Using IFML for user interface modeling

Randerson Queiroz, Anna Beatriz Marques, Tayana Conte
{rsq, tayana}@icomp.ufam.edu.br
{beatriz.marques}@ufc.br

USES Research Group
Federal University of Amazonas (UFAM)
Institute of Computing (IComp)
Manaus – AM, 69077-000

Federal University of Ceará (UFC)
Campus Russas
Russas – CE, 62900-000

USES Technical Report
TR-USES-2018-005
May 2018

Institute of Computing (IComp)
Federal University of Amazonas (UFAM)
Manaus, Amazonas 69077-000
URL: http://uses.icomp.ufam.edu.br/relatorios-tecnicos/

Using IFML for user interface modeling

[1]Randerson Queiroz, [2]Anna Beatriz Marques, [1]Tayana Conte
{rsq, tayana}@icomp.ufam.edu.br
{beatriz.marques}@ufc.br

[1]USES – Usability and Software Engineering Group
[1] Federal University of Amazonas – UFAM
[1]Manaus AM, 69077-000

[2]Federal University of Ceará – UFC
[2]Russas, CE

May 2018

Abstract

This technical report presents the artifacts used in the empirical study we conducted about IFML for interface modeling. The study aims to analyze the quality of the models in terms of correctness and completeness. We defined the following research questions: Can the user interface be modeled in a complete way using IFML in traditional software development? What is the correctness of the IFML diagrams created to represent the user interface?

SCENARIOS USED BY GROUPS A AND B FOR MODELING TASKS WITH IFML NOTATION

The subjects used functional requirements of web systems as the basis for modeling the interface. The requirements were described as scenarios. The scenario that group A received described a system of an airline company, that could be used to track flights to its destination. The scenario that group B received described a website that provided tips about restaurants by area of the city. Both scenarios had the same number of requirements (four requirements).

SCENARIO 1

The scenario of group A contained the following requirements: 1) to access the system with login and password; 2) to track previously registered flights; 3) to register flights to be tracked; and 4) to configure notifications with flight route updates.

Integral scenario: Carol would like to follow the flight path of her mother, who will make a long international trip. She is worried about the trip and she wants to follow the flight route. Thus, she decides to use an airline system that allows her to register a flight to follow its route

to the destination. By accessing the system with login and password, Carol can follow the previously registered flights for tracking. If she wishes, she can register a new flight for tracking by informing the flight number, the city that the flight departures, the city that the flight will arrive and a local made available in the purchase of the ticket. Whenever she wishes, Carol can register or remove one or more flights to be tracked. By saving the data, the system updates the flights registered for tracking by Carol. The, she can choose a flight to follow the route updates. The route update contains the flight number, departure time, arrival forecast, flight locator, airline name, description of the update with the date and time of the last update. Carol can choose to receive notifications on her cell phone with each update of the flight path. The notifications can be sent when each scale or connection occur. Whether she prefers, Carol can set a time interval to receive updates. To enable notifications, she needs to inform her mobile number. If this option is already enabled, Carol can also turn off notifications.

SCENARIO 2

The scenario of group B contained the following requirements: 1) to search for restaurant per area; 2) to bookmark a chosen restaurant as favorite; 3) to view tips about the park nest to a favorite restaurant; and 4) to confirm a restaurant booking.

Integral scenario: Felipe would like to eat in restaurants that are not in the downtown. However, he knows few options and he was interested in knowing a site that provides tips about restaurants in the city. When he opens the site and chooses a city zone, he visualizes a list of restaurants, its location, images of the place and an option for making a reservation. Felipe can change the city zone in order to see other restaurants. When he chooses a restaurant to make the reservation, Felipe needs to inform his name, his phone number, the time and date for reservation and the number of people. After the data validation, Felipe visualizes the reservation confirmation with the details about the local and phone number of the restaurant. Felipe can cancel the reservation or make a new reservation for other restaurants. By clicking on the restaurant that is already booked, Felipe can see parking options that are located until 1km to the restaurant. Each parking has a location, a estimated price by hour and images.

