

Adaptive Mobile Web Applications Through Fine-Grained Progressive Enhancement

Heiko Desruelle, Dieter Blomme, Frank Gielen

Ghent University – IBBT

Dept. of Information Technology – IBCN, Ghent, Belgium

{heiko.desruelle, dieter.blomme, frank.gielen}@intec.ugent.be

Abstract—The availability of mobile devices is growing at an incredible pace. This trend has set a need for applications being available at anytime, anywhere, and on any device. As most mobile users carry their device at all times, being truly mobile provides users an unprecedented freedom. Nevertheless the clear advantages, application developers are facing challenges due to device fragmentation. Current application development solutions are insufficiently prepared for the high variety of mobile platforms and hardware characteristics. In this paper we propose a platform for the development and delivery of adaptive mobile applications. An adaptive application composition approach is introduced, capable of autonomously bypassing fragmentation related issues. This goal is achieved by incorporating fine-grained progressive application enhancements through a quantitative evaluation strategy.

Keywords-mobile web, adaptive mechanism, progressive enhancement, quantitative evaluation.

I. INTRODUCTION

Mobile is a powerful mass medium, with a greater reach and faster growth than any other known media type [1]. Furthermore, the technology itself is rapidly maturing. Advanced features such as mobile internet access and integrated sensors have become a standard capability of devices throughout all user segments. As users carry their mobile device practically at all times, a need has been established for mobile applications and services being available anywhere and at any time. However, various technological challenges reside in the development of applications that automatically cover many types of devices [6]. This barrier is a result of the heavily fragmented mobile landscape. In order to attain a sustainable share of the mobile market, applications need to be made adaptable to various combinations of hardware, operating systems, APIs, etc. The absence of appropriate platform and tool support make it the developer's responsibility to deal with adaptability requirements, which considerably drives up development costs and narrows target markets. Against this backdrop, the use of the web as an application platform is gaining momentum. Device independent web technologies such as HTML, CSS and JavaScript offer application developers an unprecedented market reach. Furthermore, the International Telecommunication Union (ITU) estimates the use of mobile internet connectivity to surpass access rates of traditional desktop-based internet by the year 2013 [7].

Web systems are traditionally engineered along three orthogonal dimensions: the development phases, the system's views, and its aspects (as illustrated in Figure 1) [11]. The phase dimension sets out the different stages of web development, ranging from analysis to implementation. Each of these phases requires a number of specific views addressing the systems content, its navigation structure, and the presentation. Finally, the aspects dimension defines the structural and behavioral aspects of each of the views. The growing presence of mobile applications emphasizes the need for fragmentation management within the web engineering model. This concern has to be handled throughout every stage of the application's development life cycle. As proposed by Kappel et al., adaptability can be considered as an additional web engineering dimension, crosscutting all other web modeling dimensions [9]. From a developer's perspective, the straightforward incorporation of adaptability remains an important challenge [11]. Even with the use of standardized web technology, efficiently managing mobile fragmentation remains an important research topic. The different mobile browsers still contain many variability points, making true mobile convergence not to be expected any time soon [6].

Since the early days of web engineering, developers have tried to cope with the differences between browsers. Graceful degradation is a widespread design strategy that focuses on providing optimal support for the most advanced browsers. Less capable browsers are only considered during the last development phase. This approach often results in a poor stripped-down version. The graceful degradation

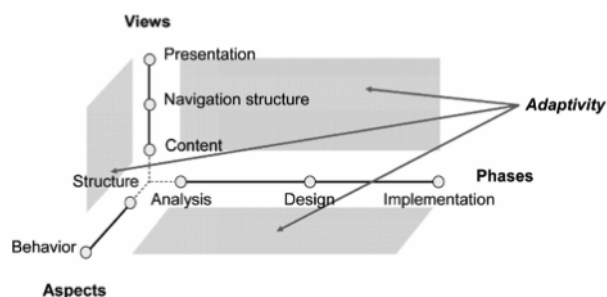


Figure 1. Adaptability as a crosscutting aspect on the traditional modeling dimensions of web engineering (from Koch et al. [10])

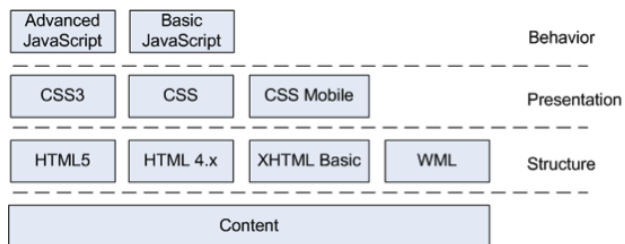


Figure 2. Fine-grained mobile progressive enhancement. A dynamic process, driven by the specific capabilities of a client's mobile device

methodology expects users to just upgrade their browser when the degraded version does not fit their needs. However, for most mobile devices upgrading the default browser is not an option.

