

Web-Marketing 3.0 in Social Network using MDA Approach and Pulse Detector

Lamlili El Mazoui Nadori Yasser*

EHEIO School of engineering
EHEIO Laboratory
Oujda, Morocco

Esbai Redouane

Mohammed First University
MATSI Laboratory
Oujda, Morocco

Lamlili El Mazoui Nadori Mohamed

Mohammed First University
LaMSD Laboratory
Oujda, Morocco

Boutayeb Wiam

Mohammed First University
LaMSD Laboratory
Oujda, Morocco

Abstract—In this paper we have focused on the evolution of the Web-marketing meaning the passage of the Web-marketing 2.0 to 3.0, baptized " Semantic Web-marketing ", the main idea being to make easier to the Internet user to use a huge and disrupting stream of information and make it organized and easily accessible.

As so, we have chosen to use the N-tiers applications and the MDA (Model Driving Architecture) transformation. In this paper we are going to present a model-driven approach to the development of N-tiers web applications based on the UML class diagram. The transformation language is the MOF 2.0 QVT (Meta-Object Facility 2.0 Query-View-Transformation) standard which defines the meta-model for the development of model transformation.

Then we introduced a webcam-pulse-detector to detect the heart beat of the clients by an imaging solution using ambient light for catching their preferences.

Keywords: *Web-Marketing, Transformation by modeling, N-tiers architecture, Transformation rules, MOF 2.0 QVT, Meta-model, webcam-pulse-detector, ambient light.*

I. INTRODUCTION

The Webmarketing 3.0 is the way to attract the clients by different ways, as so we introduced a innovated technique called "Detection of the cardio-vascular pulse wave" and can be done by means such as variations in air pressure, impedance, or strain. Photo-plethysmography (PPG)[1], introduced in the 1930's using light reflectance or transmission and is the least expensive method and simple to use. PPG is based on the principle that blood absorbs light more than surrounding tissue so variations in blood volume affect transmission or reflectance correspondingly. Applications of PPG include monitoring of oxygen saturation (pulse oxymetry), heart (HR) and respiration (RR) rates, blood pressure, cardiac output, assessment of autonomic functions and detection of peripheral vascular diseases. Remote, non-contact pulse oxymetry and PPG imaging have been explored only relatively recently [2, 3]. In this work we are going to transform an UML Model concerning an advertisement in a social network to generate a source code.

In a recent work [21], the authors have developed a source and a target meta-models. The first was a PIM meta-model specific to class diagrams. The second was a PSM meta-model for MVC2 (Model-View-Controller) web applications (particularly Struts), then they have elaborated a transformation rules using the approach by programming. The purpose of our contribution is to produce and generate an N-tiers PSM model, implementing MVC2, DI (Dependency Injection) and DAO (Data Access Object) patterns, from the class diagram. In this case, we elaborate a number of transformation rules using the approach by modeling and MOF 2.0 QVT[4], as transformation language, to permit the generation of an XML file that can be used to produce the required code of the target application. The advantage of this approach is the bidirectional execution of transformation rules[5].

II. MODEL DRIVEN ARCHITECTURE (MDA)

In November 2000, OMG, a consortium of over 1 000 companies, initiated the MDA approach. The key principle of MDA is the use of models at different phases of application development. Specifically, MDA advocates the development of requirements models (CIM), analysis and design (PIM) and (PSM) code [6].

The MDA architecture is divided into four layers. In the first layer, we find the standard UML (Unified Modeling Language), MOF (Meta-Object Facility) and CWM (Common Warehouse Meta-model). In the second layer, we find a standard XMI (XML Metadata Interchange), which enables the dialogue between middlewares (Java, CORBA, .NET and web services). The third layer contains the services that manage events, security, directories and transactions. The last layer provides frameworks which are adaptable to different types of applications namely Finance, Telecommunications, Transport, medicine, E-commerce and Manufacture, etc.).

