useKit * – Lightweight Mashups for the Personalized Web

Sven Rizzotti
Department of Computer Science
Klingelbergstrasse 50
4056 Basel, Switzerland
Sven.Rizzotti@unibas.ch

Helmar Burkhart
Department of Computer Science
Klingelbergstrasse 50
4056 Basel, Switzerland
Helmar.Burkhart@unibas.ch

ABSTRACT

In this paper, we describe a platform that allows users to personalize a menu bar with lightweight Mashups. These Mashups (useKit Missions) can integrate static and dynamic content and tools from various Web sources and let users profit from a more efficient workflow in the Web. useKit has similarities to browser plugins but differs from plugins in the way that useKit does not need installation effort and is browser- and operating system independent. Furthermore useKit allows taking advantage in the combination of Missions and collaboration among different users.

Categories and Subject Descriptors

H.3.5 [**Information Systems**]: Information Storage and Retrieval—*On-line Information Services, Web-based services*; D.2.6 [**Software**]: Software Engineering—*Programming Environments, Integrated environments*; H.5.2 [**Information Systems**]: Information Interfaces and Presentation—*User Interfaces, Interaction styles*

General Terms: Individual service composition

Keywords: Web3.0, Mashup, Personalization

1. THE PROBLEM

Very often Web content is not available in a form that supports a users need. This problem is addressed with mashups that combine existing Web content and Web services in a new way. But Mashups need technical expertise to implement. Browser extensions can offer similar functionalities to Mashups but need to be installed, maintained and are limited to a single browser instantiation.

2. USEKIT - PERSONALIZED MASHUPS

useKit is a platform that focuses on non-expert users and collaborating people and empowers them to use individual selected services, named *Missions*. Missions are a combination of services and play the role of lightweight Mashups. Missions make workflows within the Web more efficient, as in the area of research or communication.

With useKit users are able to select Missions and apply them to any Web content they are looking at. Users do not have to bother with software installation or maintenance nor do they have to understand varying user interfaces. Users also do not need to hassle with different authentication mechanisms and registration procedures. Additionally, they find a central place to discover suitable Missions to choose from.

Whenever a user involves a Mission, it allows useKit to get a better understanding about relevant content to the user. This allows improving the entire system with suggestions and optimizations that better fits the need of the user.

The useKit menu bar can be added to a Web page through a browser extension or by manual activation of useKit through a previously stored bookmark. Both variants bring the Mission selection of the user right into the visited Web page.

An alternative to the user driven activation is the possibility to include the useKit menu bar constantly to a specific Web site. This gives Web site providers the opportunity to add their own Missions to the menu bar. Every user who visits the Web site will have access to provided Missions but still has access to the personal selection.

2.1 useKit Mission

A useKit Mission is a pluggable operation that can manipulate presentation or content of a Web page. A Mission can combine several Web services to fulfill a specific task. The translation Mission for example replaces the text of a webpage with a translation into another language. The Mission offers translations from different origins such as Google[1] or Bing[2] if the translations turn out to be different.

A Mission does not connect directly to an external service but rather contacts to the useKit server first. The server initializes invocations of external services which gives a chance to select alternatives based on quality of service, response time or user preferences. It also allows tracking Mission activities and combine different Missions with each other. The tracking of Mission activities is kept in a notification stream, named LifveLog, which is described in section 2.4.

An example of a Mission is the *Cut'nGo* Mission, which allows storing Web clippings for further usage.

The Cut'nGo Mission allows cutting out data parts by selecting relevant areas and stores the cuts into a global accessible repository. The Mission can also be used to cut off information that is not relevant which can save paper when printed out.

2.2 Mission patterns

Missions follow one of three patterns: the presentation, content, or integrating pattern. *Presentation Missions* are

*An alpha version is available at `http://useKit.com`

WWW2010, April 26-30, 2010, Raleigh, North Carolina.

[1] `http://translate.google.com`
[2] `http://www.microsofttranslator.com`

used to change a Web site's appearance. Rearranging Web content, changing element sizes, fonts or colors or introducing new functionality to the user interface are examples of this kind. Enhancements in this pattern only include changes to the visual component of a Web site without consulting other Web services. Extensions of interactive elements such as formulas or static elements such as images, hyperlinks or table structured data also belong to this pattern.

Missions following the *content pattern* use external services or information providers to add additional functionality to a Web site. These Missions build a Data Mashup[6] where data from the current Web page can be intermixed with data coming from external services.

