

Mash-ups and CSCW: opportunities and issues

M. Cameron Jones

University of Illinois at Urbana-Champaign
Graduate School of Library and Information Science
501 E. Daniel St. Champaign, IL 61820 USA
+1-217-333-3280

mjones2@uiuc.edu

Michael B. Twidale

University of Illinois at Urbana-Champaign
Graduate School of Library and Information Science
501 E. Daniel St. Champaign, IL 61820 USA
+1-217-333-3280

twidale@uiuc.edu

ABSTRACT

In this workshop, we explore emerging research themes in web mash-ups which are relevant to CSCW research. Web mash-ups are web sites which combine data or services from more than one source into an integrated whole. The mash-up community resembles open-source communities and likewise could benefit from further study. We explore several themes in and related to mash-ups which connect to core research areas in CSCW. Mash-ups have applicability to rapid prototyping and rapid application design methods and could be useful in participatory design practices. Through presentation and analysis of real mash-ups and discussion with mash-up developers, workshop attendees will help explore research opportunities for mash-ups in CSCW.

1. INTRODUCTION

The World Wide Web is undergoing a drastic change from a content storage and delivery system to an active platform which supports participation, expression, development, innovation, and creativity [11]. At the forefront of application development in this new era of web as platform for development are web mash-ups, websites or applications that integrate content and services from more than one source into a single, integrated whole. Mash-ups caught the attention of the HCI community at this year's CHI conference in Montreal, Canada [9]. We believe that many CSCW researchers will also find mash-ups an intriguing topic to study, or apply as a method.

The phenomenon of web mash-ups encompasses an array of research topics, many of which are already core subjects of CSCW research. The whole concept of a mash-up entails both explicit and implicit coordination among service providers and developers. Where data and service providers have not, in the past, explicitly embraced or supported mash-up activities, developers have co-opted the technology to do so. For instance, early versions of Google Maps did not have an open access API. After Paul Rademacher reverse engineered the Google Maps API for <http://www.housingmaps.com/> and other developers copied his hack, Google released a formal, public API to control and facilitate further mash-up development.

Google Maps has become so popular, in fact, that numerous sites have been created to make building Google Maps mash-ups as easy as filling out a web form, without requiring the user to have any programming knowledge (e.g., <http://www.mapbuilder.net>). Thinking beyond Google Maps, we can envision similar frameworks for other services and data sources which would allow users to create a variety of mash-up applications. By lowering the barriers to development, mash-ups could empower end-users with the ability to program customized solutions similar to how spreadsheets enabled end-users without specific

programming skills to quickly and easily create and share applications such as payrolls, budgets, and numerical models [9] by providing a reusable framework for rapidly testing and developing numeric applications.

The minimal amount of coding required to build a mash-up and the relative speed with which they can be coded makes mash-ups useful in rapid prototyping, allowing developers to quickly build a system with high levels of functionality and fidelity with minimal investment of time and effort [4]. Currently, the authors are exploring the accessibility of mash-up prototyping in their research and teaching activities. Undergraduate students without prior programming experience enrolled in a web technologies course have constructed prototypes of web mash-ups in just three weeks, using primarily HTML and JavaScript. Most of the mash-up programming has been completed by students copying other people's examples and mash-up code and modifying it, much like how many people learn HTML or how open-source communities share code [14]. Very little original code has been written by the students themselves.

In the authors' previous research [3], [4], [5], [6], a mash-up design pattern has been used to design and build web-based systems for synchronous and asynchronous collaboration. By combining applications, data, and services together, researchers have the ability to rapidly explore a design space utilizing participatory methods of involving users in application development and evaluation (see Figure 1). Unlike other

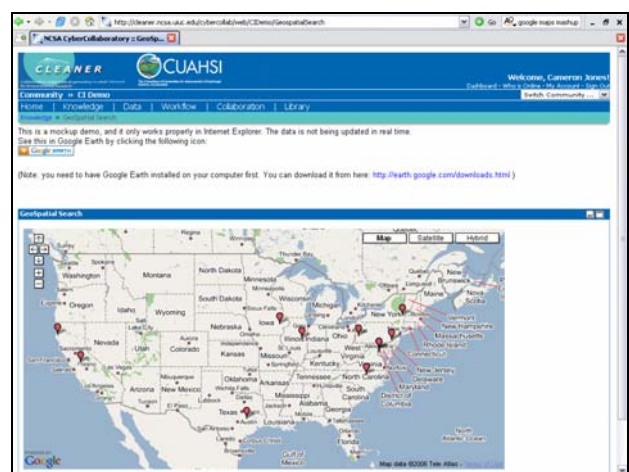


Figure 1. Cybercollaboratory Prototype: The cybercollaboratory uses mash-ups to prototype a “GeoSpatial search engine”. This site also uses a generalized mash-up design pattern to prototype a collaborative writing system using MediaWiki.

