Enhancing Interaction Flow Modeling Language Metamodels for Designing Features of Rich Internet Applications

Karzan Wakil¹²³*, Dayang N. A. Jawawi¹, Haris Rachmat⁴

¹Department of Software Engineering, School of Computing, Faculty of Engineering, University Technology Malaysia, Skudai 81300, Malaysia.
²University of Human Development, Iraq.
³Sulaimani Polytechnic University, Iraq.
⁴School of Industrial Engineering, Telkom University, 40257 Bandung, West Java, Indonesia.

Received 28 June 2018; accepted 5 August 2018, available online 24 August 2018

Abstract: Rich Internet Applications (RIAs) became standard of interactive web applications on the internet fastly. It is a complex application with a rich user interface that distributed the data between client and server also allowing an asynchronous communication between them, but web engineering methods are not able to design and implement these features impeccably. The recent web engineering method is Interaction Flow Modeling Language (IFML) which adopted by Object Management Group (OMG). It has many features for developing interactions in web application compared with other web engineering methods but also has limitation for designing RIA features. In this paper, we enhance IFML method through extension the metamodels by using UML extension mechanism, in which, we define new IFML metamodel and some new elements to cope RIA features designing in data distribution between client and server. The results show that this enhancement enables IFML to develop the new types of web applications efficiently.

Keywords: Renewable fuel, ethanol, numerical simulation, auto ignition, single-step mechanism

1. Introduction

Rich Internet Applications (RIAs) is a new type of a web application that offers more efficient graphical features and promotes the fusion of traditional applications and the client-server application. In addition, provided furnish convivial and interactive interfaces like the desktop application. RIA’s were proposed to respond to these kinds of features and have collective affluence and connectivity of the desktop interfaces into the web distribution form. On the other hand, design and the implantation of the Graphical User Interface (GUI) for RIA are well-known for its complication and hard to use tools that are sometimes can be complex, time wasting and requires more input which impacts negatively on the excellence of the resulting RIA [1]. Adopting such model-driven methods in the making of the RIA can considerably lessen the risk of reworking; better the overall quality of the application as well as the UI in particular [2].

One of the challenges faced by the Rich User Interface (RUI) is the absence of precise approaches for developing the RUI [3]. In the past years, the most excellent web engineering methodologies have been challenged by the new RIA architecture. Moreover, the markets of the developing tools remain more centered on the execution technology, and therefore at this stage, it’s vital to establish a reflection of new architecture and features of RIA on the previous web engineering development methodologies and establish if the experience. Methods and modern tools can be taken up for new requirements RIA advancement or an entirely new improvement of development tools being made [4, 5].

In recent times some web engineering methods have attempted to deal and offer the developing support for RIA development. Busch and Koch (2009) categorized web engineering methods for supporting RIA for four types [6]: 1) Improving current methods with RIA features for reproduction, further transformation and creates a precise method like OOHRIA, OOHDM, UWE-R, and WebML) Merging of the web engineering methods with other methods of user interfaces plan and development such as RUX combined with WebML and...
methodology to accomplish this goal. Also, they use this evaluation process to obtain the suitability of a features which should be modeled in RIAs and proposed multimedia, and hypermedia methodologies to demonstrate that each one accomplishes only few RIA features, the authors analyzed the result is "the main engineering methods compared for developing RIA, and the works on extension metamodel this section, we review existing work on IFML for solving RIA complexity in web engineering area as applications is RIA, and many researchers attempted to develop RIA by web engineering methods [12]. In this section, we review existing work on IFML for developing RIA, and the works on extension metamodel in web engineering methods for developing RIA, as well as some essential works on RIA by other web engineering methods. In the excellent work by Preciado et al. [13] web engineering methods compared for developing RIA features, the authors analyzed the result is “the main features which should be modeled in RIAs and proposed an evaluation process to obtain the suitability of a methodology to accomplish this goal. Also, they use this process to evaluate the suitability of several existing web, multimedia, and hypermedia methodologies to demonstrate that each one accomplishes only few RIA features, so new methodologies or extensions of the actual methodologies become necessary”. Based on Table 1 previous web engineering methods had a weakness in supporting RIA, for that reason most web engineering methods improved for developing RIA features in different ways. In [13] made a comparison between web engineering methods based on ten criteria, the result present that these methods had a limitation (LIM) or partially (PAR) support for selecting criteria, also some criteria from these methods no supports (NON).

