A Metamodel Proposal for a Web Development Code Generation Tool

Cristina Magaña, Zuriel Morales, José Alfonso Aguilar[*], Aníbal Zaldívar-Colado, Carolina Tripp-Barba, Eduardo Zurita and Omar Vicente García

Cuerpo Académico Señales y Sistemas, Facultad de Informática Mazatlán, Universidad Autónoma de Sinaloa, Ave. De Los Deportes S/N, Ciudad Universitaria, C.P. 82000, Mazatlán, Sinaloa, México

Abstract: Code generation is an important part of Model-Driven Development as well as the abstract representation of the software functionality. One of the strong deficiencies in this paradigm is the lack of research projects regarding to code generation that uses the terminology applied in the industry. In order to ameliorate this scenario, in this work is presented a metamodel proposal for a Web development code generation tool. This proposal is based on the terminology used in the industry for Web development in the region of Sinaloa, México. This is an extension and updated explanation of our work previously presented in a conference.

Keywords: Model-driven web engineering, Tools, Code generation, MDWE, Modeling.

1. INTRODUCTION

Model-Driven Development (MDD) is a software development paradigm which uses the key idea that software development processes must be guided by the abstract representation of its functionality. In response, the OMG (Object Management Group) proposes a set of guidelines named Model-Driven Architecture, a three-layer architecture for software development. The first one is the Computational Independent Model (CIM), needed in order to know the software requirements, commonly UML (Unified Modelling Language) models are used to specify the requirements as Use Case Diagrams. The second level is conformed by the Platform Independent Model (PIM), the theory stablishes that the requirements specified in the CIM are –transformed- in PIM models, *i.e.*, from the requirements is possible to obtain a Class Diagram leaving aside technical specifications as well implementation technology. The last level is the Platform Specific Model (PSM), the set of transformation rules are defined and executed in order to generate the source code of the software, this are Model to Text transformation rules (M2T). Going deeper, MDD requires the solution of several challenges for its implementation in the software industry, besides there are tools used in real projects such as Webratio (http:www.webratio.com), it is necessary the development of interoperable modelling tools, code generation tools and methods that indicates how to specify requirements in a model or set of models with tool support.

This ongoing work presents the definition of a metamodel for the development of a code generation tool. Metamodeling or meta-modeling is the analysis, construction and development of the frames, rules, constraints, models and theories applicable and useful for modeling a class of problems. Metamodels are of many types and have diverse applications. The advantages were detailed in our previous work presented in [17]. In this work, we updated the metamodel for code generation with SQL (Structured Query Language) language, this database management system is used in México in most of the micro and small software factories. Moreover, with regard to MDWE methods, we found that SQL is not present in the most of them according to [1], *i.e.*, Navigational Development Techniques (NDT) [3], UML-Web engineering (UWE) [4] and Web Modeling Language (WebML) [5]. The developed for support code generation on each method is updated based on our previous work as well the background section.

The paper is organized as follows: the background regarding to metamodels in MDD is presented in Section 2. In Section 3, the core of our code-generation proposal is explained. The conclusion and future work are presented in Section 4.

2. THE RELATED WORK

Since the goals [8] of MDD are cost and effort reduction, achieving profits in the project budget, among others, Web Engineering methods are now focused on adapting its approaches in to MDD

[*]Address correspondence to this author at the José Alfonso Aguilar, Facultad de Informática Mazatlán, Universidad Autónoma de Sinaloa, Ave. De los Deportes S/N, Ciudad Universitaria, C.P. 82000, Mazatlán, Sinaloa, México; Tel: +52 669-981-1560 Ext. 209; Fax: +52 669-981-1560;
E-mail: ja.aguilar@uas.edu.mx

© 2016 Cosmos Scholars Publishing House

paradigm, but this transition, until now, has not been easy. This is especially true with regard to tool support for code-generation, in particular for the user interface components by code-generation due to controls for the Rich Internet Applications (commonly named RIA's) are more complex to build that controls used in Web 1.0 applications. One solution proposed to facilitate this work f are the so called metamodels, also named Domain Specific Languages (DSL's) in the context of MDD. Next, is described the background regarding the development of DSL's for code generation and user interface modelling.

