

Widget User Interface considerations for ROLE widgets

Martin Wolpers, Martin Friedrich
Fraunhofer Institute for Applied Information Technology
{martin.wolpers, martin.friedrich}@fit.fraunhofer.de

Abstract

The development of computer-based learning environments aims to create a truly personal and individual learning experience. A learning environment should be self explaining and easy to use. Thus, the design and development of the user interface plays a major role in the learning experience. Furthermore, personal learning environments (PLEs) provide the learner with the ability to build their own learning environments by compiling respective services, contents and peers. In this context, the European project ROLE focuses on the development of an architecture to enable learners to compile their learning environment within the browser using widget technology. In this paper, we briefly discuss the suggestions and best practices of the ROLE project for the design of widget interfaces to run within the ROLE enabled PLEs.

Keywords:

PLE, widgets, human computer interface

1 Introduction

Learning is an individual activity. Hence, the need arises that learning environments provide support to satisfy personal requirements, demands and goals. Consequently, the learners' preferences and needs are of crucial importance to personalisation and adaptation of learning environments (Harmelen, 2006).

Following this requirement, learners should be able to create their own computer-based learning environment, e.g. by mashing up the services, contents and peers in their personal learning environment (PLE), to improve their learning efficiency. The PLE may reside either in social networks, within collaboration and portal platforms and, possibly, in an enterprise setting (Drachsler et al., 2009). Especially in lifelong learning transition phases when inhomogeneous groups of learners are treated in a one-size-fits-all way, the PLE approach becomes highly relevant as it provides ways to respond to their individual strengths and weaknesses (Attwell et al., 2009).

The most prominent current approach to PLEs aims to create responsive open learning environments (ROLEs): ROLEs enable learners to adapt the learning environment to their own, very specific needs. Furthermore, they enable the system to adapt its functionalities and components to the concrete and individual demands concerning learning environment and learning strategies (Kirschenmann et al., 2010). The core functionality of ROLEs focuses on creation of new learning services, e.g. through mashing up existing services and functionalities on the basis of existing web technologies. Such a generic framework provides the learners with adaptivity and personalisation in terms of content and navigation as well as adaptivity and personalisation of the entire learning environment and its functionalities (Wolpers et al., 2010).

From a technical point of view, ROLES are built on a generic framework that enables the flexible composition of technologies which allow the client-side mash up learning services and technologies by the user.

In this paper, we focus on presenting considerations for the design of web widgets which can be mashed up in browser-based ROLES.

2 Web-Widget specifications

2.1 Widget definition

Widgets are mini-application designed for a specific task. They can appear in different forms, such as Desktop- or Web-Widgets. The ROLE project is based on Web-Widgets, which are implemented in HTML for the definition of static content, but can be extended with JavaScript, Flash or Silverlight for dynamic behaviors. There is no uniform specification for Web-Widgets existing, but the most common specifications are listed in the following section.

2.2 Web-Widget specifications

The following Web-Widget specifications represent the most common approaches:

- W3C Widgets 1.0 (<http://www.w3.org/TR/2009/WD-widgets-20091029/>), specified by the World Wide Web Consortium (W3C)
- OpenAjax Metadata 1.0 Specification (http://www.openajax.org/member/wiki/OpenAjax_Metadata_Specification), developed by the OpenAjax Alliance
- OpenSocial Gadgets (<http://code.google.com/apis/opensocial/articles/tutorial/tutorial-0.8.html>)

Previously, it has been a vendor specific format, mainly supported by Google (Google Gadgets). It is now an independent specification managed by the OpenSocial Foundation. OpenSocial widgets are able to connect to the social graph (if existing) of the widget container. The specification is supported by a number of existing widget stores, thereby providing an already rich base of existing widgets. Examples are iGoogle, Netvibes, and learning management systems like Clix and Moodle. The Apache Shindig development is an OpenSocial container (reference implementation) which provides a well defined environment for the execution, view management, communication, etc. of widgets.

The ROLE project focuses on OpenSocial widgets, but also works on the integration of W3C widgets. For an in-depth explanation on how to develop widgets, please see (Lal and Chava, 2010).

3 Considerations for user interfaces of widgets in PLEs

3.1 General considerations

Web-Widgets are small programs that share the browser window with possibly numerous other widgets. Hence, a widget per definition provides functionality to solve a simple task. Usually, this requires the widget to offer one service only.

By simplifying the widgets purpose to one single task, the user interface should become self-explaining. Hence, the user does not need to learn how to use the widget in order to solve the task at hand but can use the widget straight away.

And finally, a widget needs to encapsulate all services needed to provide its functionality. No further widget should be required to use the widget in question meaningfully.

