

# Comparative Analysis of Interaction Flow Modeling Language Tools

<sup>1</sup>Naziha Laaz, <sup>2,3,4</sup>Karzan Wakil, <sup>1</sup>Samir Mbarki and <sup>4</sup>Dayang N.A. Jawawi

<sup>1</sup>Department of Computer Science, Faculty of Science, Ibn Tofail University, Kenitra, Morocco

<sup>2</sup>Department of Information Technology, Technical College of Informatics, Sulaimani Polytechnic University, Sulaimani, Iraq

<sup>3</sup>University of Human Development, Sulaimani, Iraq

<sup>4</sup>Department of Software Engineering, Faculty of Computing, University Technology Malaysia, 81300, Johor, Malaysia

## Article history

Received: 14-06-2018

Revised: 15-07-2018

Accepted: 02-10-2018

Corresponding Author:

Naziha Laaz

Department of Computer Science, Faculty of Science, Ibn Tofail University, Kenitra, Morocco

Email: laaznaziha@gmail.com

**Abstract:** Modeling approaches based on standards are of paramount importance in the field of front-end design for web and mobile applications. Problems are often encountered during the selection of tools for designing applications for developers and researchers, particularly applications that are related to time and cost in the market and academia. The Interaction Flow Modeling Language (IFML) is a recently standardized modeling language designed for managing the content expression, user interaction and behavior control of front-end applications. IFML brings several benefits to the development process of web and mobile front-end applications. Thus, several tools have been developed for the exploitation of technical artifacts offered by current specifications. In this study, we discuss the tools proposed for modeling IFML to provide a comparative analysis while considering various criteria. The result can be used as a basis of tool selection for specific aspects.

**Keywords:** IFML, Modeling, Modeling Tools, Comparison

## Introduction

Model-Driven Engineering (MDE) methods are aimed at providing tools and techniques for simplifying the design, development and evolution of software applications. Tools play an important role in MDE adoption given that traditional software engineers usually rely on the use of case tools to boost their productivity (Rossi *et al.*, 2016). Web applications and web modeling languages are increasing in complexity and thus demand robust process and tool support, which is one of the critical factors for completing a web development lifecycle, especially during model design and code generation.

The use of Interaction Flow Modeling Language (IFML), a new web engineering method, has recently been proposed for web and mobile application development (Brambilla and Fraternali, 2014a). It is an Object Management Group (OMG) standard that supports the platform-independent description of graphical User Interfaces (UIs) for desktop computers, laptops, cellular phones and tablets (Brambilla, 2013). Several tools, such as WebRatio (Acerbis *et al.*, 2015a),

IFMLEditor (2018) and IFMLEdit.org (CarloBernaschina, 2018), have been proposed for designing IFML models and generating codes.

IFML tools can design standard models and sometimes provide the final version as generated code. Each tool is used for a particular concern with a special philosophy. Moreover, IFML is new and thus not applicable to all types of web applications (Wakil and Jawawi, 2017a). The problem is IFML designers and developers have difficulty selecting tools during the development process.

Focusing on the topic of comparison and evaluation techniques of modelling methods, several works have been done, the most relevant are (Wakil and Jawawi, 2017a; Toghroli *et al.*, 2016; Siau and Rossi, 2011; Wakil *et al.*, 2015; Wakil and Jawawi, 2014). Regarding IFML tools, no comparison or evaluation were proposed. However, Hamdani *et al.* (2018) presented a systematic review of many works based on IFML and, suggested to establish a comparison between IFML tools, but they still have not complete it. In this study, we compared IFML tools used for developing web and mobile applications. Important criteria were selected as the

bases of the comparisons. The results are deemed useful to developers and researchers when they are selecting tools for a particular purpose. Finally, this study answers the following questions in industry and academic fields: Which tool supports the entire lifecycle phases of web engineering? Which tool rapidly uses the design stereotype of modeling? Which tool easily supports code generation? Which tool is best for a particular concern?

The rest of the paper is structured as follows. Section 2 presents previous design IFML models using the different tools and related works on the available tools. Section 3 explains IFML elements. Section 4 analyzes current IFML tools and roles to support IFML. Section 5 defines the criteria as the primary comparison and focuses on the comparison and analytical discussion of available IFML tools based on criteria. Section 6 designs a case study as an evaluation of the tools. Section 7 presents the conclusion and provides some ideas for future work.

## Background

This section presents the tools available for IFML and cites different IFML-based approaches for the model-driven development process, which creates IFML models using either IFML tools or the classical method, which involves the creation of a dynamic instance directly from the IFML metamodel syntax. Finally, this section tabulates the existing work based on the IFML metamodel with several descriptions, such as (process development approaches, technology type, application type, implementation technology and publication year).

IFML have built by developers with an extensive experience on using WebML and WebRatio for over 10 years' experience. It was adopted as a standard by the OMG in March 2013 (Bernaschina *et al.*, 2017a). IFML brings several benefits to the UIs development process of the web, desktop and mobile applications. IFML visual syntax is one of the five artifacts offered by the IFML specification (Acerbis *et al.*, 2015a). However, the visual syntax is compact and familiar to developers, which explains why several tools have been developed to design IFML diagrams.

One of the main model-driven application development tools that mark IFML editing is not of the famous WebRatio (2018). It is an integrated platform that supports the applications that model with BPMN, IFML and UML class diagrams and their implementation with model-driven code generators (Acerbis *et al.*, 2015b). WebRatio is generally used for creating robust and composite enterprise-class mobile and web applications. It is a consolidated industrial reality and allows the creation of any model for the

generation of custom web or mobile UIs. Additionally, it enables the deployment of on-premises or cloud applications (Brambilla and Fraternali, 2014b). Another solution is an open source IFML Editor (2018) by Eclipse; this editor is a plugin based on the Eclipse Modeling Framework (EMF) and Sirius technology. The editor provides a modeling environment for expressing the content, user interaction and control behavior of software application front-ends (Ed-Douibi and Brunelière, 2015). Moreover, the IFML editor supports bindings between the front-end layer and the two other layers: The persistence and logical business layers.

