

# A Tool to Support the Collection of User Preferences and Device Characteristics to Enable UI Adaptability in Web 2.0 Applications

Philip Ackermann, Carlos A. Velasco, and Evangelos Vlachogiannis

Fraunhofer Institute for Applied Information Technology FIT  
Schloss Birlinghoven, 53757 Sankt Augustin, Germany  
{philip.ackermann, carlos.velasco,  
evangelos.vlachogiannis}@fit.fraunhofer.de

**Abstract.** We present in this paper a tool that supports the collection of user preferences within the set of components developed in the i2web project to enable UI adaptability of web 2.0 applications. The tool (called Model Management System, MMS) is based upon a semantic web modelling framework to describe user preferences and device capabilities, combined with state-of-the-art models of web applications. This work builds upon previous efforts of the authors [1] leveraged with the use of the semantic framework Composite Capability/Preference Profiles (CC/PP [2]). The MMS gathers information on user preferences and supports user and application providers in discovering the characteristics of the device utilized by the user when accessing the application.

## 1 Introduction

The web is not any more a static hypertext system and has evolved into a platform for distributed and ubiquitous applications that emulate the functionality of desktop applications. These applications are known as Rich Internet Applications (RIAs) [3,4]. The concept is known as well, depending on architectural considerations, as cloud computing and web 2.0 applications.

This trend poses severe obstacles to accessibility. The absence of semantics in the custom graphical components often prevent assistive technologies from proper rendering. Several accessibility issues of RIAs are identified in [5]: lack of semantic cues and metadata in HTML content, its re-purposing for formatting and presentation, the lack of keyboard access, etc. These issues have been addressed in the previous years by adding semantic annotations to (X)HTML via WAI-ARIA [6], which adds roles, states and properties to its DOM elements. These enhancements are partially obsolete by the enriched semantics of the new HTML5 [7] elements.

The participative nature of web 2.0 complicates this landscape, where users can aggregate and combine themselves content and applications via mashups, creating a myriad of possibilities (described as  $2^W$  in [8]).

From a socio-economic standpoint accessibility and usability issues are becoming critical, considering the demographic change worldwide. When we add to these figures the percentage of the population afflicted with a disability, we get an idea of the relevance of the problem.

The paper is organized as follows. Section 2 briefly reviews the state of the art in this area. Section 3 describes the modeling framework used in the project. Section 4 presents the Model Management System per se. Section 5 briefly discusses the user evaluation results. And finally, section 6 presents our conclusions and future work.

## 2 State-of-the-Art

The state-of-the-art of the different modeling perspectives for users, devices and web applications is reviewed in detail in [1]. More recent approaches like those of the Cloud4all and Prosperity4all projects [9] are by far more complex and they do not define an explicit modeling framework that can be used by the different actors (users, developers, content providers, etc.) and how the components can interoperate with each other. In the authors' opinion that may hinder a wider adoption by industry.

The approach followed by us diverges from those described there because do not focus on the description of the user physical or cognitive abilities, but on the description of the user preferences and the device capabilities of the dispositive utilized to access the application.

## 3 Modeling Framework

Our modeling framework is based upon the Composite Capability/Preference Profiles (CC/PP) 2.0 [2] RDF framework [10]. 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. In the following, we describe shortly the main components of our framework. The architecture of the framework is described in more detail in [1].

One of the novelties of our approach was the definition of a decoupled Web Technology model,<sup>1</sup> which could be used by any other component of the framework. This approach offers several advantages:

- (i) it allows the expression of device capabilities for either hardware or software components;

---

<sup>1</sup> Available at: <http://i2web.eu/ns/2012/technology#>

- (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.

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 [11]), 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 smart phones is minimal. Although our approach for the device model is similar to that of UAProf, we decided not to extend it but to design our own vocabulary,<sup>2</sup> denoting a semantic equivalence or difference where appropriate.

