

# A Web Application Mashup Approach for eLearning

Felix Mödritscher<sup>1</sup>, Gustaf Neumann<sup>1</sup>, Víctor Manuel García-Barrios<sup>2</sup>,  
Fridolin Wild<sup>1</sup>

<sup>1</sup> Institute for Information Systems and New Media,  
Vienna University of Economics and Business Administration,  
Augasse 2-6, 1090 Vienna, Austria  
{felix.moedritscher,gustaf.neumann,fridolin.wild}@wu-wien.ac.at

<sup>2</sup> Institute for Information Systems and Computer Media,  
Graz University of Technology,  
Inffeldgasse 16c, 8010 Graz, Austria  
vgarcia@iicm.tu-graz.ac.at

**Keywords:** eLearning, Web 2.0, web application mashup, Ajax, XoWiki

## 1 Introduction

The common understanding of learning shifted over the last decades from the traditional behaviorist approach to paradigms like constructivism or discovery learning [1] and, further, started to take into account aspects like heterogeneity of tools and information resources [2]. Referring to [3], we assume that prominent Internet phenomena like Google or Wikipedia nowadays influence our learning habits. Against this background, learning platform developers start to consider Web 2.0 features [4], i.e. by adding tagging functions, collaborative elements, Wiki and Blogging components, etc. [5]. Still, many developments restrict themselves to a style platform and even a simple installation, while learners continue working with the applications and websites they are used to, even if they are encouraged to use a certain eLearning system. Moreover, learning facilitators might also want to recommend certain applications for specific activities, which, in sum, lead to a “*heterogeneous space of software, tools and services*” [6].

Full-featured heterogeneous eLearning systems are thus, hard to integrate in single solution frameworks. In order to integrate e.g. distinct open source initiatives, Web 2.0 applications as well as personalizable or customizable tools, platform developers try to provide functionalities allowing users to (re)assemble or merge external systemic entities into the own platform, as can be observed in the field of enterprise information systems [7]. In this paper, we present a mashup approach which is based on the XoWiki component [8] and enables a user to add web-based applications to Wiki content and to use and arrange these tools in a comfortable way. In the following, the paper gives a short overview on the topic at hand, describes critical challenges regarding the development of application mashups and – based on that – derives special requirements for eLearning. Thereafter, various solutions and

development frameworks are inspected according to our special needs, and a concrete implementation, namely the XoMashup component, is described closer. Finally, a possible application scenario is shown, and the next steps moving towards a semantic layer are outlined.

## 2 The Concept of a Web Application Mashup

According to [9], a mashup comprises an application that “*combines multiple sets of data streams into a unified user experience*”. Mashup approaches can be observed in many different fields, e.g. for enterprise information systems [7] or digital libraries [9]. In the field of eLearning, mashing up different learning experiences is related to the following aspects. Within the context of learning object repositories, aggregation of information junks is realized according to instructional models and on terms of standards and specifications, such as LOM or SCORM [10]. Addressing information push strategies, content syndication deals with aggregating information streams, or feeds, into one view, normally by using a XML-based metadata standard like RSS [11]. Yet, these approaches primarily focus on the combination of information chunks and streams into one application.

However, for eLearning it can be necessary to utilize various tools in order to achieve a competence [2]. For instance a group of learners might want use different web-based tools such as WordPress ([wordpress.org](http://wordpress.org)), Wikipedia ([wikipedia.org](http://wikipedia.org)), ObjectSpot ([objectspot.org](http://objectspot.org)) and XoWiki ([openacs.org/xowiki](http://openacs.org/xowiki)) in order to collaborate on a joint paper. Thus, such an environment must allow the execution of several web-based tools next to each other, taking into consideration that each web application has its own idiosyncrasies (e.g. scripts, style-sheets, databases, etc.). The most important difference between a mashup of web-based tools and a mashup or portal application comprises syndication. While a mashup aggregates different data streams and a portal even requires own rendering methods for the portlets [12], a mashup of web applications aims at providing a secure environment for each tool, whereby syndication is only possible via cross-domain scripting [13]. However, our proposed solution should also be capable of not losing the user interaction state with all tools during a session. As a conclusion, we foster the idea of a web application mashup rather than a(nother) mashup or portal approach.

