Will Personal Learning Environments Become Ubiquitous through the Use of Widgets?

Behnam Taraghi
(Graz University of Technology, Graz, Austria
b.taraghi@tugraz.at)

Herbert Mühlburger
(Graz University of Technology, Graz, Austria
muehlburger@tugraz.at)

Martin Ebner
(Graz University of Technology, Graz, Austria
martin.ebner@tugraz.at)

Walther Nagler
(Graz University of Technology, Graz, Austria
walther.nagler@tugraz.at)

Abstract: Nowadays Personal Learning Environments (PLEs) are described as the use of different social software tools for learning and teaching. The use of common web browsers often is a precondition to fulfil the requirements of social software tools. In this publication a new perspective as well as a couple of prototypes is presented to allow a more independent solution in this context. Using the new JavaFX technology so called widgets are implemented which can be used on different devices as well as different operating systems. Furthermore also the web browser can be replaced by a browser instance. It can be summarized that through the use of widgets the learning environment of the future can be more individualized and personalized.

Keywords: uLearning, Personal Learning Environment, Widget, learning
Categories: H.3.5, K.3.1, L.3.0, L.3.6

1 Introduction

Web 2.0, first announced by Tim O'Reilly [Reilly, 05], changed the way humans are dealing with the Web. So called social software is widely used to connect people of same interest and helps to exchange thoughts, opinions by sharing and collecting.
Bearing in mind these new possibilities for communication and collaboration also learning and teaching is changing radically. Stephen Downes described it as e-Learning 2.0 [Downes, 05] and pointed out that a future learning environment "becomes, not an institutional or corporate application, but a personal learning center, where content is reused and remixed according to the student's own needs and interests. It becomes, indeed, not a single application, but a collection of interoperating applications—an environment rather than a system". In the same way Wilson [Wilson, 05] is talking about a user-centred Personal Learning Environment

(PLE). As follow up numerous publications are dealing with the idea of a highly individualized system. Harmelen [Harmelen, 06] expressed that Virtual Learning Systems (VLE) and Learning Management Systems (LMS) are not dealing well with the needs of the learners. Without any doubt from a teacher and learner perspective, the personalization of educational environment will enhance the learning and teaching behaviour. But there are still some problems we have to overcome.

Introducing PLEs by using common social software as mentioned by some research works [Attwell, 07a] [Attwell, 07b] [Schaffert, 08] is maybe not satisfying due to the fact of different restrictions: On the one side the use of different web applications often implicates problems of service stability, accessibility and data protection. On the other side, the implementation of own systems needs high effort and technical support [Green, 06] [Pearson, 08]. A very interesting approach seems to be the use of Mash-Up technology for PLEs as described by Wild [Wild, 08]. But even this approach narrows the user down to a web browser. Due to the fact that ubiquitous learning (u-Learning) is defined as [Zhan, 05]

u-Learning = {u-Environment, u-content, u-Behaviour, u-Interface, u-Service}

interoperability will be one of the major components of personalisation.

In this publication we describe the use of standardized widgets technology and a first idea of how JavaFX (see section 2.3) can be used successfully for a future device-independent PLE.

2 Widgets

Widgets (in some applications also known as gadgets) are small client side applications than run on desktop or a web page. In other words widgets are a portable chunk of code that can be installed and executed on any html based webpage or on a desktop. The code is hosted on the client side. It can be a simple JavaScript code or a java applet or an Adobe Flash code for embedding media players. This code usually is embedded in the <body>-tag of an html document on the client side. In this way it provides the graphical user interface of the widget and also its functionality. Widgets enable a very simple distributed knowledge transfer and diffusion. The code that implements the widget can be on any server that is accessible through the WWW. As a result many different distributed services can be provided client-sided such as a CMS or a LMS without any further technical efforts. Widgets are also very often used in personalized homepages, personal desktops or in the PLEs where users are supported to aggregate and create their own configuration of widgets. iGoogle[1], Netvibes[2], Protopage[3] and Pageflakes[4] are some examples of such personalized desktops. The following are some projects, which provide end users with many various widgets. Konfabulator of yahoo widgets, Dashboard of apple project, Desktop widgets of Opera and Google gadgets are the most famous projects which also provide developers with some tools to implement the widgets. The only characteristic that these projects have in common is the lack of interoperability for desktop widgets.

---

1 http://www.google.com/ig [last visited: 09.03.2009]
2 http://www.netvibes.com/ [last visited: 09.03.2009]
3 http://www.protopage.com/ [last visited: 09.03.2009]
4 http://www.pageflakes.com/ [last visited: 09.03.2009]

They do not follow any open standard development infrastructure. Therefore for each project a corresponding widget engine must be installed at first to be able to run widgets on desktop.

In order to overcome this lack of interoperability a standard has been recommended by W3C as widgets 1.0 family of specifications which will be discussed in the following chapter.

