

Personal Learning Environment

A Conceptual Study

[doi:10.3991/ijet.v5s1.1195](https://doi.org/10.3991/ijet.v5s1.1195)

B. Taraghi, M. Ebner, G. Till and H. Mühlburger
Graz University of Technology, Graz, Austria

Abstract—The influence of digital technologies as well as the World Wide Web on education rises dramatically. In former years Learning Management Systems (LMS) were introduced on educational institutes to address the needs both their institutions and their lecturers. Nowadays a shift from an institution-centered approach to a learner-centered one becomes necessary to allow individuality through the learning process and to think about learning strategies in general. In this paper a first approach of a Personal Learning Environment (PLE) is described. The technological concept is pointed out as well as a study about the graphical user-interface done at Graz University of Technology (TU Graz). It can be concluded that PLEs are the next generation environments, which help to improve the learning and teaching behavior.

Index Terms—e-learning, personal learning environment, widget, PLE

I. INTRODUCTION

Web 2.0, first coined by Tim O'Reilly [1], changes how we interact and deal with the WorldWideWeb. Social Networking Software is growing rapidly and helps to connect people worldwide to share their thoughts, collaborate on topics of same interest and discuss in the virtual space.

Of course these changes influenced the learning and teaching behaviour in a new and complete different way. Bearing in mind that learning proceeds through communication [2], Web 2.0 technologies offer much more possibilities for interaction, collaboration and conversation. Mixing and connecting content by using the MashUp principle will allow learners to build their own learning environment, which is described by Stephen Downes [3] as “future learning environment which becomes not an institutional or corporate application, but also a 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”.

At the TU Graz (Graz University of Technology) research work is taking place to investigate how so called PLEs (Personal Learning Environments) can improve the daily learning and teaching behaviour. Therefore a closer look must be taken to address to the core issues. Wilson [4] expressed the user-centred philosophy of PLEs and Harmelen [5] agreed by pointing out that traditional LMSs (Learning Management Systems) are not flexible enough or are not addressing to the individual needs of a specific learner. A further approach is the use of common, internet based Social Software as mentioned in some other research [6] [22] to create an individual pool of information

and knowledge. The implementation of own systems became interesting and necessary, not only because of the learner’s arbitrariness but lots of problems concerning service stability, accessibility and data protection have been occurring. Nevertheless own systems often need high effort and technical support [7] [8]. Finally, Wild [9] mentioned that MashUp technology would help to create next generation learning environment.

However, even current research cannot point out what a highly personalized learning environment should look like in detail. Furthermore from an institutional point of view the question “How adaptable must a PLE launched by a University be?” is from high interest. The approach described in this publication is following the thoughts of Schaffert & Hilzensauer [10], who pointed out that LMS and PLE are both technological concepts and that both allow several pedagogical methods or personal learning strategies. They also defined seven crucial aspects for the shift from a LMS to a PLE:

- The role of the learner
- Personalisation
- Content
- Social involvement
- Ownership
- Educational & organisational culture
- Technological aspects

According to these aspects this publication describes the general technical issues, how such an environment can be realised and introduces to first graphical design studies. In the end the first prototype and future remarks are presented.

II. PERSONAL LEARNING ENVIRONMENTS – A FIRST TECHNOLOGICAL CONCEPT

A PLE that can be seen as a personalized individual website is not a new idea. This idea has been applied as contact widgets in some social networks such as Facebook¹ to let users get in touch with other registered users within the platform. Nowadays PLEs have gained increasing attention with the growth of Web 2.0 as described in section 1 and Rich Internet Application (RIA) technologies. It allows developers to build more dynamic and stable client side applications with a flexible Graphical User Interface (GUI) and programming logic. The programming logic is most often fully integrated in the presentation layer on the client side that results in reducing the server code, server response time and achieving a higher performance. The server side logic is no more responsible

¹ <http://www.facebook.com> (September 2009)

for the presentation layer. Its only task is reduced to retrieve the raw data from data resources and offering them back to the clients through a specific Application programming Interface (API). Thor [11] describes three main components of a mashup application: data extraction that is done on the server side, data flow and presentation which are accomplished on the client side.

