

# Visual Tools for Accessible Computer Supported Collaboration

Antti Raike, Joanna Saad-Sulonen, Jürgen Scheible,  
Roman Suzi, and Tarmo Toikkanen

University of Art and Design Helsinki TaiK,  
Hämeentie 135 C, 00560 Helsinki, Finland  
<http://www.taik.fi/>

**Abstract.** This paper discusses device-agnostic technologies and the use of visual mashups in augmenting accessibility in computer supported collaboration. The principles of Design for All (DfA) could be easily taken into software development if participants are allowed to contribute with whatever devices they have at their disposal or are able to use. However, device-agnostic services should rely on open standards, agile development and accessibility guidelines to allow participation of diverse user communities. We present three cases for further considerations of DfA in software development. We aim to promote the learning dimension of the visual tools; the ability of peers to learn and let colleagues learn, collaborate and innovate.

**Key words:** Accessibility, Agile Development, Co-Design, Collaboration, Design for All, Map mashup

## 1 Introduction: Designing Tools for Visualisation

The term '*mashup*' stands for a web application that combines data from more than one source into a single integrated tool. Map mashups offer a possibility to create visual tools, using various existing data brought together in a meaningful visualisation. Map mashups have become popular with the online maps and because some online systems and platforms offer open application programming interface (API), making it possible to combine location data and other information and visualise it on the map. A map mashup can for example use cartographic data from Google Maps, combined with other information, making it possible to add location information to a film scheduling and budgeting data, and thereby creating a new web service for a film crew that was not originally provided by either source. Other examples include future mobile services that combine the virtual and physical space, offering mobile and web users a seamless collaborative experience in real-time. More software examples to the needs of special-purpose designs are available at Programmableweb.com [1] and Web-mashup.com [2]: Both WWW sites contain information of a large diversity of web mashups. Mashups are also a good example of tools for rapid prototyping. The fact that relevant prototypes of a system being developed can quickly be mashed

up using existing applications, data sets and tools, makes them valid tools for collaborative design. Mashups are also moving to the sphere of the mainstream, with "do it yourself" (DIY) mashup sites offering users with no programming knowledge the possibility to quickly create their own map mashups [3]. The effective use of mashups nevertheless requires knowledge about open technologies for sharing information, the existing and developing practices related to the use of these technologies, the providers of these technologies and their goals, and finally, IPR issues related to the use of these technologies. Attention should also be paid to technological issues such as data-capture, encoding, data loss, white noise, visual clutter [4], and especially standards for accessibility [5].

However, the diversity of user groups provides even more challenges for the developers. Joanna Saad-Sulonen and Roman Suzi [6] point out that benefit of a collaborative design process is not always clear for the users. Hence we have become aware that it is essential to understand human issues such as perception and cognition, language and culture along the development of new devices and services. The multiple pathways used to derive meaning from culturally loaded visualisations have yet to be fully understood [7–9]. Thus agile development [10] methodologies challenge designers and programmers to understand users diverse thinking, collaborate more effectively with them and consult peers during iterative development process if needed. All in all the odds are against an individual designer or developer if he or she is unable to collaborate with users or consult peers during the design process or production. A solution could be based on device-agnostic services which rely on open standards and accessibility guidelines thus reinforcing the principles of Design for All (DfA) by allowing users to utilize whatever tools they have at their disposal.

