

Generation of an E-learning Application Through Model Programming



Aziz Srail, Fatima Guerouate, Hilal Drissi Lahsini

Abstract: Online learning or e-learning is a growing industry, whose beginnings date back to the 1980s and even before. The model of learning in history has always been quite simple. Until the early 2000s, education took place in a class of students with a teacher leading the class. The physical presence was not in question, and any other type of learning was, at best, questionable. Then the Internet appeared, and the rest is part of the story. On the other hand, the development of computer applications to ensure this type of education is also growing, but the problem is that each application is developed according to a given technology. Developing an e-learning application for each technology requires a lot of human resources and technical knowledge and always requires a migration of each application developed to other implementation platforms as needed. The model-driven architecture approach offers full or partial development of an e-learning application while remaining independent of the implementation platform of this application. Therefore, to solve this problem we propose a development of an e-learning application according to a model-driven architecture approach.

Keywords: Metamodel, models, MDA, e-learning, QVT.

I. INTRODUCTION

Some of the most important developments in training have occurred since the launch of the Internet. Nowadays, learners are used to using smartphones, web messaging and the Internet. So participating in an online course and leading it became easy. Chat systems, social media, and various other means of online communication allow learners to stay in touch and discuss course-related topics, while creating a sense of community. In the rapidly changing world of e-learning, the technologies available to make a course new and exciting are constantly evolving, and course content can and should be updated quickly to keep students up to date with the latest information. This is especially important if online training is provided to employees in an industry where keeping abreast

of industry developments is of the utmost importance. This is one of the reasons that many companies now offer training through e-learning, other reasons include the low costs and the ability for employees to study at their own pace and from their home.

Overall, traditional learning is expensive, time consuming, and results can vary. The importance of online learning is now a given and it can offer a much faster, cheaper and potentially better alternative. Our contribution in this paper is the generation of an e-learning application through model development.

This paper is organized as follows: we begin in the first section with an introduction. The section 2 discusses the works that are related to our theme. Section 3 presents the concepts of the MDA approach (Model Driven Approach). Sections 4 and 5 present our proposed solution to develop E-learning platform. The final section concludes this paper, and outlines future work.

II. RELATED WORKS

The authors in [Srai,2017] propose an application of the MDA approach to generate a PSM web model for E-learning platform respecting n-tiers architecture.

The author in [Bizonova,2007] proposes a framework based on OMG's Model Driven Architecture. The Platform-independent framework can classify future Learning Management System (LMS).

The goal of the work of [Wang,2003] is to develop E-learning applications through the MDA approach.

The authors in [Zhang,2010] propose a model-driven approach to develop E-learning platform with the implementation of the EJB beans. The transformation rules have not been cited.

III. MODEL DRIVEN ARCHITECTURE

The MDA approach is a component meta-model. It does not specify new middleware architecture but defines a representation of the abstract architecture and independent of the technical platform, while associating it with a multitude of business services. Fig .1, presents a description of the MDA metamodel.

The terms PIM, PSM, explained in the next section, represent models respectively independent and specific to the technical platform. The PIM, PSM and mapping techniques are based on a meta-model preferably represented with the standards defined by the OMG (UML, MOF, CWM and XMI).

Revised Manuscript Received on December 15, 2020.

* Correspondence Author

Dr. Aziz Srail*, is a research Doctor at LASTIMI Laboratory, Superior School of Technologies of Sale, Mohammadia School of engineering, Mohamed V University city of Rabat, Morocco.

Pr. Fatima Guerouate, is a Research Professor at LASTIMI Laboratory, Superior School of Technologies of Sale, Mohammadia School of engineering, Mohamed V University city of Rabat, Morocco.

Pr. Hilal Drissi Lahsini, is a Research Professor at LASTIMI Laboratory, Superior School of Technologies of Sale, Mohammadia School of engineering, Mohamed V University city of Rabat, Morocco.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Generation of an E-learning Application Through Model Programming

The goal of MDA is to create an uml representation of business logic and to associate MDA features with it. Tools should make it possible to automatically generate components according to the chosen component architecture (EJB, CORBA, .NET or WEB SERVICES). A finalization work will be necessary to refine the model obtained according to the chosen context. The migration of an application from one infrastructure to another consists of requesting, from the MDA model of the business logic, a generation of the model specific to the new target infrastructure. The automation of the generation should make it possible to reduce the duration and costs of migration. In addition, a publisher can more easily envision editing software for the technical platforms supported by the MDA.