One remarkable work in the metamodel domain is Xactium on providing support for engineering domain-specific languages [6], and the work at Vanderbilt University on the Generic Modeling Environment (GME) [7]. Both approaches are based on the MOF standard of MDA and provide support for the definition of metamodels. The MOF architecture is used to build models, referred to as metamodels, that define the abstract syntax of modeling languages. While these approaches utilize MDA standards they do not necessarily restrict their modeling viewpoints to the CIM, PIM and PSM. The Xactium approach, in particular, is based on an adaptive tool framework that uses reflection to adapt a development environment as its underlying modeling language changes: If extensions are made to the modeling language the environment is made aware of it through reflection and can thus adapt. In [8], the authors propose a meta-model for abstract web applications that can be mapped to multiple platforms. We extend a UML-based model to support specific features of Web 2.0 as well as to establish a bridge to functional and usability requirements through use cases and user interface (UI) prototypes. The metamodel also helps to avoid a common MDD-related problem caused by name-based dependencies. Finally, mappings to a number of specific web platforms are presented in order to validate the appropriateness of the meta-model as an abstract web model. This is focused in requirements-usability leaving aside code generation. In [9], the authors present a metamodel for requirements specification, this metamodel is for the NDT methodology. On the other hand, in [10], the work presented by the authors is based on the usability patters applied in Model-Driven Web Engineering, they focus on the properties for for usability support.

The authors of [15] have been working with the Architecture Analysis and Design Language (AADL) modeling notation for use in real-time embedded systems, safety-critical systems, and other high-assurance systems. Their work has focused on analysis and an up-front assurance that the system will function as intended, with less emphasis on code generation. This latest MDE effort focuses more on code generation, specifically in the context of business systems with code generation benefits realized by the developer. In [16], the authors present the unified metamodel of object system which can be used for Metamodel-Driven Design (MMDD) of information system. The metamodel is realized by the author in own environment of development SharpArchitect RAD Studio. Moreover, this work describes a practical experience of the implementation of the authors of information system that automates the activities of a fast food restaurant.

There have been many attempts for the implementation of MDD for Web and Software Engineering regarding user interface development, but most of the approaches are not supported by a methodology covering MDA and its three levels to code-generation, the support tool they provides generates JAVA based applications as Webratio [11], others look for its integration with another proposals such as [12] and NDT [13], that's the reason why we believe that our work is driven for a short learning curve for software developers not familiarized with code-generation and modeling tools. According to [14], there is a big gap in the learning curve of current software developers regarding to modeling tools for software modeling.

3. CODE-GENERATION METAMODEL FOR A SMALL SOFTWARE FACTORY

The definition of a metamodel for code generation in the context of Model-Driven Web Engineering is described. The classes defined for each purpose are described in detail (see Figure **1**). The Components are defined considering six main meta-classes. The first meta-class is named *Website*, is the main class and contains all the web site elements that will be defined in the model. It has one attribute named -name- for the web site name. The second meta-class is called *Page*., this is one of the most important class since it generates the structure of a web page based on the hypertext markup language HTML (Hypertext Markup Language). The properties defined for this class are: -title-, for the page title and -accessibility level- for accessibility according to WCAG 2.0 (as a future work this function will be enabled). The class *Display Table* allows to show data from a database table. The properties are: -title-, for the table title, -table- for the data binding and -edit- and -delete- by attributes. Another important meta-class is the *Form* class; this is used to design a HTML form. It provides as properties the form -name-, the name of the button, the send -method- (POST or GET) and the table -name- on which the data from the form will be stored or obtained. The other meta-class defined is named Database, this one is for specifying the parameters for database connection such as the server name, user and password and the database name. This class allows to select among different connectors. The -*Input*- class is the main class used to select among different HTML 5 controls. It defines properties used in HTML 5 controls, these are: -id- for the control identifier, -name-, for the control name, -label, for the control label, -attribute- for the control attribute such as -String- and -field type- used to define the data type for the database binding.

Also, seven meta-classes are defined in order to specify HTML 5 controls and attributes for HTML tables, these are: -Text-, Password-, -Radio-, -Combobox-, -Checkbox-, -Option- and -Attribute-. These are basic HTML 5 controls, that's the reason why the explanation of each one is out of the scope of this work, see the reference about HTML 5 on World Wide Web Consortium for further information. The technology used for the definition of our baseline metamodel is the Eclipse Modeling Project by means of the implementation of the MOF architecture as a ECORE in Eclipse, we defined an Ecore for the meta-classes definition.

