

A Model Driven Approach based on Interaction Flow Modeling Language to Generate Rich Internet Applications

Sarra Roubi¹, Mohammed Erramdani², Samir Mbarki³

^{1,2}MATSI Laboratory, Mohammed First University, Oujda, Morocco

³Departement of Computer Science, Ibn Tofail University, Morocco

Article Info

Article history:

Received Mar 20, 2016

Revised Aug 13, 2016

Accepted Aug 27, 2016

Keyword:

Code generation

Graphical user interface

Interaction flow modeling

language

Meta model

Model driven engineering

ABSTRACT

A Rich Internet Applications (RIAs) combine the simplicity of the hypertext paradigm with the flexibility of desktop interfaces. These applications were proposed as a solution to follow the rapid growth and evolution of the Graphical User Interfaces. However, RIAs are complex applications and their development requires designing and implementation which are time-consuming and the available tools are specialized in manual design. In this paper, we present a model driven approach to generate GUI for Rich Internet Application. The approach exploits the new language IFML recently adopted by the Object Management Group. We used frameworks and technologies known to Model-Driven Engineering, such as Eclipse Modeling Framework (EMF) for Meta-modeling, Query View Transformation (QVT) for model transformations and Acceleo for code generation. The approach allows to quickly and efficiently generating a RIA focusing on the graphical aspect of the application.

Copyright © 2016 Institute of Advanced Engineering and Science.

All rights reserved.

Corresponding Author:

Sarra Roubi,
MATSI Laboratory,
Mohammed First University,
BP 473 Complexe universitaire Al Qods,
Oujda-60000, Morocco.
Email: s.roubi@ump.ac.ma

1. INTRODUCTION

In recent years we have noticed a rapid growth of Web applications with sophisticated behavior of the user interface. These applications, called Rich Internet Applications (RIAs), brought the richness of desktop application interfaces to the world of Web browsing. Many frameworks and technologies have emerged to facilitate the construction of this web software, such as Ajax or JavaFX. However, RIAs remain complex applications and their development requires designing and implementation which are time-consuming. Also, we can say that the available tools are specialized in manual design.

In return, the Model Driven Development has shown its contributions and efficiency to the field of engineering and helps improving the quality of applications as well as time savings and increasing the productivity. That is why we can say that modeling approach represents a way to master complexity and ensure consistency of applications.

In this paper, we propose a Model-Driven Approach for RIAs development based on the OMG standard language called Interaction Flow Modeling Language (IFML). The proposed method automates the development process and hides complexe technological specifications.

Indeed, we propose a model-based approach that takes as input a simplified model, respecting the IFML, which will be transformed into an instance of the proposed meta model for MVP RIAs. This resulting model is the input for the code generator developed to generate the views of RIAs while taking into account the events and user interactions besides their connection with the two layers of the MVP pattern.

The paper is organized as follows: Section 2 summarizes the Model Driven Engineering approach and introduces the IFML. Section 3 and 4 present respectively the work elaborated and the running example; while Section 5 reviews the related work while Section 3. Finally, Section 6 concludes.

2. MODEL DRIVEN ENGINEERING

2.1. Meta Modeling and Transformation Process

MDE introduces an approach related to system specification which relies on the separation into three different layers of abstraction [1]:

- Computing Independent Model (CIM): It shows exactly what the system is supposed to do, but hides all the technology specifications.
- Platform Independent Model (PIM): It exhibits a sufficient degree of independence so as to enable its mapping to one or more platforms.
- Platform Specification Model (PSM): It combines the specifications in the PIM with the details for a particular type of platform which leads to include platform specific details.

The transition between the CIM, PIM and PSM models and finally the Code is possible through the execution of models transformation. A transformation converts models with a particular perspective from one level of abstraction to another, usually from a more abstract to less abstract view, by adding more detail supplied by the transformation rules. Transformations can be:

- Model to Model: it concerns the transition from CIM to PIM or from PIM to PSM.
- Model to Text: it concerns the generation of the code from the entry model (the PSM) to a specific programming language as a target.