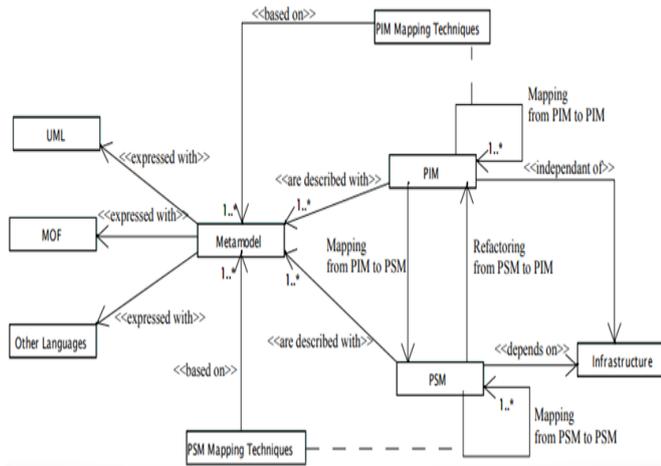


Fig. 1. MDA Meta-Model.

A. Computation Independent Model (CIM)

The CIM is the abbreviation for Computation Independent Model. The system requirements are modelled in this model which describes the situation in which the system will be used. Such a model is sometimes referred to as a Business Model or Domain Model. It does not show the details of the structure of the system. Typically this model is independent of the system implementation. The CIM corresponds to the modeling of the company not to mention the computer system.

The CIM consists of two uml models, company views and information. The CIM plays an important role in bridging the gap between experts, trade and requirements, and artifact design experts, who together meet business requirements.

B. Platform Independent Model (PIM)

The term PIM is the acronym of the Anglissism Platform Independent Model or the platform-independent model. This means that this type of model has no dependence with the technical platform. It describes the system but does not show the details of its use on the platform. The PIM represents the business logic, specific to the system, but independent of technique and technology. The PIM is the modeling of the system in a platform independent manner.

This model is concretely represented by a class diagram in uml. The PIM gives semantics to classes using stereotypes according to the MDA meta-model and uml profiles.

C. Platform Specific Model (PSM)

The PSM, for Platform Specific Model, is, on the other hand, a model dependent on the technical platform. This type

of model is used primarily as a basis for generating executable code. This model, produced from a transformation of the PIM, is a model of the same system as the PIM but specified with respect to the platform. It also describes how this system will use the chosen platform.

D. Platform Description Model

This model is referred to as the PDM for Platform Description Model. It corresponds to a transformation model from PIM to an implementation PSM. The architect must choose one or more platforms for the system implementation with the desired architectural qualities. This platform-specific model is useful for transforming PIM to PSM. The MDA approach is thus based on the detail of the models depending on the platform.

IV. PROPOSED METHODOLOGY

A. Generation of an E-learning platform from an MDA approach

To Generate an E-learning platform through an MDA approach we have considered the class diagram Fig. 2. We consider the class diagram illustrated in this figure sufficient to apply the MDA approach on E-learning platforms, and letting the paper quite understandable and clear.

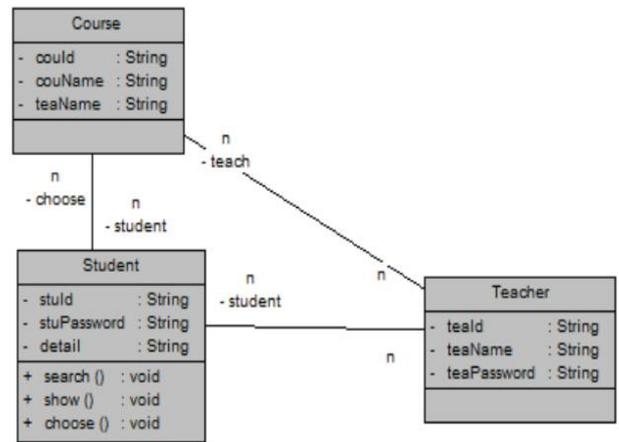


Fig. 2. Class diagram of the E-learning Application.