As mentioned earlier, we focused on the modeling of the user preferences and not on the modeling of her physiological characteristics. On that way, we could focus on transforming different UI components to enable adaptivity. Our work is based upon the CAMELEON Unified Reference Framework [12,13], which “... is a framework that serves as a reference for classifying user interfaces supporting multiple targets, or multiple contexts of use in the field of context-aware computing.” Furthermore, following the work of the SERENOA project [14], we have newly introduced a semantic container that holds categories of pre-defined transformations, which can be mapped to algorithms similar to those of the library of algorithms for advanced adaptation logic defined by SERENOA. Due to the abstraction level introduced by the Web Technology model, we can avoid redundancies in their classification due to different content-types.<sup>3</sup>

Application modeling generally involves three major aspects: architecture, implementation and interaction. Ensuring a good level of accessibility for an application is a task that has impacts on all the three aspects. Accessibility obviously affects the user interaction, but it affects also the implementation and architectural choices, because sophisticated and adaptive user interaction can scale only if the underlying architecture and implementation patterns are appropriate. Since it is realistically impossible to impose architectural and implementation patterns that adapt to all application types, the i2web application models are a set of guidelines that implement application modeling best practices for incorporating accessibility into a Web application. The model is based upon the WAI-ARIA recommendation [6], concentrates on the semantics of the application and does not contain any domain-dependent information. This approach was validated by expert and user tests.

With this approach we have been able to generate successfully inference rules customized to three industrial-size web applications: a multiplatform e-banking framework; a multiplatform e-government application; and a multimedia content management system.

---

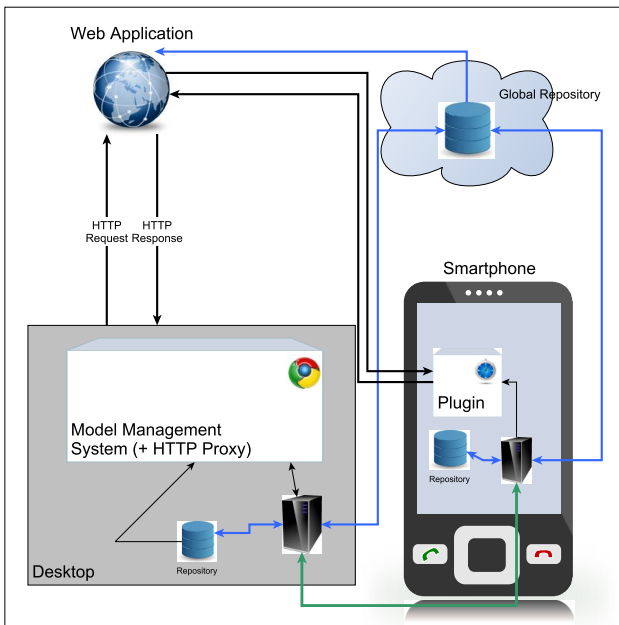
<sup>2</sup> Available at: <http://i2web.eu/ns/2012/devicemodel#>

<sup>3</sup> The user preferences model is available at: <http://i2web.eu/ns/2012/usermodel#>

## 4 Model Management System

The Model Management System (MMS) of the i2web project is an assistive system with two main functions:

- (i) support the user in configuring her preferences when interacting with different web applications; and
- (ii) extend the functionality of standard web browsers (user agents) to facilitate the communication with the different web applications so that they receive information on the user's device, her assistive technology and her preferences.