IFML Edit.org is a web-based open-source IFML tool (Carlo Bernaschina, 2018) developed with Agile Model Transformation, which is an agile modular framework that allows developers to rapidly bootstrap and evolve their MDD tools (Editor and Code Generators) (Carlo *et al.*, 2017). It allows the rapid prototyping of web and mobile applications. Also, in (Bernaschina, 2017) they are extended these tools to support other features.

In recent years, several research projects (research works) using model-driven approaches for the development of UIs based on the IFML standard have been proposed. Given the tools mentioned above, IFML has quickly made its way in the software industry and academia under the impulse of several approaches. Many researchers have exploited IFML to specify the domain model and user interaction models for applications have been conducted. Each of these works use one of the tools mentioned before. Laaz and Mbarki (2016a; 2016b; Rhazali *et al.*, 2016; Rodríguez-Echeverría *et al.*, 2016; Roubi *et al.*, 2016; Sachdeva and Singh, 2017; Yigitbas *et al.*, 2015) focus on the web. Acerbis *et al.* (2015a; Bernaschina *et al.*, 2017b; 2017c; Brambilla *et al.*, 2014; Brambilla, 2015; Umuhoza *et al.*, 2015) focused on mobile applications. Other research works have exploited IFML in different domains, such as interactive systems (Gotti and Mbarki, 2016a), big data (Bernaschina *et al.*, 2017a), execution models (Gotti and Mbarki, 2016b) and user behavior analysis (Bernaschina *et al.*, 2017a). Table 1 highlights the existing works based on IFML and the case tool.

### Interaction Flow Modeling Language (IFML)

IFML is compatible with the platform autonomous depiction of graphical user links for applications deployed in every technological and system platform (Bernaschina *et al.*, 2017a). End users have observed that the behavior and structure of a front-end application is the aim of the depiction. The typical purposes of the IFML are managing the control behavior, user interaction and the expression of the content of the front-end application of a software.

IFML facilitates the description of the following:

- The logic that determines the sequence of actions to perform in response to an event
- The bindings among displayed contents, objects/data and events
- The objects/data displayed by the application or managed by business logic
- The pages/windows through which the application interacts with the user

According to the standard model-view-controller model of a communicating application, IFML primarily focuses on the view portion. IFML models support the following design perspectives, whose notation is summarized in Table 2.

(1) The view structure description contains the definitions of the view containers. (2) The view content description contains the definitions of the view components, that is, data entry elements and content publishing contained in view containers. (3)



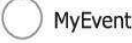



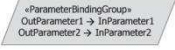
The event's description contains the definition of events, which originate from external agents, application logic, or users' interaction and which may influence the state of the UI. (4) The event transition description contains the definition of the impact of an event on a user interface. (5) The parameter binding description contains the definition of the input-output reliance between model elements.

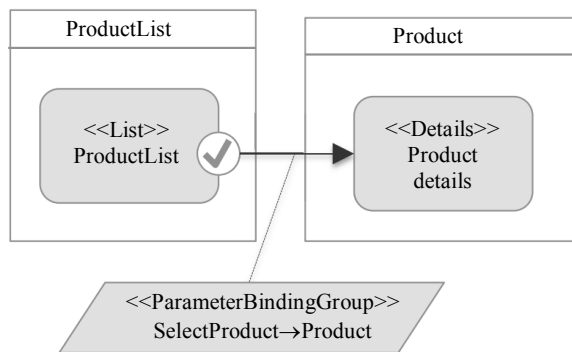
Moreover, IFML can be supplemented with peripheral models for linking to any form of the dynamic model, which explains the business idea responsible for the application front end and any form of the content model (in place of file systems, ontologies, databases, or any other resource). Figure 1 illustrates a direct instance of the IFML diagram, where an initial page shows a list of products and a target page, which show the information of the selected product, upon user selection (Acerbis *et al.*, 2015b).

**Table 1:** Presenting existing work on IFML and Case Tool

Studied Articles	Technology type	Application type	Tool Support	Year
MDA approach based on IFML and ODM for generating RIA interfaces (Laaz and Mbarki, 2016a; 2016b)	Adobe Flex, html5	RIA	IFML Editor	2016
MDA approach based on IFML for RIAs (Roubi <i>et al.</i> , 2016)	JavaFX	RIA	IFML Editor	2016
An abstract presentation of interactive systems based on ADM approach (Gotti and Mbarki, 2016a).	Not specified	interactive	IFML Editor	2016
Extending IFML for web and mobile Platforms (Brambilla <i>et al.</i> , 2014)	HTML5, CSS3 and JavaScript	Mobile app	WebRatio	2014
Automating IFML Specification of CRUD Operations (Rodríguez-Echeverría <i>et al.</i> , 2016)	Data entities	Data-intensive web app	WebRatio	2016
Extending Support of IFML to RIA Constructs (Sachdeva and Singh, 2017)	AJAX, HTML5	RIA	Not specified	2017
customized MDUID process integrating HCI Patterns for generating RIAs (Yigitbas <i>et al.</i> , 2015)	XML, JavaServerPages, JavaScript and CSS	RIA	IFML Editor	2015
Toward IFVM Virtual Machine: A Model-Driven IFML Interpretation (Gotti and Mbarki, 2016a)	Java	Execution app	IFML Editor	2016
Model-driven development of cross-platform mobile applications with webRatio and ifml (Acerbis <i>et al.</i> , 2015a; Brambilla, 2015; Umuhoza <i>et al.</i> , 2015)	HTML5, CSS, JavaScript	Mobile app	WebRatio	2016
IFMLEdit.org: Model-driven rapid prototyping of mobile apps (Bernaschina <i>et al.</i> , 2017b)	Node.js	Mobile app	IFMLEdit.org	2017
Online model editing, simulation and code generation for web and mobile applications (Bernaschina <i>et al.</i> , 2017b)	Not specified	Web and Mobile app	IFMLEdit.org	2017
A Big Data Analysis Framework for Model-Based Web User Behavior Analytics (Bernaschina <i>et al.</i> , 2017a)	Not specified	Web app	WebRatio	2017
A model transformation in MDA from CIM to PIM represented by web models through SoaML and IFML (Rhazali <i>et al.</i> , 2016)	Not specified	Web app	IFML Editor	2016
Extended IFML metamodels and designed an example after extension (Wakil and Jawawi, 2017b)	Not specified	Web app	WebRatio	2017
Analyzed IFML phases and added to Web Engineering Mapping (Wakil and Jawawi, 2017c)	Not specified	Web app	WebRatio	2017
Model Based Rapid Prototyping and Evolution of Web Application (Falzone and Bernaschina, 2018)	Not specified	Web app	IFMLEdit.org	2018
UML and IFML Mapping to GraphQL (Cabot, 2018)	Web APIs, JSON	Not specified	Not specified	2018
Formal semantics of IFML for mobile and rich-client application MDD (Bernaschina <i>et al.</i> , 2018)	JavaScript/Java/C#, CSS and HTML5, etc.	Web and Mobile app	IFMLEdit.org	2018