## 2 Design Research Methodology and Agile Development

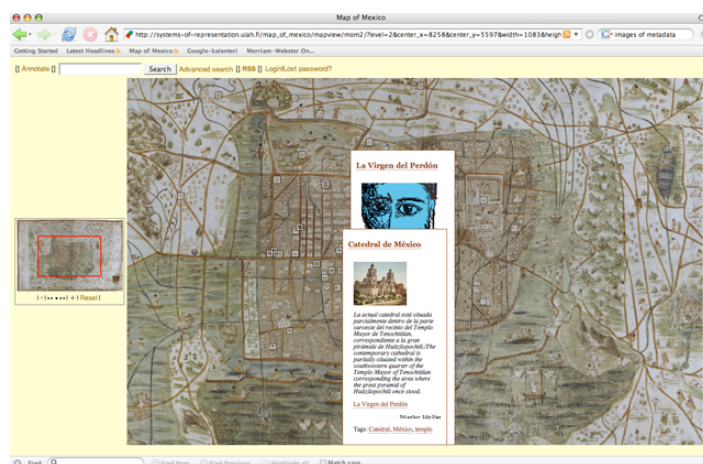
Collaborative design (Co-design) refers both to hardware and software design as well as collaborative design focusing on human and social factors in design. Design research, such as co-design and agile development, integrates the development of digital technologies with theory and practice focusing on the variation of users [11, 12]. Graham [13, 14] describes software design as a process where the aim is to find simple solutions to overlooked problems that need to be solved, and deliver them as informally as possible, starting with a crude first version, then iterating rapidly. Software co-design methodology bears many similarities with agile methodologies, as reported in the online document Manifesto for Agile Software Development [15]. Agile methodologies value customer collaboration and responding to change over following a plan, which are quite similar to the ideas of collaborative design and continuous design through use.

According to Saad-Sulonen and Suzi [6] a co-design work seems to help designers and programmers in development if a dialog happens between all the engaged stakeholders. Appropriate software prototypes can facilitate the process, because potential users can express their ideas with the common language of such prototypes. Saad-Sulonen and Suzi [6] observed that software designers

are specialists in building information models of any problem domains and that ability may lead to more streamlined designs. Likewise, HCI specialists possess practical knowledge on improving interfaces, and thus the feature requests by the user are not handled one by one but as representatives of underlying integral model. While certain qualities of agile methodologies are welcome in the design research, co-design can not be purely adaptive to the reality. The efforts described by Saad-Sulonen and Suzi [6] were combination of predictive and adaptive approaches. The design of the concepts defines the predictive backbone, while concrete implementation is adaptive and attentive to the stakeholder's own design efforts. However, the area needs more exploration and future research. The following chapters will present three examples of projects developed at the University of Art and Design Helsinki, where co-design and agile development have been used as a part of the design research.

## 2.1 Case 1: ImaNote

ImaNote (Image and Map Annotation Notebook) is a web-based multi-user tool that allows a group of peers to display a high-resolution image or a collection of images online and add annotations and links to them (fig. 1).



**Fig. 1.** ImaNote annotating a cultural artefact, an ancient map of Mexico City.

A user may mark an area on the image (that can be a map) and write an annotation related to that area. It is also possible to add links and multimedia to the annotations. Users can use RSS (Really Simple Syndication) to keep track of the annotations added to the image and also to use the annotation feed in mashups. They can also make links pointing directly to specific annotations on their own blog, web site and/or email. ImaNote is Open Source and Free Software and the released under the GNU General Public Licence (GPL). ImaNote is a

Zope product, written in Python. Zope and ImaNote run on almost all operating systems (GNU/Linux, MacOS X, \*BSD, Microsoft Windows, etc.). Imanote was developed as a collaboration between the Systems of Representation and Learning Environments research groups at the University of Art and Design Helsinki [16].

## 2.2 Case 2: Development of the Urban Mediator

Urban Mediator (UM) is a platform that provides a way for communities to mediate local, location-based discussions, activities, and information. UM uses a map-portrayal service as means for representing location-based information and complements it with a set of tools for users to process, share and organise this information. The software and related web-based services enable users (citizens and city administration) to obtain and share information about a city neighbourhood. Urban Mediator is available in a desktop and mobile version [17]. TheUM software and related services are being developed within the context of the EU-funded Innovative Cities for the Next Generation (ICING) project [18].

