W3C WIDGETS
A Solution for Implementing Platform-independent Mobile Applications

Stefan Schäfer, Stefan Christmann
Application Systems and e-Business, University of Göttingen, Platz der Göttinger Sieben 5, Göttingen, Germany

Svenja Hagenhoff
Institute of Media Management, University of Applied Sciences St. Poelten, St. Poelten, Austria

Keywords: Widget, Mobile application, Platform-independency.

Abstract: The mobile internet is very heterogeneous at all levels - this is true for end devices as well as for operating systems and runtime environments of mobile applications. Therefore all software providers face the challenge of developing platform-independent applications and making them usable on a maximum number of end devices. One possible way of doing this is provided by Widgets - small software programs with a limited functional scope that are executed by a Widget-engine. This article looks at the state of development of the most important W3C Widget-standards and analyses their prospects of success.

1 INTRODUCTION

The mobile internet poses great challenges for software providers as well as for customers, because it is a strongly fragmented market with a multitude of different producers of end devices and platforms for the production and supply of software (Gartner, 2010; Admob, 2010, p.6). Customers who change to a new device often find it difficult or even impossible to use their former applications on the new end device and to transfer their data. This is also a problem for the producers of software, as they have to provide a great variety of software for different platforms in order to retain customers. This leads to a high demand on development resources and subsequently to increased software production costs.

Solutions for this are projects like PhoneGAP or Appcelerator, which translate web applications written in (X)HTML, CSS and JavaScript into native applications for mobile devices (PhoneGap, 2010). J2ME Polish – as an alternative – is an open source extension for Java Micro Edition (JavaME) which makes it possible to design the user interface of JavaME-based applications through CSS. It also includes a database of devices providing an easy way for developers to adapt applications to several hundreds of mobile phone types (Müller & Reiprich, 2008, p.2). Both products are mainly solutions for the existing variety of different platforms. W3C, on the other hand, tries to counterbalance the great variety of software development solutions by its Widget-standard and to find platform-independent solutions for applications with a low functional scope in order to access a maximum number of end users.

Jaokar and Fish (2006, p.99) define a Widget as a "downloadable, interactive software object that provides a single service such as a map, news feed etc". "Widget" is an artificial word combining *window* and *gadget.* This goes back to the use of Widgets in the MIT Athena Project, where a window was associated with an object and called Widget (McCormack, et al., 1989, p.4). The W3C defines Widgets as applications executed on the client side, but developed using web standards. Widgets are usually downloaded and installed on an end device where they run as independent applications. They may also be integrated into web sites and accessed through a web browser (Cáceres & Priestley, 2009). Therefore, a great variety of runtime environments is available. According to the W3C definition of Widgets they may be called mobile applications if they are executed independently on a mobile end device or in a web browser. Such a runtime environment is called a Widget-engine. It is software that can execute small applications (Widgets) in

Schäfer S., Christmann S. and Hagenhoff S..
W3C WIDGETS - A Solution for Implementing Platform-independent Mobile Applications.
DOI: 10.5220/0003196101150118
In *Proceedings of the 7th International Conference on Web Information Systems and Technologies* (WEBIST-2011), pages 115-118
ISBN: 978-989-8425-51-5

their context and present them to the user (Sachse, 2010, p.7).

2 W3C WIDGET-STANDARD

In August 2006 W3C published a document called Web Applications Packaging Format Requirements, defining the requirements towards a uniform packaging format for web applications (Cáceres, 2006a). This working group further specified the requirements, leading to the final requirement document Widgets 1.0 Requirements. Also the elements for the execution of Widgets were identified (cf. Figure 1).

Technologies such as XMLHttpRequest or ECMAScript (also called JavaScript) are well-established and need no further standardization in the process of Widget-specifications. However, the packaging format for Widgets, a uniform MIME-type (media type), content and construction of the configuration document, and an API for Widgets need to be standardized. This Widget-API does not aim at providing access for device-specific functions, but only access to the Widget and its meta data. Over time, different working groups were established for the development of device-specific functions.

