

# Mixing Content and Endless Collaboration – MashUps: Towards Future Personal Learning Environments

Andreas Auinger<sup>1</sup>, Martin Ebner<sup>2</sup>, Dietmar Nedbal<sup>1</sup>, and Andreas Holzinger<sup>3</sup>

<sup>1</sup> Upper Austria University of Applied Sciences, Department for e-Business, Steyr, Austria  
andreas.auinger@fh-steyr.at, dietmar.nedbal@fh-steyr.at

<sup>2</sup> Graz Technical University, Social Learning, Austria  
martin.ebner@tu-graz.at

<sup>3</sup> Medical University Graz, Research Unit HCI4MED,  
Institute of Medical Informatics, Statistics and Documentation, Austria  
andreas.holzinger@meduni-graz.at

**Abstract.** The recent movement by major Web services towards making many application programming interfaces (APIs) available for public use has led to the development of the new MashUp technology, a method of merging content, services and applications from multiple web sites. The new technology is now being successfully applied in the academic community to enrich and improve learning and teaching applications. This paper examines its implementation and use, discusses methods and styles of usage and highlights the advantages and disadvantages of client and server application, based on related work and recent experiences gathered with a large university-wide open learning management system (WBT-Master/TeachCenter of Graz University of Technology), which allows lecturers to use diverse web resources.

**Keywords:** mashups, personal learning environments, learning content, collaboration.

## 1 Introduction

The ubiquitous availability and pervasive use of new media and the web dominate our social life, our working environment and especially our teaching and learning behavior [1], [2], [3]. The Web is turning increasingly into a community of loosely connected participants, who share their knowledge and educational interests, hence communicate and collaborate in the course of the Web 2.0 trend [4], [5], [6]. Consequently, people combine the role of producer and consumer – they turn into *prosumers* [7] on a market of widely free available content and collaboration tools, provided by the Web 2.0.

In recent years, major Web services have opened their systems to outside use through the implementation of public application programming interfaces (APIs) and data sources, resulting in Web 2.0 MashUps.

MashUps are appearing on the web at an extremely fast rate, typically, three new MashUps appear on the web each day. Some of the newest ones are listed on the

“ProgrammableWeb”<sup>1</sup> site. MashUps are considered to be both a concept and technology for merging content, services and applications from multiple web sites in an integrated, coherent way [8], [9]. In principle, no special knowledge is required to create a MashUp; however, because high level programming languages are required to integrate different APIs and data sources, creating a MashUp still requires at least some basic programming expertise [10]. MashUps hold a great potential to connect content, collaboration and communication, which are considered to be the key objectives of successful virtual learning processes [11], [12] and consequently are crucial for e-learning 2.0 scenarios.

There are many different ways to use MashUps for e-learning 2.0. E-learning 2.0 is characterized by bringing social networks to the classrooms as well as using different Web 2.0 technologies for teaching and learning [7]. Bearing in mind that traditional learning management systems (LMS) are closed and form rigid platforms with tight restrictions concerning exchange and collaboration with the Web [13], MashUps can contribute towards creating an *open* learning environment. A complex system enabling endless and personal learning by the use of different web resources such as provided through MashUps is described as a Personal Learning Environment [14]. The opportunities are manifold; for example, freely available context-relevant interaction services (including Wikipedia, Google Maps, etc.) or communication platforms (Blogs, Microblogs) can be integrated into the Learning Content Management System (LCMS) in order to enrich the traditional content or learning examples [7], [15]. Another method is to let the students create their own MashUp pages for their course [16]. Hence, different learning scenarios require different technical solutions.

Technologically, MashUps are designed to be implemented on servers (server-side style) or clients (client-side style). For server-side implementation MashUps on the e-learning webpage, Yahoo Pipes, Microsoft Popfly, Google MashUp Editor and many others are available for free. To create a client-side MashUp, free tools such as iGoogle or Oosah can be used.

