MMW-4S: a model-based approach for generating context-aware user interfaces for the mobile Web

Javier R. Escolar, Cristina G. Cachón, Ignacio Marín, Nicanor Gutiérrez
Fundación CTIC Centro Tecnológico
Parque Científico y Tecnológico de Gijón
c/ Ada Byron, 39 Edificio Centros Tecnológicos
33203 Gijón - Asturias - España
{javier.rodriguez, cristina.cachon, ignacio.marin, nicanor.gutierrez}@fundacionctic.org

ABSTRACT
The creation of mobile web User Interfaces (UIs) implies great difficulties for developers. They need not only to address device and browser fragmentation, but also to cope with the variety of circumstances in which users may interact with their applications (delivery context). In order to reduce the development efforts in such a complex scenario, the scientific community has been studying the possibility to apply model-based techniques to the creation of context-aware UIs. Although this approach has proved to be promising, there is still a reduced amount of software tools covering the complete development lifecycle. In this article we introduce MMW-4S, a Runtime UI Generation Engine (RUIGE) fully compliant with the Serenoa reference framework.

Author Keywords
Mobile web; context-awareness; model-based.

ACM Classification Keywords
D.2 Software Engineering: Design tools and techniques. H5 Information interfaces and presentation: Group and organization interfaces.

General Terms
Design; Languages.

INTRODUCTION
During the last years, the usage of Internet has experienced a dizzying growth and, accordingly, Web technologies have evolved at a rapid pace. Nowadays, users commonly access the Web from a diverse set of mobile devices, such as smartphones, tablets, book readers or even personal gaming consoles among others. Moreover, they are increasingly demanding rich applications able to tailor the UI in accordance to their specific context: location, time, user preferences, environment conditions, device capabilities, network status and so on.

From the developer perspective, programming mobile Web applications poses great challenges. Firstly, they need to cope with device heterogeneity: different screen sizes, multiple text input mechanisms, diversity in memory availability and processing capabilities, etc. Secondly, they need to deal with browser fragmentation. In spite of the fact that standards are conceived to reduce interoperability issues among systems, not all browsers are able to offer the same level of support for the incoming standards, thus causing several inconveniences for developers. Finally, developers need to handle context-awareness in order to maximize the user experience. In this regard, the Device APIs Working Group [1] is actively working in the creation of client-side APIs that will enable the development of context-aware Web applications by extracting information from device sensors, such as location, battery level, network status, vibration level, proximity events, ambient light, temperature, humidity, atmospheric pressure, etc.

A promising research approach to mitigate the complexity of the development of UIs is the Model Based User Interface (MBUI) design methodology. In this article we introduce MMW-4S, a RUIGE that is fully compliant with the Serenoa reference framework [2] and aims at facilitating the creation of context-aware UIs for the mobile Web. In order to show the applicability of our solution and how it handles context-awareness, we present four adaptation use cases where the corresponding UIs are able to react to context changes.

RELATED WORK
One of the most well-known frameworks that defines the fundamentals model-based UI design is *CAMELEON* [3]. This framework structures the development life cycle into four abstraction levels, from task specification to the finally UIs. The *Task and Concepts* level considers the logical activities to be performed in order to reach users' goals and the domain objects manipulated by these tasks. The *Abstract User Interface* (AUI) level is an expression of the UI in terms of interaction spaces or presentation units, regardless of the interaction modalities (graphical, vocal, haptic, etc.) and the interactors available. The *Concrete User Interface* level (CUI) expresses the UI in terms of "concrete interactors" that depend on the type of platform and media available. Lastly, the *Final User Interface* level

(FUI) consists of source code to be interpreted or compiled by actual target platforms.

Various models have been proposed to represent different aspects at different levels of abstraction. For instance, *ConcurTaskTrees* [4] and *ANSI/CEA-2018* [5] are intended to define task models. *MARIA* [6] and *USIXML* [7] aim at representing AUI models and *IDEAL2* [8] proposes a CUI model tailored to mobile Web UIs. In what regards to model-based approaches for multi-device web development, *W3C* has recently created the MBUI Working Group [9] in order to create standards as a basis for interoperability across authoring tools.