**Table 2:** Basic Core IFML concepts: Description, notation and platform-specific cases (Brambilla and Fraternali, 2014a)

Concept	Meaning	IFML Notation	PSM Example
View container	An element of the interface that comprises elements displaying content and supporting interaction and/or other View Containers.		Web page Window Pane.
View component	An element of the interface that displays content or accepts input		An HTML list. A JavaScript image gallery An input form.
Event	An occurrence that affects the state of the application		Select a line from a list
Action	A piece of business logic triggered by an event		A database update. The sending of an email.
Navigation Flow	An input-output dependency. The source of the link has some the output that is associated with the the in put of the target of the link		Sending and receiving of parameters in the HTTP
Data Flow	Data passing between View Components or Action as the consequence of previous user interaction.		-----
Parameter Binding Group	Set of Parameter Bindings associated to an Interaction Flow (being it navigation or data flow)		-----



**Fig. 1:** Example of IFML model showing a list of products and the details view (Acerbis *et al.*, 2015b)

**IFML Tools**

In the development of web and mobile applications, some tools have been invented for code generation and IFML models design. This section describes the current tools, including IFML editor, ifmledit.org and WebRatio.

**WebRatio**

WebRatio Policy (Mia-Software, 2018) is a low-code model-driven development tool related to IFML. It

features two editions and concentrates on the design and development of mobile and web applications. WebRatio Policy applies the web-extended form of IFML and aims precisely on web applications. WebRatio offers the following environments:

- The modeling environment supports the integration of IFML diagram specification into BPMN diagrams for the description of business process features, UML ER diagrams (class diagrams) for information design and the view definition
- The development environment supports the application of custom elements, thereby enabling the tailoring of modeling language extensions according to vertical requirements or natural responsibilities and system integration and custom data requirements
- The style design and layout template environment facilitate the greatest possible degree of UI complexity given the full support of JavaScript-based styling, CSS and HTML

WebRatio offers lifecycle management, group work support, full code generation and model checking that concerns the inputs provided by such environments. The generated code is composed of an automatic cloud-

deployed Java EE code that involves back- and front-end web applications for the WebRatio web version and ready-to-deploy cross-platform mobile applications for the mobile version within the PhoneGap distribution, based on the open-source Apache Cordova framework. The styling and layout of the target applications can be described by defining CSS and the HTML5 templates. WebRatio mobile develops applications by default that are completely working even when offline. It also offers an integrated solution for reading and writing data synchronization between the back end and the mobile device. The design of the mobile application is based on de facto and open standard libraries, which include Ionic, JData and Angular.js.

### *Ifmledit.Org*

IFML edit is an online tool. Making multifaceted configurations is unnecessary in the creation of IFML models. IMFL edit is easy to manipulate but does not support XMI interchange (Hamdani *et al.*, 2018). It is focused on mobile applications and quick web prototyping. IFMLedit.org is related to IFML, which is the OMG standard interface between applications and users (Brambilla and Fraternali, 2014a). It is an online tool that generates codes for web and mobile architecture, the investigation of their properties using mapping to Place Chart Nets (PCN) (Acerbis *et al.*, 2015a) and the description of IFML models. The environment supports the following workflow:

- Mobile emulator: This JavaScript constituent emulates a mobile cross-platform situation (currently Cordova). It supports the prompt implementation of the produced cross-platform mobile code
- Browser-server emulator: This emulator is a wholesome JavaScript module capable of emulating Node.js servers, web browsers and request–response sequences that links the two. This emulator is applied to support online and offline functionalities flawlessly
- Model-to-JSON transformation framework is a basic rule-related transformation framework built in JavaScript that. It is provided as an input of a model entity and it creates an output of a JSON assembly that epitomizes a model-to-model transformation model, a model-to-text transformation (file system structure), or anything that can be defined as a JSON object
- Model editor: This editor is a multipurpose visual editor that enables developers to insert (through dragging and dropping) components in a model, correct their properties and link them

The system provides the developer with the following tools (Brambilla and Fraternali, 2014a):

- IFML editor: This is a modification of the basic model editor to obtain IFML models

- PCN generator and simulator: This is related to the model-to-JSON transformation framework and the IFML model editor. It enables the production of a PCN from the IFML model and simulation of its performance
- Web code generator and emulator: This generator and emulator is related to the model-to-JSON transformation framework and the browser-server emulator. It enables an individual to develop a Node.js app from the IFML model and install it in the browser
- The mobile code generator and emulator: This generator and emulator is based on a model-to-JSON transformation framework and a mobile simulator. It supports the creation of a Cordova application from the IFML model and its simulation in the browser

### *Ifml Editor*

IFML editor is an open source Eclipse editor for IFML. It is derived from Sirius API, an eclipse project and facilitates the modeling of an editor's creation. The IFML editor results from a partnership between the INRIA research team AtlanMod, WebRatio and Politecnico di Milano.

