International Journal of Software Engineering and Its Applications
Vol. 9, No. 2 (2015), pp. 219-224
http://dx.doi.org/10.14257/ijseia.2015.9.2.19

A Context-based Framework Generating Personalized User Interface

[1]Jaewon Moon and [2]Dong-Hee Shin, Ph.D

[1]9FL, Electronics Center, #11, World cup buk-ro 54-gil, Mapo-gu, Seoul, Korea
bunnyjw@gmail.com
[2]Dept. of Interaction Science, Sungkyunkwan University 90326, International Hall, 53 Myeongyun-dong 3-ga, Jongno-gu, Seoul, Korea

Abstract

With recent advances in the technology of big data, many service providers have been trying to derive meaningful contexts from such data. Context-aware services can offer high quality service from the consumer's perspective, since most services are used in varying contexts and environments. The results from many studies on adaptive user interface have shown that context-aware user interfaces can be faster and preferable to non-adaptive counterparts. In this paper, we propose a novel framework providing a context adaptive user interface which automatically chooses appropriate Resources from a UI Page-Set consisting of multiple Resources. The proposed framework aimed to provide service directors who are not knowledgeable about UI development with opportunities for making context-aware UI, and to achieve high-quality user experience in complex services.

Keywords*: Context, Context-Awareness, Adaptive User Interface*

1 Introduction

Advances in the processing technology for big data, private data collected and minded by social services and private mobile devices, and public data provided by the government and public institutions, have enabled many service providers to improve the quality of their service. Many service providers have tried to derive meaningful contexts from big data and use them to make better services. Personalized services customized to the user's environment have become highlighted as a key area of user-context computing. It is essential for context-aware technology to be developed into more intelligent and meaningful services through wide application to a variety of sectors and domains.

To provide personalized service, where, when, and in what circumstances it is necessary to adapt the particular context of users should be considered [1]. Context is any information that can be used to characterize the situation of an entity that is considered relevant to the interaction between users and services. In this sense, a context-aware service is considered as a smart web service that can understand the situational context and share that context with other services [2]. High demand for the improvement of user experience has led to the development of context awareness and personalized service.

As mobile devices have also continued to evolve and spread, the design and development of user interfaces (UI) compatible with a diverse range of devices is required. Web UI, which refers to any interaction between a user and an application running on a web server, is a good solution to provide user interface without considering the platform in comparison to native user interface. Responsive Web Design is the general key approach suggesting that web user interface should respond to various characteristics such as the resolution and orientation of devices [3]. It is a combination of fluid grids and

ISSN: 1738-9984 IJSEIA

responsible contents with the media queries of CSS to change layout based on the many characteristics of a device. Web UI should automatically change to adapt for resolution, image size and device abilities based on Responsive Web Design techniques. It allows users to access a single URL for an adaptable content, without taking the trouble of having to type different URLs depending on the device used, such as mobile, tablet, desktop, *etc.*, However, client-side adaptation not only requires a careful approach, but can also face difficulties in optimization of source order, contents, application design and other user contexts within only client-side adaptations such as Responsive web design. In the field of Web technology, the use of Context Management Systems (CMS) that are information delivery systems, including web-based systems that organize content, is becoming increasingly common. CMS such as Wordpress [4], Drupal [5] and Zoomla [6] are widely used by many users as effective web management tools. In addition, the technology of web development tools which help to create UI, such as Dreamweaver and Aptana, is improving steadily. However, a common problem for these systems and softwares is that they are only fit for the generation of general web solutions.

In addition, the results from studies on adaptive user interface have suggested that an adaptive interface can reduce search time, cognitive load, and motor movement. Studies have shown that when the adaptation is successful, an adaptive interface can be faster, and is preferred over non-adaptive counterparts [7-8]. Gajos, Webbrock, and Weld's SUPPLE system [9] adjust the layout, size, and structure of interface elements to accommodate both device and user characteristics. Adaptive graphical user interfaces have the potential to improve performance and user satisfaction by automatically tailoring the presentation of functionality to each individual user.