Some development platforms aimed at facilitating mobile Web development offer declarative models to describe UIs in a device independent manner. For instance, *AWS Open Source* (previously *Volantis Mobility Server*) [10] and *IBM Portal Accelerator* [11] uses *XDIME2* [12], a standard-based *XML* vocabulary based on *XHTML* and *XForms*, and *Netbiscuits* [13] uses *BiscuitML* [14] for the same purpose. However, any of these platforms follow a pure Model-Based approach.

BACKGROUND

MyMobileWeb (*MMW*) [15] is an open-source standards-based software platform released under *LPGL* version 3 license. It has been designed to facilitate the development of applications and portals for the mobile Web. It intends to provide an advanced environment for the development of rich mobile Web applications which maximize user experience regardless of the device used to access the Web. It has been conceived to solve inherent problems of mobile Web development in a holistic way and has been declared as a reference development platform by *mTLD* through their *dotMobi* initiative.

Serenoa is a model-based framework aimed at developing a reference platform for enabling the creation of context-sensitive service front-ends (*SFEs*). From the point of view of Serenoa, a context-sensitive *SFE* provides a UI that exhibits some capability to be aware of the context and to react to changes of this context in a continuous way. As a result, such a UI will be adapted to a person's devices, tasks, preferences, and abilities, thus improving people's satisfaction and performance compared to traditional *SFEs* based on manually designed UIs.

This paper presents the evolution performed by the authors over the *MMW* platform in order to adapt it to a pure model-based approach following the guidelines established by the Serenoa reference framework (MyMobileWeb For Serenoa: *MMW-4S*).

RUNTIME USER INTERFACE GENERATION ENGINE

RUIGE is one of the modules proposed by the Serenoa reference framework. It is a modular engine responsible for the creation of final applications adapted to the context starting from the AUI level by means of *ASFE-DL* [16].

RUIGE is composed of various sub-modules, each of them in charge of generating different outputs, such as mobile interfaces, vocal systems, avatar engines, etc. New sub-modules might be added to the architecture in order to support additional interaction modes and target platforms for the same application definition.

Each RUIGE sub-module follows several stages in order to generate the final application based on the abstract definition. Each stage is associated to a specific module:

- *Transformer*: component in charge of the process of converting the abstract language (*ASFE-DL*) into a generic language for the representation of the CUI of each sub-module. This language will be the input to the next stage.
- *Generator*: this module analyses the output of the transformer and generates the executable code, which will be used at runtime.
- *Runtime*: this component is intended to provide support for the execution of applications. Note that some sub-modules may require both deployment and execution stages. This component is in charge of deploying the application, if it has not been done before by the generator. For instance, uploading web applications in a web server, deploying an application in a servlet container or even installing (or running) a file in the user device. After the deploy stage, the application is executed by an execution platform; for instance, a web browser.

MMW-4S

MMW-4S is an open-source module[1] fully compliant with RUIGE and with the Serenoa reference framework. It is aimed at generating mobile Web applications for multiple platforms and devices. Moreover, *MMW-4S* is able to interoperate with the Adaptation Engine Serenoa component in order to adapt the generated UIs to the context, thus trying to maximize the user experience. Figure 1 shows a simplified version of the *MMW-4S* architecture and its integration in the Serenoa platform. These sub-modules are highlighted in blue color. Basically, *MMW-4S* receives the abstract description of the application from the Serenoa Service Repository, which subsequently gets such description from the Authoring Tools. The following sections provide details about the three main sub-modules of *MMW-4S*: transformer, generator and runtime.

Note that *MMW-4S* runtime needs to query the Serenoa Adaptation Engine in order to know the most appropriate adaptation actions that the Runtime must apply in each context. To accomplish that goal, the Runtime needs to produce/reference ad-hoc software modules (Context Delegates) that are the responsible for the extraction of

[1] All MMW-4S code has been released as open-source under the LGPLv3 license and it is available at the Morfeo Forge: https://svn.forge.morfeo-project.org/serenoa/trunk/

contextual information coming from device sensors at runtime. These Context Delegates propagate the information to the Context Manager, which is in charge of storing all the context properties and values for each device instance. The Adaptation Engine, in order to decide the most appropriate adaptation that should be applied in each specific context, needs to process adaptation rules expressed in *AAL-DL* [16] format and should communicate to the aforementioned Context Manager.