For this work, we used the QVT-Operational mappings language [2] implemented by Eclipse Modeling Framework [3] for the Model To Model transformation and the Acceleo standard for code generation. The models in the Model Driven Approach can be read in a textual format, which is the XMI standard as defined by the OMG [4].

2.2. Interaction Flow Modeling Language

The Interaction Flow Modeling Language (IFML) is designed for expressing the content, user interaction and control behavior of the front-end of software applications. In other words, it supports the platform independent description of graphical user interfaces for software applications that can be accessed or deployed on various systems as desktop computers, laptop computers, PDAs, mobile phones, and tablets. Besides, an IFML model should supports several design perspectives, among them:

The view structure specification, which consists on the definition of view containers, their nesting relationships, their visibility. The view content specification that treats the content and data entry elements contained within view containers. The events specification that can be produced by the user interaction, by the application, or by an external system and may affect the state of the user interface. The parameter binding specification, which consists on the definition of the input-output dependencies between view components and between view components and actions.

Figure 1 shows a simple example of IFML model where the user can search for a product by entering some criteria in the Product Search Form. The matching items are shown in a list. The selection of one item causes a delete action to be triggered on it. When the deletion is completed, the updated list of products is displayed again. IFML concepts can be stereotyped to describe more precise behaviors [5].

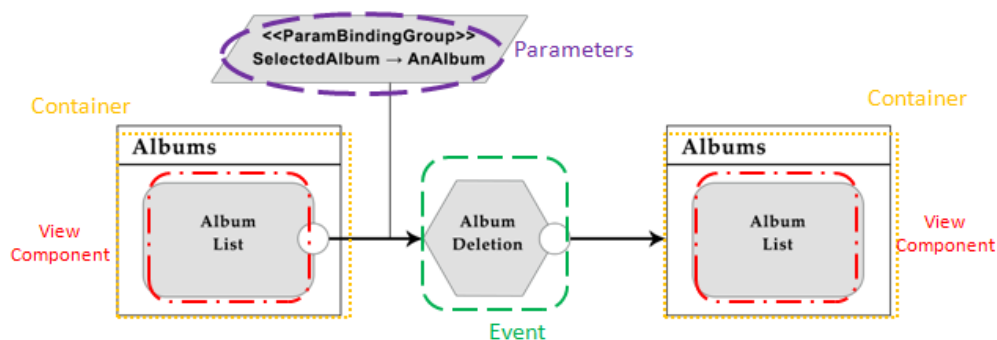


Figure 1. IFML Example: Product Search, Listing and Deletion

3. THE PROPOSED MODEL DRIVEN APPROACH

In this paper, we propose an approach adopting Model Driven Engineering technologies order to enable the automatic generation of a MVP Rich Internet Application. The entry point of the automated process is a simplified designed model using the new OMG standard; IFML. Indeed, we simplified the IFML and we kept only items that describe sufficiently the application and removed all details that can weigh the input model creation.

When we adopt a model driven development approach, a variety of technologies and frameworks are available. These tools and technologies can be effectively used to achieve the final goal which is the automatic generation of code. The whole process we adopted is described in Figure 2.

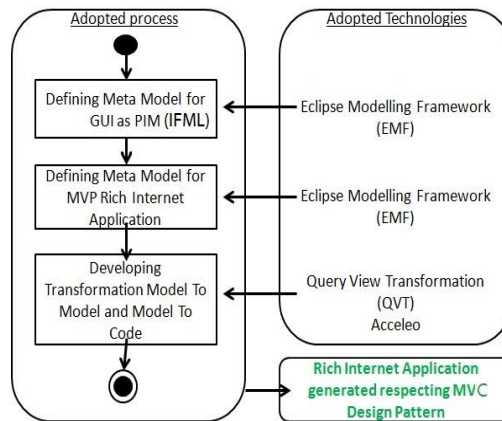


Figure 2. Process and Technologies Adopted in the Proposed Approach

3.1. IFML for RIA

As mentioned, IFML was adopted to define the input instances for our approach. Indeed, when we used IFML we had the elements to define an input model that describes the graphical interface of RIA on one hand, and the interaction of events, on the other. Also, to use a model driven approach, we adopted several standards for the proposed approach. Indeed, we defined the model instances of the IFML using EMF; the Eclipse framework. Also, for the transformations, we used the OMG standards; the QVT for the Model To Model transformation and Aceleo for the Model to Text transformation. The whole process allows having an almost complete RIA application respecting the MVP Design Pattern.