In 2008 the Web Application Formats Working Group of W3C was merged with the Web API Working Group to form the Web Applications Working Group. Its task is to further develop the specifications on APIs and Widgets. This working group has developed the following specifications on Widgets, which can be regarded as the main content of the standard (W3C, 2010).

Widget Packaging and Configuration (P&C): This specification stipulates, among other things, the structure of configuration files, the packaging, the inner structure and the possibilities for the internationalization of Widgets.

The Widget Interface (TWI): An API that facilitates the access to metadata of the Widget such as author or name in a simple way and provides a DOM-method to access URLs within Widgets.

Widget Access Request Policy (WARP): WARP is a safety model that regulates the access to network resources through settings of the configuration file and through providing methods that provide a wider access of Widgets to network resources.

Digital Signatures for Widgets: This specification describes how Widget packages may be digitally signed using XML signature syntax in order to guarantee the reliability of the Widget's source.

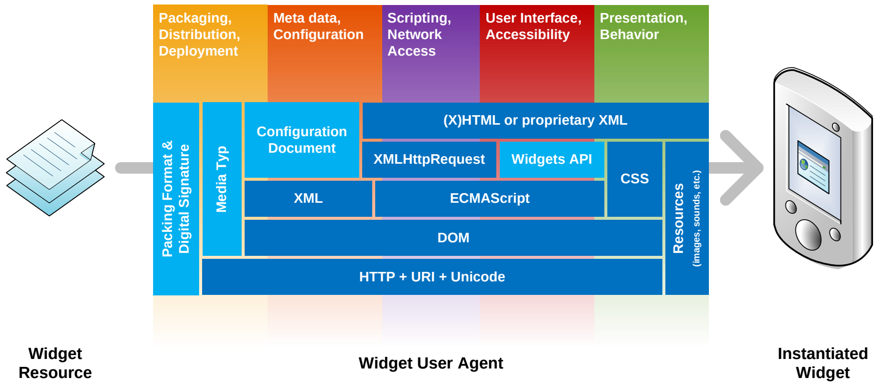
Widget URIs: This specifies the addresses through which resources within Widgets may be identified and accessed.

View Mode Media Feature: This Feature describes how Widgets behave in certain views. As Widgets are executed on a great variety of end devices, the space available on the screen varies greatly. On the other hand, they may be executed for example as full-screen applications but also as minimized applications taking only a small part of the screen.

Widget Updates: This specification describes the update mechanism that is needed in order to distribute new versions of Widgets to users.

3 IMPLEMENTATION AND USE

When the requirements towards a Widget-standard were stipulated, the existing Widget-landscape was analyzed and a multitude of Widget-platforms was identified (Cáceres, 2008). The existing implementations are described in the following chapter.



light blue areas require standardization

Figure 1: Required elements for running Widgets (cf. Cáceres 2008).

3.1 Implementations

Currently, implementation experiences are gathered for the Widget specifications P&C, TWI and WARP. For each implementation there is a number of test cases and an implementation report (Cáceres, et al., 2010). BONDI's Widget-engine was the first to fulfill all test cases of the Widget-standard 100%. Aplix Web Runtime and the Wookie-Project of Apache were the next to fulfill the standard 100% (W3C, 2009). Moreover, there are other implementations with expanded functionalities. Table 1 gives an overview.

BONDI is an initiative started by Open Mobile Terminal Platform (OMTP) in 2008. OMTP is a forum created by network providers with a total of 38 participants from phone producers to chip providers and software/operating systems developers (OMTP, 2010). BONDI is not an acronym but is named after the Bondi Beach in Sydney. In addition to implementing the Widget standard, BONDI provides new interfaces to access the functions of mobile phones. These are functions such as localization, start of programs and access to the address book in a secure manner. The platforms currently supported include Windows Mobile, Android, BREW, iOS, WebOS, BlackBerryOS and Symbian. BONDI belongs to the mobile sector.

