SMART UBIQUITOUS APPLICATION DELIVERY BASED ON USABILITY RULES

Dieter Blomme, Heiko Desruelle, Frank Gielen and Filip De Turck
Ghent University – IBBT
Dept. of Information Technology – IBCN, Ghent Belgium
{dieter.blomme; heiko.desruelle; frank.gielen; fillip.deturck}@intec.ugent.be

ABSTRACT

Today there is a growing diversity in devices that support software applications in sectors such as mobile, automotive and home entertainment systems. Developing and testing an application for each existing and future device is inefficient and not scalable. We state that the web with its standardized technologies and powerful components will be the converged platform for ubiquitous application delivery. In this paper we propose a novel approach to ubiquitous application delivery using usability rules that combine the delivery context and web application semantics. We give an overview of the parameters that need to be taken into account when delivering applications ubiquitously.

KEYWORDS

Ubiquitous, mobile, Semantic Web

1. INTRODUCTION

Mobile is predicted to be the seventh mass medium (Ahonen 2008). Automotive devices and home entertainment systems also start to have internet access capabilities. The anytime/anywhere-paradigm envisioned at the start of the internet is finally becoming a reality due to the ubiquitous nature of internet enabled devices, cloud computing and web service mash-ups.

However, it is not efficient to redesign, develop and test an application for each type of existing and future device that can access cloud applications. The web will be the platform to support application development for ubiquitous devices. The combination of standardized technologies such as HTML, CSS and Javascript with powerful rendering and interpreter components ported to all these devices will enable true convergence. But even with the standardized nature of web technologies, the device and browser diversity on the market today make it almost impossible to create web applications that are optimized for every device.

On the other hand, a lot of these devices have additional capabilities that allow them to provide extensive context information. On device sensors such as GPS, motion sensor, camera, microphone, etc. allow a web application to be tailored not only to the target device, but also to the user using the web application.

The goal of our research is to provide *a smart ubiquitous application delivery platform* from the cloud to a broad range of devices in a seamless manner.

1.1 Research Challenge

In order to achieve our goal, we need to tackle a number of issues and challenges.

- Device diversity entails a broad range of presentation and interaction capabilities. The application should adapt it's look-and-feel and interaction design to match these capabilities.
- Ubiquitous devices, especially mobile phones, have the possibility to capture extensive context information. The behaviour of the application has to be changed to fit the delivery context.
- The behaviour of the application can also be augmented by taking the additional capabilities of the device, such as the GPS for location-aware mashups, into account.

1.2 Proposed Method

We propose an approach that guarantees an optimal ubiquitous application delivery based on the following principles:

Context detection Ubiquitous context identification goes beyond the traditional device detection which focuses on hardware parameters like screen size. To provide the user with an optimal experience, detection of software capabilities such as support for web technologies and available API's and detection of user settings such as security profiles are needed.

Semantic analysis Previous efforts to automatically transform content for the mobile internet have had little success because they are only looking at the format and the structure of content. We base our application delivery on a semantic analysis of the web application. This will allow reasoning and dynamic application delivery that matches the delivery context of the device and user.

Usability The key success factor for ubiquitous web application is usability. We use a rule based system to match the semantic data from the application with the usability guidelines that match the delivery context. Transformations for web applications can be divided in two categories: object-level and structure-level transformations. The first type handles individual blocks in the application with a specific purpose on a page. Examples are a menu, a login form and input fields. The second type are transformations that redefine the structure of a web application. This can be splitting out content over multiple pages, rearranging the content on an individual page, rearranging table cells to fit the table to the width of the screen or changing a process flow.