We propose the structure for a novel context-based framework generating personalized user interface. This framework collects unit Resources matching specific conditions, according to user contexts. In so doing, it improves adaptive device features in real-time. By automatically generating adaptive user interfaces tailored to an individual's contexts, the proposed framework aims to achieve high-quality user experience for complex service.

2 Proposed Framework

3 Framework Introduction

The proposed framework, which automatically generates adaptable user interface according to contexts, aims to achieve high-quality user experience in real time. First, the definitions of terms related to this framework need to be examined.

- **Page** is defined as a whole UI representing service itself, and the personalized user interface rendered by this framework. Page can choose only one of the design layouts predefined by the web designer.
- Each **layout** consists of more than one type of layer. The main design depends upon the characteristics of a specific layout.
- **Layer** determines how the Resources look. We defined 6 layer types for this framework.
- **Resource** is the smallest unit of a user interface. Each Resource can contain significant information such as design, function and content properties. In addition, Resources can contain descriptions and other basic information. It may be defined as the smallest unit that can be reused by this framework to make various UI based on the setup properties of each Page.

As a result, a Page is comprised of one or more Resources. The primary contribution of this framework is that the final user interface may be generated according to the real-time contexts of users. Each Resource in a Page is connected with a specific context, and the final UI contains filtered Resources from the Page based on user environment. Various contents with specific types and properties can be defined by Resources. This framework uses webpage, mark-up language (HTML, XML), multimedia content (video, image,

audio), design content (CSS, Layout) and function (JavaScript) as Resources.

It is easy to make an adaptive user interface with the proposed framework. There is no need to understand programming languages used for authoring user interface. Pages can be generated not from general programming, but from connections between contexts and Resources such that service providers without extensive knowledge of high-level technology can easily generate UI with the proposed authoring module.

4 Framework Architecture

The proposed framework is divided into two main frameworks. The first main framework is the ROF (Responsive UI Organizing Framework), which contains the page authoring module. The ROF figures out the message from the RPF and generates the proper user interface. It has roughly two functions: the authoring module which makes Pages through connections between Resources and specific values of the context, and the generating adaptive UI module which collects significant Resources according to the real-time contexts of the user. The second framework is the RPF (Responsive UI Framework), which handles all of the processes for service requests from the user and interprets user context. Figure 1 shows the flow of the proposed system.

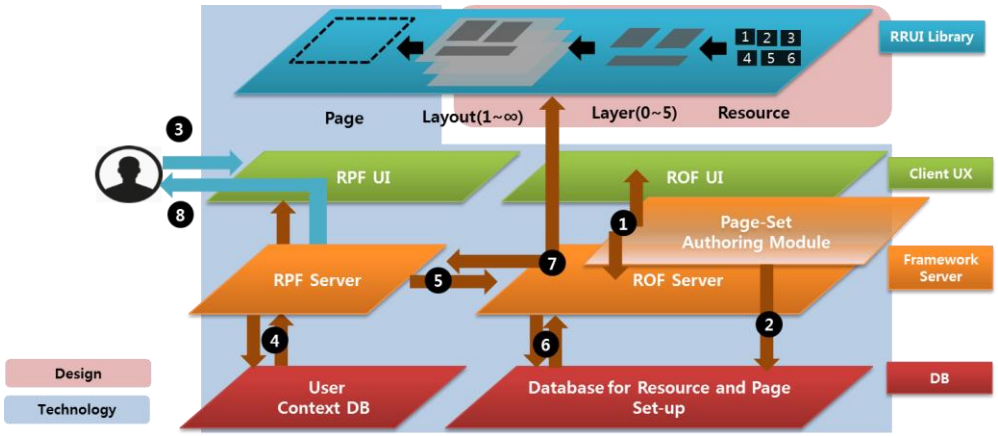

Figure 1. Flow of the Proposed Framework

1) Content provider registers various contents as Resources using the authoring module in the ROF framework. Resources can include webpages, mark-up languages, media content, design content and function modules. The service provider also sets up and makes Pages correlated to the context and its value using registered resources through authoring module.
2) The database for Resources and Page-Set stores the information by content and service provider generated through authoring module (1).
3) A user logs into the RPF Server and tries to access a specific Page.
4) The RPF Server queries user context using the user context DB.
5) The RPF Server requests the ROF server for an adaptive user interface with the value of user context and page address information.
6) The ROF Server collects the appropriate Resources according to user contexts in real time.
7) The ROF Server generates pages based on the filtered Resources and information of Page, Layout and Layer.
8) Finally, RPF transfers the final UI to the user from ROF server.

