Web Application Modeling with the CMS-ML Language. In INForum. pp. 461–472.

Schwabe, D. & Rossi, G., 1995. The Object Oriented Hypermedia Design Model. Communications of ACM, 38, pp.45–46.

Shreves, R., 2011. Open Source CMS Market Share W. & S. White paper, ed., Bali, Indonesia.

Souer, J., Honders, P., et al., 2008. A Framework for Web Content Management System Operations and Maintenance. Journal Of Digital Information Management, 6(4).

Souer, J., Honders, P., et al., 2007. Defining operations and maintenance in web engineering: A framework for CMS-based Web applications. In 2nd International Conference on Digital Information Management. IEEE, pp. 430–435.

Souer, J., Luinenburg, L., et al., 2008. Engineering a Design Method for Web Content Management Implementations. In 10th International Conference on Information Integration and Web-based Applications & Services. pp. 351–358.

Souer, J. et al., 2011. Engineering Web Information Systems: A Content Management System-based Approach. In 13th International Conference on Information Integration and Web-based Applications and Services. pp. 5–7.

Souer, J. et al., 2009. Model-Driven Web Engineering for the Automated Configuration of Web Content Management Systems. In Springer-Verlag, ed. ICWE 2009. Heidelberg, pp. 124–135.

Souer, J., Weerd, I. Van De, et al., 2007. Situational Requirements Engineering for the Development of Content Management System-based Web Applications. International Journal of Web Engineering and Technology, 3(4), pp.420–440.

Souer, J. & Kupers, T., 2009. Towards a Pragmatic Model Driven Engineering Approach for the Development of CMS-based Web Applications. In 5th Model Driven Web Engineering Workshop. pp. 31–45.

Valverde, F. et al., 2007. A MDA-based Environment for Web Applications Development: From Conceptual Models to Code. In 6th International Workshop on Web-Oriented Software Technologies.

Vidgen, R., Goodwin, S. & Barnes, S., 2001. Web Content Management. In 14th Bled Electronic Commerce Conference. Slovenia.

Vlaanderen, K., Valverde, F. & Pastor, O., 2008. Improvement of a Web Engineering Method Applying Situational Method Engineering. In 9th International Conference on Enterprise Information Systems. pp. 147–154.

Vlaanderen, K., Valverde, F. & Pastor, O., 2009. Model-Driven Web Engineering in the CMS Domain: A Preliminary Research Applying SME. In 10th International Conference on Enterprise Information

Systems. Heidelberg, pp. 226–237.

Weerd, I. Van De et al., 2006. A Situational Implementation Method for Web-based Content Management System-applications: Method Engineering and Validation in Practice. Software Process: Improvement and Practice, 11(5), pp.521–538.