This modeling device has all the advantages of IFML, which is incorporated on Eclipse as a plugin. The major elements of the IFML editor are the full coverage of the OMG IFML standard, code generation-friendliness (e.g., Acceleo Manifesto) and extensibility to different domains based on EMF (Eclipse inside). The open source IFML editor can be found on GitHub or the Eclipse Marketplace in cooperation with the projects gallery and the Sirius installation guide.

The IFML editor implements EMF as new software. Eclipse EMF (IFMLEditor, 2018) and models are described below:

- Data model: This refers to a domain model and stands for data that are being worked on
- EMF: This refers to a set of Eclipse plug-ins that can be applied in data modeling and the generation of some output based on this model or code. EMF has a difference between the actual and meta-model, which illustrates the model's structure. A model refers to a concrete case of this meta-model
- Generating data from an EMF model: The stored information in the EMF models can be applied in the generation of the derived output. For instance, EMF can be applied in the definition of an application's domain model and equivalent Java implementation classes can be created from this model. The EMF framework shows that the created code can be securely expanded with a hand

- Metamodels-Genmodel and Ecore: The EMF meta-model comprises two sections, namely, Genmodel and Ecore description files
- Ecore description file: The Ecore file provides room for the definition of the following elements:
- EClass stands for a class that has zero or more references and zero or more attributes
- EAttribute stands for an attribute that comprises a type and a name
- EReference stands for one end of an association amid two classes. It comprises flags that point out if it stands for a reference class and containment to which it is pointing
- EDataType stands for the kind of an attribute, for instance, end date, int, java, or float
- Ecore description file: A graphical representation of an accessible ecore model can be generated with the use of the context menu of an Ecore file and through the selection of an initialized Ecore diagram

### *Mia Studio*

Mia Studio refers to a code generation solution that maximizes the characteristics provided by the Eclipse platform. Mia Studio, which is meant for developers and architects, refers to a workshop for the execution and creation of code generators with the use of models that conform to an enterprise modeling language like (DSL) or UML type models (Mia-Software, 2018). The community version is available.

Mia Studio refers to a set of plugins that enable an Eclipse-based development environment to generate and apply model transformers and code generators. The developer can access all the power of Mia software products incorporated into their Eclipse workstation.

The Mia Studio Integrated Development Environment (IDE) provides a code generation Android SDK that enables the development of the user interface layer (Mia-Studio, 2018). The generation rules UML/IFML to Android are intended to make a few parts of the native Android application using one UML model (OMG-UML, 2018) linked with one IFML model (OMG-IFML, 2018). The generated code refers to an operational android SDK application, despite the code that must be completed by developers for obtaining superior UI feel and look and executing a number of UI treatments, particularly business services layer invocation.

### *Comparative Analysis of Ifml Tools*

This section compares available IFML tools. Currently, available IFML tools are WebRatio, IFMLedit.org, IFML Editor and MIA studio (Section 4). We selected the most relevant criteria after analyzing the different tools of IFML and gave an overview of each of them:

- Open source: This criterion for IFML allows tools to be updated, have new capabilities, or fix bugs through a wide community
- Multi-platform: This criterion permits to target many users by allowing the use of the tools despite the computer architecture or operating system
- User-friendly: This criterion provides practical experience to users through a well-established GUI. Moreover, it should be efficient, intuitive and effective error handling
- Extensibility: This criterion can be extended to fulfill the desired requirements. It is a principle that considers the evolution of tools
- Tool speed: This criterion represents the response time that the tools take to achieve tasks
- Support code generation: This criterion can remove a lot of necessary time doing manual coding tasks
- Standard-compliant: This criterion requires a modeling device to be derived from the IFML standard. It compels the creation of standards-compliant models. The creation of non-standard models brings the danger of its distinctive interpretation
- Teamwork: Intricate systems necessitate large teams. Thus, the chosen device must give room for numerous people who work on a similar model by putting off conflicts as it allows model branches. This criterion may not require a team developer to be capable of modifying high-level architecture
- Simulation: System simulation is the best way to ensure that it works as the model describes
- Analysis/validation functions: The device offers numerous views of the model by giving room for the users to examine the model from diverse points of view by automatically validating the model and giving room for the user to discover inconsistencies or model gaps
- Support for all types of related models: Numerous types of models must be created using diverse modeling devices. This implies that users fight with the synchronization of data and traceability
- Document output: A lot of paperwork circulates in the world of IT. A carefully chosen modeling device should circulate paperwork for users. The information is within the model and users only want to convert it into paper format
- Centralized repository: Users should not look for the newest version of models on the network. Integrating transformations to the newest is not easy when users work with an old model
- Graphical modeling: Developers can design and model the IFML models with the use of IFML specification's visual syntax
- Support context of applications: A tool employs appropriate context modeling and reasoning techniques to facilitate the design of models for the application development process

**Table 3:** Comparison between IFML Tools

Criteria	WebRatio	IFMLedit.org	IFML Editor	MIA-Studio
Open Source	N	Y	Y	Y
Multi-Platform	Y	Y	Y	Y
User-friendly	Y	Y	Y	Y
Extensibility	Y	Y	Y	Y
Support Code generation	Y	Y	Y	Y
Standard-Compliant	Y	Y	Y	Y
Support Teamwork	Y	N	Y	Y
Simulation	N	N	N	N
Analysis/validation functions	N	N	N	N
Support for all types of related models	N	N	N	N
Graphical modeling	Y	Y	Y	N
Support context of applications	N	N	N	N
Document output	N	N	N	N
Y: Supported	N: Not Supported			

Table 3 presents a comparison between the tools based on criteria in process development applications and designer experience.