Aplix Web Runtime is a runtime environment for Widgets for the systems Android (Google) and Maemo (Nokia). It supports the JavaScript-APIs by BONDI and JIL (Judge, 2010). Aplix Web Runtime also belongs to the mobile sector.

Wookie is a server application providing an opportunity to upload and provide Widgets. The Wookie-project was supported by the IST-program of the European Union as a part of the TENCompetence project and was exclusively developed by developers from the University of Bolton. The Wookie-server is an application developed in Java with a Widget-engine and other plug-ins that can integrate Widgets into web applications such as Wordpress or the course management system Moodle (Wilson, et al., 2009). Wookie is an implementation for web applications that facilitates the embedding of Widgets into web sites.

3.2 Dissemination and Acceptance

When the first document with requirements towards Widgets was published, it triggered very mixed reactions. They ranged from enthusiasm about the possibility to write a single application and use it on several platforms to doubts whether platform-independent standards would be acceptable to total rejection. In general, however, the first attempt at overcoming the fragmentation in Widget-development was favourably received (Cáceres, 2006b). Many of the important platforms for mobile end devices appreciate and support the Widget-standard, as shown by a great number of implementations. However, there are not yet any implementations for Apple devices. Besides the Aplix Web Runtime, there is no other implementation for the Android-platform. Very probably, Aplix Web Runtime will not be available as open source, but will be licensed to network providers (Wilson, 2010). Provided a wider acceptance and distribution of the Widget-standard, Apple and Google may well yield to the public pressure and develop more Widget-engines implementing the W3C standard for Apple as well as for Android platforms.

4 CONCLUSIONS

The clear advantage of W3C Widgets is that they are platform-independent and therefore universally applicable. They can be used as small desktop applications, as mini-applications integrated into web sites or as part of a Widget-engine running on mobile end devices. Meanwhile, developers from

Table 1: Available Implementations of W3C Widget Standard.

Implementation	Implementation of standard	Area	Runtime Environment
BONDI	100% compatible	Mobile	E.g. Windows Mobile, Android, BREW, iOS, webOS, BlackberryOS.
Applix Web Runtime	100% compatible	Mobile	Android, Maemo.
Wookie	100% compatible	Web	Java-based operating systems, web browsers.
Palette-Portal	divergent/extended	Web	Web browsers; integrated into learning portal.
Opera	divergent/extended	Mobile & Web	Stationary operating systems with Opera browser starting from version 9.0, Windows Mobile, S60.

many areas and companies are involved in the development of the standard. One of the great disadvantages of the Widget-standard is the lack of device APIs. The Device APIs and Policy Working Group is working on a specification, but there have been no tangible results yet. Although Widgets are supposed to be mini-applications with just a small scope of functions, they have the disadvantage of not including any code that is executable on the server side and can therefore not generate dynamic content without using JavaScript. This disadvantage can be compensated partly by providing JavaScript within the Widget-engine.

The implementation and use of the W3C Widget-standard by major companies and consortia is of vital importance for its success. Furthermore, a generally accepted specification of device APIs is an indispensable condition for the harmonization of mobile applications. The markets for mobile operating systems and browsers currently undergo diverging developments. While in the area of browsers, systems based on the rendering engine WebKit develop into the dominant platform and fragmentation is decreasing, in the area of operating systems the publishing of a growing number of new systems leads to more and more fragmentation in this area. The lower level of fragmentation in the browser area could promote a development trend competing with W3C Widgets: the use of web browsers as runtime environment. The new standard HTML5 makes it possible to continue using web applications offline using local SQL databases (W3C, 2008). The future of the W3C Widget-standard remains open. If it is consistently developed and supported by the producers of operating systems, it may become a runtime environment for mobile applications to be reckoned with.