We begin by meta-modeling the source UML meta-model presented in the fig.3.

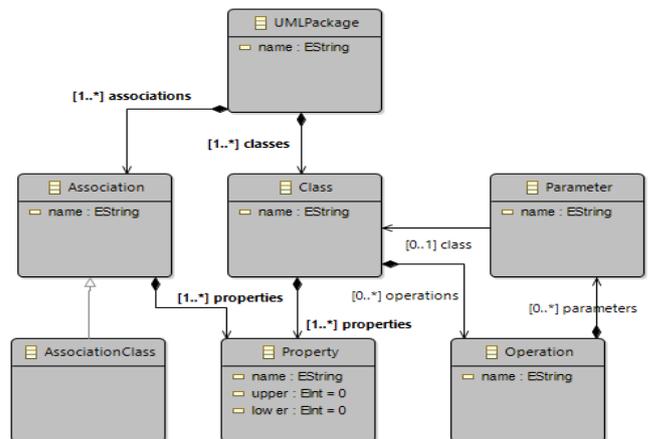


Fig. 3. UML Meta-Model.

Next we define the web service meta-model (Fig.4), EJB meta-model (Fig.5), and the Data Base meta-model (Fig.6) that allow implementation of the CIM, PIM, and PSM meta-models.

We have defined also the different transformation rules in Query View Transformation language. The different algorithms which translate these transformation rules are described in Fig.7, Fig.8, Fig 9, and Fig 10.

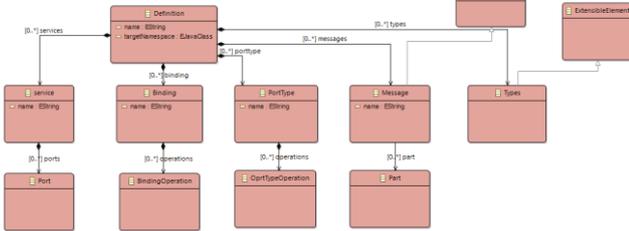


Fig. 4. Web service Meta-Model.

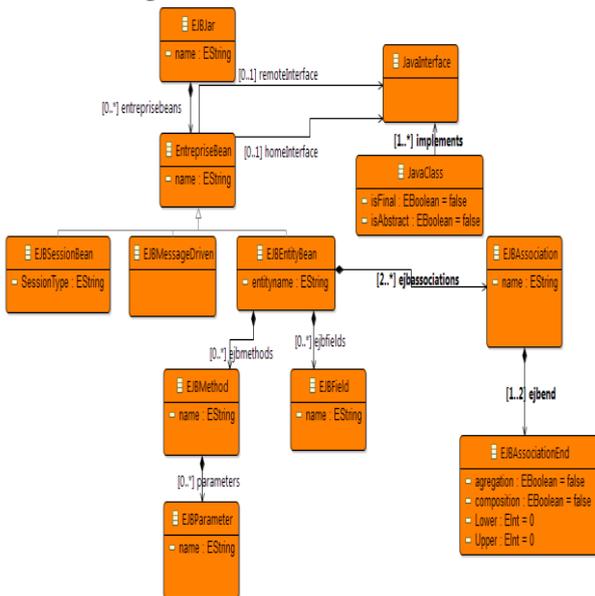


Fig. 5. EJB Meta-Model.