Progressive enhancement (PE), on the other hand, reverses the graceful degradation approach and aims at maximizing accessibility over browsers with different capabilities [13]. Progressive enhancement tries to achieve this goal by forcing developers to take the less capable devices into account from the very start of the development process. First, a basic markup document is created, providing an optimal experience for devices with the lowest common denominator (LCD) of available capabilities. Incrementally and unobtrusively, one or more layers of structural, presentational, and behavioral enhancements are added in function of the browser's specific capabilities.

The progressive enhancement methodology can be applied in a mobile context to tackle fragmentation related issues. However, when turning the theoretical approach into actual practice, a considerable number of challenges come into play. Today, the use of CSS3 Media Queries [15] and externally linked resources are the most common practice for selecting appropriate enhancement layers. The number of detectable variability points is limited, as adaptation can only be performed based on the device's screen capabilities and coarse-grained styling and scripting support. Compared to desktop browsers, the mobile ecosystem contains far more combinations of browsers with graded CSS and JavaScript support. To provide optimized end-user usability, progressive enhancement should also reckon with the different interaction methods and hardware characteristics offered by mobile devices. For example, a touch-based device will often require an additional presentational enhancement layer, providing a user interface with more space to accurately click buttons, links, etc.

In order to create a viable progressive enhancement solution, it has become increasingly important to support the use of more fine-grained enhancement layers. As shown in Figure 2, an intelligent mechanism is needed, supporting the automated creation of progressive enhancement stacks based on the specific capabilities of a user's mobile device. Within this context, the goal of our research is to

introduce such a mobile progressive enhancement platform. To support developers in the creation of adaptive mobile applications, we propose a method on how to extend existing application frameworks with mobile progressive enhancement capabilities. Furthermore, we introduce an adaptive application composition algorithm, which will be at the heart of composing optimal progressive enhancement stacks. With this approach, we propose a robust and future proof method for the flexible composition of web applications based on the specific capabilities of a user's mobile device.

The remainder of this paper is structured as follows. Section II discusses the algorithmic structure of our approach. Section III deals with the architectural aspects of extending application frameworks with mobile progressive enhancement capabilities. In section IV we discuss the proof of concept implementation of the architecture, followed by some evaluation results. Finally, future work and our conclusion are presented in Section V.

II. ADAPTIVE APPLICATION COMPOSITION ALGORITHM

In this section we propose an adaptive application composition algorithm, realizing the above defined objective to enable the capability-driven progressive enhancement of web applications. Mobile applications should provide users with an optimal experience based on the specific capabilities of their device. In order to cope with the wide variety of mobile characteristics, we introduce a quantitative evaluation algorithm derived from the Logic Scoring of Preference (LSP) method. This adaptive application composition algorithm is designed to support fine-grained progressive enhancement and is capable of suggesting a stack of layers that optimally fits the user's mobile device.

A. Basic Algorithm

LSP is a quantitative decision method, proposed by Dumjovic [3]. It is designed to assist decision makers in the evaluation, comparison, and selection of complex hardware and software systems. The method has shown its use in various domains, especially concerning situations with large and complex solution spaces. To evaluate a set of candidate solutions, LSP starts by assessing n individual performance variables. These variables define the n properties that an ideal solution is expected to have. As the algorithm deals with complex decision problems, most candidate solutions will not perfectly match the preset criteria. Nevertheless, such candidates should not be rejected, as their overall evaluation might still lead to an acceptable solution. LSP addresses this issue by taking into account how well a candidate matches the different performance variables. For each variable i , a degree of suitability $E_i \in [0,1]$ is calculated. This score expresses the similarity between a candidate solution and performance variable i , ranging from 0 to 100%. In order to attain these scores, LSP requires a predefined mapping function for each performance variable

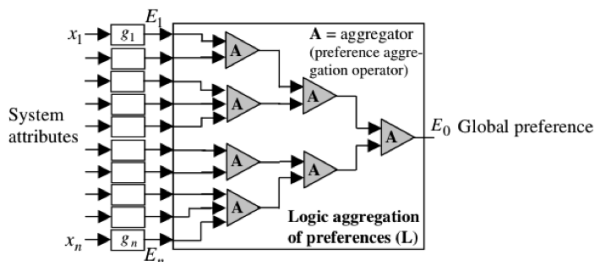


Figure 3. LSP score aggregation in order to derive an overall suitability score from the elementary degrees of similarity (from Dujmovic et al. [4])

[4]. Both Table I and II below illustrate such mapping functions in a mobile context.