Several web engineering methods have been used in RIA designing including [7, 14-20]. RIAs have introduced powerful novel functionalities into the Web architecture, borrowed from client-server and desktop applications. The emergent platforms provide designers with the opportunity to enhance the experience of users besides exploiting the potential client-side data, synchronous and asynchronous events, bi-directional client-server communication, computation and rich interference widgets [4]. Melia et al. [14] recommended an approach by the name OOH4RIA that proposed a development process that is model-driven otherwise referred to as the OOH methodology. For purposes of business and data logic modeling, a mix of the UWE was propositioned. Consequently, the RUX-Method was proposed in RIAs user interface modeling and is essentially model-driven with the ultimate aim being the development of RIA [19].

Using the OMG standard IFML, Roubi S. et al. [21] developed a model-driven approach for generating RIA. It is important to note that the RIA metamodel respected the Model View Presenter (MVP) pattern besides built up the transformation engine that facilitated the automatic production of the model that was otherwise recognized as the output. The model generated acts as the input model when considering the text producer and result in a near ready RIA that can be deployed; this occurs on the one hand through when one focuses on the graphical aspect of the application while the aspect of the user handling the application is the other angle of evaluation. The strength of the recommended model lays in IFML simplification thus minimizing the complexity of the abstract details making it easier for the designer to effortlessly reproduce the RIA devoid of the need to comprehend the specifics of the technical aspects of the MVP application [21].

A study by Laaz and Mbarki [9, 22] defined a new approach that is essentially a Model Driven Engineering (MDE) methodology in the development of GUIs drawn
from models that are abstract. GUIs have dynamic and structural features that are modeled to show complete interfaces of RIA. Further, the process of development that is model-driven is hinged on Ontology and IFML. Ontology helps in the logical description of the UI features while IFML aids in capturing of the interactions therein. The process suggested in this context is one where the input is largely abstract. In addition, transformations applied in these models generate a code that exemplifies interfaces that are Flex rich. An example of such an approach is illustrated by an e-commerce website interface [9, 22].

In the last work, for improving IFML to support RIA done by Sachdeva and Singh [5], the authors differentiate their constructs of RIAs from the traditional web applications as well as an add-on to the literature on IFML concerning the available constructs. Among the proposed constructs analyzed include drag and drop, data distribution, asynchronous requests of the page, partial page computation as well as multimedia features and push technology. The recommended standards of language modeling will be useful in the automatic generation of codes by CASE tools that are edging closer to the final deliverable code of the application. The benefits associated with this are improved productivity and application efficiency which is the bedrock of model-driven approach [5]. IFML was realized after ten years of WebML utilization. To this end, IFML is recognized for fast development of web applications but has limitations of RIA process development.

Previous works as cited in this section proposed or suggested a new approach for supporting RIA by web engineering methods including IFML. Some of them combined with other web methods or enhanced the metamodels for solving the issue. However they implemented their approaches successfully, but they presented RIA features partially, in the next section we analyze RIA features and IFML metamodels for defining a new approach for developing RIA features.

3. Analyzing IFML and RIA Features

In this section, we analyze process development web applications by IFML for presenting architecture of this method in the process development applications as well as metamodels, in one hand. In another hand, we extract RIA architecture for highlighting RIA features in the engineering domain. However, in previous work, we present IFML method has weakness in the process development modern web application features including RIA features [23, 24]. The result of this analyze become to guide for improving IFML to develop RIA features.