As it is not possible to integrate the entire set of services in the presentation layer, the PLE server serves as a single entry point to provide the client programming logic with small applications or services. Such small applications are called widgets. Distributed resources and services are each presented in a widget and come together as an end-user-mashup in a PLE [12] according to the user customization and benefits. In other words, the user can customize his learning environment by adding or removing the widgets he is interested in.

Hoyer [13] describes mashup tools with different emphasis such as Yahoo Pipes² and iGoogle³. The following sections discuss the widgets in general as well as the widget specification of World Wide Web consortium (W3C) in this regard. Furthermore a widget project that has been already implementing the W3C widget 1.0 specifications is introduced. Based on this implementation a PLE is designed to be applied at Graz University of Technology that will be introduced at the end of this paper.

A. Widgets

Widgets are small embeddable applications that can be included in a HTML-based web page or executed on the desktop. This client side code can be a simple JavaScript, Java-applets or whatever that can be embedded in a valid HTML or XHTML document. It assumes building the GUI of the widget itself and the logic to retrieve or update data and use services provided by the PLE server. Applying widgets in a PLE may have different advantages. First of all widgets can be implemented independent of the platform. The W3C widget specification, which is going to be explained in the next section, facilitates it as a standard way of widget development. Widgets can be embedded in any application which supports that specification. This results in many open source widgets, for instance representing learning objects, made useable in PLEs. Another issue is the distributed knowledge transfer and diffusion. The service used by the widgets must not be necessarily on the same PLE server. Because of the security restrictions of the XMLHttpRequest (XHR) object used by Ajax, which only allows to access files on the same server, a proxy script is used on the PLE server. The proxy passes all requests and replies unmodified from the client to the remote server and vice versa. Using JSONP (JSON⁴ with Padding) as a technique would even bypass the security restriction of XHR and enable client-side cross-site requests without applying any proxies. As a result many different distributed remote services can be provided within a PLE without any technical effort. Personalization is a very important factor for a PLE. As short lightweight applications are put in widgets, users can organize and personalize the applications (widgets) of their interest among the huge possible MashUps.

The following section goes briefly through the W3C Widgets 1.0 family of specifications which is a step forward to gain a standard in widgets development.

1) W3C Widgets 1.0 family of specifications

According to Widgets 1.0 Packaging and Configuration [14], 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, news casters, games and weather forecasters, to complex applications that pull data from multiple sources to be “mashed-up” and presented to a user in some interesting and useful way.”

It standardizes a zip packaging format, which includes the whole widget source code with a specified file structure and an XML-based configuration file with some mandatory and none mandatory elements. This specification determines also a series of steps that should (must) be followed by developers while they implement widgets. Finally the behaviour and means of error handling for the widget user agents are also specified.

The Widgets 1.0 APIs and Events specifications [15] define a set of APIs and events for the widgets’ functionality. They define the means to access the metadata declared in a widget’s configuration document and to receive events related to changes in the view state of a widget. They also determine the locale under which a widget is currently running and the notification mechanism of events relating to the widget being updated. Other points in these specifications are how to invoke a widget to open a URL on the system’s default browser and how to request the user’s attention in a device independent manner.

Widgets 1.0 Digital Signature [16] is another W3C widget specification which deals with the digital signing of widgets. Widget URIs [17] is still not a specification but a working in progress which tries to define a URI schema for use inside widgets.

Next section describes briefly a project of the European Union called “IST Palette” which has already implemented the draft W3C Widgets 1.0 specifications.

2) IST Palette project

Next to the TenCompetence [19] project the IST Palette [18] is one of the two projects of the European Union focusing on educational applications; it has already implemented the draft W3C Widgets 1.0 specifications (both packaging and API). The Palette Service portal represents a web portal users can customize by adding and removing widgets according to W3C described specifications. The W3C built-in widget engine enables installation and integration of any widget that is compatible with W3C specification. IST Palette has extended the widget manifest file described in [14] and added some additional default user preferences values. These values may be modified by the users to customize the widgets according to their own needs. As the Palette preferences are added under a separate namespace it remains compatible with [14].