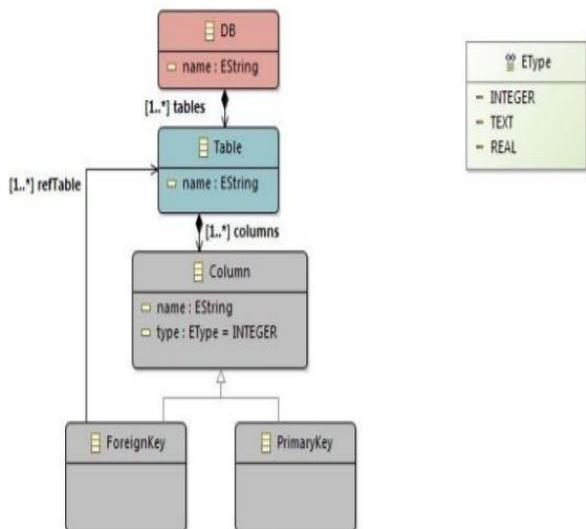


Fig. 6. Data Base Meta-Model.

```

4 transformation umlToWebService(in umlModel :umlMM, out webserviceModel: webserviceMM);
5 main() {
6 umlModel.rootObjects()[UMLPackage]-> map UMLPackageToDefinition();
7 }
8 mapping UMLPackage::UMLPackageToDefinition() : Definition {
9 result.name=': '+self.name;
10 result.services+=self.classes-> map ClassToService();
11 }
12 mapping Class::ClassToService() : service {
13 result.name=': '+self.name;
14 operations+=self.operations-> map ToServiceoperation();
15 }
16 mapping Operation::ToServiceoperation() : serviceoperation {
17 result.name=': '+self.name;
18 parameters+=self.parameters-> map ParametersToparameters();
19 }
20 mapping Parameter::ParametersToparameters() : serviceparameter {
21 result.name=': '+self.name;
22 }
23 }
    
```

Fig. 7. Algorithm represents the different transformation rules between the uml source meta-model and the web service target meta-model.

```

4 transformation EjbTransformation(in umlModel : umlMM, out ejbModel : ejbMM);
5 main() {
6 umlModel.objects()[UMLPackage]->map umlPackage2EnterpriseBean();
7 }
8 mapping UmlPackage::umlPackage2EnterpriseBean(): EJBJar {
9 result.name='EJBJar : '+self.name;
10 result.interfaces+=self.interfaces->map toJavaInterface();
11 }
12 mapping Interface::toJavaInterface(): JavaInterface {
13 result.name='javaInterface-Name : '+self.name;
14 }
    
```

Fig. 8. Algorithm represents the different transformation rules between the uml source meta-model and the EJB target meta-model.

```

4 transformation DBTransformation(in umlModel : umlMM, out dbModel : dbMM);
5 main() {
6 umlModel.objects()[UMLPackage]->map umlPackage2DB();
7 }
8 mapping UmlPackage::umlPackage2DB(): DB {
9 result.name='DB name : '+self.name;
10 result.tables+=self.classes->map toTable();
11 }
12 mapping Class::toTable(): Table {
13 result.name='Table-Name : '+self.name;
14 result.columns+=self.properties->map toColumn();
15 }
16 mapping Property::toColumn(): Column {
17 result.name='Column-Name : '+self.name;
18 }
19 }
20 }
    
```

Fig. 9. Algorithm represents the different transformation rules between the uml source meta-model and the Data Base target meta-model.



Generation of an E-learning Application Through Model Programming

```

6 main() {
7     beanModel.objects()[PackageBeans]->map beanPackage2dbPackage();
8
9 }
10 mapping PackageBeans::beanPackage2dbPackage(): DB {
11     result.name='Data-Base-name : '+self.name;
12     result.tables+=self.beans->map BeanToTable();
13 }
14 mapping Bean::BeanToTable(): Table {
15     result.name='Table-name : '+self.name;
16     result.columns+=self.files->map FiledToColumn();
17 }
18 mapping Filed::FiledToColumn(): Column {
19     result.name='Column-name : '+self.name;
20
21     if ( self = PrimitifType::int ) then {
22         result.type := EType::INTEGER;
23     }else{
24         if ( self = PrimitifType::String ) then {
25             result.type := EType::TEXT;
26         }endif;
27     }endif;
28 }endif;
29 }
30 }

```

Fig. 10. Algorithm represents the different transformation rules between the EJB source meta-model and the Data Base target meta-model.

V. RESULT ANALYSIS

All algorithms presented above are used to generate the following e-learning application (Fig. 11)

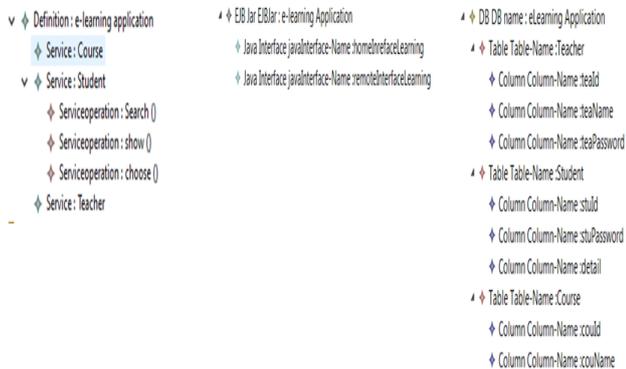


Fig. 11. Generated PSM E-learning model.