3.1 Analyzing RIA Features

In the year 2002 Jeremy Allaire in a white paper introduced the word “Rich Internet Application” for the first time [25], he defined RIA as a new generation of online applications that make use of some technologies to offer a refined user experience above the open architectural standards on the internet. The nearly all notable innovation of RIAs is founded in the dominant connection mechanism of the interface. For example native multimedia support, transition effects, animations, widgets, drag/drop and so on, together with a flexible separation of the work between the user and the server, in comparison to that of pre-web client and server application. The natural way of separating work between the client and the server is what makes RIA more successful. By using the Web as a back-end keeps hold of all the gains that come with the open, low cost, free set up of the architecture. At the same time as the computation power of the user increases the server guarantees the best quality connection that an up to date desktop applications and operating system can present. The interaction between the Web server and the applications running on the user’s desktop is one of the basics of architecture in RIA, and that is correctly indicated in Fig.1. There are various technologies the put these applications into operation while inside the browsers such as HTML5, JavaScript, JSON, and XML. Applications make use of numerous paths to interact with one another; such paths include synchronous the user’s server with HTTP, asynchronous the user’s server with AJAX or Web Socket, and peer to peer with real-time web Communication [8].

New opportunities in respect to the untainted HTML as well as HTTP architecture arise from having an application runtime at the client-side: 1) The execution of the logic can be carried out from either side be it on the clients or the server. 2) There can be a separation of the business logic amongst the client and the server. 3) Information can be stored at both levels. 4) There is the flexibility of communication between client to server, server to client and client to client. 5) The client can still work without the connection to the server. From a developer’s point of view, RIAs bring in a range of new architectural patterns, designs decisions, and execution of languages. If any event was to be raised a client or informed by the server, or even by another user it will be implemented close by the user, handed over to the server or better still be implemented at the two levels. In summary, New and more competent Web applications where the transfer of the information and the business logic amongst the user and the server as well as where interaction amongst the user and the server can still happen in both directions are all facilitated by the new RIA architecture. The negative aspect is the increased complexity of the software and the rise of languages, data formats, and communication protocols, which in turn makes the model-driven development of RIA fascinatingly possible [8].
3.2 Analyzing IFML Metamodels

IFML [8] has been completed to improve the platform-independent definition of GUI with applications installed on systems like smartphones, tablets, desktops, laptops, and PDAs. Moreover, it focuses on the behavior of application and structure as observed by the end user. In the following, we extract IFML metamodels in the process development web applications.

The development of web applications defined by interactivity is normally handled with agile methodologies, which navigate consists of three phases of requirements, analyzes/design, and implementation. The iteration of the development method derives a partial version or a prototype of the system, Fig.2 presents structural of development process IFML within the activity flow.

**A. IFML-Artifacts**

OMG officially defined IFML as standard web engineering method, the most technical artifacts of this method represented below [8]:

- The IFML metamodel; specifying the structure and relations between the elements.
- IFML is a profile of UML; in this case, IFML uses UML diagrams and concepts.
- IFML has visual syntax; meaning it riched by graphical notations to represent the elements and models.
- The IFML model run and exchange format, for tool portability.

Altogther, these artifacts compose the IFML language specification. Also specified based on OMG standards:

- The metamodel is defined through the MOF metamodeling language (an equivalent ECORE definition is available too).
- The UML profile is defined according to UML 2.4 profiling rules.
- The visual syntax defined through Diagram Definition (DD) and Diagram Interchange (DI) OMG standards.
- The model serialization and exchange format are defined based on XMI.

**B. IFML Metamodel**

IFML metamodel defined as best approach and described as the best language in practice. Incorporate abstraction, modularization, extensibility, and reuse. When consists of three packages; Core Package, Data type Package and Extension Package [8].

Core package has the concepts that build the infrastructure of the language that are Interaction Flow Elements, Interaction Flow, and Parameters. Data Type package contains the custom data types that defined by IFML. Extension Package can extend the concepts of Core package [8].

**C. Extension IFML Metamodels**

In this part, we explain extensibility of IFML metamodels for solving the gaps and the challenges the IFML, in the process designing and implementing web development as well as web applications. In section 2, we presented some researchers extended IFML metamodels for different concern, here we discuss how these extensions help IFML for developing web applications including RIA.

Previous works for addressing the issues of web engineering methods including IFML use one of the following ways: 1) Integrating current method with other web engineering methods. 2) Integrating current method with other models. 3) Defining a new method through the extension of the original method like [26].