The major objective of MDA is to develop sustainable models, those models are independent from the technical details of platforms implementation (J2EE, DotNet, PHP

or other), in order to enable the automatic generation of all codes and applications leading to a significant gain in productivity. MDA includes the definition of several standards, including UML [8], MOF [12] and XMI [14].

III. N-TIERS ARCHITECTURE

N-tiers application architecture provides a model for developers to create a flexible and reusable application and provides some advantages that are vital to the business continuity of the enterprise. Typical features of a real life N-tiers may include the Security, Availability, Scalability, Manageability, Easy Maintenance and Data Abstraction. To most people, an N-tiers application is anything that is divided into discrete logical parts. The most common choice is a three-part breakdown presentation, business logic, and data access although other possibilities exist. In this paper, we are using the following layers:

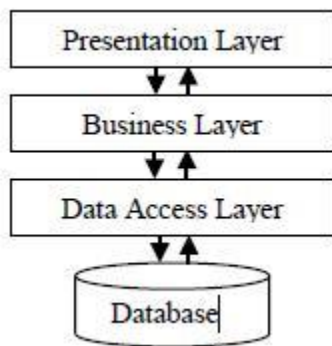


Fig.1 N-tiers Layers

Each Layer can be developed independently of the other provided that it adheres to the standards and communicates with the other layers.

A. The presentation Layer with MVC2 pattern

Along time ago, web applications were very simple and the technology that was used to develop them was Common Gateway Interface (CGI). As applications became more complex, the defects and limits of this technology have emerged. Slowness and considerable consumption of memory. Therefore, the J2EE platform applies the architecture MVC2 [7]. In this paradigm, the model represents the information system consisting of javaBeans. The view represents the HTML pages returned to the user, and consists of JavaServerPage (JSP). The Controller is the glue between the two and it is composed of servlets. In short, during the early 80's with smalltalk, MVC was widespread in the field of object development. Many frameworks that implements MVC2 pattern have emerged, among them: Struts [1], PureMVC [19], Gwittir [14], SpringMVC [15], Zend [18], ASP.NET MVC2 [5]. Struts remains the most mature solution that has earned the trust of most developers, that is why we have taken it into account in our source meta-model.

B. The Business layer with Data Transfer Object and Dependency Injection patterns

In an article written in early 2004, Martin Fowler asked what aspect of control is being inverted. He concluded that it is the acquisition of dependent objects that is being inverted. Based on that revelation, he coined a better name for inversion of control: dependency injection [19]. In other words, Dependency Injection is a worthwhile concept used within applications that we develop. Not only can it reduce coupling between components, but it also saves us from writing boilerplate factory creation code over and over again. Many frameworks that implements DI pattern have emerged, among them: Spring [15], Symphony dependency injection [17], Spring.NET [6], EJB, PicoContainer [13]. (We have used some Spring classes in our source meta-model).

IV. THE WEBCAM-PULSE-DETECTOR PROCESS

A python code that detects the heart-rate of an individual using a common webcam or network IP camera. Tested on OSX 10.7 (Lion), Ubuntu 13.04 (Ringtail), and Windows 7. Inspired by reviewing recent work on Eulerian Video Magnification, with motivation to implement something visually comparable (though not necessarily identical in formulation) to their pulse detection examples using Python and OpenCV. This goal is comparable to those of a few previous efforts in this area (such as <https://github.com/mossblaser/HeartMonitor>). This code was developed at NASA Glenn Research Center in support of OpenMDAO, under the Aeronautical Sciences Project in NASA's Fundamental Aeronautics Program, as well as the Crew State Monitoring Element of the Vehicle Systems Safety Technologies Project, in NASA's Aviation Safety Program. A list of other open-source NASA codes can be found at code.nasa.gov.