In the context of self-directed learning [2], such a web application mashup solution should support the following use cases: First of all, a learner might want to add and launch new applications, whether from the set of pre-defined tools or own ones. With respect to mashup visualization techniques [14], the mashup page should allow to arrange the learning tools according to the basis of a grid layout. Additionally, learners might want to maximize a tool over the full screen in order to work only with this one or to minimize or close another tool to keep track of more important spots. For the analysis of the interaction patterns, user interaction with single tools and with the whole mashup may be logged, as this information serves as central source for further system’s enhancements, such as improving semantic models, identifying and solving usability problems, inspecting potential privacy invasion issues, and so forth.

Analyzing existing mashup frameworks, we hardly could find generic solutions which are capable of launching different web applications as a part of the browser screen. Technically, such a solution requires primarily two functions: (1) the embedment of the web application within an iframe [15] and (2) the prevention of document object model (DOM, [16]) manipulations with this iframe element in order to not reset the state of the tool by reloading the original URL. From a broad range of existing providers listed by [17], only one solution, namely Protopage ([www.protopage.com](http://www.protopage.com)) fulfils these essential requirements, while other providers (iGoogle, MyYahoo, Netvibes, Pageflakes, etc.) rather offer a portal solution by means of mashing up xhtml snippets.

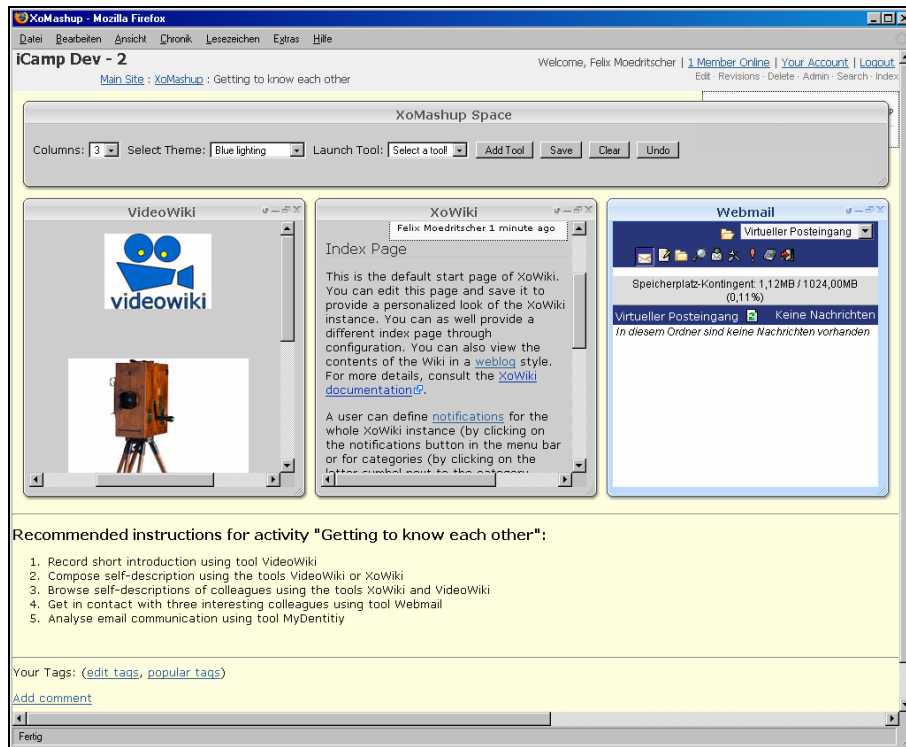
### 3 Development and Application of the XoMashup Component

Unfortunately, Protopage is a commercial system, and the portal solutions require that each web application has to render itself with an own method. Similar restrictions apply to the only published mashup component of the OpenACS community, the Solution Grove Web Desktop [18]. Thus, we had to evaluate existing Ajax frameworks [19] to build up a mashup component according to the above requirements. We examined seven of the most popular frameworks, namely Prototype Windows, Script.aculo.us, Dojo, Yahoo! UI, Google Web Toolkit, DHTML-Window, and Qooxdoo. Overall, only two of them (DHTML-Window, Prototype Windows) do support our requirements regarding iframe-capability, DOM manipulations, compatibility for the most important browsers and an open source license.