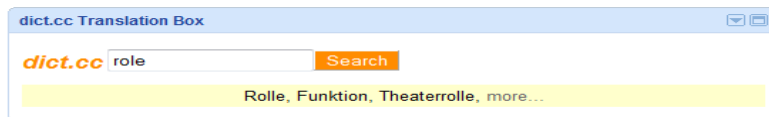


Figure 1: English-German translation widget

The general considerations are described in figure Figure 1. A simple freeform text field of one line, the button labeled “search” next to it and the result display provides intuitive functionality. Of course, it helps to relate user interface concepts to designs and metaphors already known to users.

3.2 Widget layout considerations

In order to avoid unnecessary dynamic behaviour of the PLE layout, widgets should use a fixed size, both in height and width. This way, the PLE layout does not change every time a widget changes its contents.

While scrollbars are sometime unavoidable, widgets should use them only when really necessary. The PLE user interface will otherwise become quite unclear when many widgets display scrollbars. The user is accustomed to use the browser window scrollbars only. Thus, the widget developer should make use of tabs within the widget to avoid scrolling.

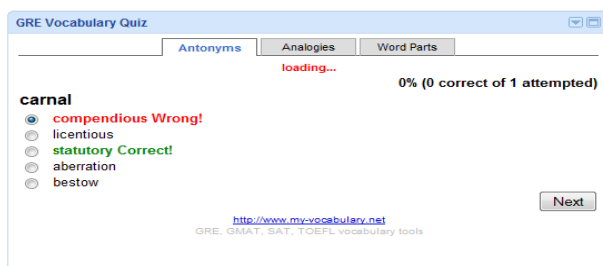


Figure 2: Vocabulary-Test widget

The widget that is shown in Figure 2 uses tabs to switch between different information clusters. It starts with its original purpose of a vocabulary test and offers related information to the word in question in other tabs.

3.3 Widget customization consideration

Users tend to change the look and feel of the widgets that they use. Hence, widgets should enable users to customize the widget at least in terms of color, background and content. Nevertheless, each widget needs to foresee default values to help the user getting started.

In addition, each widget should provide a “welcome” mode that briefly explains the purpose of the widget. An often seen implementation is that the welcome message is shown at the loading time of the widget but disappears quickly thereafter.

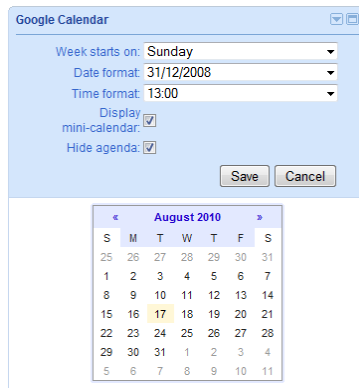


Figure 3: Google Calendar widget

The widget shown in Figure 3 exemplifies how a user can customize the look and feel of the Google Calendar widget. The user is able to specify the display of contents especially so that if she can tailor it to her own needs.

4 Conclusion

In this brief paper, we describe some lessons learned from implementing and using widgets in the various testbeds of the ROLE project. Quite clearly, the main objective for widget design is the simplification of the user interface up to the point, where the user can use the widget without having to learn how to handle it. At the same time, a widget must give as much freedom in the customization of its content as needed to enable the user to tailor it to her personal needs.

5 Literature

Attwell, G., Cook, J., Ravenscroft, A. (2009): Appropriating Technologies for Contextual Knowledge: Mobile Personal Learning Environments. In: Best Practices for the Knowledge Society. Knowledge, Learning, Development and Technology for All Communications in Computer and Information Science, Volume 49, Part 1, 15-25.

Drachsler, H. et al. (2009): ReMashed – Recommendations for Mash-Up Personal Learning Environments. In: Cress, U., Dimitrova, V., Specht, M. (eds.): Learning in the Synergy of Multiple Disciplines – 4th European Conference on Technology Enhanced Learning, EC-TEL 2009. LNCS 5794, Springer, Berlin. pp. 788-793.

Harmelen, M. V. (2006): Personal Learning Environments. In: ICALT '06: Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies. IEEE Computer Society, Washington, DC. pp. 815-816.

Kirschenmann, U. et al. (2010): Demands of Modern PLEs and the ROLE Approach. In: Sustaining TEL: From Innovation to Learning and Practice. Proceedings of the 5th European Conference on Technology Enhanced Learning (EC-TEL 2010), LNCS 6383, Springer, Barcelona, Spain. pp. x-y.

Wolpers, M. et al. (2010): Early Experiences with Responsive Open Learning Environments. Proceedings 10th International Conference on Knowledge

Management and Knowledge Technologies, 1–3 September 2010, Messe Congress Graz, Austria.

Lal, R. and Chava, L. C. (2010): Developing Web Widget with Html, Css, JSON and Ajax: a Complete Guide to Web Widget. CreateSpace.