For the case study we present later, we generate a contacts' address application. So, the IFML language allows us to define the graphical part with containers and their components. Also, for each element of interaction, we define the navigation flow in terms of events and resulting views. Figure 3 shows the input model, instance of the IFML meta model.

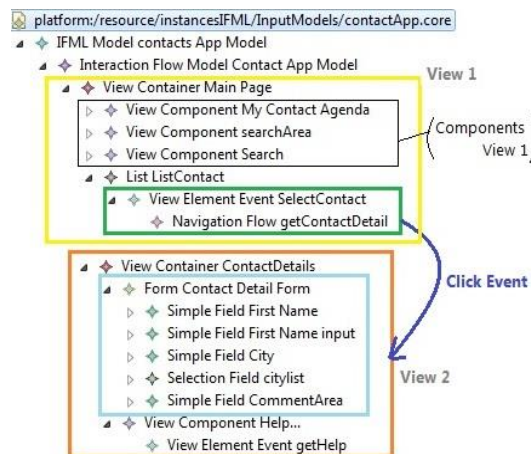


Figure 3. Input Example Instance of the IFML

The input model does not describe only the graphical elements needed to define the graphical part of the interface, for example, the form and all its fields, but also defines the flow of user interactions in a very simplified manner that does not require knowledge of the target technology. This model will be the input for our automated process and is first transform to a model respecting the MVP meta model defined for RIA.

3.2. Proposed Meta Model for MVP RIA

We choose the RIA as a target platform and precisely the JavaFX technology, so we defined the PSM meta model adopting the MVP as a core architectural pattern. As shown in Figure 4, we have the three packages:

- View Package; containing meta-classes to represent Views and the graphical components.
- Model Package; representing the domain or the business layer of the application.
- Presenter Package; containing presenters to ensure the connection between the tow layers.

In our meta model, the View/Presenter layers are responsible for describing the structure and content of views in terms of behavioral elements while the navigation flow is ensure through the presenter's handler that are connected to the specified services from the model layer. Further more, the view gathers the element describing the graphical interface; we find the main element as the scene which is composed of graphical components, named controls. The controls can be simple grahical components or containers as Roots; this relationship is defined in the meta model using the composite pattern.

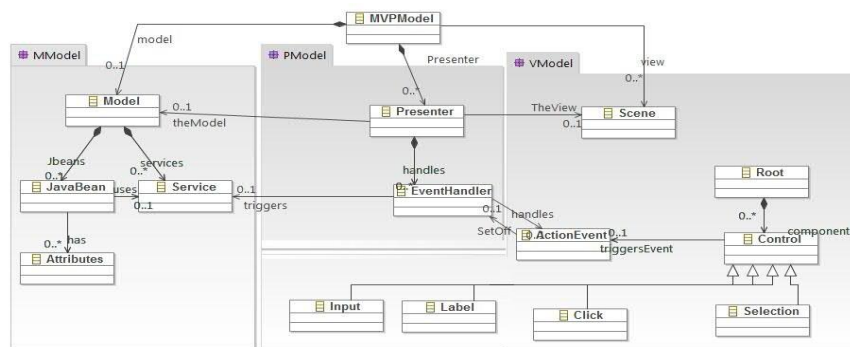


Figure 4. Proposed Meta Model for MVP RIA Simplified

3.3. Transformation process

3.3.1. Model to Model

Once the Modeling phase established, we define the transformation rules. The idea is to connect each element of the input model to its equivalent in the output model, according to the conditions and rules for transmitting all necessary information. The entry point of the transformation is the main method. This method makes the correspondence between all elements of the IFML model of the input model and the elements of type JavaFXPackage output model. For instance, for each interactionFlowModel we create the equivalent ViewPakage that will gather all the graphical aspect of the RIA. To that, we connect the Presenter and the Model which gather the services and method for the application.