5 User Interface Organizing Process

In this section, the main organizing processes generating the UI are introduced. To make interpretable Pages with the proposed framework, an authoring module which can

organize the Page for the UI needs to be used. Setting up the Page with context values and Resources must take precedence in order for the generation of the final UI. Therefore, a contents provider needs to register their contents as Resources. Resources can contain information such as basic profile (ID, name, description), contents, related data (tag, property), size, provider name and type.

Next, the service provider generates the UI Page-Set using links between Resources and contexts. Figure 2 shows the results of a UI Page-Set with the authoring module. The service provider needs to make new UI Page-Sets with unique id numbers and to define the contexts including related values used in the UI. The service provider can then select the values of the context and appropriate Resources.

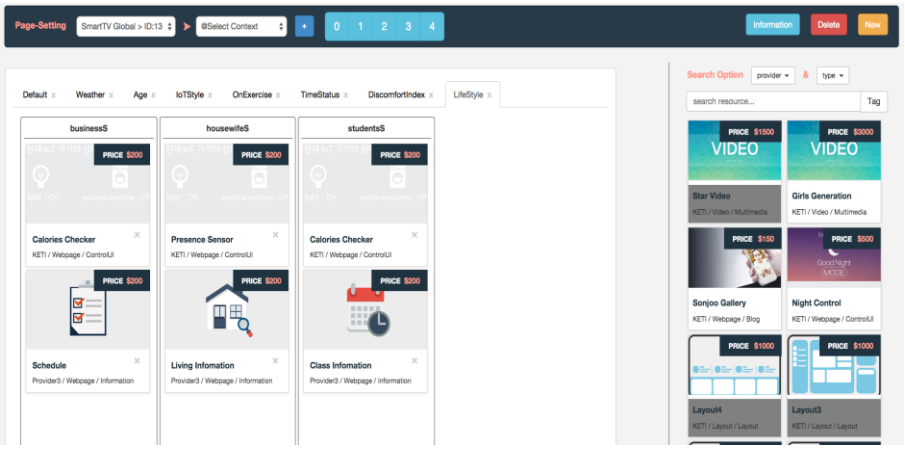

Figure 2. UI Page-Set with Authoring Module

6 Implementation

A UI Page-Set was made with the novel authoring module, and an adaptive user interface was generated to evaluate the proposed framework by using it as input according to user contexts. Contents providers can independently register their own content as Resources on the proposed cloud authoring system. Resources can be CSS influencing the design of the UI, JavaScript controlling UI function and IOT devices, webpage-Link presenting independent web contents and fragments of HTML. The structure of sample Page, ID:13, is shown in Figure 3. This page had 8 different contexts with respective values: Age (7 values), Discomfort Index (11 values), Perceived IOT Style (2 values), Perceived LifeStyle (3 values), Presence of exercise (2 values), Time (3 values) and Weather (4 values) information and default contexts. Each value of the contexts in the UI Page-Set had its own Resources. For evaluation of the proposed framework, it was assumed that there were 5 users in different circumstances: User1, User2, User3, User4, and User5. Table 1 shows a comparison of the individual context value of the users.