IST Palette has extended the specifications described in [15] too. By these extensions a new way of communication between widgets has been enabled. Widgets can add listeners to events or fire events to trigger some events in other widgets. This can be graphically realised by a simple

² <http://pipes.yahoo.com> (September 2009)

³ <http://www.google.com/ig> (September 2009)

⁴ <http://de.wikipedia.org/wiki/JSON> (September 2009)

drag & drop between widgets. As a very simple example, setting a location in a map widget can fire an event that triggers the other widgets within the web page. A weather forecast widget can adjust its content after being triggered by the map widget to show the weather status of the selected location on the map. IST Palette distinguishes between local widgets which are deployed within the platform and the remote widgets which are stored on a remote server. While local widgets are static client-side applications that are compatible with W3C widget 1.0 specifications and implemented in HTML and JavaScript, remote widgets may include server-side programming languages that dynamically produce the widget content. In theory it is possible to implement any type of widget using both approaches.

Section 2.3 introduces the technological concepts of the designed PLE which bases on the widget engine implemented by IST Palette project described above.

B. JavaFX

“JavaFX⁵ is a rich client platform for building cross-device applications and content. Designed to enable easy creation and deployment of RIAs with immersive media and content, the JavaFX platform ensures that RIAs look and behave consistently across diverse form factors and devices.” [20] JavaFX was announced by Sun Microsystems⁶ in 2008. Through the widget implementation following the standards described in section 2.1.1 it is possible to integrate a lot of different widgets in a PLE and let the user decide which ones to use in order to achieve a specific learning goal.

JavaFX features the development of widgets which can be embedded into an HTML web site, a mobile device like a mobile phone and in future releases even televisions and media devices. The usage of widget is not only restricted to the web browser. Users can access the mobile version of the widget through a mobile device such as a mobile phone and access their PLE from everywhere. PLEs become ubiquitous through the distributed services and mobile clients. The following aspects give example of PLEs supported by JavaFX.

1) JavaFX for PLE widgets

Through the JavaFX scripting language it is possible to develop widgets which can be embedded into PLEs. The integration of JavaFX widgets into PLEs enables developers to build up a PLE as a Mashup of different widgets.

Widgets can be implemented as well in HTML supported by AJAX, as an Adobe AIR or as a Microsoft Silverlight application. All of them have in common that the widget can be implemented independently of the underlying widget engine. That means for example that the widget can be developed and deployed as a small component, regardless the underlying PLE.

Microsoft Silverlight or Adobe AIR applications require a runtime environment being installed on the client machine. This approach narrows you down to use a browser in order to run the application or to a standalone application. You are not able to switch the application embedded in the browser window directly to your operating system's desktop.

JavaFX widgets require a runtime environment as well. But JavaFX has a big advantage over the other technologies: JavaFX widgets run in a Java Virtual Machine (JVM) and can therefore be executed in a broad range of devices supporting JVM. Personal Computers (PCs) as well as mobile devices that support JavaFX technology (such as mobile phones, PDAs, etc.) of different vendors are able to execute JavaFX widgets.

2) Java plus JavaFX User Interface

Another important issue is the integration of the Java programming language. JavaFX script is compiled in Java Byte Code which is then interpreted on the Java Virtual Machine at runtime. This means that the developers can develop their whole business logic in Java and integrate a JavaFX User Interface (UI) for end users. All Java specific technologies can be used through the development cycle of a JavaFX widget with some restrictions regarding Java Applets.

