Developing a Semantic User and Device Modeling Framework that supports UI Adaptability of Web 2.0 Applications for People with Special Needs

Philip Ackermann
Fraunhofer Institute for Applied Information Technology FIT
Schloss Birlinghoven, D53757 Sankt Augustin (Germany)
philip.ackermann@fit.fraunhofer.de

Carlos A. Velasco
Fraunhofer Institute for Applied Information Technology FIT
Schloss Birlinghoven, D53757 Sankt Augustin (Germany)
carlos.velasco@fit.fraunhofer.de

Christopher Power
Department of Computer Science
University of York
Deramore Lane, York YO10 5GH
cpower@cs.york.ac.uk

ABSTRACT

The introduction of user and device models to customize applications has been the subject of research for decades. This paper presents a modeling framework that supports dynamic adaptation of the UI of web 2.0 applications. This work builds upon previous efforts of the authors [22] leveraged with the use of the semantic framework for Composite Capability/Preference Profiles (CC/PP), which allows the matching of device capabilities and user preferences arising because of functional restrictions. The combination of these models with those of the corresponding web applications, enables an adaptive transformation process that facilitates access to users with special needs derived of their functional restrictions or because of context-related handicapping situations. We argue that this approach will enable a user-centric access to the web, including mobile and ubiquitous delivery of services and applications.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: User-centered design; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia; K.4.2 [**Computers and Society**]: Social Issues—*Handicapped persons/special needs*

General Terms

Interface Adaptivity, Model Driven Engineering

Keywords

web accessibility, user model, device model, application model, CC/PP, interface plasticity, web 2.0, mobile web

1. INTRODUCTION

W4A2012 - Communications, April 16-17, 2011, Lyon, France. Co-Located with the 21st International World Wide Web Conference.

The web does not resemble any more its origins as a global and static hypertext system and has evolved into a platform for distributed applications that emulate the functionality of desktop applications. These applications were originally known as Rich Internet Applications [9, 13], but the term has evolved depending on architectural considerations into cloud computing applications and web 2.0 applications. Early in the previous decade, they were thought to be run on proprietary sandboxes in different user agents. However, the introduction of the XMLHttpRequest (XHR) object in the early 2000s originated a new breath of web applications. XHR defines an API [8] that provides scripted client functionality for transferring asynchronously data between a client and a server. The combination of (X)HTML and CSS (presentation layer), some metadata describing interaction components and Javascript increases the application responsiveness and improves the user experience [15].

This trend poses severe obstacles to accessibility. The absence of semantics in the custom graphical components prevented the assistive technologies from proper rendering [3]. These issues have been addressed in the previous years by adding semantic annotations to (X)HTML via WAI-ARIA [4], which adds roles, states and properties to its DOM elements. These enhancements are partially obsolete by the enriched semantics of the new HTML5 [6] elements.

This landscape complicates itself with the participative nature of web 2.0, where users themselves can aggregate and combine content and applications via mashups, creating a myriad of possibilities. This obsoletes the traditional "inverted-V" accessibility components model [11], because users become de-facto developers.

The paper is organized as follows. Section 2 briefly reviews the state of the art in this area. Section 3 describes the architecture of our system and the adaptation process. Section 4 presents the underlying modeling framework. And finally, section 5 presents our conclusions and future work.

2. STATE OF THE ART

User modeling is an active research topic for more than four decades. Since the beginnings of the 1970s many dedicated and general user modeling systems have been developed to enable user-adaptation in different domains (see, e.g., [14, 16]). Recent approaches manifest a convergence of the adaptive web and the semantic web [21] by utiliz-

ing standardized web ontology languages and their software infrastructure. Among them, we can highlight the VUMS Cluster,[1] which aims at a harmonization and standardization of machine-readable, scalable and reusable user models. Following a declarative approach the user is described by a set of fine-graded properties defined in a shared hierarchical meta-model (a taxonomy of physiological and cognitive variables). Their particular impact on the application behavior or concrete interface rendering depends on the subsequent accumulation and interpretation by the adaptive system. For instance, [18] presents a virtual human model suitable to the development of products in CAD environments.

The projects ACCESSIBLE[2] and AEGIS[3] have developed large ontologies targeting users with special needs. These are modeled at a higher level in terms of their capabilities or functional limitations. Various stereotype profiles were provided. Due to their holistic and declarative nature these models tend to become too complex and demanding with respect to management and run-time processing. Therefore we investigated more tailored approaches for explicitly stating user preferences and adaptation needs by means of a dedicated ontology, which deems the interpretation of declarative models unnecessary.

