

Mobile Displays in Global Software Development: Opportunities and Limitations

Alexander Boden¹

Frank Rosswog²

Gunnar Stevens²

Volker Wulf^{1,2}

¹Fraunhofer Institute FIT
Schloss Birlinghoven, D-53754 St. Augustin
alexander.boden@fit.fraunhofer.de
volker.wulf@fit.fraunhofer.de

²Information Systems, University of Siegen
Hölderlinstr. 3, D-57076 Siegen
frank.rosswog@uni-siegen.de
gunnar.stevens@uni-siegen.de

ABSTRACT

This paper discusses opportunities and challenges of using mobile displays for supporting coordination and learning in distributed software teams. We argue that the growing availability of mobile devices such as tablets and smartphones provides interesting design opportunities for addressing challenges of distributed collaboration such as lack of awareness and limited knowledge exchange between cooperating teams. Based on empirical studies on Global Software Development (GSD) and a previously published design study, we present key opportunities and challenges for future work in this area.

Author Keywords

Global Software Development, mobile devices, public displays, group awareness.

ACM Classification Keywords

H.5.3 [Group and Organization Interfaces]: Computer Supported Cooperative Work.

INTRODUCTION

Distributed work such as in the context of Global Software Development (GSD) is characterized by a number of challenges that have to be overcome for successful cooperation. Global software teams are known to communicate less and suffer from limited inter-team awareness [11], hindered knowledge exchange [6], and different kinds of social and cultural misunderstandings [3], just to name a few. CSCW research has been interested in obtaining a detailed understanding of the implications of geographical, temporal, organizational and temporal distribution on team collaboration in order to design software tools that help to overcome the limitations of distributed cooperation. In our own research in this domain, we have found that one important issue distributed teams have to face is the need to mediate between the various formal and informal tools used to coordinate distributed software development work [5]. This mediation can be understood as a form of invisible meta-coordination (or *articulation work*) that is needed to make the distributed cooperation work.

In this position paper, we want to share our experiences with using lightweight mobile displays for supporting the kind of meta-coordination needed for distributed cooperation. Our argument is that the growing availability

of mobile devices such as tablets and smartphones creates interesting new design possibilities as the lightweight form factor and usability of the devices are well suited for use cases that need a certain informality, flexibility and peripheral forms of usage. At the same time, we see that the properties of the devices create certain limitations that can affect matters of privacy, change the complex interrelations between the “formal” and the “informal”, and pose certain limitations for the applications that can be deployed on such devices.

BACKGROUND

Tools for supporting coordination of software development work can roughly be divided into two different areas [1, 4]: the first one includes tools for information management such as bug-trackers or version control systems. They provide the basic infrastructure for working on source-code and managing tasks for software developers. The second one includes tools for personal communication such as Instant Messengers, micro-blogs or email. In our studies, we found that, while formal systems played a very important role for the coordination of the development work, their use was highly embedded in informal discussions which took place before, during and after the tasks in the formal systems were accomplished. Our findings resonate with related work from CSCW and Software Engineering with regard to supporting continuous coordination [14] by combining formal and informal forms of coordination, which in the case of distributed software development can be scattered across the tool infrastructure used.

While the used systems are usually accessed from the desktop, mobile touchscreen-based devices such as smartphones and tablets are increasingly used in workplace-related contexts [9]. They provide additional possibilities for accessing emails, manage tasks in the calendar, or even taking notes in cloud applications during meetings and in all kinds of other situations that were previously dominated by laptops or hand-written notes. The growing availability of such devices, in the form of private smartphones and tablets brought to the workplace or company-owned devices, creates new ways of interaction and cooperation between co-located and distributed co-workers.

In order to explore further possibilities of using mobile

devices for supporting cooperative work, we have developed a prototype based on a lightweight mobile device. The system is based on a server that aggregates relevant information from coordination tools such as bug-trackers, version control systems, or development blogs. These information are presented as sticky notes on a iPad serving as a mobile display (see Figure 1). The design follows a bulletin board metaphor and allows for different types of interaction. Users can drag messages around, rearrange them and organize them on the screen, similar to projects like the Visual Knowledge Builder [15]. Users can also post new messages, mark certain notes as important (which doubles them in size), or delete notes from the screen. Posted notes become more and more transparent over time and are automatically deleted after a while. New messages always appear on top of the pile and are color-coded depending on their source (e.g. new bugs from a bug-tracking system appear as red). While the display itself aims at supporting a local team, multiple displays can be connected to the server in order to support distributed work groups. To support inter-team interaction, the displays can also be synchronized with each other.



Figure 1. An iPad as a mobile display.

Our design idea has been inspired by approaches that use public displays to support cooperative work. For example, approaches such as CommunityMirrors support peripheral awareness and serendipity [13] and provide a shared working context for cooperating actors [10]. In software development, a special approach of using public displays has been tried in the form of “dashboards” that represent project-related information [2, 16].