These two different MashUp styles are shown briefly within a technological outline in this paper and their benefits and weaknesses are discussed. Furthermore, the potential of web MashUps for Personal Learning Environments is discussed and accordingly shown by example lectures at Graz University of Technology.

## 2 Technological Outline

The two active ingredients of web MashUps are the data and application programming interfaces (APIs), which provide an interface to allow non-programmers to gain access to a malleable form of the data.

Both, data and APIs, can be public or private. MashUps have gained attention because of the creativity involved in their development and the functionality they afford users. If the Internet is thought of in superseding layers - *physical* (the wires), *logical* (the protocols), *content* and *social* - MashUps fit between the content and social layers, changing the way in which individuals relate to content [17].

---

<sup>1</sup> <http://www.programmableweb.com> (last visited: January, 5, 2009).

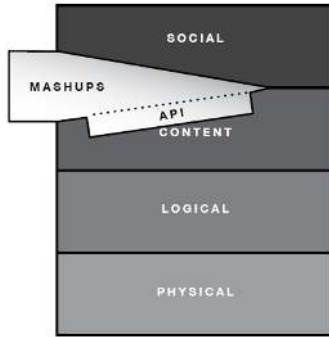


Fig. 1. MashUp Layers [16]

## 2.1 MashUp Styles

The two primary MashUp styles are server-side MashUps and client-side MashUps. As you might expect, server-side MashUps integrate services and content on a server and client-side MashUps on the client. Typically, MashUps use different channels like RSS-Feeds, APIs, Widgets or Web service interfaces to link various services. Servers act as a proxy between a client's web browser and the source web site that takes part in the MashUp. In a server-side MashUp all the requests from the client go to the server, which acts as a proxy to make calls to the other web site. Consequently, in a server-side MashUp, the work is either requested by or pushed to the client browser [9]. Client-site MashUps do not use a particular server that acts as a proxy, but combine different data and components within the client browser.

**Server-Side MashUps.** Usually, a server-side MashUp mixes the content on the web server and transfers it to the client via HTTP Protocol (e.g. RSS Feed). The following steps are performed during this process [9]:

- (1) A user generates an event on the client's web browser. The event triggers a JavaScript function on the client.
- (2) The client makes a request to the server ("proxy server") that provides the web site. The request is typically an AJAX ("Asynchronous JavaScript And XML") request. The main benefit of an AJAX request is that data is retrieved from the server asynchronously in the background without interfering with the display and behavior of the existing webpage.
- (3) A web component on the server receives the request and calls a method which encapsulates the code to connect and interact with the other web site in the MashUp.
- (4) The proxy server opens a connection to the MashUp site, i.e. the web site that provides the needed service.
- (5) The MashUp site receives the request, processes the request and returns data to the proxy server.
- (6) The proxy server receives the response and may transform it to an appropriate data format for the client.
- (7) The proxy server returns the response to the client.
- (8) The AJAX function updates the client's view of the page with the results from the server.

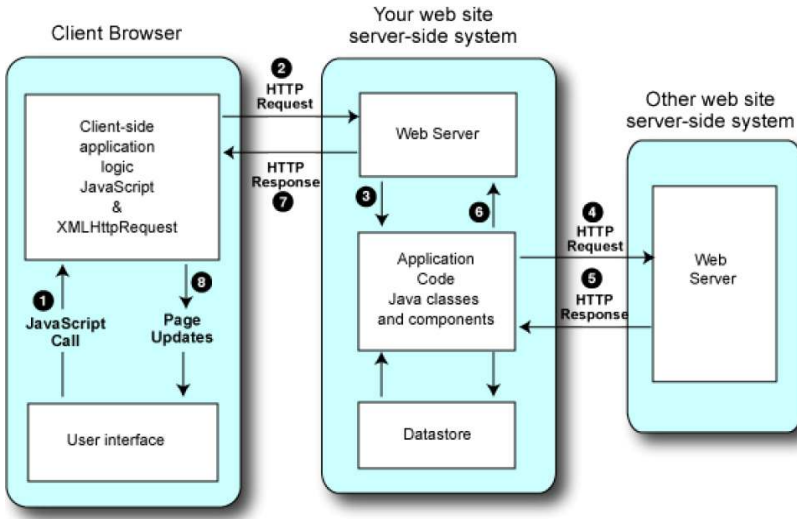


Fig. 2. Server-side MashUp concept [9]