After obtaining the elementary degrees of satisfaction, all individual matching scores are to be combined into one objective overall suitability score. This aggregated score is used to determine the best-matching candidate. LSP supports the use of aggregation networks, expressing the mutual relationships between individual scores and how to calculate the overall score (see Figure 3). The standard aggregation operators in LSP are based on the superposition of fundamental Generalized Conjunction Disjunction (GCD) [2]. These operators enable aggregations in terms of partial conjunction, full conjunction, partial disjunction, full disjunction, and neutrality in a single operator. Moreover, a GCD supports the specification of aggregations in terms of 17 graded combinations of conjunction and disjunction. A frequently used implementation for GCD are Weighted Power Means (WPM)

$$WPM(x_1, x_2, \dots, x_m; W_1, W_2, \dots, W_m; r) = (W_1x_1^r + W_2x_2^r + \dots + W_mx_m^r)^{\frac{1}{r}} \quad (1)$$

The variables \$W_i\$ in Equation 1 represent the relative weight for each elementary degree of suitability \$x_i\$, where \$W_1 + \dots + W_m = 1\$. The exponent \$r\$ is determined in function of the aggregation's desired degree of conjunction or disjunction. This approach allows an evaluator to precisely couple the mutual importance of individual suitability degrees. The calculated aggregation network results in an objective overall suitability score

$$E = L(E_1, \dots, E_n) \quad (2)$$

where the function \$L\$ is a combination of one or more GCDs using the individual suitability degrees as input parameters. After calculating \$E\$ for each of the candidates, conclusions regarding the best-matching solution can be drawn. The LSP approach selects the candidate with the highest overall suitability score as the optimal choice.

B. LSP in a Mobile Context

LSP has the ability to flexibly, yet objectively, evaluate systems under various circumstances. We propose a

modification of the LSP method in order to support the adaptive composition of mobile web applications. In this particular case, the stacks of progressive enhancement layers are considered to be the candidate solutions. Each candidate must define the conditions in which it should be able to contribute to an application's optimization and to what extent these conditions are strictly required, or rather optional. As in the standard LSP approach, this degree of desirability is expressed in terms of a GCD.

The available stacks of progressive enhancement layers are in turn individually evaluated by matching their desired conditions to the mobile device's specific capabilities (e.g., available interaction methods, web technology support, etc.). The incorporation of this LSP step in a mobile context requires a defined set of mobile-relevant mapping functions. The mapping functions specify the similarity between performance variables and the actual device capabilities. To illustrate the concept, both Table I and II contain an implementation of a mapping function that compares the performance variable "stylus interaction" with a device's actual interaction method. The function in Table I uses Boolean logic, which implies that only a perfect match is scored. The one in Table II, on the other hand, uses fuzzy logic. The latter approach makes much better use of the available scoring interval by also grading the less-than-perfect matches. Such examples highlight the importance of carefully thought through mapping functions. In this context, the W3C Mobile Web Best Practices Working Group, as well as Web Accessibility Initiative contributed significantly with their efforts in the mobile web usability and accessibility areas [17][18]. The published set of recommendations is an excellent example of a potential source from which we are able to extract usable mapping functions.

Furthermore, the elementary scores that resulted from matching the candidate progressive enhancement stacks with the device's specific capabilities are aggregated into an overall suitability score. The overall score is attained by applying the candidate's predefined GCD network. Once all

Table I
SAMPLE OF A BOOLEAN LOGIC MOBILE MAPPING FUNCTION

Interaction capability	Match
Touch	0 %
Stylus	100 %
Joystick	0 %
Click wheel	0 %

Table II
SAMPLE OF A FUZZY LOGIC MOBILE MAPPING FUNCTION

Interaction capability	Match
Touch	75 %
Stylus	100 %
Joystick	30 %
Click wheel	10 %