3) Drag-to-Install

Further it is important to enable the end user to personalize the learning environment as much as possible. Using JavaFX in PLEs has the advantage that the user is able to drag the embedded widgets out of the PLE (e.g. the browser window) and drop them onto the desktop. This feature is called “Drag-to-Install” and was introduced by JavaFX technology. A JavaFX widget is running in an own thread which makes it possible first to drag the widget out of the browser and second even to close the browser window while the widget is running. As a result a shortcut is created on the user's operating system desktop and the widget is still running, without the browser window. Afterwards it is possible to start the widget by clicking on the shortcut created former on the desktop. As the end user drops the widgets out of the browser window onto the desktop it ends up as a PLE constructed desktop by the end user itself.

C. Technological concept

Extendibility is a very significant issue in the design phase of each project that must be considered precisely. IST Palette widget engine supports next to the local widgets that are W3C compatible, also remote widgets that can be implemented using any server-side programming language for dynamic content generation. Looking at the wide scope of learning objects and services with their eventual dynamic processing requirements in background, it is clear that the remote widgets provided by IST Palette can be applied very usefully in this context. The widgets variation used in PLE can be increased and extended to many learning services on remote servers in form of remote widgets.

Looking at the definition of widgets in [14] it is obvious that the idea behind the W3C specifications is not restricted only on widgets running within browsers. If wanted, widgets should be able to run as stand-alone applications on desktops, mobile phones or other client machines too. A Firefox extension called “Prism” [21] provides the possibility to run HTML pages as stand-alone applications on the desktop. This approach requires Firefox being installed on the client side and works only for desktops. JavaFX which is described in section 2.2 does not have these restrictions. Widgets (local or remote) developed with JavaFX can be easily dragged to the desktops or used in mobile clients respectively.

⁵ <http://www.sun.com/software/javafx/> (September 2009)

⁶ <http://www.sun.com> (September 2009)

The designing process consists of a detailed GUI design and some performance issues in the IST Palette portal which would serve as the widget container for the designed PLE. Packing and minifying the JavaScript and CSS files dynamically and using gzip⁷ to reduce the load time are some performance issues which are taken into consideration. Furthermore all JavaScript libraries including the widget engine and GUI specific ones as well as required CSS files are minified into one output buffer. Furthermore they are loaded together to reduce the number of client requests to one for JavaScript and one for CSS. Caching is another important issue in this regard. The required scripts are cached on the server to decrease the load time. Besides that, ETags⁸ are set in HTTP headers to let the browser use the cached content if not modified.

The following section describes the way TU Graz is going to realize a PLE for its students on base of modern widget technology.

III. FIRST DESIGN STUDIES

There are many e-Learning services that are already provided by the TU Graz, including course administrations, course learning materials such as e-books, podcasts etc. and user generated contents as well as user contributions such as blogs, bookmarks and files posts. All these services are going to be integrated in the designed PLE as widgets. Therefore it is necessary to design a coherent GUI to avoid the possible usability and consistency problems that could occur.

The process of designing and developing the GUI itself has not been yet completed. At present it is merely possible to partly describe the momentary output of the work progress. A more profound description of the GUI in respect to form and colour would go beyond the scope of this explanation.

The conceptual design of the GUI is oriented at the following central ideas:

1. "less is more" - Whenever working with information the amount of information shown on the screen at a certain point of time is crucial. Concepts which give the user the possibility to influence the information density could lead to an information flooding. In order to avoid this the finite space for the arrangement of information is the efficient instrument.
2. "tidy not cluttered" - A user interface moulded by a reduced, straight forward and structured design looks tidy and supports the user in an optimal way to grasp the depicted information. In order to enable the user to arrange his information with a maximum of freedom but at the same time to avoid a cluttered information space an appropriate information carrier was structured.
3. "the best of two worlds" - New concepts like a widget-based user interface are always a challenge and introduce a process of re-adjustment. In this case the aim of the GUI has to cover two aspects: to present the new technology and even more to support the user to get used to working with this new technology as fast and easy as possible. Very often a mix of familiar and unfamiliar elements as well as the func-

tional linking of both lead to the above mentioned desired results and finally the familiar elements can be shifted more and more to the background.