**Client-Side MashUps.** In contrast to server-side MashUps, client-side MashUps integrate the data and components directly at the client's web browser [18]:

- (1) The browser makes a request to the server in the client's web site for a web page.
- (2) The server web site loads the page into the client. The page usually references a JavaScript library of functions from the MashUp site - for example from Google Maps - to enable the use of the MashUp site's service. The referenced library is then loaded into the web page.
- (3) Some action in the browser page (e.g. change of the map's zoom level) calls a JavaScript function from the MashUp site's library.
- (4) A request (typically an AJAX-message as well) is made to the MashUp server.
- (5) The MashUp site processes the request and returns data in the requested format.
- (6) The callback function updates the client's view of the page by updating the necessary parts of the client's webpage (e.g. uploading more detailed images of the map according to the current zoom level).

## 2.2 Pros and Cons of Web 2.0 MashUp Creation Tools

For creating server-side MashUps, Yahoo Pipes<sup>2</sup>, Microsoft Popfly<sup>3</sup>, Google MashUp Editor<sup>4</sup> and many other server-side tools are available for free on the web. To generate

<sup>2</sup> <http://pipes.yahoo.com> (last visited: August 2008).

<sup>3</sup> <http://www.popfly.com> (last visited: July 2008).

<sup>4</sup> <http://editor.googlemashups.com> (last visited: July 2008).

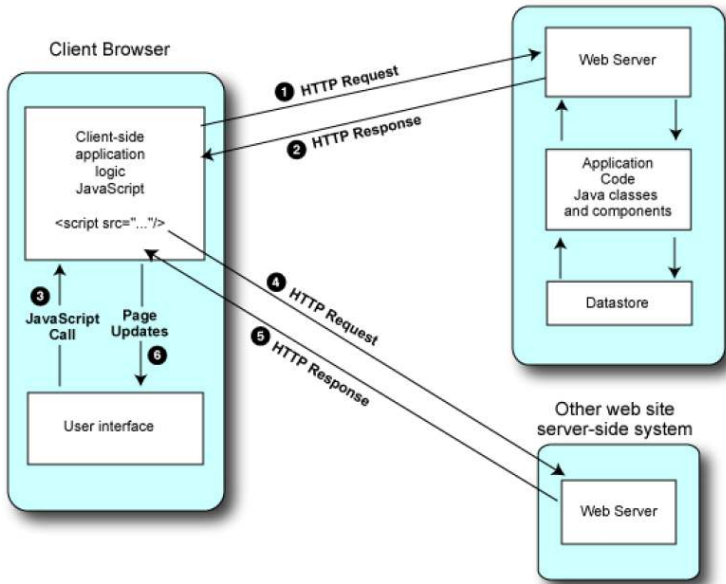


Fig. 3. Client-side MashUps concept [18]

client-side MashUps, simple MashUp platforms<sup>5</sup> or Oosah<sup>6</sup> can be utilized. These freely available Web 2.0 MashUp tools or sites have several strengths but also some weaknesses. The **pros** are [19], [20], [21]:

(i) *Applicability*: The tools available for MashUps are useful for diverse areas and support a great variety of input and output data types, from comma separated value (CSV) formats to structured XML and semantically rich RDF. Common MashUp tasks such as data integration by mapping identities can be easily performed, e.g. with Yahoo Pipes even without explicit coding.

(ii) *Ease of use*: The tools in general have intuitive designs and easy-to-use web interfaces that require little training for beginners. Extraction, recombination and integration of data with these tools is easier than writing code in a particular programming language. This helps unskilled users to gain some benefit with little effort and helps skilled users to do more powerful things.