3.3.2. Model to Text

The first transformation generates the model file for RIA respecting the MVP design pattern; the next step is to generate the whole code of the application. To do this, there are several techniques; using scripting languages, XML parsing or model transformation language. Since we adopted a model driven approach, we opted for a model to text transformation. So, the M2T focuses on generating textual artifacts from model elements. We used Aceleo that is implementation of the Object Management Group (OMG) MOF Model to Text Language (MTL) standard. The operating principal is to use templates to generate code from a model depending on the chosen platform. The realization of a new generation module therefore requires the creation of templates. So, we defined templates for the JavaFX platform to automatically transform models obtained in the first transformation phase.

The execution of these templates we developed gives the source code of the application with Java and XML files for the views, the presenters and the models. With these generated files we are able to create

an MVP JavaFX project that give us the graphical interface with all the components as desired and also all the connections with the applications three layers.

4. RUNNING EXAMPLE

In order to validate our approach, we applied it to generate a Contact application for searching and editing contacts information. The application enables its users to: View the contacts in a list, select a contact and edit its information. We provide the design model, instance of IFML, as described above, which is basically the only input file to our generator. We defined the two views that we want for our application with IFML. Also, we find the model elements that will be used in the processing of information in the application. Figure 3 shows the input model. Once the application has been sufficiently modeled using the IFML, the file is used as an input for the transformation engine developed for the approach. Indeed, we will first generate a MVP model for the contact Application that respects the PSM defined meta model. Then, it will become the input for the code generation part. The resulting file is a model respecting the Meta Model for RIA implementing the MVP Design Pattern, Figure 5.

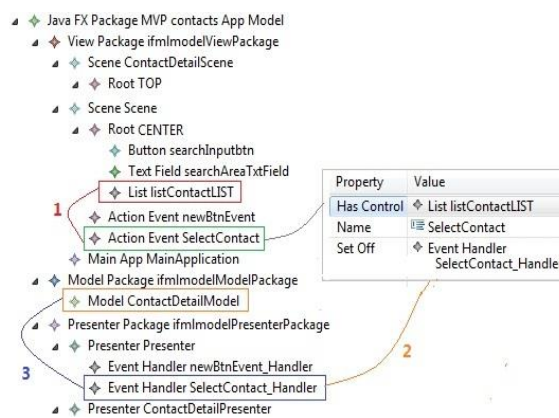


Figure 5. The Generated File for MVP RIA

This file will be considered as the input for the code generator, and will provide all the source code files needed for the application to be almost running after being loaded into Eclipse IDE, see Figure 6. The graphical aspect of the application and the user interaction are the major focus on this approach without neglecting the Model and Presenter layers.

By using this approach, we reduced significantly the development time compared to a classic method of programming. Compared to a classic approach that requires developing manually all the views, the model and the controller, this method generates the final files code automatically. Besides, if the user wants to add, remove or update graphical information for the application, the model is updated with the changes and the whole code is re generated with no supplementary effort needed.