The criteria of the tools described in the table have many similarities. All the tools are multi-platform, user-friendly, extensible and standard-compliant. However, the tools do not support simulation, analysis or validation functions, all types of related models, the context of applications and document output. In the following, we detail the comparison between the tools

#### *Industrial and Academic*

The adoption of criteria is the first step in the simplification of the tool selection process. We usually recommend general-purpose tools in the early stages of modeling and situation-specific tools are applied in a precise and carefully defined way, such as MIA Studio for mobile applications and WebRatio for web applications. For code generation, only WebRatio and MIA-Studio can generate the code directly after modeling the IFML diagram:

- IFML is dedicated to researchers and the industrial world. However, WebRatio is developed to be used primarily in industries. IFMLedit is designed for research due to the limitations of this tool
- Industrial: For developer or designer
- Based on our research, we recommend the use of Mia Studio and WebRatio for industrial purposes given the multiple functionalities as presents and the criteria cited above

#### *Academic: For Researcher*

As for academic purpose, the user can use all the tools cited above except IFML.edit which still a mature project and not as powerful as the others.

If the user targets a specific platform, the Mia Studio or WebRatio is the best choice, but if the user wants to customize his output, then the IFML editor is

the best to go because it has a much abstract aspect that gives the user multiple choices when it comes to customization.

#### *Open Source*

The analysis table shows that all tools are open source, except WebRatio, which comes in two editions: A community one, which is free of use and a professional one, which is a commercial edition. The community edition does not provide sufficient functionalities (WebRatio-Platform, 2018):

- Building custom styles
- Local back-end testing environment
- Back-end production environment on-premises or in any public or private cloud and team working

#### *Support Multiplatform*

IFML editor and Mia Studio are tools that run in eclipse as plugins and eclipse is a cross-platform IDE. Both tools support multiple platforms:

- IFMLedit.org is a web-based tool and can be used in all the platforms that have a browser
- WebRatio is built on the Eclipse editor, which is a cross-platform IDE. Therefore, WebRatio is also considered a multi-platform IDE
- Thus, each IFML modeling tool runs on different devices

#### *Speed*

The speed criteria invokes the time response that each tool takes to begin and use:

- Mia Studio and IFML editor are Eclipse dependent tools. It means that their speed is relative to eclipse IDE speed. Eclipse has more plugins and can be slow in the starting phase

- WebRatio is based on Eclipse IDE. Thus, its response time is as fast as an eclipse
- IFMLedit.org is considered the fastest among other tools because it is a web-based tool. Therefore, it depends only on Internet speed

### Technical Difficulties:

- WebRatio is the richest tool among the presented tools but consists of Java 1.7 version, which is no longer maintained by the Oracle Corporation
- Mia Studio is an eclipse plugin and can be found in the eclipse marketplace
- IFML editor is a plugin built for eclipse and compatible with the latest version. It is not embedded in the marketplace, similar to Mia, but can be installed as new software. It is easy to install and presents no difficulties

### Code Generation

Many tools cannot afford this criteria. Of the listed tools above, only webradio can generate the code because WebRatio has specific platform support (mobile and web), noticeable in its environment by

many components related to the target platform, unlike other tools that present an abstract definition of the IFML diagram.

IFML editor does not provide code generation but has the ability to obtain this function added using Aceleo. This promising tool gives the user the power to extend and specify the output for the output model.

### Case Study and Evaluation

This section designs a case study by the IFML editor. The case study is an Amazon website with the following scenario.

Scenario: In this design, the case study is on the Amazon website. Customers after login in the following scenario as shown in Fig. 2:

- Search for a book- customer searches for a book in Amazon bookstore
- Order book- customer orders the book
- Make payment- customer pays online for the book

The IFML diagram is designed by the Eclipse-based editor “IFML Editor” but can be created with all the tools mentioned above.