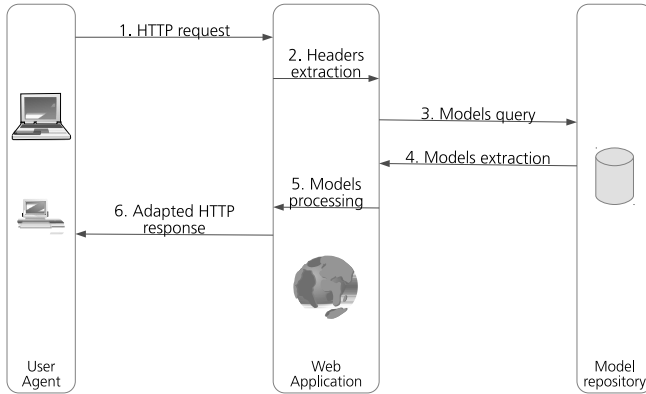


**Fig. 1.** General overview of the Model Management System (MMS)

Fig. 1 presents a high level overview of the system. The target is that the MMS is able to run in different devices (desktop, smart phone, etc) and at the same time, it takes care of the synchronization of the user preferences between them and the global repository (which location does not need to be necessarily on the cloud). The MMS also supports the user in deciding which preferences fit her needs.

Additionally, it provides to the user's device a HTTP proxy so that the anonymized profile location is transmitted via HTTP headers to the web application. The web application communicates with the global model repository

and provides to the user an adapted web application. This HTTP proxy component defines clearly an interaction sequence between the user agent in the corresponding client device and the web application at the server-side. This sequence consists of six stages (see Fig. 2):



**Fig. 2.** Client-server interaction sequence diagram

1. The client device sends in addition to the normal HTTP requests, two supplementary headers that represent pointers to model instances of the user preferences and her device profile.
2. These headers are extracted by the web application.
3. The server queries the global models' repository to receive the relevant information for the web application.
4. The global models' repository sends the requested profiles.
5. The web application processes this information and prepares the corresponding adapted response.
6. The HTTP response is sent to the client.

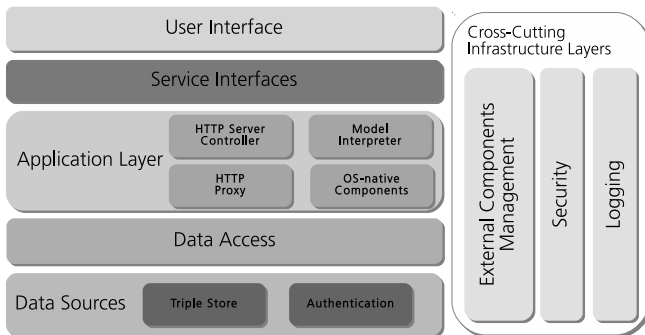
It must be highlighted that due to the asynchronous nature of modern web 2.0 applications, this sequence is transparent to the user and a great effort was invested to avoid performance penalties.

Fig. 3 presents an overview of the architecture of the system via a layered diagram. Its main components are:

- Cross-cutting Infrastructure Layers: these are components related to common functionalities (e.g., logging services or security/authentication) or with third-party services (like communication with the semantic repositories).
- User Interface Layer: this component includes interactions with end users in regard to the configuration of their preferences and of the client.
- Service Interfaces Layer: components related to different applications services.

- Application Layer: these components include the core functionality of the MMS and have 4 elements:
  - HTTP Server Controller: client-side HTTP server with small memory footprint responsible of controlling the local web application utilized by the user to configure the system among other tasks.
  - Model Interpreter: component responsible of bidirectionally “translating” modeling information.
  - Operating System Native Components: set of specific components tailored to the different operating systems to detect relevant information to the MMS, such as installed software and assistive technology.
  - HTTP Proxy: this component is responsible of inserting at runtime the i2web HTTP headers that contain the relevant information of the user preferences and her device settings. This component can be used in two versions, as a standard HTTP proxy at the client side or as a browser plug-in. This dual implementation allowed a great flexibility in the user testing phase. For instance, some screen readers do not function properly with given browsers.
- Data Sources/Data Access: these are components related to the storage of information, be it the semantic models representing device information or user preferences or authentication information that protects the privacy of the users.

These components are developed in a service-oriented architecture and communicate with each other via REST-based services.