ORACLES USED TO AID IN THE ANALYSIS OF THE COMPLETENESS AND CORRECTNESS OF THE MODELS

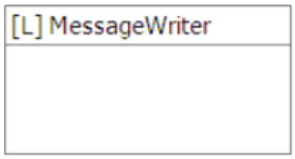
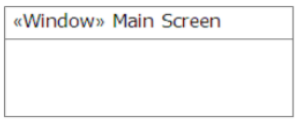
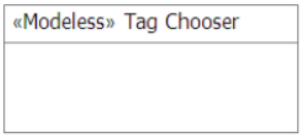
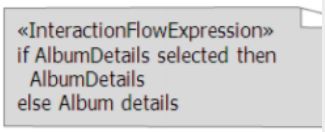
To achieve the completeness and correctness of each model, we elaborated oracles in order to support the analysis. The oracle corresponds to a possible solution for the scenarios modeling and defines a set of IFML elements that can be used in the solution. In the analysis, we used the oracles as basis for analyzing the elements used by the subjects, the elements not used and the elements that they could use in the model. Each oracle has specific requirements for each scenario. For each requirement, we listed which elements were necessary to model the front-end related to the requirement described.

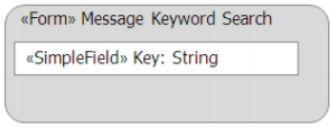
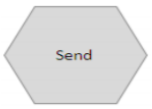
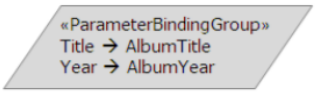
Group A – Scenario A	
Access System	Flight Tracker
ViewContainer	ViewComponent List
ViewComponent Form	Select event
Submit event	Action
Type of Data	ViewComponent Details
Action	Type of Data
Parameter Binding	Parameter Binding
	ViewContainer
Register flight	Configure Notification
ViewContainer	Action
ViewComponent Form	ViewComponent Form
Subimit event	Select event
Parameter Binding	Subimit event
Type of Data	ViewContainer

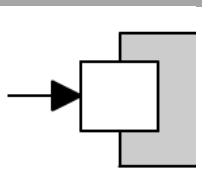
Group B – Scenario B	
Make reservation	Change zone
Select event	ViewContainer
Action	ViewComponent List
Parameter Binding	Select event
ViewComponent Form	Type of Data
Type of Data	
Subimit event	
ViewContainer	
Reserved restaurants and parking	View confirmation
ViewComponent List	ViewContainer
Select event	ViewComponent Details
Action	Action
ViewComponent Form List	Parameter Binding
Type of Data	Type of Data
ViewContainer	

GUIDE WITH THE ELEMENTS OF THE INTERACTION FLOW MODELING LANGUAGE (IFML)

In order to assist the subjects during modeling, we developed a guide of IFML elements.

Name	Concept	Notation
ViewContainer	An element of the interface that comprises elements displaying content and supporting interaction and/or other ViewContainers.	
XOR ViewContainer	A ViewContainer comprising child ViewContainers that are displayed alternatively.	
Default View Container	A ViewContainer that will be presented by default to the user, when its enclosing container is accessed.	
Landmark View Container	A ViewContainer that is reachable from any other element of the user interface without having explicit incoming InteractionFlows.	
Window	A ViewContainer rendered as a window.	
Modal Window	A ViewContainer rendered in a new window that, when displayed, blocks interaction in all other previously active containers.	
Modeless Window	A ViewContainer rendered in a new window, that when displayed, is overlaid all other previously active containers, which remain active.	
Activation Expression	A boolean expression associated with a ViewElement, ViewComponentPart or Event: if true the element is enabled.	
Interaction Flow Expression	It determines which of the InteractionFlows are going to be followed as consequence of the occurrence of an Event.	

Name	Concept	Notation
List	A ViewComponent used to display a list of DataBinding instances.	
Form	A ViewComponent used to display a form that is composed of Fields.	
Details	A ViewComponent used to display details of a specific DataBinding instance.	
Event	An event that affects the state of the application.	
Action	A piece of business logic triggered by an event; it can be server side (the default) or client-side, denoted as [Client].	
	It indicates only the continuation of Interaction.	
Navigation Flow	It indicates an update of the interface elements in view or triggering of an action caused by the occurrence of an event. Data may be associated with the flow through parameter bindings.	
Data Flow	It indicates a data passing between ViewComponents or Action as consequence of a previous user interaction	
Select Event	An event denoting the selection of a single item of the user interface.	
Submit Event	An event that triggers a parameter passing between interaction flow elements	
Parameter Binding	A set of ParameterBindings associated to an InteractionFlow (being it navigation or data flow).	
Module	A piece of user interface and its corresponding actions, which may be reused for improving IFML models maintainability.	

Name	Concept	Notation
Input Port	An interaction point between a Module and its environment that collects InteractionFlows and parameters arriving at the module.	
Output Port	An interaction point between the Module and its environment that collects the InteractionFlows and parameters going out from the module.