UM has been developed following an iterative co-design approach, which means that designers and programmers would not first gather all the requirements and then build software for the rest of the time (purely predictive method). The approach rather emphasised the involvement of a variety of stakeholders in the area of Arabianranta in Helsinki (active residents, local actors, workers in city administration, local pupils and teachers), throughout the development and design process. Co-design workshops and exercises involved paper and pen prototyping sessions as well as the use of lightweight software prototypes [6]. The choice of using light-weight prototypes meant that making quick changes in response to participants experience of the prototype and their comments was possible. The software development tools chosen to build UM were web.py (web framework written in Python) and MySQL database.

Saad-Sulonen and Suzi [6] followed guidelines of Floyd [3] and made constant attention not to overlook existing online components that could complement and enrich UM functionality, or provide it with spare parts. Some of the prototypes used through the development of UM included mashups. One prototype mashup example consisted of using web feeds containing geodata, from the photo sharing platform Flickr [19], as an example of data that can be harvested by Urban Mediator and that enriches the collections of location-based information gathered in a particular UM instance.

Another mashup prototype combined location data gathered using UM with Google maps map portrayal service [20], making it possible at early stages of development to convey the idea of the representation of data on online maps. Later, the possibility of using Google for representations of UM data came also handy in explaining how UM map widgets can be integrated in websites (fig. 2). Such prototypes, rapidly created, have proved very useful in discussions with stakeholders as they give a concrete tangible example of the possibilities of the software being developed.



**Fig. 2.** A Google map showing UM data is embedded along with UM web widgets on a website (in this case the UM development blog).

Using mashups can lead to flexible designs benefiting from rigid, simple building blocks. This enormously facilitates overall design efforts as building blocks become more comprehensible by stakeholders than traditional building blocks found in software design. These extra components also served as means to illustrate possible courses of development and showcase for encouraging more collaborative, user-designable, ecosystemic software [21].

### 2.3 Case 3: Mobile Python

Jürgen Scheible and Ville Tuulos [22] describe in 'Mobile Python' how rapid prototyping of applications on the Mobile Platform can be realized with Python (for S60). They explain a wide range of easy to program smartphone functionalities such as camera, sound, Bluetooth, GPS, messaging, Internet as well as 2D and 3D graphics. Using Python for S60 [23] and its open source code allows fast and easy development of many kinds of applications such as visual tools for collaboration in which mobile devices play a central role. An example was Manhattan Story Mashup [24] which combined personal mobile phones, the web and a large public display into an interactive, collaborative visual storytelling tool. Visual stories were jointly created by people writing stories on the Internet and by people taking photos in New York with camera phones to illustrate the stories in real-time (fig. 3).

Manhattan Story Mashup worked in such a way, that individual key-words of textual stories written by web users were presented, one word at a time, to mobile users for the purpose of taking a matching photo. The story writers could see their illustrated own story in their web browser where each sentence was represented with one photo and the graphically over-layered text. The best visual stories were displayed on eleven large public displays in Times Square.

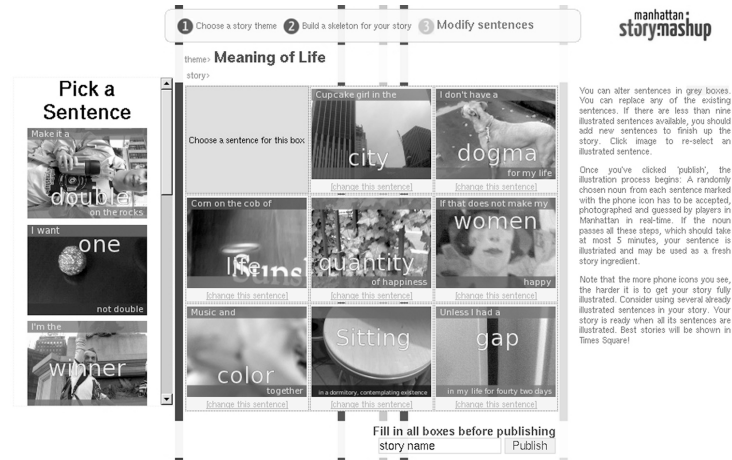


Fig. 3. Storytelling tool in the web.