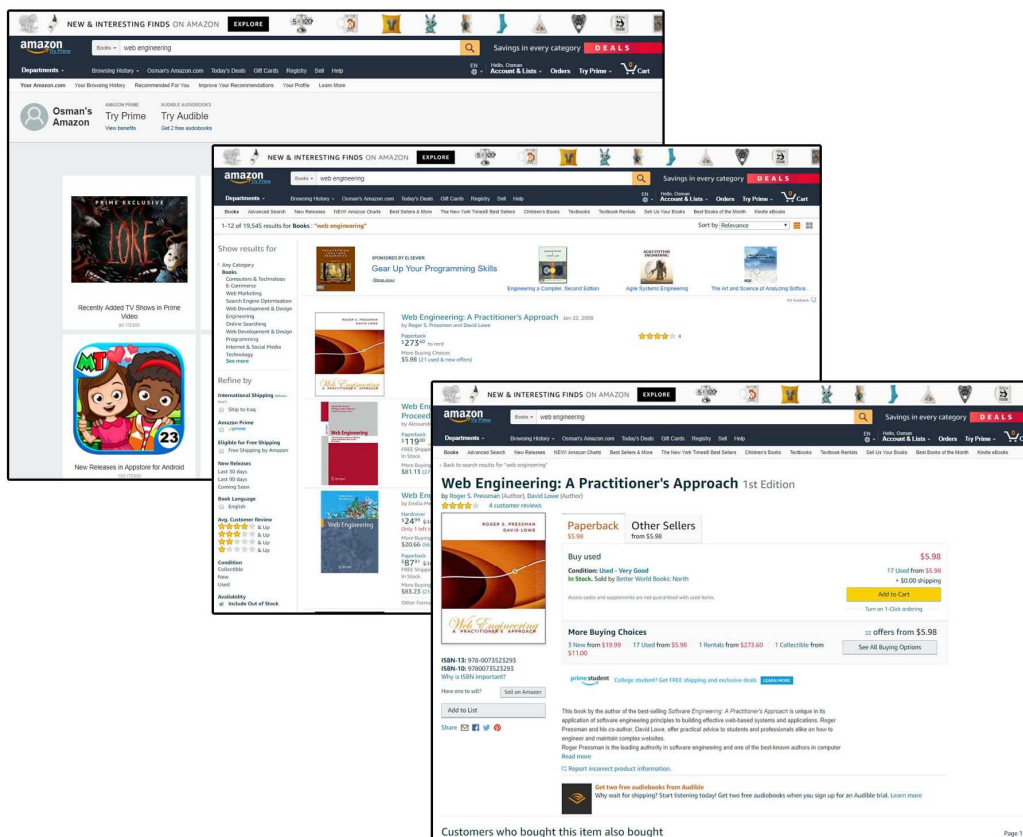


Fig. 2: Amazon website scenario process search, order and payment



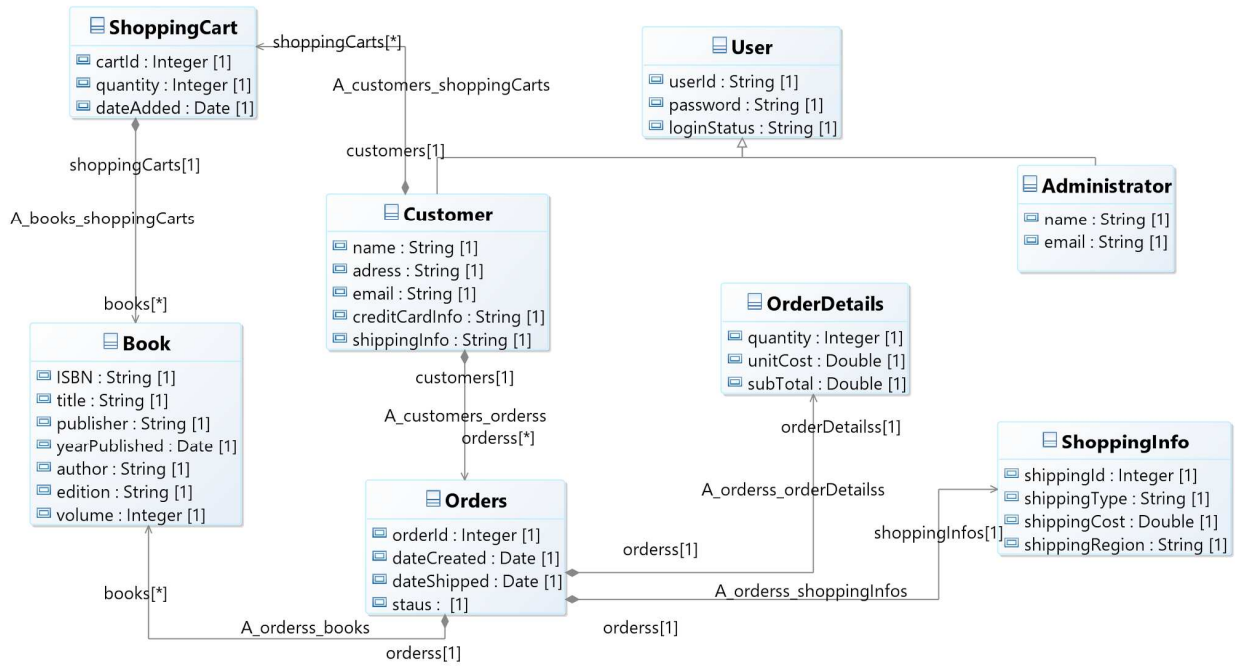


Fig. 3: Data model for Amazon website scenario

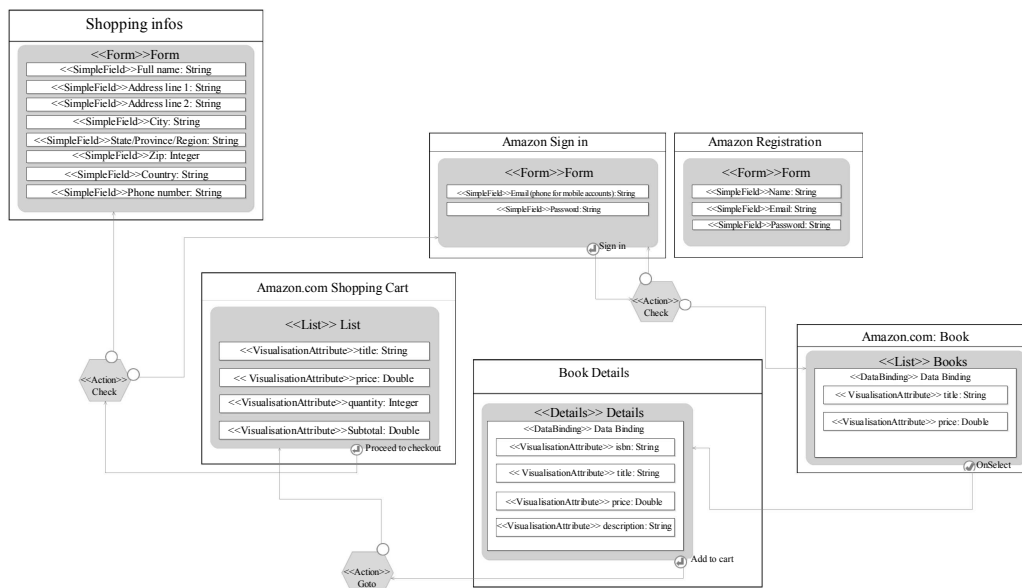


Fig. 4: IFML, Front-End Model for Amazon WebSite Scenario

Content Model: In this example, a customer creates an account for logging on the website. Then, the customer searches for a book in the website. The website lists relevant available books based on the customer’s search. The customer selects a target book from the list of the books, then orders the target book. The system requests the customer to add a credit card. Subsequently, the customer adds a credit card in order to buy the current book order, prior to being given a

permission to download the target book. The system then allows the customer to either leave the website or continue searching and shopping on the system. Figure 3 shows the Content Model for Amazon Book following IFML method by using IFML Editor Tool.

The important diagram is the IFML model capturing the whole scenario. By using IFML Editor as case tool, the IFML model is designed for Amazon book. IFML has a number of elements to design interaction models.

Additionally, IFML Editor also has the ability to design these models. Figure 4 presents the IFML front-end model for Amazon book by using IFML Editor Tool.