OPPORTUNITIES

Our prototype has been designed on the basis of empirical studies in global software teams in small enterprises [5] and evaluated in a design study that has already been published [7]. In the following part, we will discuss opportunities that we have encountered in working with mobile devices in our study.

Supporting Peripheral Awareness

Using mobile devices as “semi-public” displays has considerable advantages compared to desktop applications,

as they allow information to be presented in an unobtrusive way, hence providing for peripheral awareness and supporting serendipitous knowledge exchange [12]. At the same time, they can be used cooperatively in contrast to the tools installed on individual development machines, which offers interesting implications for designing novel supportive tools for planning work, annotating documents in meetings, and sharing informal information about what is going on in the company [cf. 17].

Mobility & Flexibility

Our studies have also revealed that the lightweight form factor has interesting implications for supporting the team work, as the display can easily be installed in various places such as offices or coffee kitchens, and as it is possible to take the device into meeting rooms or to a customer site. In that regard, we made good experiences in using Velcro fastener to be able to flexibly attach the device to walls in different places (see Figure 2). This flexibility challenges notions of ownership and allows actors to expand the information space into different contexts, raising questions about how the mobility and different locations of the device affect the work practices in contexts such as software development [8]. Taking the device to different rooms or to the customers’ site allows for a nuanced use and has interesting implications for the dimensions of “awarenesses” that you might support, with clear implications for the design of mobile display applications for GSD.

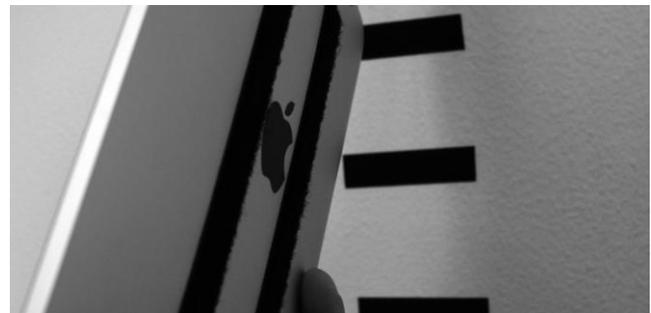


Figure 2. Velcro fastener as a flexible fixation solution for mobile displays.

Filling gaps in the tool landscape

Our study has shown that the design metaphor of a mobile display is well suited for making invisible parts of the work more prominent and bridge formal and informal types of information, as both kinds of messages can be presented in a shared context. Posting automated messages from different relevant sources together with user-generated comments on such a device creates new possibilities for contextualizing junks of information, independent from the different tools. The annotations can be used to provide context to the otherwise often decontextualized information in the coordinative tools. In doing so, mobile devices offers a level of integration and ease-of-use in both formal and informal channels and can serve as a central medium for filling gaps in the usually desktop-oriented tool landscape.

Hence, they can allow for meta-coordination mechanisms to emerge in the context of articulation work of distributed software development [7].

LIMITATIONS

The use of mobile devices as displays in GSD also comes with risks and has limitations which need to be kept in mind by designing such a solution.

Limitations of the form factor

We found that the visualization of the content was of essential importance for a successful usage of a tablets computer as a mobile display. Readability of content is a challenge given that public displays need to be seen from medium- as well as close-range. As smartphones and tablet GUI libraries are usually targeted for single users of the displays, a more cooperative use scenario calls for different design and presentation metaphors that can create a use experience that is different from what users expect from such devices. At the same time, there is the challenge of creating simple log-in mechanisms in order to allow for personalized settings and the posting of user messages.

Formalizing the informal

Using shared tablets as mobile or semi-public displays to create rooms for informal articulation work within and across teams also creates the risk of changing the perceived nature of the “informal” information that is posted there. It has to be ensured that informal aspects of articulation work are not formalized too much by the mobile system because such a change could be perceived as unnecessary bureaucratic burden by the practitioners (possibly even destroying the informal character of the coffee kitchen). At least, such effects need to be taken into account in the design of applications for the mobile devices.

Privacy issues

Exploiting the growing number of mobile devices also comes with the risk of invading the privacy of team members. Private tablets and smartphones, even when used for work, are often highly individualized tools which contain personal data. Including personal devices into work-related contexts can invade the privacy of workers, and create barriers for sharing the tool with co-workers. While buying a dedicated device for the team is an obvious solution for this problem, there can still be pressure to install the work-related tools on private devices, leading to ethical implications for ensuring privacy and limiting data access from within the deployed applications.