At the end, we decided to utilize Prototype Windows (<http://prototype-window.xilinus.com/>) for realizing the client-sided functionality of our mashup solution due to the framework's object-oriented nature and the dirty coding of DHTML-Window. Based on this client-sided framework and XoWiki, the XoMashup component is implemented in the following way:

- The **presentation layer** consists of a grid-based screen separated in a certain number of (sortable) columns (see also Fig. 1). One column, in the style of a portlet-based portal, is able to hold various windows, each one representing an own web application or information stream. A window can be reloaded, minimized, maximized or closed. Further, each window can be dragged to any place on the screen, whereby it is always dropped into one of the columns. Finally, it is also possible to change the theme of a page by exchanging the decorations of all windows (though not of the tools).
- The **application and data layer** is realized on the basis of the XoWiki module and allows setting up a mashup space. Therefore, administrators can create template pages containing tools, which can be started or are already launched, as well as any Wiki-based content. Entering the XoMashup application, registered users receive an overview of available templates and their personal mashup pages, which is realized by two includelets (“`{{my_xomashup_pages}}`” and “`{{available_xomashup_templates}}`”). Similar to XoWiki, a user can create a new mashup page on the basis of one pre-defined template. The initial state of such a personal mashup page is duplicated from the template. All other user actions are

persisted, even by using the versioning functionality of XoWiki, and can be revisited by them. Personal mashup pages are created with a restriction for other users, which is guaranteed by an own creator privilege in the access control policy. Later on, it is planned to store all user interactions in the mashup page in order to build up a semantic layer, i.e. to be able to recommend tools for given user scenarios.



**Fig. 1.** This XoMashup page contains instructions for an exemplary learning activity (at the bottom) as well as three required tools (VideoWiki, XoWiki, Webmail).

At this point, the following problematic aspects of the current XoMashup implementation have to be outlined. Firstly, certain html-elements, like hyperlinks with the property “target=\_top”, may destroy the whole mashup page by loading the URI address into the top window. Hereby, heuristics for web-based tools to be included into XoMashup have to be determined to avoid such problems. Secondly, web applications should be customizable in terms of its display size in order to allow users to work with a tool in the original windows. As one remedy, a web application could provide parameters to initiate the display size, which we successfully realized for the ObjectSpot application mentioned in section 2. In both cases, the technological flexibility of this approach is restricted due to the demand for certain requirements of the targeted tools. Thirdly, combining different tools can be realized either on the basis of client-sided user actions, like copy-pasting content from one to another web

application, or by cross-domain scripting. Finally, the issue of common third-party functionality is still an open issue. For instance, user authentication could be implemented by web-based mechanism like OpenID [20] or in the application layer of XoMashup and the tools. Similarly, logging has to be considered for user interactions – which is our current topic of interest – and systemic events.

Despite these problematic aspects, we believe that our solution approach can be of benefit for many different application areas, and it is planned to be used within the iCamp project, an EU-funded research project in the field of higher education (<http://www.icamp.eu>). This mashup approach should serve as the basis for the so-called “iCamp Space”, an attempt to create a supportive environment for eLearning scenarios by means of combining learning content and web-based tools (e.g. the iCamp building blocks [2]). In short, the proposed result allows facilitators (administrators) to prepare learning templates consisting of Wiki-based content and web-based tools which either have already been started on the screen or which are to be launched by a learner on demand. Selecting one of these templates, learners (registered users) can create a new personal mashup page and work with it by using the tools, re-arranging the windows and modifying the Wiki content. All changes to such a personal learning page are persisted. Due to functionality provided by the XoWiki module and the OpenACS content repository, the mashup content and state are even under version-control allowing the learner to reproduce each learning step.