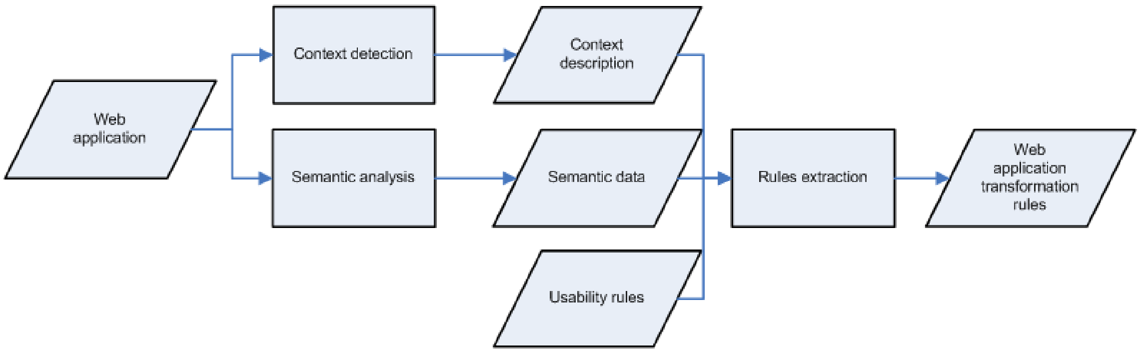

Figure 1: The proposed method.

2. STATE OF THE ART

This section gives an overview of prior research in our domain on which our work will build.

Semantic page analysis Some works try to divide a page in segments based on their semantic relationships (Xiang, P. 2006; Baluja, S. 2006). By using the structured data from the web page and visual hints extracted from a rendered version of the page, it is divided into it's structural blocks.

Context delivery Mobile systems offer large possibilities for context-aware applications. By using microformats, small object-level transformations can be made to adapt community dependent data (e.g. dates, addresses) to the local semantics (Mrissa, M. et al. 2008). The Web 2.0 trend of mashups can be extended to creating context-aware mashups targeted to mobile devices (Brodt, A. et al. 2008). Personalization in general web-based information systems (Tvarozek, M. et al. 2007) can also be used and extended to take a mobile device's added capabilities into account.

Usability Vigo, M. et al. 2008 provide a method to evaluate web pages adapted for mobile, using the MobileOK tests from the W3C as a basis. These tests are extended to take the device characteristics into account. Device features are retrieved from heterogeneous repositories.

3. SEMANTICS FOR (X)HTML

An optimal transformation of a web application can only be obtained when the source HTML file contains unambiguous, machine-interpretable semantic information about both the object and structure level.

At the present, there are three methods to annotate or enhance an HTML page, that are both standardized (or are in the process of standardization) and are already being used in existing web applications: Microformats, RDFa an HTML 5. Based on these methods, we develop a fourth, novel approach for our system.

Microformats Microformats are tiny pieces of information that are injected into (X)HTML code by use of the class, rev and rel attributes. This way it promotes data reuse while reusing existing technologies. It is the most popular way to provide semantic annotations on a web page. Web 2.0 sites such as Twitter, Flickr, LinkedIn and Yahoo have already adopted microformats.

Microformats are developed according to an open standard called the microformat specifications (Allsop, J. 2007; Khare, R. 2006). The invention of custom microformats outside of the community is discouraged, meaning a strict but limited vocabulary is available. This makes it easy to parse, but limits the extendibility in terms of our problem.

RDFa RDFa (Adida, B. et al. 2008a; Adida, B. et al. 2008b) is a modular extension of XHTML based on RDF, developed and proposed by the W3C. It allows a set of XML attributes to carry metadata about values in the human-readable text of a HTML page. Because it uses XML specific properties, such as namespaces, it is unusable for the non-XML compliant HTML standard. RDFa is based on external domain vocabularies and thus promotes data reuse. It is still in an early adoption phase, but the number of RDFa enabled websites will rise quickly with the integration in the new versions of popular content management systems such as Drupal.

Unlike microformats, RDFa provides publisher independence and self containment. It allows application developer to develop and publish their own vocabularies, extend others and let a vocabulary evolve over time while still allowing maximum interoperability. The HTML and RDFa data are also separated.

HTML5 with microdata HTML 5 is a standard in development by the W3C and is intended to be the successor for both HTML 4.01 and XHTML 1.0. It aims to reduce the need for plug-in based RIA technologies such as Flash, Silverlight and JavaFX, indirectly improving the possibility to search and adapt rich internet applications.