CONCLUSION

The experiences from our studies hint at opportunities and limitations of using mobile devices as lightweight semi-public displays for distributed software teams. While our first results are promising, further research is needed to better understand how mobile devices can fill gaps in the existing tool landscape, how they can avoid creating new bureaucratic burdens for the cooperation, and how the limitations that result from the form factor and design metaphors of mobile operation systems can be mitigated,

particularly in situations where several teams use several displays to cooperate in a remote manner. After all, introducing additional devices and applications into the already highly complex software ecosystems poses its own challenges and may introduce new problems by trying to solve the existing ones, as our own experiences also showed. In summary, we believe that investigating the potentials and implications of using mobile devices in the context of GSD in more detail is an interesting field for further CSCW studies and that a better understanding of the issues can help to develop new tools for coordination support that can be also useful in other domains of cooperative work.

REFERENCES

1. Aranda, G.N., Vizcaino, A., Cechich, A. and Piattini, M. (2006): ‘Technology selection to improve global collaboration’. In *Proceedings of the international Conference on Global Software Engineering*, IEEE Computer Society Press, pp. 223-232.
2. Biehl, J. T., Czerwinski, M., Smith, G. and Robertson, G.G. (2007): ‘FASTDash: A visual dashboard for fostering awareness in software teams’. In *Proceedings of the International Conference on Human Factors in Computing Systems (CHI '07)*, ACM Press, pp. 1313-1322.
3. Bjørn, P. and Ngwenyama, O. (2009): ‘Virtual Team Collaboration: Building Shared Meaning, Resolving Breakdowns and Creating Translucence’. *Information Systems Journal*, vol. 19, no. 3, pp. 227-253.
4. Boden, A., Nett, B. and Wulf, W. (2009): ‘Trust and social capital: Revisiting an offshoring failure story of a small german software company’. In *Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW '09)*, London, pp. 123-142.
5. Boden, A. (2011): ‘Coordination and learning in global software development. Articulation work in distributed cooperation of small companies’. Univ. Diss., Siegen. <http://dokumentix.ub.uni-siegen.de/opus/volltexte/2012/598/>.
6. Boden, A., Avram, G., Bannon, L. and Wulf, V. (2012): ‘Knowledge sharing practices and the impact of cultural factors: Lessons from two case studies of offshoring in SME’. *Software Maintenance and Evolution: Research, and Practice*, vol. 24, no. 2, pp. 139-152.
7. Boden, A., Rosswog, F., Stevens, G. and Wulf, V. (2014): ‘Articulation spaces: bridging the gap between formal and informal coordination.’ *Accepted for publication in Computer Supported Cooperative Work*.
8. Büscher, M. and Urry, J. (2009): ‘Mobile methods and the empirical’. *European Journal of Social Theory*, vol. 12, no. 1, pp. 99-116.
9. Chen, C. and Zhang K. (2012): ‘Continuous awareness: a visual mobile approach’. In *Proceedings of the*

- International Symposium on Visual Information Communication and Interaction*, ACM Press, pp. 69-76.
10. Dewan, P., Agarwal P., Shroff, G. and Hegde, R. (2009): 'Distributed side-by-side programming'. In *Workshop-Proceedings Cooperative and Human Aspects on Software Engineering (ICSE)*, IEEE Computer Society Press, pp. 48-55.
 11. Gutwin, C., Penner, R. and Schneider, K. (2004): 'Group awareness in distributed software development'. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work (CSCW '04)*. ACM, New York, NY, USA, 72-81.
 12. Koch, M. and Richter, A. (2007): 'Enterprise 2.0: Planung, Einführung und erfolgreicher Einsatz von Social Software in Unternehmen'. Oldenbourg Verlag, München, Germany.
 13. Ott, F. and Koch, M. (2010): 'CommunityMirrors – Large interactive screens as natural user interfaces for cooperation systems'. In *Workshop-Proceedings of the International Conference on Human Factors in Computing Systems*, ACM Press, pp. 1-5.
 14. Redmiles, D., van der Hoek, A., Al-Ani, B., Quirk, S., Sarma, A., Silva Filho, R.S., de Souza, C. and Trainer, E. (2007): 'Continuous coordination: A new paradigm to support globally distributed software development projects'. *Wirtschaftsinformatik*, vol. 49, no. 3, pp. 28-38.
 15. Shipman, F.M. III, Hsieh, H., Maloor, P. and Moore, J.M. (2001): 'The visual knowledge builder: a second generation spatial hypertext'. In *Proceedings of the 12th ACM conference on Hypertext and Hypermedia (HYPERTEXT '01)*. ACM Press, pp. 113-122.
 16. Treude, C. and Storey M.A. (2010): 'Awareness 2.0: Staying aware of projects, developers and tasks using dashboards and feeds'. In *Proceedings of the International Conference on Software Engineering*. ACM Press, pp. 365-374.
 17. Wilson, S., Galliers, J. and Fone, J. (2006): 'Not all sharing is equal: the impact of a large display on small group collaborative work'. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '06)*, ACM Press, pp. 25-28.