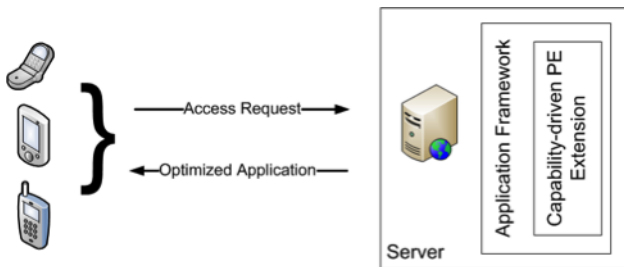


Figure 4. A high level system overview. Integrating the fine-grained progressive enhancement approach within existing web application frameworks

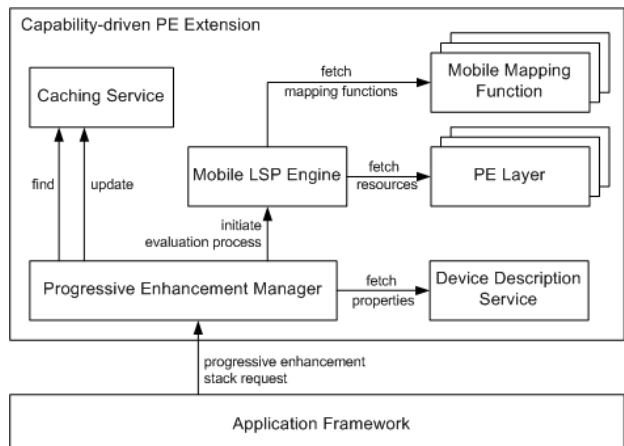


Figure 5. Detailed system architecture. Enabling an application framework to request a stack of progressive enhancement layers that optimally suits the user’s device characteristics

combinations of available candidates have been evaluated, the layer selection process is concluded by selecting the highest scoring solution and applying it to the application.

III. ARCHITECTURE AND DESIGN

This section discusses the architectural structure that is needed to integrate the proposed adaptive application composition algorithm within an application development framework. In general, developers dedicate substantial efforts in mastering a specific application framework. It is thus desirable for our system to support existing frameworks rather than introducing a completely new framework. Hence, support for fine-grained mobile progressive enhancement is provided as a generic plug-in for web application platforms. A high level overview of our approach is depicted in Figure 4. The proposed capability-driven progressive enhancement extension interacts with web application frameworks through a web interface. Calling this interface will generate and return a progressive enhancement stack that is optimized for the particular capabilities of the end-user’s device. In result, application developers are no longer required to manually support the wide range of mobile variability points, as mobile fragmentation issues are handled by the proposed framework extension. What follows is a short elaboration

on the main architectural components of our system, which are shown in Figure 5.

The progressive enhancement manager is the system’s interface to the outside world. It delegates all incoming requests from the application framework to compose an optimized stack of progressive enhancement layers. The manager starts by accessing the Device Description Service component, which returns detailed information on the current device context. The contextual description contains data regarding the capabilities and features of mobile devices (e.g., supported CSS and JavaScript versions, available device APIs, etc.). In turn, the manager addresses the Caching Service component to find out whether the result for this particular device has previously been stored. Due to performance considerations, the caching of results for popular devices is an important part of the system (see performance evaluation discussion in Section IV). In case of a cache-miss, the adaptive application composition algorithm will have to be executed. This process is managed by the Mobile LSP Engine component.

The Mobile LSP Engine is responsible for establishing an optimal progressive enhancement stack. This component is at the heart of the proposed system, as it objectively evaluates and combines the applicability of candidate progressive enhancement layers. The engine starts by fetching all available progressive enhancement layers. Next, the mobile mapping functions are retrieved, specifying the degree of similarity between desired device properties and the actual device capabilities. The Mobile LSP Engine calculates the overall similarity score for each candidate progressive enhancement stack. Finally, the engine then selects the stack with the highest overall score, thus the combination of progressive enhancement layers that best matches the capabilities of the client’s device. This final selection is passed back to the Progressive Enhancement Manager component, which will cache the result and deliver it to the application framework.

IV. EVALUATION

This section evaluates the algorithm and system architecture introduced in Section II and Section III respectively. The proposed capability-driven progressive enhancement extension has been implemented for both the Drupal and Joomla web application frameworks [5][8]. The prototype implementations are used to validate our adaptive application platform objectives and to perform a series of usability and performance evaluations.