VI. CONCLUSION

In this paper, we have applied an MDA approach on an E-learning platform in order to facilitate the development of this type of application and to remain independent of the implementation technology. We can support programming by model because it saves time and costs during the entire development cycle. In perspective, we are working to develop a Model To Text transformation using the Acceleo code generator to generate the entire application.

REFERENCES

1. A. Sraï, F. Guerouate, N. Berbiche, H. Drissi, "Generated PSM Web Model for E-learning Platform Respecting n-tiers Architecture,"

2. Z. Bizonova, D. Ranc, M. Drozdova, Proceedings of the EC-TEL 2007 PROLEARN Doctoral Consortium, Crete, Greece, September 18, 2007.
3. Hong Wang, Dong Zhang, Jun Zhou, MDA-based development of e-learning system, Computer Software and Applications Conference, 2003. COMPSAC 2003. Proceedings. 27th Annual International, December 2003. DOI: 10.1109/CMPSAC.2003.1245417.
4. X. Zhang et al., A Model Driven Architecture Approach for Developing E-Learning Platform, (Eds.): Edutainment 2010, LNCS 6249, pp. 111-122, 2010.
5. Aziz Sraï, Fatima Guerouate, Hilal Drissi Lahsini, Generated Psm Multi-Layered Model Using Mda Approach, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8 Issue-4, April 2019.
6. A. Sraï, F. Guerouate, N. Berbiche, H. Drissi, "MDA Approach for EJB Model," 6th IEEE International Conference on Multimedia Computing and Systems (ICMCS'18). DOI: 10.1109/ICMCS.2018.8525924.
7. Gharavi, V., Mesbah, A., Deursen, A. V., "Modelling and Generating AJAX Applications: A Model-Driven Approach," Proceeding of the 7th International Workshop on Web-Oriented Software Technologies, New York, USA (Page: 38, Year of publication: 2008, ISBN: 978-80-227-2899-7).
8. XML Metadata Interchange (XMI), version 2.1.1, December 2007, <http://www.omg.org/spec/XMI/>.
9. Acceleo, viewed September 2014. <https://eclipse.org/acceleo/>.
10. EMF, Eclipse Modeling Framework, viewed September 2014. <http://eclipse.org/modeling/emf/>.
11. Liliana Favre, Liliana Martinez, Claudia Pereira, "Modernizing software in science and engineering: From C/C++ applications to mobile platforms," Conference: ECCOMAS Congress 2016 European Congress on Computational Methods in Applied Sciences and Engineering, DOI: 10.7712/100016.2402.4906.
12. Czarnecki, K., Helsen, S., "Classification of Model Transformation Approaches," in online proceedings of the 2nd OOPSLA'03 Workshop on Generative Techniques in the Context of MDA. Anaheim, October, 2003.
13. J.Abd-ali, Meta modélisation et transformation automatique de PSM dans une approche MDA, Maitrise en Informatique, Université du Québec en Outaouais, Canada.
14. Bezivin, J., Hammoudi, S., Lopes, D., Jouault, F., "Applying MDA approach for web service platform," In EDOC'04 preceedings of the 8th IEEE International Enterprise Distributed Object Computing Conference, pages 58-70, 2004.

AUTHORS PROFILE



Dr. Aziz Sraï, is a research Doctor at LASTIMI Laboratory, Superior School of Technologies of Sale, Mohammadia School of engineering, Mohamed V University city of Rabat, Morocco.

Pr. Fatima Guerouate, is a Research Professor at LASTIMI Laboratory, Superior School of Technologies of Sale, Mohammadia School of engineering, Mohamed V University city of Rabat, Morocco.

Pr. Hilal Drissi Lahsini, is a Research Professor at LASTIMI Laboratory, Superior School of Technologies of Sale, Mohammadia School of engineering, Mohamed V University city of Rabat, Morocco.