approaches to participatory design, a mash-up-based approach can put real functionality in the hands of the users and get real-time feedback of user needs as the tools can be embedded in existing work activities [4], [5].

Indeed mash-ups can also be used to enhance existing collaborative systems by reusing and remixing data in new ways, combining it ways the system wasn't originally designed to support, bringing in external data from other systems, or exporting data to other applications.

In the course mentioned above, one group of students is exploring how the university course timetables could be remixed in a new registration mash-up to improve the class registration interface. Another group of students is taking apartment listings and restaurant reviews from the student newspaper and creating a campus-town mash-up where students can find information on local resources and provide comments, ratings and reviews.

2. WORKSHOP THEMES

There are numerous themes to explore, far too many to fit in any one workshop. In the following we outline some major themes in mash-ups we see as particularly interesting and relevant to CSCW research which we wish to explore further in the workshop discussions.

A central theme in mash-ups is sharing; mash-up developers share their applications, experience, examples, documentation, and so on. One main forum for sharing is the website <http://programmableweb.com/> where developers can publish their mash-ups, review and comment on each others mash-ups, and generate ideas for future development. There are also related open source projects for developing toolkits in numerous languages to access various APIs. These collaborative activities surrounding mash-ups warrant further investigation.

The emergence of standards, protocols, APIs and practices in the mash-up universe is decentralized. This environment is similar to open-source development and further parallels need to be explored. How are these de facto standards formed and then adopted? What are the communication processes and dissemination patterns within this community? How can we apply the insights from the analysis of the processes of open source development to the processes of the development, use and innovation involving mash-ups [12]?

Jones and Twidale [7] reported on the effects of publishing personal activity. The decision to publish the information using RSS afforded its inclusion in mash-ups and supported activity-awareness and collaboration. Similar features could be added to other aspects of personal information managements and could be extended to team information management systems. This leads to two research themes; one, how publishing personal information streams can be used to design collaborative applications and afford collaborative interaction; and two, how systems in general can be designed to be mashed-up.

In a similar theme, mash-ups may have value as examples of technology appropriation. When data and/or systems are combined and remixed it may violate their designed or intended purpose, allowing for an investigation of the dynamics of technology production and consumption [2]. Beyond the examination of mash-ups as *artifacts of appropriation*, mash-ups may also provide a viable platform for supporting the “distributed cooperative appropriation, circumvention, [and] modification”

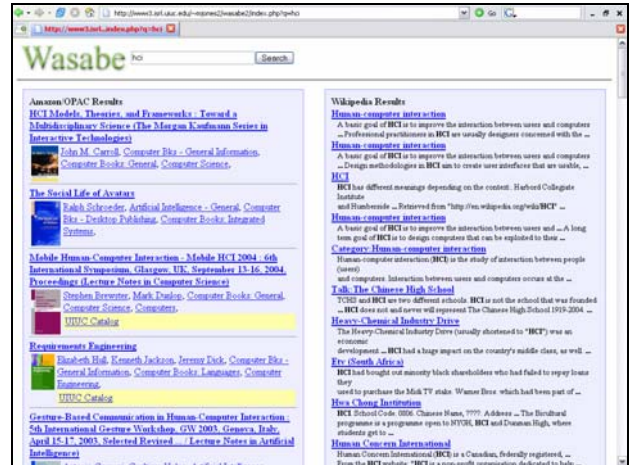


Figure 2. Wasabe: A mash-up search engine which searches the library OPAC using Amazon and Wikipedia.

described by Schmidt as a central challenge in CSCW [13]. In other words, a mash-up design approach may actually support further appropriation. Connected with concepts of appropriation is the notion that mash-ups can be a powerful tool in end-user innovation, allowing users to creatively combine technologies to create tailored solutions to their immediate needs [15].