(iii) *Direct manipulation*: Users are able to work directly with the data they are interested in, without having to think about abstract concepts such as programs.

(iv) *Reusability and extensibility*: MashUp tools are explicitly designed for sharing and reusing. The author of a MashUp can describe and publish his MashUp and others can reuse previously done work to add new features or customize and combine it into something new.

<sup>5</sup> <http://www.google.com/ig> (last visited: August 2008).

<sup>6</sup> <http://ww.oosah.com> (last visited: July 2008).

(v) *Interoperability*: Different tools can be easily combined to enhance the potential use of MashUps. For example, Dapper<sup>7</sup> can be used to fetch data in formats that are not supported by Microsoft Popfly on default.

(vi) *Active participation of users*: The participation of users is a major benefit in web 2.0 tools. Activities, such as reporting bugs, suggesting or even implementing new features by users, facilitate the improvement of applications much more rapidly than with traditional software engineering methods.

Despite these strengths some issues arise when using these MashUp tools. In general MashUps have to deal with five basic issues: (i) *data retrieval* can be complicated over multiple web pages, (ii) *source modeling* is needed to deduce relationships between existing data sources and new ones, (iii) *data cleaning* of new data sources might be necessary to conform to existing data, (iv) *data integration* is the process of combining several data sources and *data visualization* often requires programming to display the final data as desired [19]. To be more specific, the **cons** of MashUp tools can be summarized as follows [3]:

(i) *Performance and scalability*: Mashing large quantities of data using Web services from different sources can be very slow. A bandwidth or server bottleneck causing limited speed of the network connections and slow retrieval of the desired data is the result.

(ii) *Security*: Users have to bear the security risks when uploading their data to web 2.0 sites. Although the users may choose not to publish the work to the public, control of the work is relinquished once it is uploaded (or if directly programmed online) to a server.

(iii) *Missing features and instability*: This is due to the fact that the available tools are relatively new and sometimes still have “beta” status, i.e. some essential features may either be not supported or have limited functionality. In addition, bugs can lead to server instabilities and downtimes.

(iv) *Flexibility*: Although these tools are considered to be useful in common data MashUp use cases, they are not as flexible as individually programmed components. For certain tasks, writing custom code may be required.

(v) *Quality*: Due to the previously mentioned issues, the final output from MashUp tools may not match the quality that “professional” users can achieve with local software development.

### 3 The Use of MashUps for Teaching and Learning Purposes

The use of MashUps for learning and teaching purposes [22] is becoming more and more efficient and effective. Beside the initially mentioned terms e-learning 2.0 and Personal Learning Environments, buzzwords such as EduPunk [23] or informal learning [24] describe a complex system or an approach to teaching and learning using different web resources.

Bearing in mind that the very first e-learning activities resulted in the use of rigid learning management systems, often called e-learning 1.0 [13], nowadays Web 2.0

---

<sup>7</sup> <http://www.dapper.net> (last visited: August 2008).

technologies allow the combination of teaching and learning with social communities. By accepting that education is based on creativity and curiosity, communication and collaboration, and most of all on *active participation* [25], [26], [27], [28] it seems obvious that the Web is able to enhance such educational settings. For example students can create a MashUp page as an alternative to a blog to spur more interest in a course and the creation of one's own applications or widgets keeps the more savvy or interested student of IT engaged in the course that might otherwise seem dry and theoretical [16]. Several projects have been conducted that aim at improving the learning experience by the use of MashUps. In the following some relevant projects are outlined.

The We-LCoME system [7], developed by the University of Bologna, aims to mashup compound multimedia potentials with the so called "collective intelligence" the Web 2.0 has revealed. The system allows the cooperative creation and sharing of SMIL-based multimedia resources. By using SMIL as a key technology describing the spatial and temporal relations inside compound multimedia is possible. The system lets users add captions/subtitles and annotations to the original multimedia contents by resorting to a wiki-like interface. We-LCoME is capable of transforming the content production workflow, involving all the different actors playing a role in improving accessibility and effectiveness of learning materials.

