

# InterAct: A Framework to Generate Device Specific Smart User Interfaces

H. Nawarathna, P. De Alwis, S. Dilhara, S. Liyanage, S. Weerawarana  
Department of Computer Science and Engineering  
University of Moratuwa  
Moratuwa, Sri Lanka

*Abstract*—InterAct is a form based User Interfaces (UI) designing framework for non-technical users. It can be used to design UIs for multiple device platforms. The framework consists of two parts, an Integrated Development Environment (IDE) and a conversion library. The designed UIs are saved in an XML format type, called IXML. These user sketches are then converted into HTML pages, which can be displayed in various devices using the conversion library. Data is bound to the UI elements at runtime. Usability is the major concern of the InterAct Framework. This aspect is addressed by applying the best practices of Human Computer Interaction (HCI) in order to provide the user a better experience.

*Keywords* – HCI; device specific; WURFL; segmentation; conversion; data binding; IDE

## I. INTRODUCTION

InterAct is a form based User Interface designing framework, which can be used to design UIs for multiple device platforms. It is specially designed for the use of people such as business analysts, who do not have technical expertise. The framework mainly consists of two parts, an IDE and a conversion library. A user can design forms by dragging and dropping the UI elements onto a canvas at design time using the IDE and the design is saved in an XML [1] format. Then it is converted into HTML [2] pages which can be displayed in various devices like a PC or a handheld device using the conversion library. Data is bound to the UI elements at the runtime.

In developing the InterAct framework, we mainly focused on developing two major components. One is the InterAct IDE, which is developed using the Eclipse Graphical modeling framework (GMF) [3]. The IDE consists of a canvas and a pallet, which includes a set of UI controls. A designer can easily design the required UIs by dragging and dropping the controls in to the canvas from pallet. As the user designs the UI the relevant XML file, which describes the UI is auto-generated by the IDE. The preview function enables the user to see how his design is displayed in the target environment such as a web browser.

The final output of InterAct IDE is saved in an XML format according to the Interact XML (IXML) schema. We designed IXML as an intermediate format considering the weaknesses of existing abstract user representation languages such as XForms [4]. It captures the necessary details in the UI

design and it facilitates the conversion into HTML and adaptation to the target platform.

The conversion library is the other major component of the framework proposed in this paper. Its main task is to convert the intermediate XML file into HTML. When the user sends a request, it detects the user device from the header information and gets the necessary details such as screen size from a web service. Then the data within IXML is parsed in to HTML and divided in to several pages according to the target environment's constraints such as the screen size.

Usability of the system is the major concern. Various design considerations are taken into account to achieve the highest possible level of usability. The InterAct IDE is designed to minimize the learning time of the user. Unnecessary complexity is deliberately avoided to make it easy to be understood by users. The preview facility helps them see the interim results so they can do the necessary changes to create the UI required.

For the development of the IDE, we used the Eclipse Graphical modeling framework (GMF). GMF is a framework within the Eclipse platform. It provides a generative component and a runtime infrastructure for developing graphical editors based on the Eclipse Modeling Framework (EMF) [5], [6] and the Graphical Editing Framework (GEF) [7]. XSLT [8] (EXtensible Stylesheet Language Transformations) is used for the conversion library.

Initial version of the implementation only supports the following UI controls: Textbox, Label, Combo box, Checkbox and Radio button.

## II. METHODOLOGY

In this section, we describe the methodology that we have followed to develop this framework. We have mentioned the problem definition, research areas and the literature surveys that we did in order to arrive at the solution framework.

### A. Problem Definition

WSO2, an open source middleware company in Sri Lanka has developed a Business Process Server (BPS) [9]. Business processes are written in WS-BPEL [10] (Web Services Business Process Execution Language) and they can be deployed in BPS. A human has to make decisions in certain business processes and rest of the business flow depends on their decision. As a result, these business processes need human interaction to proceed. These human tasks are written

in the middle of a business process using WS-Human Task [11]. Human task invokes the WSDL (Web Services Definition Language) when a human interaction is needed for decision making.