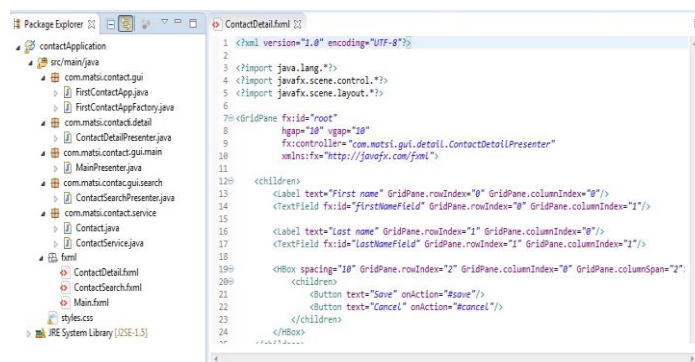


Figure 6. Code Generated as Loaded in Eclipse

5. RELATED WORK

On the one hand, Along with the evolution of the Model Driven approach, several works dealing with computer science problems used this approach. Among them, DSRM [6] is an ontology Driven Domain Specific Data retrieval Model that allows describing user query and queried data. Also, regarding the importance of graphical user interface, several works focused on performances for specific domain, like for [7] that presents a developed graphical interface for power system stability. On the other hand, many works dealing with GUIs automatic generation have emerged and tried to combine the tow communities. In [8], the methodology proposes a model based approach and advocates in favor of a complete separation of the web application functionality from the context adaptation at all development phases.

Also, a Rich Internet Application for web based product development was presented in [9]. Besides, an approach called OOH4RIA [10] proposes a model driven development process that extends OOH methodology for GWT applications. Also, a combination of the UML based Web Engineering (UWE) method for data and business logic modeling with the RUX-Method for the user interface modeling of RIAs was proposed as model-driven approach to RIA development [11]. In [12], the research was based on the plasticity of User Interface and the application of MDA concepts for the purpose of unifying the modeling for GUI. Another related work for designing Graphical User Interfaces of RIAs and uses iterative series of XSLT transformations based on XML User Interface description languages using XSLT as the transformation language between the different levels of abstraction [13].

In addition to that, and specifically for generating web application, a MDA approach for AJAX web applications [14] was the subject of a study that proposes a UML scheme by using the profiling for modeling AJAX user interfaces, and report on the intended approach of adopting ANDROMDA for creating an AJAX cartridge to generate the corresponding AJAX application code, in ICEFACES, with back-end integration. A meta model of AJAX has been defined using the AndroMDA tool. In addition to that, an extension of the IFML was proposed for generating mobile application front end [5].

In our work, we propose an approach based on the IFML standard applied to RIA and focusing on the graphical components on the one hand and interactions with the whole application's layers on the other, which gives us a complete generated RIA. In comparison to the approaches proposed for RIA, as it is the case for OOH4RIA [11], the user should be aware of the technology specification on the one hand. On the other hand, if the user wants to specify the flow process of the application, he should add a new model for navigation. With the IFML and the proposed approach, a single model can contain all the information to conceptualize the flow process along with the graphical aspect of the RIA. Also, another important aspect is the application of the OMG standards. Indeed our approach proposes a set of model-to-model and model-to-text transformations and allows having a traceable process; however the RUX method [11] uses code generators to obtain the final implementation. In addition to that, the OOHDm [15] does not follow any OMG standard, also RUX only uses UML for WebML models, whereas in our approach we use several standards from the OMG; MOF, QVT, OCL and Acceleo languages.

6. CONCLUSION

In this paper we presented a new model driven approach for generating RIA based on the new OMG standard IFML and a new MVP proposed meta model. To establish this approach, we first study and simplified the IFML to represent our input models that describes the GUI and the interactions also. Second, we defined a meta model for RIA, taking as a implementation technology the JavaFX. This meta model respects the Model View Presenter design Pattern and uses also the composite pattern for the graphical components modeling. After that, the transformation engine was developed for both Model to Model and Model to Text transformations. As result, we were able to generate a complete RIA respecting MVP pattern but focusing on the graphical aspect of it on one hand, and the user interactions on the other.

The major contribution is he use of a simplified version of the IFML as input besides the abstraction of all technical details that a RIA development requires. Also, the time can be significantly reduced using this approach. Future works will cover the implementation of more refined code generator and the application of the proposed method to estimate how this approach scales in large projects. Also, we aim at enrich the IFML and target several platform for mobile, desktop and Web application starting from the same input model.

REFERENCES

- [1] J. Miller, *et al.*, "MDA Guide Version 1.0.1," OMG, 2003.
- [2] Meta Object Facility (MOF), "2.0 Query/View/Transformation (QVT), Version 1.1," OMG, 2009.
- [3] The Eclipse Modeling Project, <http://www.eclipse.org/modeling>.