A. Proof of Concept Implementation

All architectural components have been implemented for the system’s prototype. The repository of device characteristics in the system’s Device Description Service component is realized by interfacing with the Wireless Universal Resource File (WURFL) [19]. WURFL is an open source project

gathering information on a vast range of mobile delivery contexts. Furthermore, various mobile progressive enhancement layers were created, as a means to validate the system’s capability to adapt to the characteristics of heterogeneous delivery environments. The created enhancements range from simple CSS styling for feature phones, to complex HTML5 and JavaScript layers for high-end smartphones.

B. Usability Evaluation

To evaluate the usability of applications enabled by our system, an adaptive m-commerce application has been built on top of the prototype framework implementations [12]. The application was evaluated using a set of automated usability tests from the W3C MobileOK test suite [16]. The MobileOK service checks the usability of web applications in a mobile context. Tests are based on the validation of markup, accessibility, content and navigation structuring, load time, and the use of network resources. The prototype m-commerce application scores perfect on all MobileOK test. Moreover, the application attains a score in the top 10th percentile of all web applications checked by this W3C service [14].

Applying our fine-grained progressive enhancement approach drastically facilitates the development of accessible and usable mobile applications. Application developer are only required to define of a basic version of their applications, intended for devices with the lowest common denominator (LCD) of capabilities. Figure 6(a) shows the basic LCD structure of the m-commerce application, containing only a simple HTML markup. For more capable devices, such as high-end feature phones, the system detects the applicability of elementary CSS and JavaScript layers. Figure



Figure 6. Adaptive mCommerce web application on two feature phones. (a) The Motorola RAZR, a low-end feature phone and (b) the Nokia N96, a high-end feature phone

6(b) depicts this scenario. The LCD version is automatically enhanced with presentational as well as behavioral layers. Furthermore, the system is capable of adapting itself to the capabilities of smartphone devices. The advanced HTML5, CSS3 and AJAX (Asynchronous JavaScript and XML) support allows the system to select complex enhancement layers that mimic an operating system’s native look-and-feel (see Figure 7(a) and 7(b)).

C. Performance Evaluation

The adaptive application composition algorithm proposed in Section II has a significant influence on the performance of our approach. As the algorithm evaluates the applicability of all possible progressive enhancement stack combinations, running time increases exponentially with the number of candidate progressive enhancement layers in the system. Moreover, if the system contains a set of n candidate progressive enhancement layers, the algorithm is expected to consume

$$O\left(\sum_{i=0}^n \binom{n}{i}\right) = O(2^n) \tag{3}$$

time for completing the evaluation process. Performance tests on the prototype implementation confirm this prediction. Figure 8 shows the average response times of our platform in function of the number of available progressive enhancement layers. Two types of tests are performed. In the first series of tests, caching is disabled, resulting in rapid performance degradation due to the algorithm’s computational complexity. For the second series of tests, on the other hand, the system’s caching capabilities are enabled. The use of a caching mechanism drastically improves performance of our system. Results show that in case of a cache hit,



Figure 7. Adaptive mCommerce web application on two smartphones. (a) An Android smartphone, the HTC Dream and (b) an iOS smartphone, the Apple iPhone

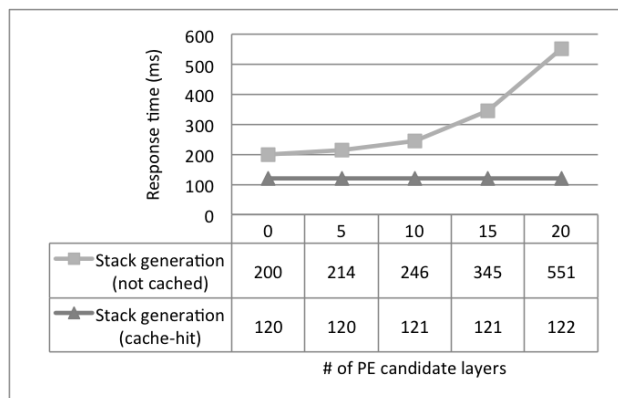


Figure 8. Performance testing results. Response time measured in function of the number of candidate progressive enhancement layers in the system

the generation of a progressive enhancement stack only requires a constant execution time, regardless of the number of candidates that have to be evaluated.