Missions that belong to the *integration pattern* integrate and combine more than one external site in order to fulfill their task. The focus of these Missions lies on the integration and controlling of the information providers. They lay out the structure for service Mashups[6].

2.3 Collaboration

Most of the Mission content can be shared with other useKit members. Friends, who are allowed to see another users activities can follow Mission actions on a timeline or may participate in collaboration. The Sticky Mission for example is an implementation of a shared notebook in the Web. It allows annotations of Web pages across different Web sites.

2.4 useKit LifveLog

The *LifveLog* represents the activity timeline of a useKit member. Every activity within useKit has a corresponding entry in the LifveLog. Most of the entries in the LifveLog should be of relevance to the user since the user has decided to use a Mission to process the content in some way. Depending on the privacy settings, other users get notified and can participate in the activities of a user. A user can review and reconstruct the work that has been done and will be able to replay it in the future.

3. RELATED WORK

Several work have addressed web automation and modifications such as changing the presentation or the content of web pages and can be grouped into client side extensions for Web browsers and proxy servers acting as mediator between the browser and contacted web sites.

Greasemonkey [3] is an example of the first group and allows users to load and install scripts that make on-the-fly changes to specific web pages. Greasemonkey scripts are persistent and changes made to the web pages are executed every time the page is opened, making them effectively permanent for the user running the script. The functionality of Greasemonkey is limited to FireFox, but other products exist for the Safari browser. useKit offers similar functionalities but puts more focus on user activities and leaves semantic analysis to the user.

Marmite [5] is an end-user programming tool, which can be used to create Mashups. Marmite supports a data flow architecture, where data is processed by a series of operators in a manner similar to Unix pipes. Marmite works by displaying a linked data flow / spreadsheet view, letting people see the program as well as the data simultaneously. The Marmite tool offers a comfortable way for building new mashups but its target audience are users willing to create new constellations rather than users who focus on usage only which lies in the focus of useKit. The Marmite project is no longer active.

Yahoo! Pipes[2] is a representative of graphical programming. It provides a user interface for building applications that aggregate web feeds and other services. These applications can combine various sources with a piping mechanism and are being published after additional filter rules have been defined. In comparison to useKit, Yahoo! Pipes follows a different approach. useKit focuses on a single Web page that is being manipulated on a users wish while Yahoo! Pipes runs a Mashup at an external site.

Intel Mash Maker[1] can take data elements from multiple websites and mash them together into a single, integrated view. Mash Maker is a browser extension that learns what information a user is interested in and automatically creates personalized mashups with content and visualizations from other sources on the web, and brings it all together in one place. Mash Maker has similarities to useKit but does not offer server side processing and data gathering as with useKit.

4. CONCLUSIONS AND OUTLOOK

We have presented useKit, a platform that permits non-experts to use and personalize lightweight Mashups in the context of existing Web applications. useKit allows to discover new Missions and to compile an individual menu bar with selected Missions. Missions can alter the presentation or behavior of any Web page a user visits. Mission data can be shared with friends who offer a collaboration area above the Web. Additionally, usage of Missions generates entries in the so-called LifveLog that represents relevant activities of a user. The activities can be shared with friends or reviewed or replayed later.

The useKit platform is currently in alpha state. Future plans include the embedding of mobile devices, which is predestinated for tools that increase efficiency[4] and the opening up with a public API.

5. REFERENCES

[1] R. Ennals and M. Garofalakis. Mashmaker: Mashups for the masses. *ACM SIGMOD International Conference on Management of Data*, pages 1116 – 1118, 2007.

[2] T. Loton. *Mashup Case Studies With Yahoo! Pipes*. CreateSpace, November 2008.

[3] M. Pilgrim. *Greasemonkey Hacks*. O'Reilly & Associates, 1 edition, November 2005.

[4] A. Voulodimos and C. Patrikakis. Using personalized mashups for mobile location based services. *Wireless Communications and Mobile Computing Conference*, pages 321 – 325, 2008.

[5] J. Wong and J. I. Hong. Making mashups with marmite: towards end-user programming for the web. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1435–1444, New York, NY, USA, 2007. ACM.

[6] G. O. Young, E. Daley, M. Gualtieri, H. Lo, and M. Ashour. The Mashup Opportunity. *Forrester Research*, May, 6 2008.