The University of Pittsburgh has developed EduLINK, a solution to automatically link community wisdom to authors and ease various difficulties in authoring [15]. To automatically provide content the system uses MashUps to pull relevant examples for the users. The aim is to utilize the flexibility of MashUps to increase the value of authoring tools. Currently, the system uses socially tagged data from three well-known websites, Wikibooks, Delicious and YouTube as the resources.

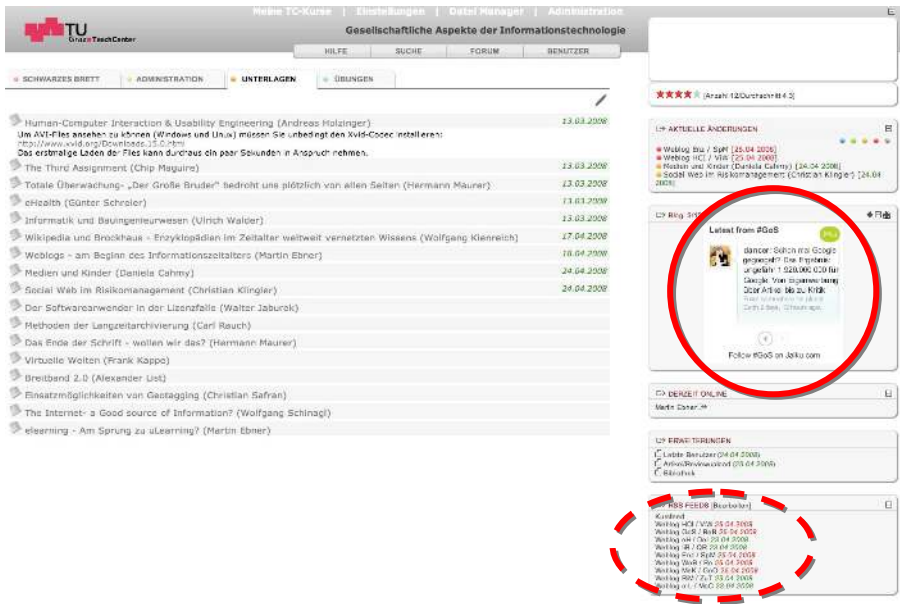
In [24] the authors describe the use of various Web 2.0 technologies for learning in a community. The referenced community, a newly formed land trust in Scotland, is provided with a MashUp of a wide array of individually proven Web 2.0 technologies (including YouTube, Flickr, Delicious, Wikipedia, Myspace, podcasting and internet broadcasting) in an innovative, inter-linking and collaborative manner. Informal learning by the community, both local and virtual, is also intended by the project.

At Graz University of Technology, the creation of a more open learning management system has been realized, allowing lecturers to use a diverse set of web resources. The university-wide system, called WBT-Master/TeachCenter [29], has been enhanced in different ways. Currently three different possibilities for mixing the content are being implemented:

(i) *RSS-Feeds*: Teachers can implement their collected RSS-Feeds into the course. All changes can be highlighted within the learning management system by aggregating it to the usual course feed.

(ii) *APIs*: Often used Web applications, for example Wikipedia, have been integrated within the LMS by using the provided API. In this particular case, students are able to search directly in Wikipedia articles via the LMS interface.

(iii) *Widgets/Web services*: Furthermore, the system allows the integration of different codes, which is provided by the origin website. The code may consist of pure html tags (Web service) as well as links to a JavaScript source (Widget).



**Fig. 4.** Learning Management System WBT-Master / TeachCenter using MashUps

Fig. 4 shows a typical course interface of an online lecture at Graz University of Technology: the solid circle on the right shows an embedded Widget from an external Web application. Students can easily follow the activities on this platform by entering the main lecture website. Furthermore, the dotted circle at the bottom of the right encloses all integrated external RSS-Feeds. These feeds are automatically added to the existing course feed.

