

Using Mash-ups for End-User Rapid and Responsive Prototyping in Collaborative Environments

Ingbert Floyd

University of Illinois at Urbana-Champaign
Graduate School of Library and Information Science
ifloyd2@uiuc.edu

ABSTRACT

It is currently difficult for non-programmers to create most kinds of web mash-ups. However, several web services have already been created which allow nearly anyone with access to the internet to create a simple google maps web mash-up. Some examples include mapbuilder.net, wayfaring.com, and quikmaps.com. Use of these sites is starting to go mainstream; for example, quickmaps.com has been used by a local TV news station to provide a map to accompany one of their stories (http://www.wgrz.com/news/news_article.aspx?storyid=39171). In addition, people are appropriating existing web services in order to create mash-ups of their own without needing to program. One excellent example of this is a Flickr-Google Maps mash-up where a Flickr user used a satellite image of his hometown obtained from Google Maps plus the annotation feature of Flickr to convey some of the memories from his childhood (<http://www.flickr.com/photos/mathowie/8496262/>). This mash-up inspired many people to copy him and do the same thing for their childhood or other memories, and another user even set up a group called MemoryMaps to collect these mash-ups together (<http://www.flickr.com/groups/51468602@N00/>). Finally, classes aimed at teaching non-programmers how to create web mash-ups are starting to be taught in universities, with significant success (Floyd, Jones, Rathi & Twidale 2007). These examples provide evidence that there is vast potential for an explosion of mash-ups created by non-programmers once mature web services to support such endeavors exist, or once knowledge of how to create simple mashups by copying and pasting webpage sourcecode becomes more widespread. (To see or contribute more examples of mash-ups by non-programmers, see the workshop wiki page: http://mashworks.net/wiki/Building_Mash-ups_for_Non-Programmers.) This position paper explores how a rich array of tools which non-programmers could use to create web mash-ups could be a tremendous boon for computer supported cooperative work (CSCW) researchers and applications developers.

The introduction of collaborative computerized work systems such as enterprise management systems has given workers the ability to process information at a speed and with functionality never dreamed of before. However, this capability has come with costs. One important cost has been the shift of the locus of control over the process from the

people who are experts in the process because they live and work it every day to a separate group of people who are experts in creating the tools needed to support the process, but have rarely if ever engaged in the actual process itself. Even more surprising, is that these new experts are often work for a totally different organization. Essentially, many modern corporations, both for-profit and non-profit, are outsourcing the decision-making of how the internal workings of their business should be structured to people whose expertise is in manufacturing the tools which are needed to structure the inner workings. This is as if business owners in the 1800s and early 1900s were to rely on paper makers and printers to decide what the sales-order processing system should be for their company. Or it is like entrusting the pipe manufacturer to design the plumbing system in your building. Nobody would ever consider it reasonable to have a generic, already-assembled water-pipe system that gets shipped to every building, which people would then have to work around in order to get their jobs done. Why is it this way with software?

The solution to this problem is unlikely to be simple. SAP and other enterprise management software companies realize that their behemoth software systems are not easy to use, and are not optimized for the various work environments found around the world. Thus, as they are spending lots of money trying to improve their systems, it is likely they would have already discovered a simple solution if one existed. However, given what works with other forms of infrastructure, a solution is likely to include at least two features. First, the system must be modularized into useful components that can be meaningfully and sensibly put together. And second, the locus of control should be given to the people who have expertise in system structure. Pipe manufacturers create the pipes, but specialized engineers figure out where they should go in a building, and plumbers install them.

One of the problems with many rapid prototyping techniques is that the locus of control over the prototype remains with the developers of the software: only the developers can modify the design of the prototype. Paper prototyping is a striking exception to this, because everyone can participate in the activity, even children (Rettig 1994; Druin 2005). However, one of the challenges of paper prototyping is that it is often difficult for users to envision what the final system's capabilities are, and it is impossible

for them to incorporate the prototypes into their daily activities in order to figure out how they might actually use the features in the current iteration of the design. Methods of high-fidelity rapid prototyping such as patchwork prototyping can provide users with the opportunity to incorporate prototypes into their daily activities (Jones, Floyd & Twidale 2007), however the locus of control still remains with the developers. An ideal solution would be for the users themselves to be able to rapidly iterate high fidelity prototypes, and this is where the possible future of web mash-ups becomes relevant.

In the near future, there may be a large number of stable, mature web services which non-programmers can mash together without programming skills. If this is the case, then it will become feasible for users to prototype and rapidly iterate their prototypes in order to explore the design space and figure out how best to meet their every-day needs. Software designers will still have a role to play, with respect to guidance, suggestions, and the incorporation of software or services which remain too difficult for the end-users to incorporate themselves. Not even paper prototyping can be successful without the participation of the designers. However, the locus of control will be firmly in the hands of the end-users once more. In effect, we will be able to return to the days when secretaries and managers were able to develop business work-flows optimized to the needs of their particular organizations. But instead of designing paper

forms and sending them off to the printer, they will be mashing up prototypes and sending them off to software developers for implementation.

WORKS CITED

1. Druin, Allison (2005). What children can teach us: Developing digital libraries for children. *Library Quarterly*, 75(1), 20-41.
2. Floyd, Ingbert R.; Jones, M. Cameron; Rathi, Dinesh; Twidale, Michael B. (2007: In Press) Web Mash-ups and Patchwork Prototyping: User-driven technological innovation with Web 2.0 and Open Source Software. HICCS Proceedings.
3. Jones, M. Cameron; Floyd, Ingbert R.; Twidale, Michael B. (2007: In Press) Patchwork Prototyping with Open-Source Software. In *The Handbook of Research on Open Source Software*, St. Amant, K. and Still, B. (Eds), Idea Group, Inc., PA.
4. Rettig, M. (1994). Prototyping for tiny fingers. *Communications of the ACM*, 37(4), 21-27.