## 4 Conclusions and Outlook

All in all, the XoMashup component comprises a personal web application mashup page, which is useful for self-directed learning, but can also be utilized in other application areas. Addressing the iCamp project, this mashup solution is a first approach towards the iCamp space. As a next step, it is planned to extend this web application mashup with a semantic layer to recommend tools and tool combinations for given learning activities. Further, the semantic model should be refined in terms of analyzing the user interaction.

**Acknowledgments.** The iCamp project is funded by the EC IST 6<sup>th</sup> Framework Project (Contract number: 027163).

## References

1. Lennon, J., Maurer, H.: Why it is Difficult to Introduce e-Learning into Schools And Some New Solutions. In: *Journal of Universal Computer Science* 9(10), 1244--1257 (2003).
2. Kieslinger, B., Wild, F., Arsun, O.: iCamp: The Educational Web for Higher Education. In: *1<sup>st</sup> European Conference on Technology Enhanced Learning*, 640--645 (2006).
3. Safran, C., Helic, D., Gütl, C.: E-Learning practices and Web 2.0. In: *International Conference on Interactive Computer Aided Learning* (2007).
4. O'Reilly, T.: What is the Web 2.0, 2005, <http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html> (2007-11-20).

5. Information Multimedia Communication AG: Start für neue Web 2.0 Open Content Community SLIDESTAR, 2007, [http://www.im-c.de/News\\_und\\_Events/Aktuelle\\_News/389/5421.html](http://www.im-c.de/News_und_Events/Aktuelle_News/389/5421.html) (2007-11-20, in German).
6. Väljataga, T., Pata, K., Laanpere, M., Kaipainen, M.: Theoretical Framework of the iCampFolio: New Approach to Comparison and Selection of Systems and Tools for Learning Purposes. In: 2<sup>nd</sup> European Conference on Technology Enhanced Learning, 349--363 (2007).
7. Jhingran, A.: Enterprise Information Mashups: Integrating Information, Simply. In: 32<sup>nd</sup> International Conference on Very Large Data Bases, 3--4, VLDB Endowment (2006).
8. Neumann, G.: XoWiki: Towards a Generic Tool for Web 2.0 Applications and Social Software. In: OpenACS and .LRN Conference (2007).
9. Kulathuramaiyer, N.: Mashups: Emerging Application Development Paradigm for a Digital Journal. In: Journal of Universal Computer Science 13(4), 531--542 (2007).
10. Mödritscher, F., García-Barrios, V.M., Gütl, C.: Enhancement of SCORM to support adaptive E-Learning within the Scope of the Research Project AdeLE. In: Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education, 2499--2505 (2004).
11. Hammersley, B.: Content Syndication with RSS: Sharing Headlines and Information Using XML. O'Reilly, San Francisco (2003).
12. Bellas, F.: Standards for Second-Generation Portals. In: Internet Computing 8(2), 54--60 (2004).
13. Jackson, C., Wang, H.J.: Subspace: Secure Cross-Domain Communication for Web Mashups. In: 16<sup>th</sup> International Conference on World Wide Web, 611--620 (2007).
14. Spoerri, A.: Visual Mashup of Text and Media Search Results. In: 11<sup>th</sup> International Conference Information Visualization, 216--222 (2007).
15. World Wide Web Consortium (W3C): Inline frames: the IFRAME element, 1999, <http://www.w3.org/TR/html4/present/frames.html#h-16.5> (2007-12-14).
16. World Wide Web Consortium (W3C): Document Object Model (DOM), 2005, <http://www.w3.org/DOM/> (2007-12-14).
17. Lennartz, S.: Jedem das Seine: Personalisierte Startseiten im Überblick, 2006, <http://www.drweb.de/weblog/weblog/?p=524> (2007-11-21, in German).
18. Solution Grove: Web Desktop, 2007, <http://www.solutiongrove.com/mashup/portal> (2007-11-21).
19. Giorgi, M.: Comparison of AJAX Frameworks: Prototype, GWT, DWR and Thinware, 2007, <http://ajax.javabeat.net/articles/2007/05/comparison-of-ajax-frameworks-prototype-gwt-dwr-thinware/> (2007-11-21).
20. OpenID Foundation: OpenID, 2007, <http://openid.net/> (2008-01-08).