A. Main GUI Elements

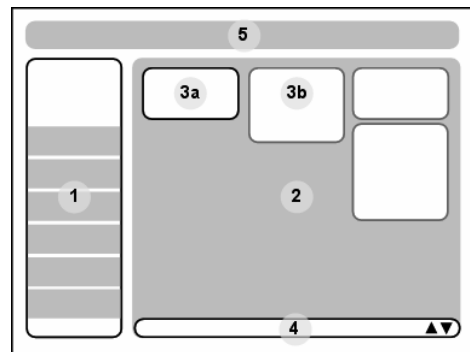


Figure 1. Main GUI Elements

1) Sidebar (1)

The sidebar is designed as a navigation element as well as an information element. It contains main categories which are defined as "main activities" provided by the university (i.e. CommunicationCentre, PersonalNotes, ...). The sidebar also updates the user on the status of these activities by means of colour indicator and a numerical indicator.

Being a familiar element in a familiar position, the sidebar supports the user's way of thinking and further more the user dealing with the unfamiliar widget-based part of the user interface - the WidgetZone. Finally the sidebar can be switched off in favour of the WidgetZone and is replaced by another navigation element (see "Workflow").

2) WidgetZone (2)

The WidgetZone contains the widgets and is structured by columns. Each activity has its allocated and specific WidgetZone. Thus the information is organised according to topics and subsequently the amount of information is increased by having more than one WidgetZone. The user can see the content on a familiar place but through a new display format - the WidgetZone.

3) Widget (3)

A widget consists of a front side and a rear side, the latter contains the preferences. Usually the widget is visible in a minimised state on the WidgetZone but it can be maximised (see "Workflow"). In this case minimised means that the information shown on the widget is in a summarised form. If the information is maximised it means the total information is displayed.

The width of the widget in comparison to the height is not variable and is defined by the columns of the WidgetZone. There are two kinds of widgets:

1. System widget (3a): A system widget is existent from the very start of the WidgetZone and its position can be shifted by the user but cannot be removed. It contains all information relevant to the specific centre.
2. Standard widget (3b): For each specific WidgetZone the user can choose from a selection of standard widgets which can be removed.

4) PersonalDashboard (4)

The PersonalDashboard is at all times available to the user independent of the different topics of the activities

⁷ <http://de.wikipedia.org/wiki/Gzip> (September 2009)

⁸ <http://de.wikipedia.org/wiki/ETag> (September 2009)

and it can be overlaid if necessary. In this way the user can add different widgets from the specific activities. In the current state of the development the Personal-Dashboard is structured in the same way as the Widget-Zone namely in columns and finite in space.

5) *Banner (5)*

The banner is not only a graphic element contributing to brand a site and helps the user to orientate and to locate. But its main purpose is to display information from the Network in a user-profile-sensitive way. It is also in context to the currently active WidgetZone.

Furthermore the user can create an own banner by means of a graphic design.

Please note, that the possibilities of such elements have not been exploited to the maximum.

B. *Workflow*

The goal of the following Workflow described below is to understand the functions of the GUI in a better way.

By clicking on a certain topic (i.e. TeachCenter) on the sidebar a switch is made feasible to the desired Widget-Zone. The user can see the widgets in a minimised state and also in an unchanged position. If preferences must be changed, the desired widget can be flipped. By this applied flip-animation the users spatial perception is undisturbed and makes the GUI more understandable (see Fig. 2).

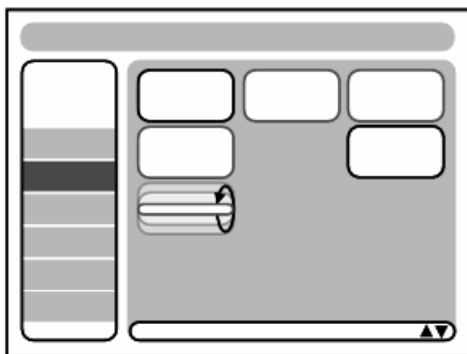


Figure 2. WidgetZone

By clicking on a certain control the widget is enlarged, that is maximised. Information displayed in the widget is more detailed. A zoom animation helps the user to understand the GUI (see Fig. 3).