V. CONCLUSION AND FUTURE WORK

In this paper, we introduced a platform in support of developing and delivering adaptive mobile web applications. The proposed method can serve as a basis for developers to create and maintain a single version of their mobile application, without being limited by fragmentation related issues. Our adaptive application composition algorithm is at the hearth of the proposed approach. The algorithm is based on a quantitative evaluation algorithm derived from the Logic Scoring of Preference (LSP) method. The proposed approach enables the automated and fine-grained progressive enhancement of web applications. The process is entirely driven by the characteristics of the user's device, in order to provide an optimal user experience.

While the extensive evaluation of our approach has yet to be carried out, initial testing of prototype implementations showed promising results. Future work includes the further validation of our proposed approach as well as the extension of our algorithm towards supporting real time application request handling. Other necessary steps in the development of this application composition method are related to broadening the scope on supported device types. The diversity of devices that enable access to third party software applications is currently extending towards home entertainment systems, automotive, etc. Such evolution further emphasizes the impact of fragmentation on application developers. This is why the applicability of our proposed application composition method should be evaluated for more ubiquitous environments.

REFERENCES

[1] T. Ahonen. *Mobile As 7th of the mass media: Cellphone, cameraphone, iPhone, smartphone*. London: Futuretext, 2008.

- [2] I. Batyrshin, O. Kaynak, and I. Rudas. "Generalized conjunction and disjunction operations for fuzzy control." in *Proc. of 6th European Congress on Intelligent Techniques and Soft Computing EUFIT'98*, 1998, pp. 52-57.
- [3] J.J. Dujmovic. "A method for evaluation and selection of complex hardware and software systems." in *Proc. of Int. Conference for the Resource Management and Performance Evaluation of Enterprise Computing Systems*, 1996, pp. 368-378.
- [4] J.J. Dujmovic, G. De Tre, and N. Van de Weghe. "LSP suitability maps." *J. Soft Computing*, vol. 14, pp. 421-434, 2010.
- [5] "Drupal." Available: <http://www.drupal.org> [Sep. 1, 2011].
- [6] G.R. Frederick, and R. Lal. "The future of the mobile web." in *Beginning smartphone web development*, Springer, 2009, pp. 303-313.
- [7] International Telecommunication Union. *Measuring the Information Society 2010*. Geneva, Switzerland: ITU, 2010.
- [8] "Joomla." Available: <http://www.joomla.org> [Sep. 1, 2011].
- [9] G. Kappel, B. Proll, W. Retschitzegger, and W. Schwinger. "Modeling ubiquitous web applications: the WUML approach." in *Conceptual Modeling for New Information Systems Technologies*, Springer, 2002, pp. 183-197.
- [10] N. Koch, A. Knapp, G. Zhang, and H. Baumeister. "UML-Based web engineering." in *Web engineering: modeling and implementing web applications*, Springer, 2008, pp. 157-191.
- [11] A. Schauerhuber, M. Wimmer, M. Schwinger, E. Kapsammer, and W. Retschitzegger. "Aspect-oriented modeling of ubiquitous web applications: the aspectWebML approach." in *Proc. of Int. Conference on the Engineering of Computer-Based Systems*, 2007, pp. 569-576.
- [12] "WAFL Adaptive e-commerce demonstration." Available: <http://www.wafl.ugent.be/demonstration> [Sep. 1, 2011].
- [13] J. Wells, and C. Draganova. "Progressive enhancement in the real World." in *Proc. of 18th conference on Hypertext and hypermedia HT2007*, 2007, pp. 55-56.
- [14] W3C. "MobileOK checker usage statistics." Available: <http://www.w3.org/2008/04/mokstats/public/global/20080401-20100110/Overview.html> [Sep. 1, 2011].
- [15] H.W. Lie, T. Celik, D. Glazman, and A. van Kesteren. "Media Queries." Available: <http://www.w3.org/TR/css3-mediaqueries> [Sep. 1, 2011].
- [16] J. Rabin, and P. Archer. "MobileOK Scheme." Available: <http://www.w3.org/TR/mobileOK> [Sep. 1, 2011].
- [17] W3C. "Mobile Web Best Practices Working Group." Available: <http://www.w3.org/2005/MWI/BPWG> [Sep. 1, 2011].
- [18] W. Chrisholm, G. Vanderheiden, and I. Jacobs. "Web Content Accessibility Guidelines." Available: <http://www.w3.org/TR/WAI-WEBCONTENT> [Sep. 1, 2011].
- [19] L. Passini. "Wireless Universal Resource File." Available: <http://wurfl.sourceforge.net> [Sep. 1, 2011].