Moreover, UML allows to extending metamodels package [27]. Also, UML extension mechanism allows designing new elements by adding new building block that is; stereotypes, tagged value, and constraints [28]. We test and approve UML extension in previous works [29-31].

Extension methods and models classified for two categorists that are: extension purpose and extension mechanism [32]. Extension purpose highlights the aim of the researcher to enhance the method, but extension
mechanism focused on how extending methods technically, as explained in [33, 34]. Finally, we can conclude this section is; IFML has extensibility for RIA features, in the next section we proposing a new approach for that aim.

4. Methodology

In this section, we define a new approach for enhancing IFML through extension metamodels to develop RIA features. Here we adopt our previous approach for extension IFML metamodels as presented in [35], before extending IFML metamodels should highlight the RIA features and explain how added to IFML models, then we can extend it based on the new features as shown in Fig. 3.

![Fig. 3: A New Approach for proposing RIA-IFML](image)

4.1 IFML Elements and Concepts

IFML concepts and elements use for designing front-end interaction model to represent IFML model diagram [8], briefly the main elements presented in Table 2:

<table>
<thead>
<tr>
<th>No.</th>
<th>IFML Concept and Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>View Container</td>
<td>An element of the interface that comprises elements displaying content and supporting interaction and/or other ViewContainers.</td>
</tr>
<tr>
<td>2</td>
<td>ViewComponent</td>
<td>An element of the interface that displays content or accepts input</td>
</tr>
<tr>
<td>3</td>
<td>Event</td>
<td>An occurrence that affects the state of the application</td>
</tr>
</tbody>
</table>

![Table 2: IFML Concepts and Elements](image)

4.2 RIA Features

RIA is a web application proposed to deliver same features desktop application on the internet. RIA generally slit the processing across between client and server. Which user activity refer to the client side, data manipulation, and operations refer to the server. Several articles highlighted RIA features. In our previous experience, we analyzed web RIAs features and summarized the features that consist of Client, Server, Rich User Interface and Rich Client [36]. In this paper our main focus on Client and Server communication.

4.3 Extension IFML metamodels for RIA

RIA features based on our paper defined how we show distribution support Client and Server communication by IFML model. In this section, we extend IFML metamodel for defining new elements to design RIA features.

IFML metamodels have the capability to extensions to define new concerns. In the following, we extend IFML Core packages for developing RIA features. In the first step, the new class defined entitled RIA models for supporting has two attributes that are Client and Server, this class defined by OCL language for the separated role of Client and Server. In the second step, the RIA model integrated with original metamodels that generated Core elements. In the third step, the new elements generated after integrating RIA model with original IFML elements. These elements have attributes existing IFML elements with RIA model attributes, the new classes presented by gray color in Fig. 4. finally, we have a new IFML metamodel for support RIA features.
4.4 Define New Elements for IFML

After extension IFML metamodels for RIA in Fig. 5, we can define the elements by using UML extension mechanism. IFML uses the native extensibility mechanisms of UML to allow the definition of stereotypes, tagged values, and constraints on existing concepts.

In this paper, we are not defined new elements; we need to add new RIA features for original elements, for this reason, we propose OCL language only for new features. In the following, we present OCL language for new elements as an independent element.

- **Stereotype:** C/S.
- **List of attributes** (Client, Server).
- **Context:** ClientServer inv: name.size()<= 50 inv: number>=0 && number<=100 inv lenient.size()<=5.

We can follow above OCL syntaxes for generating all elements that defined during extension IFML metamodel. The graphical icon for new elements is one of the requirements define new elements, after applying above OCL codes for all elements we get new elements for supporting RIA features. In the Table 3, list of new elements tabulated with descriptions. In this table not described the elements because already described in Table 2, here only presented the new description hat are Client and Server feature.