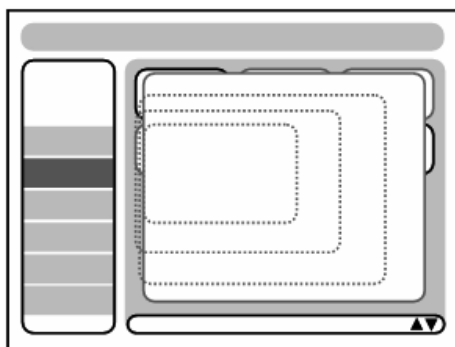


Figure 3. Widget zoom

By clicking on a certain control the PersonalDashboard is displayed by an overlay animation. At the same time the remaining GUI is shaded (see Fig. 4).

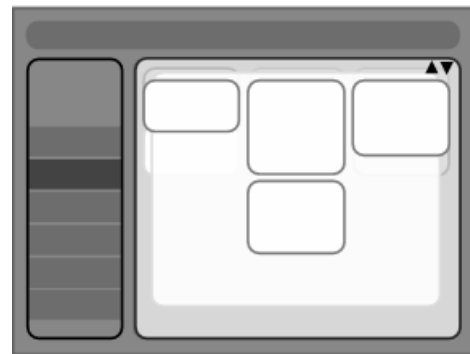


Figure 4. WidgetZone full screen

As mentioned previously the sidebar can be turned hidden. In doing so the area of the WidgetZone is enlarged and likewise the PersonalDashboard. The sidebar is replaced by another navigation and information element, which is overlaid in the lower part of the WidgetZone. In the Fig. 5 the circles of the element refer to the main topic, namely the centres.

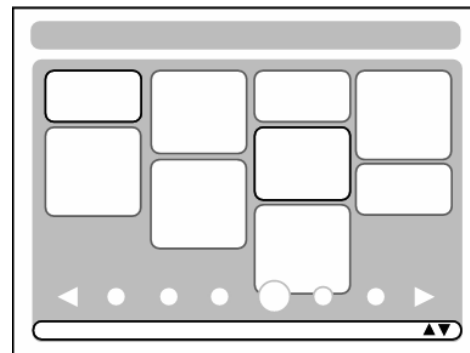


Figure 5. WidgetZone full screen

A GUI of such a design as seen in the illustration could be applied by the user to create additional WidgetZones according to his specific needs.

IV. CONCLUSION

In this paper we described the first approach and design studies towards a Personal Learning Environment at Graz University of Technology. By using widgets for instance with JavaFX technology a flexible individual website can be provided to the students and teachers. From the institutional point of view the PLE is going to serve students as a tool to use both the services of the university as well as all other services available in the World Wide Web in a personalized and customizable manner. Students will be able to use the required university services which are partially decentralized in different faculties in their own personal learning environment, customized according to their actual needs. The numerous resources in the WWW can also be provided to students for general learning goals or specific courses. Students can select the ones they are interested in, customize the preferences, as far as possible and apply them according to their learning process. This publication gave a first overview about the technological concept and the conducted design studies as well as usability tests. The next step will be the development and implementation of the prototype within the university environment to gather experiences how PLEs can be used in real-life in order to carry out different didactical scenarios.