Finally, it is important to mention that the main idea behind this work is to produce a complete Model-Driven tool for semi-automatic web application generation for a local software factory under open source initiative.

A set of transformation rules were defined in order to obtain source code from a model. The transformations were designed with QVT language (Query-View-Transformations) and implemented with Acceleo, this is an eclipse project based on templates, is a pragmatic implementation of the Object Management Group (OMG) MOF Model to Text Language (MTL) standard. Next is presented an example of a model created with our metamodel (see Figure **2**).

Figure **2** shows a model representing a website with three pages. One contains a login form, the second contact information and the third page is a profile page. Applying the Acceleo template defined we generated the website source code in PHP, HTML 5 and SQL languages.

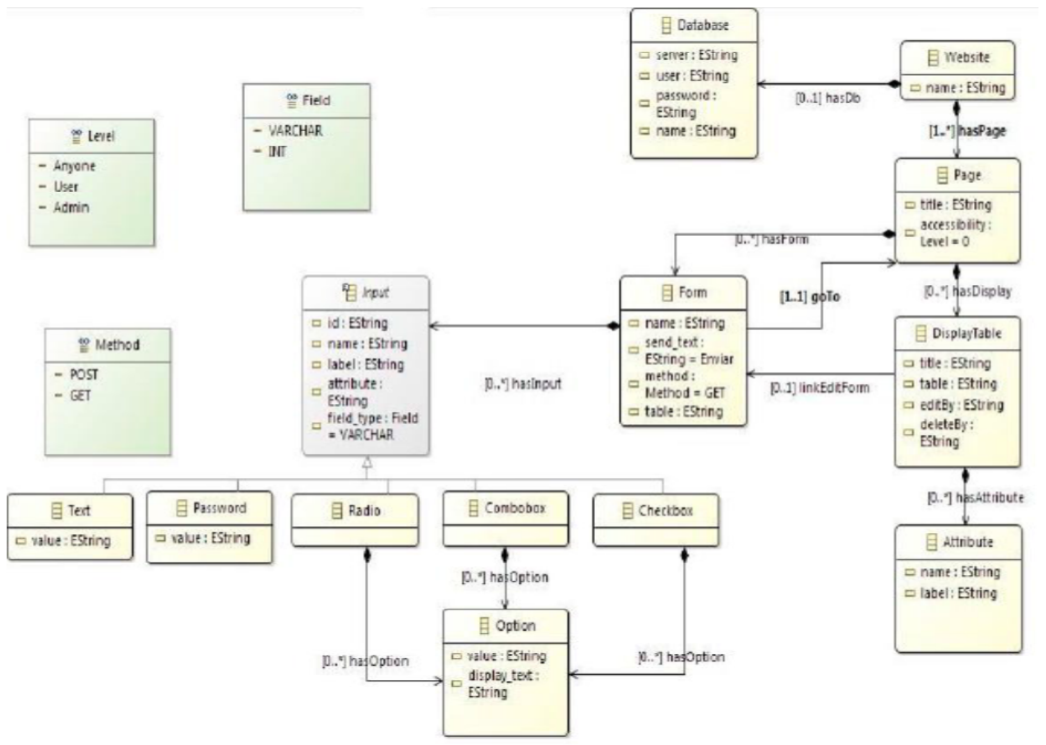
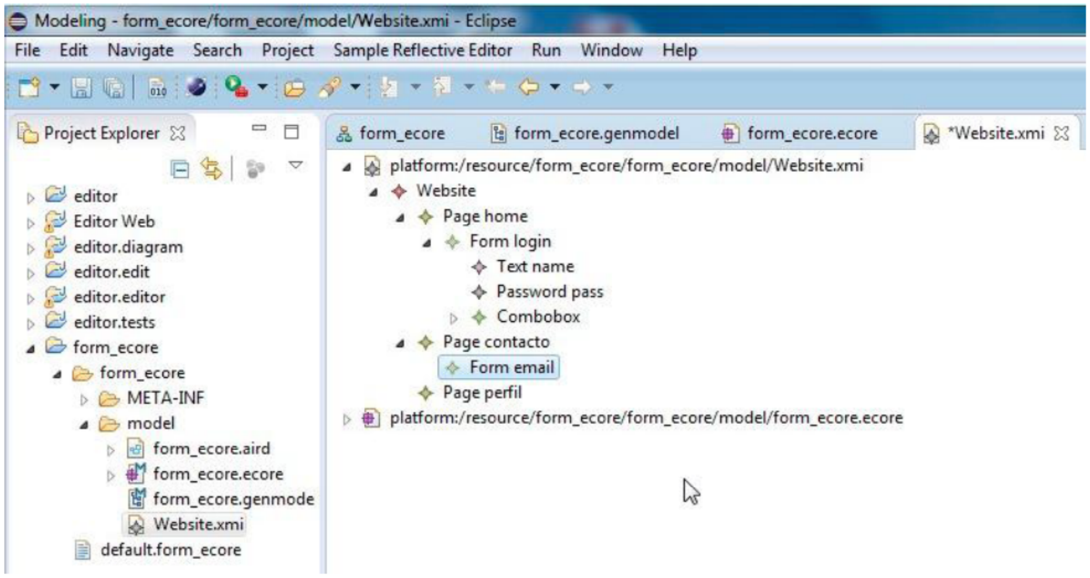