This application uses OpenCV to find the location of the user's face, then isolate the forehead region. Data is collected from this location over time to estimate the user's heart rate. This is done by measuring average optical intensity in the forehead location, in the subimage's green channel alone (a better color mixing ratio may exist, but the blue channel tends to be very noisy). Physiological data can be estimated this way thanks to the optical absorption characteristics of (oxy-). With good lighting and minimal noise due to motion, a stable heartbeat should be isolated in about 15 seconds. Other physiological waveforms (such as Mayer waves) should also be visible in the raw data stream. Once the user's heart rate has been estimated, real-time phase variation associated with this frequency is also computed. This allows for the heartbeat to be exaggerated in the post-process frame rendering, causing the highlighted forehead location to pulse in sync with the user's own heartbeat.

Support for detection on multiple simultaneous individuals in a single camera's image stream is definitely possible, but at the moment only the information from one face is extracted for analysis.

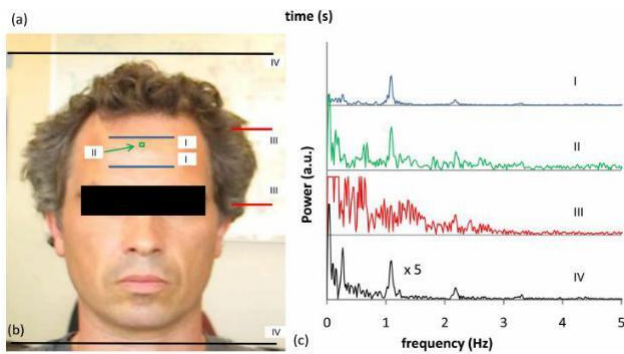


Fig 2. Signals corresponding power spectra

V. Transformation process from UML to N-tiers implementation

CRUD operations (Create, Remove, Update, and Display) are most commonly implemented in all systems. That is why we have taken into account in our transformation rules these types of transactions. We first developed EMOF models corresponding to our source and target meta-models, and then we implemented the algorithm using the transformation language QVT Operational Mappings. To validate our transformation rules, we conducted several tests. For example, we considered the class diagram (see Figure 7). After applying the transformation on the UML model, composed by the classes User and advertisement, we generated the target model.



Fig. 4 UML instance model

This Figure represents the first part of the code of the transformation of UML model source to N-tiers target model.

```

1 Transformation UML2CRUD(in umlModel:UML, out crudModel:NTierE):
2
3 MAIN() {
4   umlModel.objects().[UmlPackage]->map UmlPackage2CrudProjectPackage():
5   }
6
7   MAPPER UmlPackage2UmlPackage2CrudProjectPackage ( : CrudProjectPackage {
8     name = 'crud-rs3f.name'
9     dPack = object DaoPackage {
10      name = 'daoPackage'
11      pojo = umlModel.objects().[Class]->map class2Pojo()
12      dao = umlModel.objects().[Class]->map class2Dao()
13      daoImpl = umlModel.objects().[Class]->map class2DaoImpl()
14    }
15
16    bPack = object BusinessPackage {
17      name = 'businessPackage'
18      dto = crudModel.objects().[Pojo]->map pojo2DTO()
19      services = umlModel.objects().[Class]->map class2Service()
20      serviceImpl = umlModel.objects().[Class]->map class2ServiceImpl()
21    }
22
23    vPack = object ViewPackage {
24      name = 'viewPackage'
25      view = umlModel.objects().[Class]->map class2View()
26    }
27
28    cPack = object ControllerPackage {
29      name = 'controllerPackage'
30      ctrl = crudModel.objects().[Controller]
31      ctrlMapping = object ActionMapping {
32        name = 'actionMapping'
33        actions = umlModel.objects().[Class]->map class2Action()
34      }
35    }
36  }
37
38 }
    
```

Fig 3. A transformation code UML2CRUD

The entry point of the transformation is the method 'main'. This method makes the correspondence between all the elements of the UMLPackage type of the input model and the element of the CrudProjectPackage type of the output model. The objective of the second part of this code is to transform a UML package into N-tiers package, by creating the elements of type package 'Dao', 'Business' and 'Presentation'. It is a question of transforming each class of package UML to Jsp page and Action in the View package, to DTO, IService and ServiceImpl in the Business package, and to Pojo, IDao and DaoImpl in the Dao package, without forgetting to give names to the different packages.