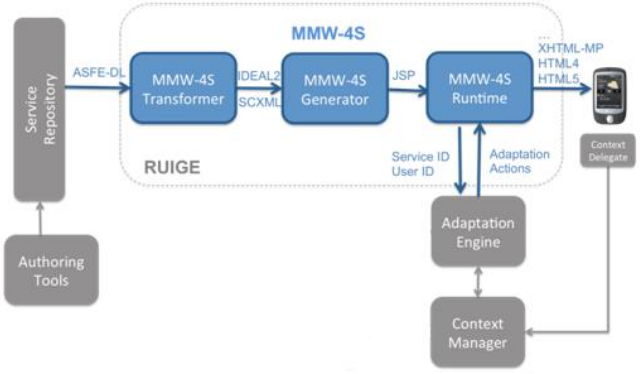

Figure 1. Architectural approach

Transformer

MMW-4S transformer carries out the transformation from *ASFE-DL* to the CUI models used by *MMW*: *IDEAL2* for the UI view definition and *SCXML* [17] for the application flow description. It has been implemented as an *XSL* transformation sheet (*XSLT*) due to the fact that both the input language (*ASFE-DL*) and the output languages (*IDEAL2* and *SCXML*) are *XML*-based. The current version of *MMW-4S* Transformer supports the last version of *ASFE-DL* available at the time of writing, the last version of *IDEAL2* and the draft version of *SCXML* published on 16 May 2008.

Generator

The UI Generator of *MMW-4S* is the part of the module in charge of generating JavaServer Pages (*JSPs*) from *IDEAL2* documents. Each UI defined in *IDEAL2* implies the generation of a specific *JSP* for each markup language in the market (*WML*, *XHTML Basic*/*MobileProfile*, *HTML4* and *HTML5*). Note that the generated *JSPs* contain server-side code that will be executed at runtime. These *JSPs* are structured to guide the execution of the *HTLM5* Rendering Engine at runtime.

Runtime

This module is intended to provide the most appropriate FUI at runtime. Whenever a client device requests a specific Web page to the server, the corresponding *JSP* will be executed. Note that *JSPs* are translated into Java Servlets at runtime. Consequently, each web page access will result in the invocation of server-side code containing the logic that generates the adapted Web content dynamically. The *HTML5* rendering engine is based on the *jQueryMobile* library. Moreover, a set of pre-generated Javascript libraries based on *jQuery* offer the possibility to perform client-side adaptations at runtime.

ADAPTATION SCENARIOS

Once RUIGE behavior has been explained, it is necessary to introduce some adaptation rules to be considered during the application lifecycle. In order to exemplify how *MMW-4S* treats adaptation rules, we will suppose the following set of rules, as coming from the Adaptation Engine. For the sake of legibility they are expressed in natural language in this article, although they have been formally defined in *AAL-DL*.

Adaptation to device features

R1: If the device is a tablet, then master-detail presentations will be rendered in one single view.

R1 must be considered during the transformation process when converting from *ASFE-DL* (AUI) to *IDEAL2* + *SCXML* (CUI) by means of an *XSLT*. This *XSLT* generates two different versions of the *IDEAL2* description. In the case of tablets, a master-detail view is expressed in just one *IDEAL2* presentation. Otherwise, in case of mobile devices, the same view is expressed in two different *IDEAL2* presentations, plus the additional flow logic expressed in *SCXML*. In order to carry out the adaptation, *MMW-4S* sends the User-Agent to the Context-Manager to identify which kind of device is accessing to the web.

Adaptation to the user profile

R2: If user is color-blind, then it will use an alternative color palette in the referenced images.

Adaptation to the status of the device

R3: If the level of the battery is higher than a prefixed threshold, then a video must be shown but if its level is lower, it renders an image.

Both rules, R2 and R3 must be considered at execution time by the runtime module in order to dynamically adjust the multimedia content (the color palette or the video/image in each case) while the user is interacting with the application. In this case, the adaptation rules received in *AAL-DL* format have been transformed into a rules language expressed in *DRL*, so as they can be executed at runtime by using the new rule engine developed within *MMW-4S*. Such rule engine is based on Drools Rule Engine [18]. In R2, the Context Manager handles user profiles in order to recognize special user features as color-blindness. In order to get the level of the battery in R3, an Android application was built. It simulates the access to the battery because, so far, there isn't any API implementation in any browser to do it.

Adaptation to the position of the user

R4: If the user is far away from the device, then bigger font-size will be shown but if the user is closer, the font-size will be smaller.