Figure 1: A baseline DSL for a code-generation tool.


Modeling - form_ecore/form_ecore/model/Website.xmi - Eclipse
File Edit Navigate Search Project Sample Reflective Editor Run Window Help
Project Explorer
form_ecore
form_ecore.genmodel
form_ecore.ecore
*Website.xmi

```
editor
Editor Web
editor.diagram
editor.edit
editor.editor
editor.tests
form_ecore
  form_ecore
    META-INF
    model
      form_ecore.aird
      form_ecore.ecore
      form_ecore.genmode
      Website.xmi
  default.form_ecore
```

```
platform:/resource/form_ecore/form_ecore/model/Website.xmi
  Website
    Page home
      Form login
        Text name
        Password pass
        Combobox
    Page contacto
      Form email
    Page perfil
platform:/resource/form_ecore/form_ecore/model/form_ecore.ecore
```

Figure 2: Modelling a website.

The proposal described in this section represents the effort made in order to bring the gap between research and its application in the software industry. The metamodel created with Eclipse Modelling Project and the transformations rules defined with Acceleo help us in or goal which is to obtain a MDD tool for code generation in the context of Web Engineering. The idea is that micro and small software factories in México decreases the time and effort expended in repetitive programming tasks.

4. CONCLUSIONS AND FUTURE WORK

The success of MDD is not depending only on resolving obvious technical issues such as defining suitable modeling languages and automatic code generation. For the experience we obtain working with small software factories with these technologies clearly indicates that solving the issues presented in our work is important. Unless the experience of applying MDD is acceptable from the day-to-day perspective of the developer, it will be rejected despite its lack of support tools for increasing of productivity and reliability. Fortunately, over the past decade, numerous commercial vendors have developed tools that address these issues successfully. Our first attempt let us move on to a real approach for practical development of Web applications since we will test its use with a real project in a small software factory as a future work.

We conclude our work with these affirmations:

- The use of common models and languages for component representation for code-generation tools is recommended due to this fact will help to the use of the same terminology for user interface modelling, because, a topic we found is the different terminology used by current Model-Driven Web Engineering methods to name their types of components for user interface modeling and code-generation. Therefore, if standard concepts are promoted in Web modelling we can establish a standard form for modeling user interface components and controls.

- The use of current Web interface techniques for modelling the behavior of the Web application is not common among the current methods for Model-Driven Web Engineering.

ACKNOWLEDGEMENTS

This work has been partially supported by: Universidad Autónoma de Sinaloa (México) by means PROFAPI 2014/002.

REFERANCES

[1] Nuseibeh B, Easterbrook S. Requirements Engineering: A Roadmap. Proceedings of the Conference on the Future of Software Engineering (ICSE). New York, USA: ACM 2000; 35-46. https://doi.org/10.1145/336512.336523

[2] Aguilar JA, Garrigós I, Mazón JN, Trujillo J. Web Engineering Approaches for Requirement Analysis-A Systematic Literature Review. Proceedings of the 6th International Conference on Web Information Systems and Technologies (WEBIST). Valencia, Spain: ScitePress Digital Library 2010; 187-190.