B.Result:

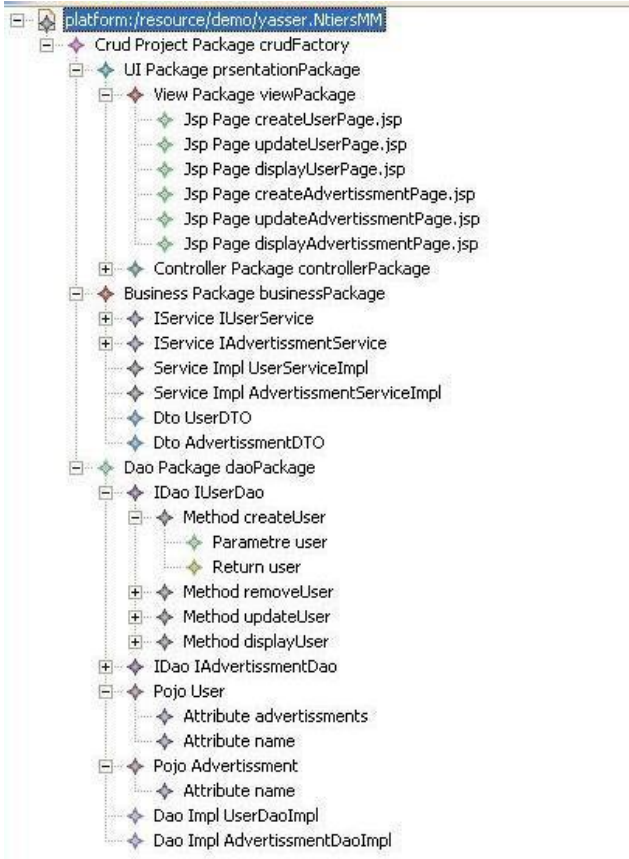


Fig. 5 Generated PSM N-tiers Web model

The first element in the generated PSM model is UIPackage which includes viewPackage that contains the JSPs, namely DisplayUserPage.jsp, Display-AdvertissmentPage.jsp, CreateUserPage.jsp, CreateAdvertissmentPage.jsp, UpdateUserPage.jsp, and UpdateAdvertissmentPage.jsp. Since the operation of the removal requires any form, we'll go to the controllerPackage element, which contains a single element ActionMapping. The latter contains eighteen delegating action proxy whose names are respectively DisplayXAction, CreateXAction, UpdateXAction, RemoveXAction, CreateXEndAction, UpdateXEndAction, where X should be replaced by User, and Advertissment. Operations for creation and update, add forms to enter new values. For this reason, we add CreateXEndAction and UpdateXEndAction.

The second element in the generated PSM model is businessPackage which includes three services' interfaces, three services' implementations and three Dtos' objects correspond to the two objects 'User' and 'Advertissment'. The last element in the generated PSM model is DaoPackage which contains three Pojos' objects that contains their attributes, three Daos' interfaces that contains methods with their parameters and their implementations.



Fig. 6 Sample of the generated Web site

VI. Conclusion

In this paper, we applied the MDA approach to generate the N-tiers web application based on UML class diagram to generate a skeleton of a social network and create appropriate advertisements to the users in function of them profiles.

This involves developing all meta-classes needed to be able to generate an N-tiers application respecting a MVC2, DI and DAO patterns, then we applied the approach by modeling and used the MOF 2.0 QVT standard as a transformation language. The transformation rules defined allow browsing the source model instance class diagram, and generating, through these rules, an XML file containing layers of N-tiers architecture according to our target model. This file can be used to produce the necessary code of the target application. The algorithm of transformation manages all CRUD operations. Moreover, it can be re-used with any kind of methods represented in the UML class diagram. In the future, this work should be extended to allow the generation of other components of Web application besides the configuration files. For instance, we will be able to provide part of user interface. Afterward we can consider integrating other execution platforms like PHP and DotNET.