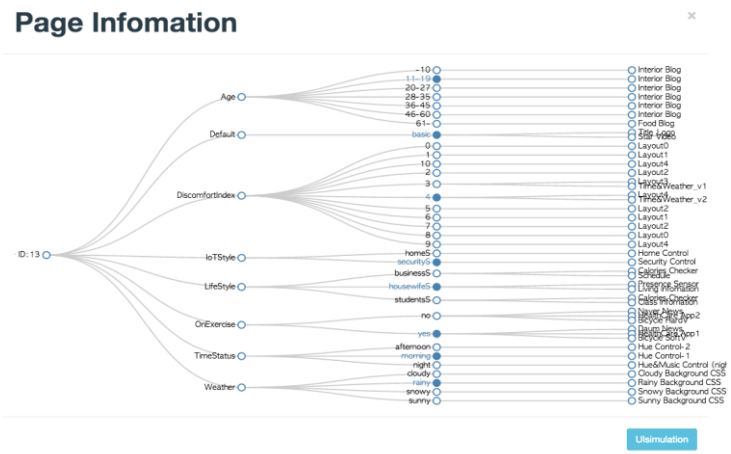

Figure 3. An Example Structure of UI-Set Page, ID:13

Table 1. Comparison of Individual Context Values of Users

Context\User	User1	User2	User3	User4	User5
Age	28-35	61-	11-19	36-45	46-60
Discomfort Index	Level 1	Level 2	Level 3	Level 4	Level 5
IOT Style	Home IOT Style	Security IOT Style	Home IOT Style	Security IOT Style	Home IOT Style
Life Style	Business Style	Housewife Style	Student Style	Business Style	Housewife Style
Exercise	No	Yes	NO	Yes	Yes
Time	Morning	Afternoon	Night	Morning	Afternoon
Weather	Sunny	Rainy	Cloudy	Snowy	Sunny

Three different UI Page-Sets were created to test the framework under various conditions. The final UI Page was successfully generated through the framework using the 3 UI Page-Sets and the 5 conditions of the different users. The results of the final UI for all users using the ID:13 Page-Set, one of the 3 UI Page-Sets, are shown in Figure 4. The proposed system was confirmed to provide various user interfaces for each user and case.

Figure 4. Results of Simulation for Each User

7 Conclusion

In this paper, a novel approach for the generation of user interface was introduced. This was achieved by registering unit Resources and mapping them with the context in the UI Page-Set. The demo showed cases of implementing the framework for several individual users. It allowed the designer or service director to make the UI for different cases without requiring expert knowledge. Furthermore, any contents such as CSS, JavaScript, HTML, Web-page and XML can be included as Resources, so that the approach is available for web platforms and applications running in a web browser. The next step is to build a more generalized framework in the field and find a way to implement intuitive UI and UX design principles. In addition, we plan to improve this system through careful consideration of complex conditions so that it can support complex UI like IOT (Internet of Things).

Acknowledgement

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2014S1A5B1014964)

References

[1]. G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith and P. Steggles, Towards a better understanding of context and context-awareness. In Handheld and ubiquitous computing, Springer Berlin Heidelberg, **(1999)**.
[2]. A. T. Manes, "Enabling open, interoperable, and smart web services-the need for shared context", In Proceedings of The W3C Workshop on Web services, **(2001)**.
[3]. B. Frain, "Responsive web design with HTML5 and CSS3", Packt Publishing Ltd., **(2012).**
[4]. "Wordpress", http://www.wordpress.com.
[5]. "Drupal", http://www.drupal.org.
[6]. "Zoomla", http://www.zoomla.com.
[7]. K. Z. Gajos, M. Czerwinski, D. S. Tan and D. S. Weld, "Exploring the design space for adaptive graphical user interfaces", In Proceedings of the working conference on Advanced visual interfaces, ACM, **(2006)**, pp. 201-208.
[8]. L. Findlater ans J. McGrenere, "Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces", In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, **(2008)**, pp. 1247-1256.
[9]. K. Z. Gajos and D. S. Weld, "SUPPLE: automatically generating user interfaces", In Proceedings of the 9th international conference on Intelligent user interfaces, ACM, **(2004)**, pp. 93-100.
[10]. T. M. T. Ma, J. H. J. Huang and G. W. G. Wu, "An architecture and supporting environment of service-oriented computing based-on context awareness", International Journal of Grid and Distributed Computing, vol. 2, **(2009),** pp. 1-8.
[11]. P. Chen, J. Gu, R. Tan and Z. Zhou, "An Open Web Service based Multi-Agent Framework for Context Aware Computing", International Journal of Grid & Distributed Computing, vol. 5, **(2012)**.
[12]. H. Lee and J. Kwon, "Combining context-awareness with wearable computing for emotion-based contents service", International Journal of Advanced Science and Technology, vol. 22, **(2010)**, pp. 13-24.