REFERENCES

- [1] T. O'Reilly, (2009, June 19) *What is Web 2.0 – Design Pattern and Business Models for the next Generation of Software*, [Online] Available: <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-2.0.html>
- [2] A. Holzinger, “*Multimedia Basics*”, Volume 2: Learning, Cognitive Fundamentals of Multimedial Information Systems, Laxmi, New Delhi, 2002
- [3] S. Downs, (2009, June 19) *e-Learning 2.0*, eLearn Magazine, [Online] Available: <http://www.eleammag.org/subpage.cfm?section=articles&article=29-1>
- [4] S. Wilson, (2005) Architecture of Virtual Spaces and the Future of VLEs, Power Point slides, [Online] Available: <http://www.cetis.ac.uk/~members/scott/resources/itslearning.ppt>, 2005.
- [5] M. van Harmelen, “Personal Learning Environments”, in *Proceedings of the Sixth International Conference on Advanced Learning Technologies (ICALT'06)*, 2006
- [6] G. Attwell, “E-Portfolios – the DNA of the Personal Learning Environment?”, in *Journal of e-Learning and Knowledge Society*, vol. 3, no. 2, pp. 33-61, 2007.
- [7] S. Green, E. Pearson, and C. Stockton, “Personal Learning Environments: Accessibility and Adaptability in the Design of an Inclusive Learning Management System,” in *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 2006, pp. 2934-2941.
- [8] E. Pearson, St. Green, and V. Gkatzidou, “Enabling learning for all through adaptable personal learning environments”, in *Proceedings ascilite Melbourne 2008*, Melbourne, 2008, pp. 742-749.
- [9] F. Wild, F. Mödritscher, and S. Sigurdarson, “Designing for Change: Mash-Up Personal Learning Environments,” in *eLearning Papers*, no. 9, July, 2008
- [10] S. Schaffert and W. Hilzensauer, “On the way towards Personal Learning Environments: Seven crucial aspects,” in *eLearning Papers*, no. 9, July, 2008
- [11] D. Aumueller and A. Thor, „Mashup-Werkzeuge zur Ad-hoc-Datenintegration im Web,“ in *Datenbank-Spektrum*, 8. Jahrgang, vol. 26, <http://dbs.uni-leipzig.de/file/aumueller-thor-2008-mashup-werkzeuge.pdf>, 2008
- [12] M.T. Gamble and R.F. Gamble, “Monoliths to Mashups: Increasing Opportunistic Assets,” in *IEEE Software*, vol. 25, no. 6, pp. 71-79, December 2008
- [13] V. Hoyer: „Ad-hoc-Software aus der Fachabteilung,“ in Report - Enterprise Mashups, iX 10, pp. 98, 2008
- [14] Widgets 1.0 Packaging and Configuration - W3C Working Draft, 2008, <http://www.w3.org/TR/widgets/>
- [15] Widgets 1.0 APIs and Events - W3C Working Draft, 2009, <http://www.w3.org/TR/widgets-apis/>
- [16] Widgets 1.0 digital Signatures - W3C Working Draft, 2009, <http://dev.w3.org/2006/waf/widgets-digsig/>
- [17] Widget URIs - W3C Editor's Draft, 2009, <http://dev.w3.org/2006/waf/widgets-uri/>
- [18] Pedagogically sustained Adaptive Learning through the Exploitation of Tacit and Explicit Knowledge, 2008, <http://palette.ercim.org/>
- [19] Wookie, 2009, <http://getwookie.org/Welcome.html>
- [20] C. Castillo, (2009, May), “What is JavaFX?”, <http://javafx.com/docs/gettingstarted/javafx/>
- [21] Mozilla Labs Prism, 2009, <http://prism.mozilla.com/>
- [22] G. Attwell, “Personal Learning Environments – the future of elearning?,” in *eLearning Papers*, vol. 2, no. 1, January, 2007

AUTHORS

B. Taraghi (b.taraghi@tugraz.at) is with the Department of Social Learning, Graz University of Technology, 8010 Austria.

M. Ebner (martin.ebner@tugraz.at) is Head of the Department of Social Learning at Computer and Information Services, Graz University of Technology, 8010 Austria.

G. Till (gtill@tugraz.at) is with the Institute of Housing Architecture, Graz University of Technology, 8010 Austria.

H. Mühlburger (muehlburger@tugraz.at) is with the Department of Social Learning, Graz University of Technology, 8010 Austria.

This article was modified from a presentation at the International Conference of Interactive Computer Aided Learning ICL2009, September 2009 in Villach, Austria. Submitted 11 September 2009. Published as resubmitted by the authors on 17 January 2010.