Device modeling started to acquire relevance as ubiquitous and mobile computing became mainstream. The most relevant industrial application has been the User Agent Profile (UAProf, [19]), which provided a framework for describing capabilities of mobile devices until the middle of the previous decade. This vocabulary became obsolete because its applicability to the recent generations of smartphones is minimal.

On the basis of a classification of input devices, [20] uses device models to categorize interaction problems that occur because of wrong use of devices. However this categorization is at a functionality metalevel and has no applicability for UI adaptation. Other strand of work is focused on the adaptation of the device to the content, like in the Universal Remote Console (URC) standard [17].

The notion of application model is multifaceted. One can address the topic from different perspectives: architecture design, implementation design or interaction design [7]. It is out of the scope of this paper to discuss in detail the different approaches. We will focus our attention on the research area of Model-Based User Interface design,[4] which aims at identifying high-level models for the specification and analysis of interactive applications from a semantic perspective (as opposed to the more traditional syntactic perspective). Under this umbrella, the more interesting approach is the CAMELEON Unified Reference Framework [10], which is similar to the decomposition selected. Furthermore, as indicated earlier, we are also making use of the WAI-ARIA recommendation [4], which specifies how to make web applications more accessible by semantically annotating (X)HTML widgets (elements) with roles, attributes and properties.

3. ARCHITECTURE

The architecture of our system is determined by the diversity of environments in the I2Web project.[5] Reflecting the richness of the ubiquitous web nowadays, we have implemented our framework in: a multiplatform e-banking framework; a multiplatform e-government application; and a multimedia content management system.

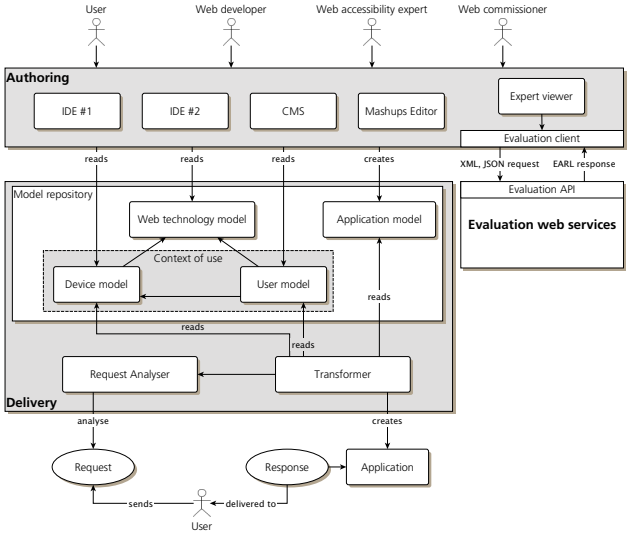

Figure 1: Architecture of the modeling framework.

The objective of the designed architecture is to cover the entire life-cycle in the development (authoring process) and delivery of a modern web application, represented in Fig. 1. For our purposes, it suffices to mention that the authoring process is used in its wider acception, including from web developers using complex IDEs to standard Internet users aggregating information via mash-ups. The architecture includes access to remote evaluation services for web accessibility. An evaluation client facilitates the access to the remote evaluation API. This client sends evaluation requests in the XML or JSON formats and parses the response, an evaluation report in the EARL format [5]. The access to the model repository allows the creation of an application model that is key to the adaptation process.

Focusing on the delivery component of Fig. 1, we can initiate the foreseen workflow when a user makes a request to the server. The request analyzer component identifies the user based on information sent inside this request. This may be an extended HTTP header (e.g., x-i2web-uprof) or her login information. Then the transformer component retrieves the appropriate user model from the model repository and stores a reference to it in the current user session.

The device model can be determined by two alternative ways: either via an extended HTTP header which identifies the device (e.g., x-i2web-dprof) or directly from the user model —if the user model contains a reference to her available devices. The first approach allows the user to switch her device on the fly. As long as the device sends the correct header, the application will adapt accordingly. The second approach requires that the user selects manually her current device.

An extension considered is a HTTP client-side proxy (initially envisaged in [22]), which automatically sends infor-

[1] `http://www.veritas-project.eu/vums/`
[2] `http://bit.ly/wqxJ8W`
[3] `http://www.aegis-project.eu/index.php?Itemid=65`
[4] See [2] and references therein.
[5] `http://i2web.eu/`

mation about the current device with every request. Note that detecting the device based on the User-Agent HTTP header is not sufficient in our case because it only contains information about the user agent and the operating system, i.e., it does not contain information about assistive technologies and other important client capabilities, and because the information inside the header is not always reliable.

After both the device and the user model are determined, the transformer component reads the application model and applies one or more transformations to create the adapted application. There are different types of transformations [12]: suppression, insertion, substitution and reorganization (see Section 4.3 for some typical examples). A single-step transformation process uses as input the three models (user, device and application) to generate the new application interface. A multi-step transformation uses the application model and at least one of the user or device models as input. Every transformation generates an application state that again can be used as input for another transformation, similar to the pipes and filters pattern.