REFERENCES

- [1] A. Jonsson, "Pressure Sore Etiology - Highlighted with Optical Measurements of the Blood Flow, Chapter 3, New sensor design made to discriminate between tissue blood flow at different tissue depths at the sacral area," PhD thesis (Mälardalen University Press, 2006).
- [2] J. Allen, "Photoplethysmography and its application in clinical physiological measurement," *Physiol. Meas.* 28, R1-R39 (2007).
- [3] L. G. Lindberg and P. A. Oberg, "Photoplethysmography II. Influence of light-source wavelength," *Med. Biol. Eng. Comput.* 29, 48-54 (1991).
- [4] AndroMDA. <http://www.andromda.org/>.
- [5] ASP.NET MVC site <http://www.asp.net/mvc/>

- [6] Blanc, X., MDA en action : Ingénierie logicielle guidée par les modèles (Eyrolles, 2005).
- [7] Bezivin, J., Busse, S., Leicher, A., Suss, J.G, Platform Independent Model Transformation Based on TRIPLE. In Middleware'04: Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware, pages 493- 511,2004.
- [8] Bezivin, J., Hammoudi, S., Lopes, D., Jouault, F., Applying MDA approach for web service platform. In EDOC'04 proceedings of the 8th IEEE International Enterprise Distributed Object Computing Conference, pages 58-70, 2004.
- [9] 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.
- [10] Cong, X., Zhang, H., Zhou, D., Lu, P., Qin, L., A Model- Driven Architecture Approach for Developing E-Learning Platform , Entertainment for Education. Digital Techniques and Systems Lecture Notes in Computer Science, Volume 6249/2010, 111-122, DOI: 10.1007/978-3-642-14533-9_12, 2010.
- [11] Distante, D., Rossi, G., Canfora, G., Modeling Business Processes in Web Applications: An Analysis Framework. In Proceedings of the The 22nd Annual ACM Symposium on Applied Computing (Page: 1677, Year of publication: 2007, ISBN: 1-59593-480-4).
- [12] Eclipse.org. ATLAS Transformation Language (ATL). <http://www.eclipse.org/m2m/at/>.
- [13] Gharavi, V., Mesbah, A., Deursen, A. V., Modelling and Generating AJAX Applications: A Model-Driven Approach. Proceeding of the7th International Workshop on Web- Oriented Software Technologies, New York, USA (Page: 38, Year of publication: 2008, ISBN: 978-80-227-2899-7)
- [14] Gwittir Source Web Site <http://code.google.com/p/gwittir/>
- [15] Hibernate Framework (<http://www.hibernate.org/>)
- [16] Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I., ATL: A model transformation tool. Science of Computer Programming-Elsevier Vol. 72, n. 1-2: pp. 31-39, 2008.
- [17] Koch, N., Transformations Techniques in the Model-Driven Development Process of UWE, Proceeding of the 2 International Workshop Model-Driven Web Engineering, Palo Alto (Page: 3 Year
- [18] Kraus, A., Knapp, A., Koch N., Model-Driven Generation of Web Applications in UWE. Proceeding of the 3rd International Workshop on Model-Driven Web Engineering, CEUR-WS, Vol. 261, 2007
- [19] Fowler, M., Inversion of Control Containers and the Dependency Injection pattern (<http://martinfowler.com/articles/injection.html>)
- [20] Mbarki, S., Erramdani, M., Toward automatic generation of mvc2 web applications, InfoComp - Journal of Computer Science, Vol.7 n.4, pp. 84-91, December 2008, ISSN: 1807- 4545.
- [21] Mbarki, S., Erramdani, M., Model-Driven Transformations: From Analysis to MVC 2 Web Model, International Review on Computers and Software (I.RE.CO.S.), Vol. 4. n. 5, pp. 612-620, September 2009.
- [22] Mbarki, S., Rahmouni, M., Erramdani, M., Transformation ATL pour la génération de modèles Web MVC 2, 10e Colloque Africain sur la Recherche en Informatique et en Mathématiques Appliquées, Theme5:Information Systems, CARI 2010.