REFERENCES

Admob, 2010. *AdMob Mobile Metrics Report* (Online) Available at: http://metrics.admob.com/wp-content/uploads/2010/05/AdMob-Mobile-Metrics-Apr-10.pdf (Accessed 26 May 2010).

Cáceres, M., 2006a. *Web Applications Packaging Format Requirements* (Online) (Updated 21 August 2006) Available at: http://www.w3.org/TR/2006/WD-WAPF-REQ-20060821/ (Accessed 28 May 2010).

Cáceres, M., 2006b. *Widgets 1.0 responses and thoughts* (Online) (Updated 17 November 2006) Available at: http://datadriven.com.au/2006/11/widgets-10-responses-and-thoughts/ (Accessed 3 June 2010).

Cáceres, M., 2008. *Widgets 1.0: The Widget Landscape* (Online) (Updated 14 April 2008) Available at: http://www.w3.org/TR/widgets-land/ (Accessed 3 June 2010).

Cáceres, M. & Priestley, M., 2009. *Widgets 1.0: Requirements* (Online) (Updated 30 April 2009) Available at: http://www.w3.org/TR/widgets-reqs/ (Accessed 20 April 2010).

Cáceres, M., et al., 2010. *Implementation Report: Widgets Packaging and Configuration* (Online) (Updated 9 February 2010) Available at: http://dev.w3.org/2006/waf/widgets/imp-report/ (Accessed 3 June 2010).

Gartner, 2010. *Gartner Says Worldwide Mobile Phone Sales Grew 17 Per Cent in First Quarter 2010* (Online) (Updated 19 May 2010) Available at: http://www.gartner.com/it/page.jsp?id=1372013 (Accessed 25 Mai 2010).

Jaokar, A. & Fish, T., 2006. *Mobile Web 2.0: The Innovator's Guide to Developing and Marketing Next Generation Wireless/Mobile Applications*. London: futuretext.

Judge, S., 2010. *Android W3C Widgets* (Online) (Updated 16 April 2010) Available at: http://www.mobilephonedevelopment.com/archives/1022 (Accessed 31 May 2010).

McCormack, J., et al., 1989. *X Toolkit: The Intrinsics & Athena Widgets.* Cambridge: Silicon Press.

Müller, N. & Reiprich, T., 2008. *J2ME Polish* (Online) (Updated 25 February 2008) Available at: http://www.mi.fh-wiesbaden.de/~barth/mobile/ws0708/J2MEPolish.pdf (Accessed 12 May 2010).

OMTP, 2010. *Current Members* (Online) Available at: http://www.omtp.org/Membership.aspx (Accessed 31 May 2010)

PhoneGap, 2010. *About PhoneGap* (Online) Available at: http://www.phonegap.com/about (Accessed 12 May 2010).

Sachse, J., 2010. *The standardization of Widget-APIs as an approach for overcoming device fragmentation.* Berlin: Grin.

W3C, 2008. *W3C: Offline Web Applications* (Online) (Updated 30 May 2008) Available at: http://www.w3.org/TR/offline-webapps/ (Accessed 30 May 2010).

W3C, 2009. *Twitter Account widgetspecs* (Online) (Updated 28 November 2009) Available at: http://twitter.com/widgetspecs (Accessed 4 June 2010).

W3C, 2010. *PubStatus – WEBAPPS* (Online) (Updated 21 May 2010) Available at: http://www.w3.org/2008/webapps/wiki/PubStatus#Widget_Specifications (Accessed 28 May 2010).

Wilson, S. B., 2010. *W3C Widgets + Android* (Online) (Updated 16 April 2010) Available at: http://metadata.cetis.ac.uk/members/scott/blogview?entry=20100416085805 (Accessed 4 June 2010).

Wilson, S. B., et al., 2009. *Wookie Proposal* (Online) (Updated 20 September 2009) Available at: http://wiki.apache.org/incubator/WookieProposal (Accessed 31 May 2010).