4. MODELING FRAMEWORK

The transformations described in the previous section are based upon our modeling framework that represents the delivery context for our web application. Our models are based upon the Composite Capability/Preference Profiles (CC/PP) 2.0 [1] RDF framework. CC/PP enables a description of device capabilities and user preferences that can be used to guide the adaptation of content. CC/PP allows to construct user and device profiles as a two level hierarchy: a profile with a number of components, which have a number of attributes. The attributes of a component may be included directly in a profile document, or may be specified by reference. This simple framework allows an efficient processing of profiles without the parsing overhead of semantic models based on OWL, for instance.

4.1 Web Technology model

One of the novelties of our approach is the definition of a decoupled Web Technology model,[6] which could be used by any other component of our architecture. This approach offers several advantages: (i) it allows the expression of device capabilities for either hardware or software components; (ii) it allows the expression of user preferences; and (iii) it simplifies the expression of matching rules for adaptation with different granularities, ranging from top components like, i.e., images or movies to HTML elements and attributes, or CSS properties, for instance. The model is characterized by a hierarchy of classes, in which subclasses like MarkupTechnology, XMLTechnology or StyleTechnology include containers for their constituting parts.

4.2 Device model

Our approach for the device model is similar to that of UAProf [19]. However, we have decided not to extend it but to design our own vocabulary,[7] denoting a semantic equivalence or difference where appropriate. The main components are:

- HardwarePlatform: the HardwarePlatform component is used as a generic platform description. It is different from HardwarePlatform in UAProf, which contains properties of the device's hardware, such as display size, supported character sets, etc.

- Hardware: this component identifies individual hardware elements like displays, graphic cards, sound cards, keyboards, mice, etc. It can also be used to identify assistive technology hardware like Braille-lines, etc.

- SoftwarePlatform: the SoftwarePlatform component is used as the operating system description of the device. Again, it is different from the UAProf SoftwarePlatform component, which contains properties of the device's application environment, operating system, and installed software.

- Software: this component identifies individual software components of the device. We distinguish two subclasses: ApplicationSoftware (software that performs some kind of activity for the user) and InteractionSoftware (software that supports the interaction of the user with the device like a screen-reader, an on-screen keyboard, etc.). A key subclass from the former is the UserAgent class, used to define software that retrieves and renders web content for users and is similar to the UAProf BrowserUA component.

4.3 User model

Our user model[8] is oriented towards an efficient mapping between device capabilities and user preferences. The model supports the four UI remoulding approaches described in [12] via 4 properties that link different components: suppression of the UI components that become irrelevant, i.e., elimination of decorative images for a blind user; insertion of new UI components to provide access to new services relevant in the new situation/context, like the addition of subtitles and text descriptions to a movie for a hard-of-hearing user or like the addition of additional help cues for an older adult; substitution of UI components when UI components are replaced with new ones (substitution can be viewed as a combination of suppression and insertion of UI components), i.e., provision of an alternative text description for a blind user; and reorganization of UI components by revisiting their spatial layout and/or their temporal dependency. This could result in simplified layouts for processes that may reduce the cognitive load for complex operations. This decomposition allows the application to quickly identify unavailable user preferences, permiting a proactive approach by informing the user of potential barriers.

4.4 Application model and integration

The aforementioned models need to be integrated with the application thus it can be rendered properly. Our initial work has been focused to the usage of the new semantic components of HTML5 [6] or enhancing existing (X)HTML applications with WAI-ARIA annotations [4].

With this approach we have been able to generate inference rules customised to the three applications described in Section 3. Further steps will include an analysis to generalize this application model and include additional input like user strategies.

[6] Available at: `http://i2web.eu/ns/2012/technology#`
[7] Available at: `http://i2web.eu/ns/2012/devicemodel#`
[8] Available at: `http://i2web.eu/ns/2012/usermodel#`

5. CONCLUSIONS AND FUTURE WORK

We have presented in this paper our approach to deliver accessible and adaptable web 2.0 applications via semantic user and device models based upon the CC/PP 2.0 framework. These models are enhanced with a decoupled "technology model," which allows the creation of efficient adaptation rules with suppression, substitution, insertion or reorganization of content (markup), style (CSS) or interaction (Javascript, server-side logic) components.

This model-driven architecture demands an application model that transforms its UI components. The implemented prototypes deal with this by using the semantics provided by HTML5 and/or WAI-ARIA annotations.