It provides semantics, by replacing certain generic block and inline elements with new semantically improved tags. This will provide extended information about the structure of a web application, allowing an improved structure-level adaptation. Because the HTML5 standard is work in progress and is now in last call phase, browser support is still limited and the number of HTML 5 sites is small. Microdata (Anonymous 2010) is a feature of HTML5 that provides another way to embed semantic annotations.

Meaningful class attributes Besides microformats, the aforementioned technologies don't enjoy a widespread use at this moment. Microformats itself only provides a limited and strict vocabulary. This gives us a very limited data set for testing. Therefore we made the choice to develop a fourth approach for semantic annotations. A lot of sites already provide meaningful class attributes to indicate semantic properties of the content as a way to separate content from presentation. By using CSS presentation can be associated to content based on semantics.

This approach fulfills our requirements. By developing a custom vocabulary, extensibility is ensured. Reusing the existing class attributes in sites promotes data reuse. A lot of sites already have meaningful class attributes, so widespread use is provided. The key quality requirement performance can also be ensured, because a lot of high performance (X)HTML and CSS parsers already exist.

4. DEVICE CATEGORIZATION

A user only perceives an application as usable when the interaction is smooth and doesn't provide frustrations. This makes it important to be able to categorize devices according to their interaction mechanisms. This chapter gives an overview of the most important interaction methods.

- The oldest type of interactive devices have a limited *keypad* and a set of *directional buttons*. In mobile, these devices are the so-called feature phones, mostly associated with the 2G GSM networks. A similar apparatus for page traversal is the remote control of an internet-enabled TV-set.
- Some devices use a *trackball*-based system to support interaction. This results in more degrees of freedom for page traversal. This navigation method can be found on a lot of blackberry phones and control sticks on most game consoles.
- Since the introduction of the iPhone, the era of *touch* phones has begun. Touch phones can be divided in two categories: *stylus* and *finger* based touch interaction. Some automotive interfaces also provide a

finger based touch interaction. Finger based interaction requires a completely different interface than a regular desktop website. At present day, most types of smartphones (iPhone, Android, WebOS, Symbian[3], etc.) even provide *multitouch* capabilities.

– The next steps in interaction method are *multimodal interfaces*. The device sensors and actuators can be used as additional input and output mechanisms, e.g. the camera to take a picture of a business card for contact information.

5. USABILITY RULES

The quality attributes that are deemed most important are usability, performance and modifiability. Based on this, the design of our top level architecture will be based on a rule based system to determine the adaptation process for a specific delivery context. The rule-base is extensible, providing the modifiability of the transformation pipeline.

We focus our efforts on object-level transformations. Based on the use cases from the previous section, we can define two types of transformations:

Base level transformations that adapt the HTML code to the target runtime. This is needed e.g. for the transformation of the menu to either a selection box or a list.

Progressive enhancement transformations that add CSS and/or javascript to the web application, optimizing it for the target delivery context. For example, javascript can be used to make the menu list expandable.

The input parameters for both transformations are (1) the semantic annotations for the object and (2) the delivery context parameters. A set of rules that are expressed as formal logic statements will select the appropriate usability rule.

The usability of the application will be guaranteed by creating a correct and complete set of rules. Rule based systems have a negative impact on performance due to a slow selection process. However, by using known variables and a conflict resolution strategy combined with the RETE algorithm, the performance of the selection engine can be significantly improved .

Figure 2 gives an example in pseudocode that indicates the transformation of a menu to a selection box if the browser does not support XHTML and only provides limited CSS2 support.

```
( object.class.contains(menu)
   AND !browser.htmlsupport.contains(XHTML)
   AND !browser.csssupport(CSS2 FULL) )
   =>
   apply(transformation.menuToSelectionBox)
```

Figure 2. Usability rule: transformation of a menu to selection box.

The second example (see Fig. 3) shows the addition of a stylesheet that makes sure all pointer elements on a finger touch based device have a minimum size.

```
Device.interaction.contains(Touch)
   =>
   attach(stylesheet.touch)
```

Figure 3. Usability rule: making the interface touch friendly.