**Fig. 3.** Layered architecture diagram of the MMS assistive system

## 5 User Evaluation

The user interface went through 4 design iterations, which included 3 expert and end-user evaluations. It has been conceived as a wizard application that

supports the user to configure her system in a step-by-step manner. In the different versions of this interface, major changes were introduced only in the sequence/number of steps, the amount of information displayed to the user, and improving the automatic detection of system characteristics to hide from the user any complexity. The key steps identified were:

- authentication,
- detection of previous configurations,
- operating system detection,
- user agent (browser) detection,
- assistive technology detection, and
- preferences configuration.

After these steps were complete, a submission to the persistence layer is done by the system to store the changes and inform the proxy of those changes.

The final user evaluation was in general very positive and the users were able to use and manage the system autonomously. Due to resources and time constraints, the system offered a limited set of adaptations to the end user. It is expected in the near future to widen these elements, so that the user had at her disposal more choices for this adaptation process.

## 6 Conclusions and Future Work

As mentioned earlier, we have successfully tested the whole framework in several environments from the technical perspective and with end users. For its applicability to a larger scale, the repository needs to include more instances of devices capabilities and user preferences. To that end, it is necessary first to extend the MMS to support different mobile devices and to test the framework with more users with special needs.

**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).<sup>4</sup>

## References

1. Ackermann, P., Velasco, C.A., Power, C., Mohamad, Y., Pullmann, J.: Developing a Semantic User and Device Modeling Framework that supports UI Adaptability of Web 2.0 Applications for People with Special Needs. In: Proceedings of the International Cross-Disciplinary Conference on Web Accessibility, Lyon, France, pp. 12:1–12:4. ACM (2012)
2. Kiss, C. (ed.): Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 2.0, W3C Working Draft 30 April 2007. World Wide Web Consortium, W3C (2007)

---

<sup>4</sup> <http://i2web.eu/>

3. Allaire, J.: Macromedia Flash MX — A next-generation rich client. Technical Report March, Macromedia, Inc., San Francisco (March 2002)
4. Duhl, J.: Rich internet applications. Technical report, IDC, Framingham (November 2003)
5. Pappas, L., Schwerdtfeger, R., Cooper, M. (eds.): WAI-ARIA 1.0 Primer - An introduction to rich Internet application accessibility challenges and solutions, W3C Working Draft. World Wide Web Consortium, W3C (September 16, 2010)
6. Craig, J., Cooper, M. (eds.): Accessible Rich Internet Applications (WAI-ARIA) 1.0, W3C Proposed Recommendation . World Wide Web Consortium, W3C (February 6, 2014)
7. Berjon, R., Faulkner, S., Leithead, T., Navara, E.D., O'Connor, E., Pfeiffer, S., Hickson, I. (eds.): HTML5 - A vocabulary and associated APIs for HTML and XHTML, W3C Candidate Recommendation World Wide Web Consortium, W3C (February 4, 2014)
8. Raman, T.V.: Toward  $2^W$ , beyond web 2.0. *Communications of the ACM* 52(2), 52–59 (2009)
9. Vanderheiden, G.C., Treviranus, J., Chourasia, A.: The global public inclusive infrastructure (gpri). In: *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility. ASSETS 2013*, pp. 70:1–70:3. ACM, New York (2013)
10. Brickley, D., Guha, R. (eds.): RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation. World Wide Web Consortium (W3C) (February 10, 2004)
11. Open Mobile Alliance: User Agent Profile – Approved Version 2.0. Technical report, Open Mobile Alliance (February 6, 2006)
12. Calvary, G., Coutaz, J., Bouillon, L., Florins, M., Limbourg, Q., Marucci, L., Paternò, F., Santoro, C., Souchon, N., Thevenin, D., Vanderdonckt, J.: The CAMELEON Reference Framework. Technical report, CAMELEON Project, IST-2000-30104 (2002)
13. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A unifying reference framework for multi-target user interfaces. *Interacting with Computers* 15(3), 289–308 (2003)
14. Motti, V.G.: Algorithms for Advanced Adaptation Logic. Deliverable 4.2.1, SERENOA project (2011)