The IFML tools cited above can design this case study. The case study is designed by the IFML editor. All IFML tools can design the main concepts of the IFML method, but each tool has a special philosophy. We modeled this scenario with the four tools used for modeling IFML diagrams. The difference resides in the speed of each tool.

For the design of our case study, the IFML diagram must be created first for the definition of the UML model, which represents the domain model. All the tools mentioned except IFMLEdit.org starts with this procedure. Figure 3 shows the domain model designed for the Amazon Website Scenario proposed. The class diagram is composed of eight classes.

The core model is created after the creation of the domain model. It is created based on the domain model. Figure 4 depicts the creation of IFML components (container, view component, event and action).

We define all views for our scenario from the abstract representation of interfaces associated with the domain model.

These scenarios are common procedures in all the tools presented above. However, WebRatio goes further and extends the IFML components to provide users the ability to produce IFML diagrams for web or mobile applications. IFMLEdit.org also provides this feature.

## Conclusion and Future Work

Modeling tools are important in the development of applications, especially web and mobile applications. In the MDWE field, 18% of the contribution type is a tool (Wakil and Jawawi, 2015). The evaluation of modeling tools is difficult. The robustness of the research and industrial often determines whether the tool is useful and effective.

In this study, we analyzed and compared IFML tools that consist of WebRatio, IFML Editor, IFMLEdit.org, or MIA-studio. Each tool has its strengths and weaknesses.

Future applications can evolve rapidly and engineering methods for modeling need to extend and define new concerns. Thus, we recommend researchers to extend the IFML tools for the development of new features based on modeling languages.

## Author's Contributions

**Naziha Laaz:** Worked on most of parts, introduction, analysis data, result description, design case study.

**Karzan Wakil:** Literature review, criteria description and finding result.

**Samir Mbarki:** Clarifying abstraction, and planning.

**Dayang N.A. Jawawi:** Research methodology and reviewing final version.

## Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and there are no ethical issues involved.

## References

- Acerbis, R., A. Bongio, S. Butti and M. Brambilla, 2015a. Model-driven development of cross-platform mobile applications with WebRatio and IFML. Proceedings of the 2nd ACM International Conference on Mobile Software Engineering and Systems. May 16-17, IEEE Xplore Press, Florence, Italy, pp: 170-171.  
DOI: 10.1109/MobileSoft.2015.49
- Acerbis, R., A. Bongio, M. Brambilla and S. Butti, 2015b. Model-driven development based on omg's IFML with webratio web and mobile platform. Proceedings of the 5th International Conference on Web Engineering, Jun. 23-26, Rotterdam, The Netherlands, pp: 605-608.  
DOI: 10.1007/978-3-319-19890-3\_39
- Brambilla, M. and P. Fraternali, 2014a. Interaction Flow Modeling Language: Model-Driven UI Engineering of Web and Mobile Apps with IFML. 1st Edn., Morgan Kaufmann, ISBN-10: 0128001089, pp: 422.
- Brambilla, M. and P. Fraternali, 2014b. Large-scale model-driven engineering of web user interaction. The Webml and WebRatio experience. Sci. Comp. Progra., 89: 71-87.  
DOI: 10.1016/j.scico.2013.03.010
- Bernaschina, C., 2017. ALMOsT. Js, an agile model to model and model to text transformation framework. Proceedings of the International Conference on Web Engineering, (CWE' 17), Springer, Cham, pp: 79-97.  
DOI: 10.1007/978-3-319-60131-1\_5
- Bernaschina, C., M. Brambilla, A. Mauri and E. Umuhoza, 2017a. A big data analysis framework for model-based web user behavior analytics. Proceedings of the International Conference on Web Engineering, (CWE' 17), pp: 98-114.
- Bernaschina, C., S. Comai and P. Fraternali, 2017b. IFMLEdit.org: Model driven rapid prototyping of mobile apps. Proceedings of the IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems, May 22-23, IEEE Xplore Press, Buenos Aires, Argentina, pp: 207-208.  
DOI: 10.1109/MOBILESoft.2017.15
- Bernaschina, C., S. Comai and P. Fraternali, 2017c. Online model editing, simulation and code generation for web and mobile applications. Proceedings of the IEEE/ACM 9th International Workshop on Modelling in Software Engineering, May 21-22, IEEE Xplore Press, Buenos Aires, Argentina, pp: 33-39. DOI: 10.1109/MiSE.2017.1