To sum it up, this concept allows the mixing of various sources on the web so that it doesn't matter whether lecturers like to use specific Web applications for their teaching in the same way as students use them for learning purposes. Due to this fact, MashUps have the capability to enhance e-learning courses on a new and useful way and YouTube, Flickr, etc. can easily be integrated within this environment.

## 4 Conclusion

By using Widgets or APIs as well as RSS technologies, online lectures are enhanced by social platforms or other applications such as Flickr, YouTube etc. After first evaluations, it can be stated that the use of MashUps for learning is changing the role of the lecturer as well as the role of the learning management system (LMS) itself. The teacher turns from a content producer to a content facilitator by using the LMS as a central environment, which is responsible for structuring and supporting the learning activities. Furthermore, online collaboration and online communication using different channels (web resources) becomes more and more important. We can conclude that the use of MashUps for learning purposes has definitive a great potential.



Especially in consideration of current research work concerning Personal Learning Environments (PLE), remixing content and communicating through different channels (web resources) will be one of the most challenging aspects in the future of technology enhanced learning. Current research is aimed at developing a client-side MashUp to support and allow students to create their own view and to access the LMS. This work can be seen as a similar approach to iGoogle, although with special APIs to university wide platforms. However, by turning towards more flexibility, personality and individualism, MashUps are indispensable. Furthermore, the next generation of learning and teaching applications will change from rigid content distributors to content aggregators of different resources.

## References

1. Chen, N.S., Kinshuk, W.C.W., Yang, S.J.H.: Designing a self-contained group area network for ubiquitous learning. *Educational Technology & Society* 11(2), 16–26 (2008)
2. Holzinger, A., Nischelwitzer, A.K., Kickmeier-Rust, M.D.: Pervasive E-Education supports Life Long Learning: Some Examples of X-Media Learning Objects (2008), <http://www.wccee2006.org/papers/445.pdf> (last access, 2008-05-05)
3. Klamma, R., Chatti, M.A., Duval, E., Hummel, H., Hvannberg, E.T., Kravcik, M., Law, E., Naeve, A., Scott, P.: Social software for life-long learning. *Educational Technology & Society* 10(3), 72–83 (2007)
4. O'Reilly, T.: What is Web 2.0 Design Patterns & Business Models for the Next Generation of Software (2009), <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html> (last access January 5, 2009)
5. O'Reilly, T.: Web 2.0: Stuck on a name or hooked on value? *Dr. Dobbs Journal* 31(7), 10 (2006)
6. Ebner, M., Holzinger, A., Maurer, H.: Web 2.0 Technology: Future Interfaces for Technology Enhanced Learning? In: Stephanidis, C. (ed.) *HCI 2007*. LNCS, vol. 4556, pp. 559–568. Springer, Heidelberg (2007)
7. Ferretti, S., Mirri, S., Muratori, L.A., Rocchetti, M., Salomoni, P.: E-learning 2.0: you are We-LCoME! In: *Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A)*, pp. 116–125 (2008)
8. Tuchinda, R., Szekely, P., Knoblock, C.A.: Building Mashups by example. In: *Proceedings of the 13th international conference on Intelligent user interfaces*, pp. 139–148 (2008)
9. Ort, E., Brydon, S., Basler, M.: Mashups Styles, Part 1: Server-Side Mashups (Sun Microsystems), [http://java.sun.com/developer/technicalArticles/J2EE/mashup\\_1/](http://java.sun.com/developer/technicalArticles/J2EE/mashup_1/) (last access January 10, 2009)
10. Zang, N., Rosson, M.B., Nasser, V.: Mashups: who? what? why? In: *CHI 2008 extended abstracts on Human factors in computing systems*, pp. 3171–3176 (2008)
11. Auinger, A., Stary, C.: Ubiquitous access to learning material in engineering. In: Stephanidis, C. (ed.) *HCI 2007*. LNCS, vol. 4556, pp. 481–490. Springer, Heidelberg (2007)
12. Auinger, A., Stary, C.: Verknüpfung von Content und Kommunikation für selbstgesteuerten, webbasierten Wissenstransfer. In: Szwillus, G., Ziegler, J. (eds.) *Mensch & Computer 2003*, Teubner, pp. 359–369 (2003)
13. Ebner, M.: E-Learning 2.0 = e-Learning 1.0 + Web 2.0? In: *Proceedings of the The Second International Conference on Availability, Reliability and Security*, pp. 1235–1239 (2007)