R4 must be considered by the generator during the generation process when converting from *IDEAL2* + *SCXML* (CUI) to *HTML5* + *CSS3* + *Javascript* (FUI). In

this case, the adaptation rule is considered in the generation process to determine the Javascript code that should be delivered. However, the adaptation will be carried out in the runtime process when the Javascript code is executed. In order to change the font-size, the frontal-web camera is used by means of *webRTC* [19]. This API allows calculating the distance between the screen and the user and this value is periodically sent to the server to update the font-size.

CONCLUSIONS AND FUTURE WORK

This article summarizes the ongoing research work aimed at creating an open-source RUIGE that uses Model-Based techniques to facilitate the development of mobile Web UIS tailored to the context. Moreover, we present four scenarios to highlight the types of adaptations considered and we provide technical details about the implementation of such adaptations.

Further research needs to be done in order to: (a) align the proposed method with incoming standards, such as those models being defined in the *W3C* MBUI Working Group; (b) exploit further context variations to perform adaptations, taking into account the evolution of the recommendations of the W3C Device APIs Working Group; (c) validate our proposal in comparison to other research works

ACKNOWLEDGMENTS

The research is being performed thanks to the support of the following research projects:

- MyMobileWeb: an open source project intended to facilitate the creation of mobile Web applications funded by Ministerio de Industra, Turismo y Comercio (MITyC) within the framework of The National Plan for Scientific Research, Development and Technological Innovation 2008-2011 and by the European Regional Development Fund (ERDF).
- Serenoa: a european project aimed at developing a novel, open platform for enabling the creation of context-sensitive SFEs. This project has received funding from the European Commission's Seventh Framework Programme under grant agreement nº 258030 (FP7-ICT-2009-5).

REFERENCES

1. W3C Device APIs (DAP) Working Group: http://www.w3.org/2009/dap/.
2. Serenoa Project web site: http://www.serenoa-fp7.eu.
3. Limbourg, Q., Vanderdonckt, J., Bouillon, L., Calvary, G., Coutaz, J., Thevenin, D.: A unifying reference framework for multi-target user interfaces. Interacting with Computers. 15, 289-308 (2003).
4. Paterno, F., Mancini, C., Meniconi, S.: ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. Proceedings of Interact 97, 14-18 July 1997.
5. Rich, C. Building Task-Based User Interfaces With ANSI/CEA-2018. IEEE Computer, Vol. 42, No. 9, August 2009.
6. Paternò, F., Santoro C., Spano, L.D., MARIA: A Universal Language for Service-Oriented Applications in Ubiquitous Environment", ACM Transactions on Computer-Human Interaction, Vol.16, N.4, November 2009, pp.19:1-19:30, ACM Press.
7. Limbourg, Q., Vanderdonckt, Q., Michotte, B., Bouillon, L., López-Jaquero, V.: USIXML: A Language Supporting Multi-path Development of User Interfaces. EHCI/DS-VIS 2004: 200-220
8. Cantera, J.M, Díaz, J.L, Rodríguez, C.: IDEAL2 Corelanguage. MyMobileWeb Working Draft, 31 December 2010, http://files.morfeo-project.org/mymobileweb/public/specs/ideal2/ideal2-20101231(2010).
9. W3C Model-Based User Interfaces (MBUI) Working Group: http://www.w3.org/2011/mbui.
10. AWS Open Source web site: http://www.antennasoftware.com/community/developers/open-source
11. IBM Portal Accelerator web site: http://www-03.ibm.com/software/products/us/en/ibmmobiportacce/
12. XDIME2 language reference: http://www.antennasoftware.com/pdf/open_source_start_here.pdf
13. Netbiscuits Platform web site: http://www.netbiscuits.com/
14. BiscuitML reference guide: http://kb.netbiscuits.com/tactile/bml/index.html
15. MyMobileWeb Project web site: http://mymobileweb.morfeo-project.org.
16. Ghiani G., Paternò F., Santoro C., Spano L.D.: A Set of Languages for Context-Aware Adaptation. CASFE 2012, CEUR WS Proceedings, Vol 970.
17. Barnett, J. et al.: State Chart XML (SCXML): State Machine Notation for Control Abstraction. W3C Working Draft 16 February 2012, http://www.w3.org/TR/2012/WD-scxml-20120216/ (2012).
18. Drools web site: http://www.jboss.org/drools/
19. WebRTC web site: http://www.webrtc.org/