Mash-ups often involve social media like wikis, social bookmarks, blogs, commenting and review systems, etc. Mash-ups may consume, process, and remix data from these social systems in new applications, or adopt components of those systems and/or their interfaces. An example of this can be seen in Figure 2, the Wasabe mash-up. Wasabe was a prototype search engine built by the authors to explore alternative ways of searching the library catalog. The system uses Amazon's search engine and Wikipedia to construct a list of books relevant to a particular search. The Amazon search engine returns results which contain links to 'similar items', the similarity of these items comes from the collective purchasing habits of Amazon's customers, acting like a collaborative filter [8]. Likewise, the Wikipedia results rely on the collaborative construction of content and links among articles.

Although the application is not intended for explicit collaboration on the end-user side, performance of the Wasabe search engine is dependant upon collaborative interactions. This use of collaboratively generated data and other artifacts of collaboration taken outside of the initial collaborative context in which they were created is an interesting issue which has not received much, if any, attention in the past. Traditionally, CSCW has been studied with respect to two main axes, time and space. Mash-ups like Wasabe which utilize collaboratively generated data add a third dimension on analysis, purpose. Artifacts of collaboration can be used for either same or different purposes than those for which they were originally created. CSCW research tends to analyze collaborative data *within* the context of the collaboration; mash-ups afford the opportunity to look across usage boundaries.

3. ACTIVITIES & GOALS

The goals of this workshop are to introduce CSCW researchers to the emerging development of mash-ups and to situate and discuss their role in CSCW research. Participants will leave the workshop with not only new knowledge of how mash-ups work and how

they may benefit their research, but also electronic and print resources to help them to build their own mash-ups. Workshop organizers will provide a CD-ROM of mash-up construction and reference materials to all participants. This CD will contain examples, web-service APIs, documentation, and other applications like the Apache web server and MySQL database server.

3.1 Morning Program

The morning portion of the workshop will focus on technical discussions of what mash-ups are and how they are built. This discussion will be designed to establish a common foundation for further discussion of research topics related to mash-ups. Discussion will focus around the cases of mash-up development. In total, four to six cases will be presented for discussion. These cases will be selected from the participants' submissions, the organizers' own development experience, and knowledge of existing web mash-ups. The cases will be selected to represent a diverse mix of topics, themes, and services.

We also plan to invite those with development experience (including one of the workshop organizers) to actually build one or more mash-ups live, during the morning session. This has been a practice at various recent mash-up camps [1]. This will afford an opportunity for the other workshop participants to observe the process of doing mash-ups and serve as an introduction to later discussion about studying the activity.

3.2 Afternoon Program

The afternoon program will be focused on establishing a map of research topics related to mash-ups which are relevant to CSCW research. Participants submitting position papers will have an opportunity to give a short presentation (approximately five minutes) after which the participants will break-up into small discussion groups. Each group will pick a theme and generate ideas, concepts and research questions related to their theme which connect to broader CSCW research goals.

3.3 Schedule of Activities

Here is a rough timeline of the activities planned for the workshop. This schedule is not to be interpreted as final, and will be adjusted in response to the submissions received.

08:30 – 09:00	Coffee and welcome
09:00 – 10:30	What are mash-ups? Introduction to mash-up concepts and technologies
10:30 – 11:00	Coffee break
11:00 – 12:00	Mash-up case presentations and discussion
12:00 – 13:30	Lunch
13:30 – 14:30	Participant presentations
14:30 – 16:30	Small group discussions
16:30 – 18:00	Small group presentations and whole group discussion

4. PARTICIPANTS

Participants will be asked to submit position papers to the organizers. Position papers could either (i) situate the participants' interests and background among the themes of the workshop, or (ii) report on preliminary research findings or case studies where the mash-up concept has been applied.

Additionally or alternatively participants may also submit, or point to and describe a mash-up that they have created. We hope to recruit a diverse set of no more than 20 participants, and intend this workshop to appeal to people with a range of research interests and theoretical traditions including:

- systems developers with some experience of mash-up or mash-up-like development;
- systems developers interested in creating better environments for mash-up development, including by less skilled developers;
- people interested in the psychology of programming;
- ethnographers interested in studies of software development;
- people interested in analyzing collective software development;
- people interested in creativity and design;
- people interested in rapid application design techniques and participatory design techniques.