At the moment of writing this paper, we are carrying tests with users to evaluate the feasibility and efficiency of our approach. The initial tests carried out by our teams have shown positive results. Next steps include: (i) the refinement of the architecture, (ii) implementation of transformations as rules and integration of rule engines, (iii) integration in our model-based simulation and evaluation framework, and (iv) the extension and generalisation of the corpora of adaptation rules.

6. ACKNOWLEDGEMENTS

Parts of this work have been undertaken within the framework of the Inclusive Future-Internet Web Services (I2Web) project, co-financed by the 7th Framework Programme of the European Commission under its ICT Programme (Grant Agreement number 257623).

7. ADDITIONAL AUTHORS

Yehya Mohamad and Jaroslav Pullmann (Fraunhofer Institute for Applied Information Technology FIT, Schloss Birlinghoven, D53757 Sankt Augustin, Germany).

8. REFERENCES

[1] Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 2.0, W3C Working Draft 30 April 2007. Technical report, World Wide Web Consortium (W3C), 2007.

[2] Model-Based UI XG Final Report, W3C Incubator Group Report 04 May 2010. Technical report, World Wide Web Consortium (W3C), May 2010.

[3] WAI-ARIA 1.0 Primer - An introduction to rich Internet application accessibility challenges and solutions, W3C Working Draft 16 September 2010. Technical report, World Wide Web Consortium (W3C), 2010.

[4] Accessible Rich Internet Applications (WAI-ARIA) 1.0, W3C Candidate Recommendation 18 January 2011. Technical report, World Wide Web Consortium (W3C), 2011.

[5] Evaluation and Report Language (EARL) 1.0 Schema, W3C Working Draft 10 May 2011. Technical report, World Wide Web Consortium (W3C), 2011.

[6] HTML5 - A vocabulary and associated APIs for HTML and XHTML, W3C Working Draft 25 May 2011. Technical report, World Wide Web Consortium (W3C), 2011.

[7] State-of-the-art of the User, Device and Application Models. Technical report, I2Web project, Sankt Augustin, 2011.

[8] XMLHttpRequest Level 2, W3C Working Draft 17 January 2012. Technical report, World Wide Web Consortium (W3C), 2012.

[9] J. Allaire. Macromedia Flash MX — A next-generation rich client. Technical Report March, Macromedia, Inc., San Francisco, Mar. 2002.

[10] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt. A Unifying Reference Framework for multi-target user interfaces. *Interacting with Computers*, 15(3):289–308, June 2003.

[11] W. A. Chisholm and S. L. Henry. Interdependent components of web accessibility. In *Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A) - W4A '05*, number May, page 31, New York, New York, USA, 2005. ACM Press.

[12] J. Coutaz and G. Calvary. HCI and Software Engineering: the Case for User Interface Plasticity. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications - Human Factors and Ergonomics Series*, chapter 56, pages 1107–1118. Taylor & Francis Press, New York, 2008.

[13] J. Duhl. Rich internet applications. Technical report, IDC, Framingham, Nov. 2003.

[14] G. Fischer. User Modeling in Human-Computer Interaction. *User modeling and user-adapted interaction*, 11(1):65–86, 2001.

[15] J. J. Garrett. Ajax: A New Approach to Web Applications, 2005.

[16] A. Kobsa. Generic user modeling systems. *User modeling and user-adapted interaction*, 11(1):49–63, 2001.

[17] B. LaPlant, S. Trewin, G. Zimmermann, and G. Vanderheiden. The universal remote console: a universal access bus for pervasive computing. *IEEE Pervasive Computing*, 3(1):76–80, Jan. 2004.

[18] Y. Mohamad, C. A. Velasco, J. Pullmann, M. Lawo, and P. Kirisci. Virtual User Concept for Inclusive Design of Consumer Products and User Interfaces. In C. Stephanidis, editor, *Universal Access in Human-Computer Interaction. Users Diversity*, Lecture Notes in Computer Science, pages 79–87. Springer, Berlin - Heidelberg, 2011.

[19] Open Mobile Alliance. User Agent Profile – Approved Version 2.0 – 06 Feb 2006. Technical report, Open Mobile Alliance, Feb. 2006.

[20] M. Schulz, S. Schmidt, K.-P. Engelbrecht, and S. Möller. Using Device Models for Analyzing User Interaction Problems. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '11, pages 303–304, New York, NY, USA, 2011. ACM.

[21] S. Sosnovsky and D. Dicheva. Ontological technologies for user modelling. *International Journal of Metadata, Semantics and Ontologies*, 5(1):32–71, 2010.

[22] C. A. Velasco, Y. Mohamad, A. S. Gilman, N. Viorres, E. Vlachogiannis, A. Arnellos, and J. S. Darzentas. Universal access to information services — the need for user information and its relationship to device profiles. *Universal Access in the Information Society*, 3(1):88–95, Mar. 2004.