Scheible and Tuulos [24] conclude that the system like Manhattan Story Mashup could well suit for educational purposes, even across city and country borders. The use of video or other multimedia pieces would increase the presentation power over still images and would open many new opportunities on how the Story Mashup system could be utilized. In addition, Tuulos, Scheible and Nyholm [25] suggest that developers have to pay more attention to matching the physical and the virtual timescales, to the natural friction between the web and the physical world. Hence further experiments are needed to find best practices and design frameworks for successful interaction between the two worlds.

### 3 Discussion: Potential of Visual Tools in DfA

How to optimise learning and knowledge sharing by visual tools? In principle, modern ICT could be used to solve varied challenges caused by the potential users. The three presented examples have led us to considerations of accessibility and DfA in software development. What kind of practice should be taken into consideration and embedded in the co-design and agile development? How to make the interactive maps accessible to the widest possible range of users? What kinds of representations are effective for collaborative situations? Experts like Tufte [27] and Zeki [28] highlight e.g. the importance of colours in envisioning information, but do the designers and programmers understand the basic elements for visualisations, like the cultural symbolism of colour [8]? Indeed it is rather easy both for people with normal vision and with vision impairment or dyslexia to adjust text sizes and colours of an interface. However, there are several types of cognitive disabilities more difficult to diagnose than e.g. hearing or vision impairment or colour blindness. These include e.g. dyslexia or autism, motor or dexterity disability such as paralysis, cerebral palsy, carpal tunnel syn-

drome, and all impairments due to ageing. Another challenge of effective DfA in ICT is to improve accessibility of people with cognitive disabilities; including those with reduced communication or reading skills.

Brown [26] makes a relevant question for developers: Would it be possible to see over the impairments differentiating people and focus on the common universal mind of humans and cultural universals? The general principles of vision hold also when it comes to complex visualisations of any kind. Hence we should also incorporate the lessons from the research on emotional and motivational systems, but the interpretation of the role and significance of the brain processes requires that the whole situational context of interaction is taken into account. Human brain is not a fixed system but rather has the ability to adjust its functions and structure according to the demands set and challenges given by the surrounding environment. Thus pervasive (ubiquitous) computing is a way to improve visual accessibility of collaboration. Small devices like mobile camera-phones offer an access to cheap, low-powered and constant networking with peers. Similarly, laptop computers and PDAs have become a common accessory almost for everyone. Such ubiquitous interaction devices extend communication and enhance novel ways to collaborate with visual tools.

## 4 Conclusion

We have seen it worthwhile to study cognitive, cartographic and social principles affecting mashups and other complicated layering used in computer displays and visualisations. Interactive visualisations would be beneficial to users of minorities who should have device-agnostic ICT augmenting accessible collaboration with peers. Finding common language is crucial to the success of co-design and agile development as it allows all engaged stakeholders to see technical and social possibilities in the solution domain. It is quite safe to claim, that almost any data could be organised and analysed to display large amounts of information in ways that are easy to understand and help reveal relationships and patterns. Visual tools help users in clarifying thoughts and assist seeing how ideas are connected and how information can be organised to construct knowledge. In addition, mobile tools can offer collaborative tools for people with various backgrounds, as was seen in the case of Manhattan Story Mashup: Strangers formed teams to achieve an ad hoc goal in a game. In further projects we will continue to seek the potential of different disciplines to understand better how knowledge building happens in and between diverse communities.