14. Schaffert, S., Hilzensauer, W.: On the way towards Personal Learning Environments: Seven crucial aspects, <http://www.elearningeuropa.info/files/media/media15971.pdf> (last access January 15, 2009)
15. Hsiao, I.H., Li, Q., Lin, Y.-L.: Educational social linking in example authoring. In: Proceedings of the 19th ACM conference on Hypertext and hypermedia, pp. 229–230 (2008)
16. Hardy, N., Pinto, M., Wei, H.: The impact of collaborative technology in it and computer science education: harnessing the power of web 2.0. In: Proceedings of the 9th ACM SIGITE conference on Information technology education, pp. 63–64 (2008)
17. Palfrey, J., Gasser, U.: Mashups Interoperability and eInnovation. Case Study. Berkman Publication Series. Harvard University Research Center of Information Law and University of St. Gallen, St. Gallen (2007)
18. Ort, E., Brydon, S., Basler, M.: Mashups Styles, Part 2: Client-Side Mashups. Sun Microsystems (last access January 15, 2009), [http://java.sun.com/developer/technicalArticles/J2EE/mashup\\_2/](http://java.sun.com/developer/technicalArticles/J2EE/mashup_2/)
19. Cheung, K.H., Yip, K.Y., Townsend, J.P., Scotch, M.: HCLS 2.0/3.0: Health care and life sciences data mashup using Web 2.0/3.0. *Journal of Biomedical Informatics* 41(5), 694–705 (2008)
20. Ennals, R., Gay, D.: User-friendly functional programming for web mashups. In: Proceedings of the 12th ACM SIGPLAN conference on Functional programming, pp. 223–234 (2007)
21. Zhang, Z., Cheung, K.H., Townsend, J.P.: Bringing Web 2.0 to bioinformatics. *Briefings in Bioinformatics* 10(1), 1–10 (2009)
22. Kulathuramaiyer, N., Maurer, H.: Current Developments of MashUps in Shaping Web Applications ED-Media 2007, pp. 1172–1177 (2007)
23. Ebner, M.: Why We Need EduPunk. *Journal of Social Informatics* (9), 1–9 (2008)
24. Mason, R., Rennie, F.: Using Web 2.0 for learning in the community. *Internet and Higher Education* 10, 196–203 (2007)
25. Dewey, J.: *Democracy and Education. An introduction to the philosophy of education* (Reprint 1997). Free Press, Rockland (1916)
26. Schank, R.C.: Active learning through multimedia. *IEEE Multimedia* 1(1), 69–78 (1994)
27. Holzinger, A.: *Multimedia Basics*, Laxmi, Delhi. *Learning. Cognitive Fundamentals of multimedial Information Systems*, vol. 2 (2002), <http://www.basiswissen-multimedia.at>
28. Holzinger, A., Kickmeier-Rust, M.D., Wassertheurer, S., Hessinger, M.: Learning performance with interactive simulations in medical education: Lessons learned from results of learning complex physiological models with the HAEMODynamics SIMulator. *Computers & Education* 52(2), 292–301 (2009)
29. Helic, D., Maurer, H., Scerbakov, N.: Knowledge transfer processes in a modern WBT system. *J. Netw. Comput. Appl.* 27(3), 163–190 (2004)