2.1 The W3C Widgets 1.0 Family of Specifications

According to Widgets 1.0 Packaging and Configuration [Widgets 1.0 PC, 08], the W3C defines a widget as follows: "*Widgets* are full-fledged client-side applications that are authored using Web standards. They are typically downloaded and installed on a client machine or device where they typically run as stand-alone applications outside of a Web browser. Examples range from simple clocks, stock tickers and news casters to complex applications that pull data from multiple sources to be "mashed-up" and presented to a user in some interesting and useful way." The specifications [Widgets 1.0 PC, 08] standardize a zip packaging format, an xml based configuration file and a series of steps which developers should follow while they implement widgets. The packaging format acts as a container for files used by a widget. The xml-configuration file declares some metadata and configuration parameters for widgets. Furthermore the expected behaviour and means of error handling for widget user agents are specified.

2.2 IST PALETTE and TenCompetence

IST PALETTE [Palette, 08] and TenCompetence[5] are two projects of the European Union which have already implemented the draft W3C widgets 1.0 specification (both packaging and API). Both projects also focus on educational applications. Wookie [Wookie, 08] is a standalone widget engine which was developed as a part of the TenCompetence project, in order to enable learning activities to coordinate the usage of different external tools. It can be integrated into any web application by using the wookie widget factory API. It has the task to instantiate the widgets within the platform and renders them to the user interface of the corresponding web applications. To simplify this integration for existing applications the TenCompetence has developed plug-ins for some famous web applications such as Wordpress, Moodle and Elgg. Unlike Wookie, the widget engine in IST PALETTE [Palette, 08] is implemented as a part of the Palette web portal which means that the integration of widgets in other applications cannot be performed by plug-ins. IST PALETTE [Palette, 08] has extended the widget manifest of the W3C [Widgets 1.0 PC, 08] and added additional default user preference values. Although this extension is advantageous for e-learning systems where widgets can be configured and customized according to their preferences, the TenCompetence's Wookie widget engine cannot handle these widgets as their manifest file contains extended elements not described in W3C standards [Widgets 1.0 PC, 08]. Both projects have extended the W3C standards [Widgets 1.0 API, 09] in order to enable intercommunications within the

[5] http://www.tencompetence.org/ [last visited: 09.03.2009]

application in the background or through drag and drop between the widgets. Events can be fired between the widgets in the application.

2.3 JavaFX and widgets

"JavaFX[6] is a rich client platform for building cross-device applications and content. Designed to enable easy creation and deployment of rich internet applications (RIAs) with immersive media and content, the JavaFX platform ensures that RIAs look and behave consistently across diverse form factors and devices." [Castillo, 09] It was announced by Sun Microsystems[7] in 2008 and supports the development of PLEs. Widgets can be implemented and integrated in PLEs and users are able to personalize their specific needs through widgets. The next chapter describes FeedBoard and AtomWidget, two widget prototypes developed using JavaFX Platform.

3 FeedBoard and AtomWidget

3.1 Introduction

The implementations of FeedBoard and AtomWidget are both aimed to run on different platforms and devices. The user interface of both applications is designed to be embedded in a browser window, on a desktop as well as on a mobile device. The ability of being platform and device independent enables developers to develop one application and run it on different infrastructure.

FeedBoard

The goal of FeedBoard is to monitor different feeds embedded in a PLE. FeedBoard is able to run in a web browser, a mobile device, a desktop computer and in future releases even on a television. The current prototype is developed to run embedded in a web based PLE and a mobile device using JavaFX and JavaFX Mobile technology support.

The first prototype just downloads a specified RSS feed, parses it and displays the results on different views. FeedBoard updates the feeds in a specified time interval. FeedBoard notifies the users about the updated feeds if new feed entries are available. Because of the use of JavaFX technology FeedBoard has a big advantage compared to current feed readers. This advantage of FeedBoard is called "Drag-to-Install". FeedBoard's difference to other widgets is the ability to be dragged-out of the browser window and being "installed" on the operating system by dropping it onto the operating system's desktop. The term "installed" means here that a shortcut is created on the desktop and that the widget (in our case FeedBoard) is added to the list of installed software on the current operating system.

The "Drag-To-Install" feature was introduced by Sun Microsystems's launch of JavaFX Platform. Users are able to select a widget of choice in the web browser, drag-and-drop it out of the browser window onto the desktop of the current operating system. The widget can be started using the automatically created shortcut placed on

---

6 http://www.sun.com/software/javafx/ [last visited: 09.03.2009]

7 http://www.sun.com [last visited: 09.03.2009]

the desktop even if the browser window has been closed. The browser window doesn't have to be started in order to make use of the widget.