A user interface (UI) is needed to get the human interaction to a business process. Currently, the UIs used in BPS are limited to a textbox, a label and a submit button. Also the UI is automatically generated using JSP [12] (Java Server Pages). These few components are not enough to capture complex user inputs. There is no validation in the current UI and it does not assist users to recover from errors. Due to those reasons, the generated UIs are not usable enough and those UIs are required to be redesigned having usability as the major concern.

Currently, the UIs are not specifically designed to adapt according to the device environment. InterAct handles this adaptation issue and provide the user a better experience.

## B. Related Work

### 1) Abstract User Representation Languages

There is an extensive body of literature devoted to the representation of UI in an abstract format, independent of the device platform [13].

We looked into the existing commonly used abstract user representation languages, which are mentioned below.

- UIML (User Interface Markup Language) [14]
- XIML (Extensible Interface Markup Language) [15]
- AIAP (Alternative Interface Access Protocol) [16]
- XForms [17]
- XUL (XML User Interface Language) [18]

These languages do not have the capability to segment the data about UI controls into several small mobile friendly HTML pages along with the facility to navigate between pages. All the abstract languages mentioned above need a server side renderer that does an extensive amount of processing to produce a concrete output.

We did not find an abstract language, which suited best for our requirements. Therefore, we decided to extract the important features such as the structure of abstract representation language, the UI controls to be added and developed our own XML schema IXML.

### 2) Existing Tools

There are many design tools available to draw form based UIs by dragging and dropping the UI controls.

Glade [19] is a Rapid Application Development tool to enable quick and easy development of user interfaces. The UIs designed in Glade are saved in XML format.

Orbeon [20] form builder is a web based form design tool which also helps to design forms by dragging and dropping the components into a canvas.

The main issue in the above mentioned designed tools is the user has to divide the canvas into sections before he/she drag and drop the components on to that. These tools are complex and include more than enough UI controls. These do not support quick design changes and they consume a considerable amount of learning time.

By considering the facts we experienced when using those tools we came with our own IDE; InterAct which does not need to partition the canvas before the design.

## C. Human Computer Interaction (HCI)

A basic goal of HCI is to enhance the usability and improve the interaction between users and computers by making computers more usable and receptive to the user's needs.

User Interface designing is a sub area of Human Computer Interaction [21]. The design should be centered on the user and his requirements. The golden rule in User Interface designing is "Know your user".

We gathered user requirements of target users of InterAct at the very beginning of the research in order to find the expectations and needs of the users. With this requirement analysis we could maximize the user satisfaction and so that the framework would not be subjected to the refusal. Following HCI aspects were considered in developing the system to achieve usability [22], [23].

### 1) Human Emotions

User should not be stressed by the UIs. He/she should be able to learn the system easily and do their task successfully. Therefore, the UI of the IDE should be kept simple.

Human are prone to errors. There are two types of errors in the study of Human Computer Interaction. Those are slips and mistakes. In slips, user knows what to do and how to do but he might do something other than the expected. The target should be enough in size to do the required operation. The controls are designed to facilitate dragging, dropping and clicks.

Users do mistakes when they do not know what to do. So we have added help pages and proper documentation to guide the user in order to reduce the learning time.

### 2) Display Devices

The designed UIs are displayed in various display devices like monitors, mobile screen and touch sensitive display. The resolution (pixel density) and aspect ratio between width and height should be considered before sending the UIs for a device. Conversion library/ renderer of InterAct will detect these details and generate the most suitable UI for the user. Colors, anti-aliasing and soften edges should be considered when developing the user interfaces.

### 3) Input Devices

As we are dealing with PCs and mobiles, we have to consider the input devices such as touch pads, track balls, cursor keys, mouse, touch screens and keyboards (QWERTY). Also, UIs should provide the facility to deal with all types of users. The beginners might be slow and after some learning,

users expect to do the same work in less time. So keyboard short cuts should be provided targeting those users. Default input data are provided with the final UI by data binding to facilitate the user in data input.