- Bernaschina, C., S. Comai and P. Fraternali, 2018. Formal semantics of OMG's Interaction Flow Modeling Language (IFML) for mobile and rich-client application model driven development. *J. Syst. Sof.*, 137: 239-260.  
DOI: 10.1016/j.jss.2017.11.067
- Brambilla, M., 2013. Interaction Flow Modeling Language (IFML). Standardization Specification Ptc.
- Brambilla, M., 2015. IFML, building the front-end of web and mobile applications with OMG's interaction flow modeling language. *Web Engineering*.
- Brambilla, M., A. Mauri and E. Umhuza, 2014. Extending the Interaction Flow Modeling Language (IFML) for model driven development of mobile applications front end. *Proceedings of the International Conference on Mobile Web and Information Systems, (WIS' 14)*, Springer, Cham, pp: 176-191. DOI: 10.1007/978-3-319-10359-4\_15
- Cabot, J., 2018. Towards a UML and IFML mapping to GraphQL. *Proceedings of the International Conference on Current Trends in Web Engineering, (TWE' 18)*, Rome, Italy, pp: 149.
- Carlo, B., C. Sara and F. Piero, 2017. FMLEdit. Org, a web tool for model based rapid prototyping of web and mobile applications. *Proceedings of the International Conference on Mobile Software Engineering and Systems, Mobilesoft, (ESM' 17)*, pp: 21-31.
- CarloBernaschina, 2018. IFMLEdit.org.
- Ed-Douibi, H. and H. Brunelière, 2015. Modeling software application front-ends, introducing the open source IFML graphical editor. In *EclipseCon North America -Modeling Symposium*.
- Falzone, E. and C. Bernaschina, 2018. Model based rapid prototyping and evolution of web application. *Proceedings of the International Conference on Web Engineering, (CWE' 18)*, pp: 496-500.
- Gotti, Z. and S. Mbarki, 2016a. Java swing modernization approach-complete abstract representation based on static and dynamic analysis. *Proceedings of the 11th International Conference on Software Engineering and Applications, (SEA' 16)*, Science and Technology Publications, pp: 210-219.  
DOI: 10.5220/0005986002100219
- Gotti, S. and S. Mbarki, 2016b. Toward IFVM virtual machine: A model driven IFML interpretation. *Proceedings of the 11th International Conference on Software Engineering and Applications, (SEA' 16)*, Science and Technology Publications, pp: 220-225.
- Hamdani, M., W.H. Butt, M.W. Anwar and F. Azam, 2018. A systematic literature review on Interaction Flow Modeling Language (IFML). *Proceedings of the 2nd International Conference on Management Engineering, Software Engineering and Service Sciences*, Jan. 13-15, ACM, Wuhan, China, pp: 134-138.  
DOI: 10.1145/3180374.3181333
- IFMLEditor, 2018. The open source IFML editor for Eclipse.
- Laaz, N. and S. Mbarki, 2016a. A model-driven approach for generating RIA interfaces using IFML and ontologies. *Proceedings of the 4th IEEE International Colloquium on Information Science and Technology*, Oct. 24-26, IEEE Xplore Press, Tangier, Morocco, pp: 83-88.  
DOI: 10.1109/CIST.2016.7805005
- Laaz, N. and S. Mbarki, 2016b. Integrating IFML models and owl ontologies to derive UIs web-Apps. *Proceedings of the International Conference on Information Technology for Organizations Development*, Mar. 30-Apr. 1, IEEE Xplore Press, Fez, Morocco, pp: 1-6.  
DOI: 10.1109/IT4OD.2016.7479284
- Mia-Software, 2018. Mia-Software.
- Mia-Studio, 2018. Mia-Studio development booster for Android native applications.
- OMG-IFML, 2018. "OMG-IFML".
- OMG-UML, 2018. OMG-UML.
- Rhazali, Y., Y. Hadi and A. Mouloudi, 2016. A model transformation in MDA from CIM to PIM represented by web models through SoaML and IFML. *Proceedings of the 4th IEEE International Colloquium on Information Science and Technology*, Oct. 24-26, IEEE Xplore Press, Tangier, Morocco, pp: 116-121. DOI: 10.1109/CIST.2016.7805027
- Rodríguez-Echeverría, R., J.M. Conejero, J.C. Preciado and F. Sánchez-Figueroa, 2016. AutoCRUD-automating IFML specification of CRUD operations. *Proceedings of the 12th International Conference on Web Information Systems and Technologies, (IST' 16)*, Rome, Italy, pp: 307-314.
- Rossi, G., M. Urbietta, D. Distanto, J.M. Rivero and S. Firmenich, 2016. 25 Years of model-driven web engineering: We achieved, what is missing. *CLEI Electronic J.*, 19: 5-57. DOI: 10.19153/cleiej.19.3.1
- Roubi, S., M. Erramdani and S. Mbarki, 2016. A model driven approach based on interaction flow modeling language to generate rich internet applications. *Int. J. Elec. Comput. Eng.*, 6: 3073-3079.  
DOI: 10.11591/ijece.v6i6.pp3073-3079
- Sachdeva, R.K. and R.K. Singh, 2017. Extending support of IFML to RIA constructs.
- Siau, K. and M. Rossi, 2011. Evaluation techniques for systems analysis and design modelling methods-a review and comparative analysis. *Inform. Syst. J.*, 21: 249-268. DOI: 10.1111/j.1365-2575.2007.00255.x
- Toghrolri, A., M. Suhatriil, Z. Ibrahim, M. Safa and M. Shariati *et al.*, 2016. Potential of soft computing approach for evaluating the factors affecting the capacity of steel-concrete composite beam. *J. Int. Manufac.* DOI: 10.1007/s10845-016-1217-y

- Umuhoza, E., M. Brambilla, J. Cabot and A. Bongio, 2015. Automatic code generation for cross-platform, multi-device mobile apps, some reflections from an industrial experience. Proceedings of the 3rd International Workshop on Mobile Development Lifecycle, Oct. 26-26, ACM, Pittsburgh, PA, USA, pp: 37-44. DOI: 10.1145/2846661.2846666
- Wakil, K. and D.N. Jawawi, 2015. Model driven web engineering, a systematic mapping study. *Einför. Soft. Eng. J.*, 9: 107-142. DOI: 10.5277/E-INF150106
- Wakil, K. and D.N. Jawawi, 2014. Metamodels evaluation of web engineering methodologies to develop web applications. *Int. J. Soft Eng. Applic.*, 5: 47-63. DOI: 10.5121/ijsea.2014.5504
- Wakil, K., D.N. Jawawi and A. Safi, 2015. A comparison of navigation model between UWE and WebML: Homepage development case study. *Int. J. Inform. Edu. Technol.*, 5: 650-655. DOI: 10.7763/IJJET.2015.V5.585
- Wakil, K. and D.N. Jawawi, 2017a. Comparison between web engineering methods to develop multi web applications. *J. Soft*, 12: 783-794. DOI: 10.17706/jsw.12.10.783-793
- Wakil, K. and D.N. Jawawi, 2017b. Extensibility interaction flow modeling language metamodels to develop new web application concerns. *Kurdistan J. Applied Res.*, 2: 172-177. DOI: 10.24017/science.2017.3.23
- Wakil, K. and D. Jawawi, 2017c. Analyzing interaction flow modeling language in web development lifecycle. *Int. J. Adv. Comput. Sci. Applic.*, 8: 286-293. DOI: 10.14569/IJACSA.2017.080137
- WebRatio, 2018. My Webratio.
- WebRatio-Platform, 2018. WebRatio-Platform.
- Yigitbas, E., B. Mohrmann and S. Sauer, 2015. Model-driven UI development integrating HCI patterns.