## References

1. ProgrammableWeb, <http://www.programmableweb.com/> (2006)
2. Webmashup.com, <http://www.webmashup.com/> (2007)
3. Floyd, I. Using Mashups for End-user rapid and responsive prototyping in collaborative environments. Presented at the 20th ACM Conference on Computer Supported Cooperative Work. Banff, Alberta, Canada. 04-08 November, 2006. Proceedings available at [http://mashworks.net/wiki/CSCW\\_workshop\\_papers](http://mashworks.net/wiki/CSCW_workshop_papers) (2006)

4. Rosenholtz, R., Li, Y., & Nakano, L. Measuring visual clutter. *Journal of Vision*, 7(2):17, pp. 1–22, <http://journalofvision.org/7/2/17/> (2007)
5. Web Accessibility Initiative (WAI), <http://www.w3.org/WAI/>
6. Saad-Sulonen, J. & Suzi, R. 2007: Designing Urban Mediator. Proceedings of THE GOOD, THE BAD AND THE UNEXPECTED Conference. Brussels: COST Action 298 – Participation in the Broadband Society (2007)
7. Poizner, H.; Klima, E. S. & Bellugi, U. *What the Hands Reveal About the Brain*. MIT, Massachusetts (1987)
8. Nisbett, R.E. *The Geography of Thought. How Asians and Westerners Think Differently – and Why*. Nicholas Brealey, London (2003)
9. Tufte, E.R. Ask E.T. Forum: Project Management Graphics (or Gantt Charts), <https://www.edwardtufte.com/tufte/index> (2002)
10. Cockburn, A. *Agile Software Development*. Addison-Wesley (2002)
11. Raike, A.; Laitinen, K. & Viikari, T. Elokuvantaju–Cinemasense, <http://elokuvantaju.uiah.fi/> (2001)
12. Raike, A. Searching knowledge: CinemaSense as a Case Study in Collaborative Production of a WWW Service in Two Universities. In: Miesenberger & al. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 568–574. Springer, Heidelberg (2006)
13. Graham, P. *Hackers & Painters. Big Ideas from the Computer Age*. OReilly, Cambridge (2004)
14. Graham, P. Six Principles in Making New Things. In [paulgraham.com](http://www.paulgraham.com/index.html), <http://www.paulgraham.com/index.html> (2008)
15. Beck, K. & al. Manifesto for Agile Software Development, <http://agilemanifesto.org/> (2001)
16. Pietarila, J.; Timonen, A.; Díaz, L.; Toikkanen, T. & Leinonen, T. ImaNote – Image and Map Annotation Notebook, <http://www.taik.fi/imanote/> (2005)
17. Urban Mediator information website, <http://um.uiah.fi/> (2007)
18. Innovative Cities for the Next Generation (ICING) project 2007, <http://www.fp6-project-icing.eu/> (2007)
19. Flickr, <http://www.flickr.com/>
20. Google Maps <http://maps.google.com>
21. Myller, M. On the System of Systems Approach to the Development of Everyday Life Applications. Master’s Thesis. University of Helsinki, Faculty of Science, Department of Computer Science and University of Art and Design Helsinki, Media Lab, ARKI Research Group. University of Helsinki, Helsinki, <http://ethesis.helsinki.fi/julkaisut/mat/tieto/pg/myller/> (2005)
22. Scheible, J. & Tuulos, V. *Mobile Python: Rapid Prototyping of Applications on the Mobile Platform*. Indianapolis: Wiley (2007)
23. Scheible, J. Mobilephone programming for Multimedia. Proceedings of the 15th international conference on Multimedia. ACM Press, New York (2007)
24. Scheible, J. & Tuulos, V. Manhattan Story Mashup – Interactive, Collaborative Street Art. In: *Space Time Play*. Birkhuser, Basel (2007)
25. Tuulos, V.; Scheible, J. & Nyholm, H. Combining Web, Mobile Phones and Public Displays in Large-Scale: Manhattan Story Mashup. Proceedings of the 5th International Conference on Pervasive Computing, Toronto, Canada, May (2007)
26. Brown, D.E. Human universals and their implications. In: N. Roughley (Ed.) *Being humans: Anthropological universality and particularity in transdisciplinary perspectives*. Walter de Gruyter, New York (2000)
27. Tufte, E. R. *Envisioning Information*. Graphics, Cheshire, Connecticut (1990)
28. Zeki, S. *Inner Vision: An Exploration of Art and the Brain*. Oxford University, Oxford (2000)