- [4] Object Management Group (OMG), "XML Metadata Interchange (XMI)," *MOF 2.0/XMI Mapping, version 2.1.1*, 2007. <http://www.omg.org/docs/formal/07-12-02.pdf>.
- [5] M. Brambilla, *et al.*, "Extending the Interaction Flow Modeling Language (IFML) for Model Driven Development of Mobile Applications Front End," 2014.
- [6] J. Li and Q. Cao, "DSRM: An Ontology Driven Domain Scientific Data Retrieval Model," *TELKOMNIKA Indones. J. Electr. Eng.*, vol/issue: 12(2), pp. 1462–1470, 2014.
- [7] G. D. Eddine, *et al.*, "A Developed Graphical User Interface for Power System Stability and Robustness Studies," *TELKOMNIKA Indones. J. Electr. Eng.*, vol/issue: 15(3), pp. 458–467, 2015.
- [8] G. M. Kapitsaki, *et al.*, "Model-driven development of composite contextaware web applications," *Inf. Softw. Technol.*, vol/issue: 51(8), pp. 1244-1260, 2009.
- [9] Z. Ahmed and V. Popov, "Integration of Flexible Web Based GUI in I-SOAS," 2010.
- [10] S. Meli, *et al.*, "A model-driven development for GWT-based rich internet applications with OOH4RIA," *Proc. - 8th Int. Conf. Web Eng. ICWE 2008*, pp. 13-23, 2008.
- [11] J. C. Preciado, *et al.*, "Designing rich internet applications combining UWE and RUX-method," *Proc. - 8th Int. Conf. Web Eng. ICWE 2008*, pp. 148-154, 2008.
- [12] J. S. Sottet, *et al.*, "A model-driven engineering approach for the usability of plastic user interfaces," *Engineering Interactive Systems, Springer Berlin Heidelberg*, pp. 140-157, 2008.
- [13] F. J. Mart'inez-Ruiz, *et al.*, "A first draft of a model-driven method for designing graphical user interfaces of Rich Internet Applications," in *LA-Web 06: Proceedings of the 4th Latin American Web Congress, IEEE Computer Society*, pp. 32-38, 2006.
- [14] V. Gharavi, *et al.*, "Modelling and Generating AJAX Applications: A Model-Driven Approach," *Proceeding of the 7th International Workshop on Web- Oriented Software Technologies, New York, USA*, pp. 38, 2008. ISBN: 978-80-227-2899-7.
- [15] M. Urbietta, *et al.*, "Designing the Interface of Rich Internet Applications," in *Proceedings of the 2007 Latin American Web Conference, 2007*.

BIOGRAPHIES OF AUTHORS



Sarra Roubi is a PhD student with the MATSI Laboratory, Mohammed First University, Morocco. She got a degree of engineer in Computer Science from the National School of Applied Science. She is focusing her researches on the Model Driven based approach applied to the automatic generation of applications from simplified models and focusing on the Graphical User Interface. E-mail: s.roubi@ump.ac.ma



Mohammed Erramdani is with the MATSI Laboratory, Oujda Institute of Technology, Morocco. He is a professor in the Department of Management at the Institute of Technology and teaches the concept of Information System. He got his thesis of national doctorate in 2001. His activities of research in the MATSI Laboratory (Applied Mathematics, Signal Processing and Computer Science) focusing on MDA (Model Driven Architecture) integrating new technologies XML, EJB, MVC, Web Services, etc. E-mail: m.erramdani@gmail.com



Samir Mbarki is with the Computer Science Department of Ibn Tofail University, Kenitra, Morocco.

e-mail: samirmbarki@hotmail.com

He is a professor in the Department of Computer Science at Faculty of Science Ibn Tofail University. His research interests include software engineering, model driven architecture, software metrics and software tests. He obtained an HDR in computer science from Ibn Tofail University in 2010.