3.2 AtomWidget

AtomWidget is the second prototype developed by TU Graz Department of Social Learning and demonstrates the use of a RESTful architecture [Fielding, 00] through a JavaFX application. AtomWidget implements HTTP methods such like GET, POST, PUT and DELETE in order to communicate with a java server called Simple Atom Server. The server of choice for the prototype implementation is called Simple Atom Server and was developed through Jersey[8] a JAX-RS (JSR 311)[9] Reference Implementation for building RESTful Web services. Simple atom server retrieves HTTP requests and returns data in ATOM[10] publishing protocol format. Resources can be added using the POST method, resources can be deleted using the DELETE method. In order to update a resource AtomWidget uses the HTTP PUT method and to retrieve a list or a specific entry AtomWidget uses HTTP GET request method. The communication between AtomWidget on the client side and the simple atom server is being managed over the HTTP protocol and the ATOM publishing protocol. AtomWidget was developed using JavaFX technology and can be integrated into PLEs. Through the "Drag-To-Install" feature of JavaFX users are able to start AtomWidget without a web browser by clicking on the shortcut placed on the operating systems' desktop.

4 Conclusions

The W3C specifications are a standard that can be a basis for all PLEs and e-learning systems. Thereby the problem of interoperability is solved. Distributed services can be easily diffused and mashed up in a PLE. As the implemented prototype demonstrates, u-learning can be easily realised by using ubiquitous services independent of the platform or device of learners. Another positive aspect is the user centred model. As [Harmelen, 06] describes, in a PLE the learn objects should be under the control of the learners themselves. The learners can personalize their learning environment by adding ubiquitous widgets relevant to their interests in a PLE in ubiquitous environments, while they are on the way (mobile) or sitting in front of PC. As software systems should always be easy to extend and flexible to changes, the implemented widget prototype meets this requirement by implementing and using a RESTful architecture.

5 Future Work

As described in section 3, it is possible in near future to run JavaFX applications on televisions. Another future work is the integration of external services and social

---

8 https://jersey.dev.java.net/ [last visited: 09.03.2009]

9 https://jsr311.dev.java.net/nonav/releases/1.0/index.html [last visited: 09.03.2009]

10 http://atomenabled.org/ [last visited: 09.03.2009]

networks such as twitter, facebook, etc. It is also planed to offer specific university services through widgets. It can be summarized that the use of widgets will enhance the personal learning behaviour towards individuality and independency.

References

[Attwell, 07a] Attwell, G., (2007) Personal Learning Environments – the future of elearning?, eLearning Papers, Vol. 2, Nr. 1, January 2007

[Attwell, 07b] Attwell, G. (2007) E-Portfolios – the DNA of the Personal Learning Environment?, Journal of e-Learning and Knowledge Society, Vol. 3, Nr. 2, pp. 33-61

[Castillo, 09] Castillo C., 2009, What is JavaFX?, http://javafx.com/docs/gettingstarted/javafx/ (last visited January 2009)

[Downes, 05] Downes, S. (2005) e-Learning 2.0, eLearn Magazine, http://www.elearnmag.org/subpage.cfm?section=articles&article=29-1 (last visited January 2009)

[Fielding, 00] Fielding R., 2000, Architectural Styles and the Design of Network-based Software Architectures, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm (last visited January 2009)

[Green, 06] Green, S., Pearson, E. & Stockton, C. (2006). Personal Learning Environments: Accessibility and Adaptability in the Design of an Inclusive Learning Management System. In E. Pearson & P. Bohman (Eds.), Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2006, pp. 2934-2941

[Harmelen, 06] Harmelen van, M. (2006) Personal Learning Environments, Proceedings of the Sixth International Conference on Advanced Learning Technologies (ICALT'06)

[Palette, 08] Pedagogically sustained Adaptive Learning through the Exploitation of Tacit and Explicit Knowledge, 2008, http://palette.ercim.org/

[Pearson, 08] Pearson, E., Green, St., Gkatzidou, V., (2008) Enabling learning for all through adaptable personal learning environments, Proceedings ascilite Melbourne 2008, p. 742-749

[Reilly, 05] O'Reilly, T. (2005) What is Web 2.0 – Design Patterns and Business Models for the Next Generation in Software, http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html (last visited January 2009)

[Schaffert, 08] Schaffert, S., Hilzensauer, W., (2008) On the way towards Personal Learning Environments: Seven crucial aspects, eLearning Papers, n.9, p. 1-10

[Widgets 1.0 API, 09] Widgets 1.0 APIs and Events - W3C Working Draft, 2009, http://www.w3.org/TR/widgets-apis/

[Widgets 1.0 PC, 08] Widgets 1.0 Packaging and Configuration - W3C Working Draft, 2008, http://www.w3.org/TR/widgets/

[Wild, 08] Wild, F., Mödritscher, F., Sigurdarson, S., (2008) Designing for Change: Mash-Up Personal Learning Environments, eLearning Papers, 9, July 2008 p. 1-15

[Wilson, 05] Wilson, S., (2005) Architecture of Virtual Spaces and the Future of VLEs, Power Point slides, http://www.cetis.ac.uk/members/scott/resources/itslearning.ppt , 2005.

[Wookie, 09] Wookie, 2009, http://getwookie.org/Welcome.html.

[Zhan, 05] Zhan, G., Jin, Q. (2005) Research on Collaborative Service Solutions in Ubiquitous Learning Environment, 6th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'05), pp. 804-806