6. USE CASE

One important use case was selected to provide an overview of progressive enhancements techniques for ubiquitous application delivery: a form which is an interaction object.

Desktop forms are designed for interaction methods based on a keyboard and mouse with field that require text input. On devices with limited input capabilities, filling out the same forms will be time-consuming and frustrating.

Two basic solutions can be used individually or combined, based on the type of device. One solution *splits the form in multiple pages*. Form data should be stored in between browsing sessions, because it is likely a user only fills out part of these pages during a short browsing session and wishes to continue the form entry at a later time.

Another solution is to *hide the optional fields* from the view. On low-end devices these fields can be put on optional, separate pages. On middle- and high-end devices this can be done using javascript to hide the optional fields from view, while still allowing the user to show them.

Automatic form-filling can be used as a layer on top of the previous solution. Ajax can be used to provide autocompletion, e.g. to show possible values for the input field. The added capabilities of some devices also allows automatic form-filling for certain types of fields. Possible values for address fields can be shown based on context information such as the user's current location or his home address.

Automotive systems should limit user interaction, so the required input does not distract the driver. *Home entertainment systems* have a very limited typing interface, usually only a remote control with a keypad and directional buttons. The interaction for these devices can be optimized by displaying only the most basic form, e.g. one with only dropdown boxes.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have developed a method for smart ubiquitous application delivery. Our approach is based on context detection that goes beyond the traditional device detection and a semantic analysis of the web application. Our method is based on a rule based system to determine the adaptation process for a specific delivery context. The rules are expressed as formal logic statements composed using known variables to increase the performance of the selection process.

Future research will include several directions. First, we will extend the rule base of our system with additional object-level and structure-level transformations. A next step is the development of the semantic analyzer for further automation of the ubiquitous delivery process. We will also investigate techniques that allow the adaptation of application logic to the delivery context.

REFERENCES

Adida, B., Commons, C. 2008a. *Rdfa in xhtml: Syntax and processing.* [Online] Available from:<http://www.w3.org/TR/rdfa-syntax>

Adida, B., Birbeck, M. 2008b. *Rdfa primer* [Online] Available from: <http://www.w3.org/TR/xhtml-rdfa-primer/>

Ahonen, T., 2008. *Mobile As 7th of the Mass Media: Cellphone, Cameraphone, Iphone, Smartphone*. Futuretext, London, UK

Anonymous. 2010. *Microdata* [Online] Available from: <http://www.whatwg.org/specs/web-apps/currentwork/
multipage/microdata.html>

Baluja, S., 2006, Browsing on small screens: recasting web-page segmentation into an eficiente machine learning framework. *Proceedings of the 15th international conference on World Wide Web*. Edinburgh, Scotland, pp. 33-42

Brodt, A., Nicklas, D., Sathish, S., Mitschang, B., 2008. Context-aware mashups for mobile devices. *Lecture Notes in Computer Science*, Vol. 5175, pp. 280-291

Fling, B., 2009. *Mobile Design and Development*. O'Reilly Media, Sebastopol, USA

Khare, R., 2006. Microformats: the next (small) thing on the semantic web? *IEEE Internet Computing*, Vol. 10, No. 1, pp. 68-75

Mrissa, M., Al-Jabari, M., Thiran, P., 2008, Using microformats to personalize web experience. *Proceedings of the 7th International Workshop on Web-Oriented Software Technologies*. New York, USA, pp. 57-62

Tvarozek, M., Barla, M., Bieliková, M., 2007. Personalized presentation in web-based information systems. *Lecture Notes in Computer Science*, Vol. 4362, pp. 796-807

Vigo, M., Aizpurua, A., Arrue, M., Abascal, J., 2008, Evaluating web accessibility for specific mobile devices. *Proceedings of the 2008 international cross-disciplinary conference on Web accessibility*, Bejing, China, pp. 65-72

Xiang, P., Yang, X., Shi, Y., 2006, Effective page segmentation combining pattern analysis and visual separators for browsing on small screens. Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, Hong Kong, China, pp. 831-840