#### 4) *Selecting Colors*

The colors are not used to convey the messages because 8% of men and 1% of women are color blind and they are not able to understand the difference between some colors.

#### 5) *Aesthetic Pleasing*

Aesthetic pleasing can be achieved by adding only the required controls and the user preferred colors. The design facilitates the designer to select the colors for the end user.

#### 6) *Fonts*

Square ended strokes are good on screens, therefore, it is recommended to use san-serif in UIs. Lowercase enhance the readability. Uppercase is only be used for individual letters and non-words.

#### 7) *WYSIWYG - What You See Is What You Get*

The designed UIs run on sandbox environment and show previews first to give the designer a feeling about how it is displayed in the real environment. Android emulator is integrated to the InterAct IDE in order to support the mobile view. Relevant HTMLs for the requested devices will be selected and displayed in the mobile environment by having paging with navigation.

#### 8) *Input Validation*

The controls are filled with data using runtime data binding and which help to do the input validation.

### D. *Eclipse Graphical Modeling Framework (GMF)*

GMF provides a set of tools that makes it easy to utilize GEF (Graphical Editing Framework) and EMF (Eclipse Modeling Framework). It includes wizards for generating intermediary graphical, tools and mapping definitions based on an initial meta model (EMF ecore), as well as the final runtime code. InterAct IDE was developed using the GMF framework.

## III. RESULTS AND DISCUSSION

In this section we will discuss the output of the InterAct framework. As mentioned in this paper, the framework consists of two major components. They are the InterAct IDE and the conversion library.

### A. *InterAct IDE*

The InterAct IDE is developed on top of the Eclipse platform and it provides the facility to design the UIs and save them in IXML format. The IDE as in Fig. 1 has following features:

- Drag and drop ability - Users can select UI controls from the pallet and drag them to the canvas to design the UI.
- Keyboard short cuts - Keyboard shortcuts are provided for frequently used common functionalities.

- Preview of the design in required format (e.g.: Web browser)
- Outline of UI design - Components of the design are highlighted and displayed in a tree view.
- Zooming
- IXML of the design is dynamically generated as user changes his design.

### B. *IXML Schema*

When the objective is to have several output formats (for different device environments) for the same user interface design, having an intermediate format is preferable. IXML is a XML schema that is used as an intermediate format within InterAct framework. It is used to capture the information of UIs designed by the IDE.

UI elements of the design are grouped in to several sections to support the navigation between pages. The set of elements that should be displayed within a single page are grouped in a one section. As a result, if the device environment constraints such as screen size do not permit two groups to display within a one page, the second group will be displayed in the next page. It avoids displaying one and half groups in a single page which gives unacceptable results.

### C. *Converter/Renderer*

Converter plays an important role in the InterAct framework. This module is responsible for rendering the IXML into different device environments. As in Fig. 2 Converter takes the intermediate XML file generated by the IDE as an input and it produces the output file as XHTML/HTML. The output file is adapted to display in a particular device environment.

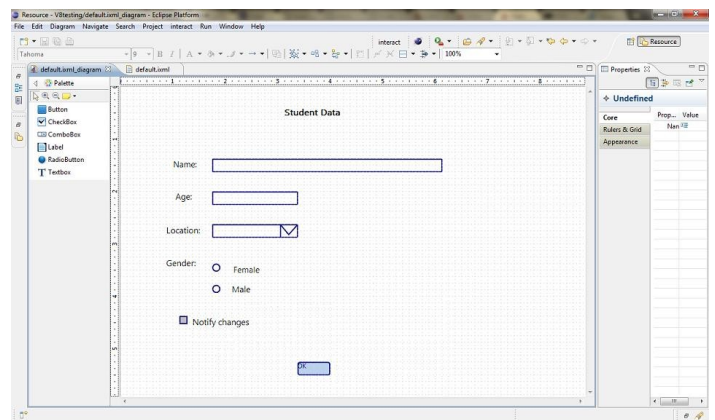


Figure 1. InterAct IDE.