<table>
<thead>
<tr>
<th>No.</th>
<th>IFML Concept and Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[C]/[S] ViewContainer</td>
<td>C/S-View Container</td>
</tr>
<tr>
<td>2</td>
<td>C/S-ViewComponent</td>
<td>C/S-ViewComponent</td>
</tr>
<tr>
<td>3</td>
<td>C/S-Event</td>
<td>C/S-Event</td>
</tr>
<tr>
<td>4</td>
<td>C/S-Action</td>
<td>C/S-Action</td>
</tr>
<tr>
<td>5</td>
<td>C/S-Navigation Flow</td>
<td>C/S-Navigation Flow</td>
</tr>
<tr>
<td>6</td>
<td>C/S-Data Flow</td>
<td>C/S-Data Flow</td>
</tr>
<tr>
<td>7</td>
<td>C/S ParameterBindingGroup</td>
<td>C/S-Parameter Binding Group</td>
</tr>
</tbody>
</table>
Finally, we get new elements for IFML that could support RIA Features in the process development RIAs, and we can call RIA-IFML. In the next section to approve our approach, we design a case study by RIA-IFML.

5. Designing Case Study

In this section, we design IFML model for a case study by using current IFML elements and RIA-IFML elements.

5.1 Design Case Study by IFML

The case study is bookstore scenario that designed by [8]. Based on the scenario the user selects a category from CategoryList a navigation event is produced, and as a result, the products corresponding to the Selected Categories are displayed. Similarly, when the user selects a product from ProductList, the details of the SelectedProduct are displayed, as shown in Fig.5.

Fig. 5: FML model corresponding to the exploration of products [8]

Fig.5 presented user interaction with the server but could not distribute client actions with Server actions, in the following same scenario designed by using new elements that proposed for RIA-IFML. The new IFML can present RIA features in IFML models.

5.2 Design Case Study by RIA-IFML

IFML-RIA has all IFML properties for presenting interaction design and RIA features at the same time. We redesign the same scenario that designed by IFML in Fig.5. In the first step, the ViewContainers (Categories) should be separate the action refers to which side. This category related to Server side, so we designed as Server action and [S] added on top of ViewContainer, also ViewContent that named (Category List) listed by the server and need to add (S:) for showing the action from Server side. In the second step, each user selects a category of the book as a client, so we need to add (C:) for showing this action from the Client side. I the third step user select a Book so need to add (C:) as Client side, Fig.7 designed the scenario through RIA-FML.

Fig: 6 RIA-FML model corresponding to the exploration of products

Now we can find a difference between designing by IFML in Fig.6 and designing by RIA-IFML in Fig.6. RIA-IFML more interactive compared with IFML, because with presenting user interaction distributed the role of Client and Server in each step, this enhancement help the developer for designing the projects more efficient in the implementation phase.

6. Discussion Result and Limitations

The new approach for developing RIA features through IFML metamodels done successfully, as explained in the previous section after designing the case study, the difference between IFML and RIA-IFML appeared. The difference is RIA-IFML can distribute the actions of Clients and Server. Currently, RIA-IFML has all elements that present web application features in rich domain and communication between Clients and Server. IFML metamodels have the capability for defining ant concepts from any applications, but we could not propose more elements for RIA-IFML, because any new elements have new challenges. For example, introducing a new element for rich user interface need to reformation model between interface and content in one hand, and need to explain how to distribute this new element between Clients and Server in another side. Also, tool support becomes another challenge in the process development. Finally, we can conclude RIA-IFML more efficient during developing RIAs but need to more extensions, in the next section we recommend for future work for more extensions.

7. Conclusion

In this paper, we enhanced IFML metamodels through extension IFML metamodels to develop RIA features. However, IFML method described as a new method for supporting RIA, but has limitation and cannot adequately support RIA features especially Client/Server. For solving this issue, we extend IFML metamodels by implementing UML extension mechanism for defining new IFML model.
that included new elements for RIA and called RIA-IFML.

Our approach implemented successfully, but this is not enough for RIA because RIA has other features such as rich user interface. We recommend for the researcher to improve IFML metamodels to design all RIA features. moreover enhancing the tool supports for presenting the new features including code generation

References


[22] Laaz, N. and S. Mbarki. A model-driven approach for generating RIA interfaces using IFML and ontologies. in Information Science and Technology (CiSt), 2016 4th IEEE International Colloquium on. 2016. IEEE.


[32] Braun, R. Towards the state of the art of extending enterprise modeling languages. in Model-Driven Engineering and Software Development (MODELSWARD), 2015 3rd International Conference on. 2015. IEEE.