[3] García-García JA, Escalona MJ, Ravel E, Rossi G, Urbieta M. NDT-merge: A Future Tool for Conciliating Software Requirements in MDE Environments. Proceedings of the

14th International Conference on Information Integration and Web-based Applications & Services (IIWAS). New York, USA: ACM 2012; 177-186. https://doi.org/10.1145/2428736.2428765

[4] Koch N, Kraus A, Hennicker R. The authoring process of the UML-based Web engineering approach. Proceedings of the 1st International Workshop on Web-Oriented Software Technology (IWWOST). 2001.

[5] Brambila M, Fraternali P. Large-scale Model-Driven Engineering of Web User Interaction: The WebML and WebRatio experience. Sci of Comp Progr Elsevier 2014; 89; 71-87. https://doi.org/10.1016/j.scico.2013.03.010

[6] Xactium [homepage on the Internet]. [updated 2016 Oct 16]: Available from: http://www.xactium.com

[7] Generic Modeling Environment [homepage on the Internet]. [updated 2015 May 10]: Available from: http://www.isis.vanderbilt.edu/projects/gme/

[8] Brown A. Model-Driven Architecture: Principles and practice. Soft. and Syst Mod 2004; 3 (4): 314-327. https://doi.org/10.1007/s10270-004-0061-2

[9] Escalona MJ, Koch N. Metamodeling the Requirements of Web Systems. In: Web Information Systems and Technologies: International Conferences, WEBIST 2005 and WEBIST 2006 Revised Selected Papers. Springer Berlin Heidelberg, Berlin, Heidelberg 2007; 267-280. https://doi.org/10.1007/978-3-540-74063-6_21

[10] Insfran E, Fernandez A. A systematic review of usability evaluation in web development. In Hartmann S, Zhou X., Kirchberg M, eds: Web Information Systems Engineering, WISE 2008 Workshops. Volume 5176 of Lecture Notes in Computer Science. Springer Berlin Heidelberg 2008; 81-91. https://doi.org/10.1007/978-3-540-85200-1_10

[11] Brambilla M, Butti S, Fraternali P. Webratio BPM: a tool for designing and deploying business processes on the web. Springer 2010.

[12] Linaje M, Preciado J, Morales-Chaparro R, Rodrguez-Echeverra R, Sanchez-Figueroa F. Automatic generation of rias using rux-tool and webratio. In Web Engineering. Volume 5648 of Lecture Notes in Computer Science. Springer Berlin Heidelberg 2009; 501-504. https://doi.org/10.1007/978-3-642-02818-2_48

[13] Robles Luna E, Escalona M, Rossi G. Modelling the requirements of rich internet applications in webre. In Software and Data Technologies. Volume 170 of Communications in Computer and Information Science. Springer Berlin Heidelberg 2013; 27-41.

[14] Ceri S, Brambilla M, Fraternali P. The history of webml lessons learned from 10 years of Model-Driven development of web applications. In: Conceptual modeling: Foundations and applications. Springer 2009; 273-292. https://doi.org/10.1007/978-3-642-02463-4_15

[15] The Latest Research in Software Engineering and Cybersecurity [homepage on the Internet]. [updated 2016 November 1]: https://insights.sei.cmu.edu/sei_blog/2015/05/model-driven-engineering-automatic-code-generation-and-beyond.html

[16] Pave P. Oleynik. Metamodel-Driven Design of Database Applications. Journal of Computer Science Technology Updates 2015; 2(1): 15-24. https://doi.org/10.15379/2410-2938.2015.02.01.03

[17] Morales Z, Maga-a C, Aguilar JA, Zaldívar A, Tripp C, Misra S, Garcia V, Zurita C. A Baseline Domain Specific Language Proposal for Model-Driven Web Engineering Code Generation. In: International Conference on Computational Science and its Applications (ICCSA). Lecture Notes in Computer Science. Springer Berlin Heidelberg 2016; 50-59. https://doi.org/10.1007/978-3-319-42092-9_5

Received on 31-10-2016 Accepted on 02-12-2016 Published on 24-12-2016

http://dx.doi.org/10.15379/2410-2938.2016.03.02.03

© 2016 Magaña *et al.*; Licensee Cosmos Scholars Publishing House.
This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/), which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.