After studying the existing technologies we found that XSLT and XPath are the most suitable technologies to develop the converter module. XSLT is used to develop all the styling rules for XML file. Saxon processor is used to automate the conversion process. We obtain an HTML file after processing the XML and the XSLT files through the Saxon processor.

Page segmentation or dividing the data in the output XML into several HTML pages is a function of the render module.

It decides which HTML to send according to the device request for the form. This method is to provide better user experience when the user is accessing the UIs through a device which has a smaller screen.

Wireless Universal Resource File (WURFL) [24] is used to get the features of devices which request the deployed form. It is an XML configuration file that contains up to date information of mobile devices. The WURFL file is frequently updated by adding the new mobile devices that are coming to the market. If we are to use the XML file for our framework, then it has to be changed every time the file gets updated. Therefore we decided to use the web service provided by the WURFL. Using the web service as in Fig. 3 reduces the overhead of changing the file frequently and it allows us to request for the device information by mentioning the user agent.

Finally we can say that InterAct framework will bridge the gap between the Abstract User Interface Representation languages and the user Interface designing. Therefore applying this framework can provide immense benefits in improving the HCI in user interfaces used in Web Services.

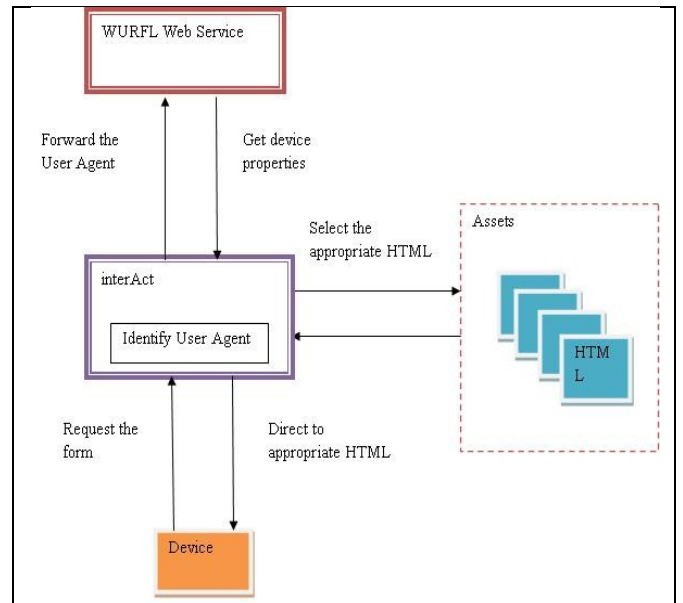


Figure 3. Direct to appropriate device specific HTML pages.

## REFERENCES

- [1] L.Quin. (2011, Apr.). *Extensible Markup Language*. [Online]. Available: <http://www.w3.org/XML/>
- [2] S. Pemberton. (2010, Dec.). *Hyper Text Markup Language*. [Online]. Available: <http://www.w3.org/MarkUp/>
- [3] M. Istria. (2011, Nov.). *Eclipse GMF Official Documentation*. [Online]. Available: [http://wiki.eclipse.org/GMF\\_Documentation](http://wiki.eclipse.org/GMF_Documentation)
- [4] J. Boyer. (2009). *XForms Specification*, Version 1.1. [Online]. Available: <http://www.w3.org/TR/xforms11/>
- [5] F. Budinsky, *Eclipse Modeling Framework: A Developer's Guide*. Addison-Wesley, 2003.
- [6] *Eclipse Modeling Framework Official Documentation*. (2011) [Online]. Available: <http://www.eclipse.org/modeling/emf/docs/>
- [7] *Eclipse Graphical Editing Framework Official Documentation*. (2011) [Online]. Available: <http://www.eclipse.org/gef/documentation.php>
- [8] J. Clark. (1999, Nov.). *EXtensible Stylesheet Language Transformations Specification*, Version 1.0. [Online]. Available: <http://www.w3.org/TR/xslt>
- [9] *WSO2 Business Process Server*. (2011). [Online]. Available: <http://wso2.com/products/business-process-server/>
- [10] *Web Services Business Process Execution Language*, Version 2.0. (2007, Apr.). [Online]. Available: <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>
- [11] *WS-Human Tasks Specification*, Version 1.1. (2007, Jun.). [Online]. Available: [http://incubator.apache.org/hise/WS-HumanTask\\_v1.pdf](http://incubator.apache.org/hise/WS-HumanTask_v1.pdf)
- [12] *JSP*. (2011). [Online]. Available: <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>
- [13] S. Trewin, G. Zimmermann, and G. Vanderheiden, "Abstract user interface representations: How well do they support universal access?" CUU '03 Proceedings of the 2003 Conference on Universal Usability, 2003.
- [14] *UIML*. (2006, Jul.). [Online]. Available: <http://uiml.org/specs/index.htm>
- [15] A. Puerta and J. Eisenstein. *XiML: A Universal Language for User Interfaces*. [Online]. Available: <http://www.ximl.org/documents/XimlWhitePaper.pdf>