Our selection criteria for the workshop will be influenced by a desire to cover a range of backgrounds including, but not limited to, those described above.

5. ORGANIZERS

Michael B. Twidale is an Associate Professor in the Graduate School of Library and Information Science at the University of Illinois at-Urbana-Champaign. Dr. Twidale is a long-standing member of the CSCW research community whose research interests include rapid prototyping and evaluation, the usability of open source software, collaborative aspects of data-quality management, applying open source models to systems analysis and design, and end-user appropriation.

M. Cameron Jones is a graduate student in the Graduate School of Library and Information Science. Under the guidance of Dr. Twidale, Mr. Jones is exploring aspects of end-user appropriation and end-user innovation in his research, looking for technologies and design methods which afford appropriation and empower users with the tools they need to create tailored solutions.

Mr. Jones is currently teaching an undergraduate course titled, "Web Technologies and Techniques", in the Graduate School of Library and Information Science. In this course students are exploring the use of HTML, CSS and JavaScript in creating simple mash-up prototypes. Mr. Jones is planning another undergraduate course for the fall 2006 semester titled "Programming Web Mash-ups" which will explore server-side programming, data modeling, and other concepts through the lens of building mash-ups.

Mr. Jones has been named a Graduate Scholar for the Academy of Entrepreneurial Leadership (AEL) at the University of Illinois at Urbana-Champaign to study mash-ups and their application to end-user innovation. As part of his AEL grant, Mr. Jones will be interviewing several key figures in the mash-up community and organizing local workshops and tutorials on mash-up development.

6. REFERENCES

- [1] <http://mashupcamp.com/>

- [2] Eglash, R. (2004). Appropriating Technology: an introduction. In *Appropriating Technology: Vernacular science and social power*, R. Eglash, J. Crossiant, G. Di Chiro, and R. Fouché (Eds.). University of Minnesota Press.
- [3] Gruzid, A. & Twidale, M. (2006) Write While You Search: Ambient Searching of a Digital Library in the Context of Writing. Submitted to the JCDL Workshop on Digital Libraries in the Context of User's Broader Activities. Available at: <https://netfiles.uiuc.edu/agruzid2/shared/CUBAGruzid.pdf>
- [4] Jones, M.C., Floyd, I.R. (forthcoming). Patchworks of Open-Source Software: High-Fidelity Low-cost Prototypes. In "Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives", K. St Amant, B. Still (Eds). Idea Group, Inc.
- [5] Jones, M.C., Floyd, I.R., Twidale, M.B. (2006). Patching Together Prototypes on the Web. Submitted as a Notes paper to CSCW 2006.
- [6] Jones, M.C., Rathi, D., & Twidale, M.B. (2006). Wikifying your Interface: Facilitating Community-Based Interface Translation. Proceedings of DIS 2006.
- [7] Jones, M.C., Twidale, M.B. (2006). Snippets of Awareness: Syndicating Copy Histories. Submitted as a Notes paper to CSCW 2006.
- [8] Linden, G., Smith, B., York, J. (2003). Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing*. January/February 2003, pp 76-80.
- [9] Metcalf, B., Taylor, B. (2006). Add a dash of interface: Taking mash-ups to the next level. In Plenary Panel, Wednesday, April 26, 2006, David Gilmore (Session Chair) at CHI 2006, Montreal, Quebec, Canada.
- [10] Nardi, B.A. & Miller, J.R. (1991). Twinkling lights and nested loops: Distributed problem solving and spreadsheet development. *International Journal of Man-Machine Studies*, 34(2), 161-184.
- [11] O'Reilly, T. (2005). What Is Web 2.0? Design Patterns and Business Models for the Next Generation of Software. Available at: <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>
- [12] Raymond, E.S. (2001). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly Media, Inc.
- [13] Schmidt, K. (1991). Riding a Tiger, or Computer Supported Cooperative Work. In the Proceedings of the European Conference on Computer Supported Cooperative Work.
- [14] Singh, V. Twidale, M.B. & Rathi, D. (2006). Open Source Technical Support: A Look at Peer Help-Giving. Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06).
- [15] Thomke, S., von Hippel, E. (2002). Customers as Innovators: A New Way to Create Value. *Harvard Business Review*. April, 2002. 80(4), 74-81.