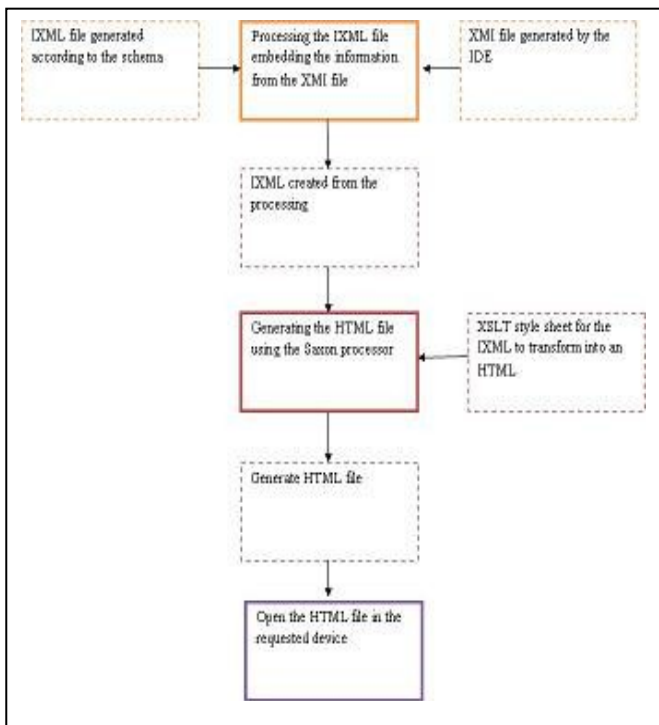


Figure 2. Page conversion.

[16]G. Vanderheiden. (2001, Jan.). Alternative Interface Access Protocols (AIAP) for Assistive Technology: Controlling Next Generation Technologies from Assistive Technologies . [Online]. Available:

[http://www.itl.nist.gov/iad/itaccess01/Presentations/Vanderheiden\\_Final\\_Presentation/tsld001.htm](http://www.itl.nist.gov/iad/itaccess01/Presentations/Vanderheiden_Final_Presentation/tsld001.htm)

[17]XForms Specification, Version 1.1. (2009, May). [Online]. Available:

<http://www.w3.org/MarkUp/Forms/2007/XForms-11-Schema.xsd>

[18]A. Mechelycnk. (2011, Jun.). XUL. [Online]. Available:

<https://developer.mozilla.org/En/XUL>

[19]J. Jardon. (2009, Sep.). Glade Tutorials. [Online]. Available:

<http://live.gnome.org/Glade/Tutorials>

[20]Orbeon Forms. (2011, Sep.). [Online]. Available:

<http://www.orbeon.com/>

[21]W. Galitz, The Essential Guide to User Interface Design, 3rd ed., Wiley Publishing Inc, 2007

[22]A. Beale and D. Finlay, Human Computer Interaction - 3rd Edition, Book Slides

[23]A. Blackwell, Human Computer Interaction, Part II Course 2004-2005, Notes, University of Cambridge Computer Laboratory

[24]WURFL Web Service. [Online]. Available: <http://wurflws.com/>