Playing the Tricky Game of Toolkits Research

Michael Nebeling
University of Michigan School of Information
nebeling@umich.edu

ABSTRACT

In this paper, I reflect on my experience from the past several years conducting toolkit driven multi-device interaction research that appeared in CHI and EICS. I discuss lessons learned and share my perspective on the larger field of user interface engineering, including what I think the main challenges and opportunities are with toolkits research and good examples of it. I hope that sharing my perspective is useful for the new generation of researchers interested in, and potentially struggling with, doing engineering research in HCI.

Author Keywords

technical HCI; systems research; user interface toolkits.

BEFORE I GET STARTED

Toolkits and system-driven research is one of the most challenging, but perhaps also the most interesting, kinds of research we have in HCI. It is challenging for many reasons: *results wise*—because it takes a lot of time and effort to create a system that can be studied to answer the research questions behind it, *process wise*—because every system is different and there are too many technical, design and evaluation challenges that cannot all be addressed at once and therefore need to be well balanced, and *publication wise*—because the resulting artifact is likely to come close to other systems and it is neither an easy task for authors to articulate the differences nor for reviewers to judge whether these are significant.

I learned this the hard way as a PhD student interested in designing systems and tools that solve real-world problems. I started out with publication attempts in web engineering and HCI conferences, and was pushed between the two worlds as neither wanted to accept my work. For web engineering, I did too much on interfaces, and for HCI it was too much engineering. I also think part of the problem was lack of a clear research method. I did not think of system building as a research method at the time; in fact, I was warned about it and, depending on who I talk to, sometimes still struggle to explain that even though my research involves a lot of engineering, it is still research. When I was introduced to Alan Hevner's design science research, I thought that is what I was doing. However, I still think that even with specific engineering committees at CHI and whole conferences such as EICS, there is still a lot of confusion about the science part, and we still have a hard time acknowledging systems research.

My PhD thesis [17] essentially developed around a set of tools [21, 24, 25] that I designed to investigate new methods and techniques to create more flexible and adaptive interfaces. I was interested in this topic because of the ongoing proliferation of new computing devices, with many new touch devices coming out in all kinds of form factors since the iPhone started the trend in 2007. This was a risky PhD topic for all the three reasons stated above; in particular, there was already a long history of research into context-aware and adaptive interfaces. However, existing research struggled to meet the needs of practitioners and industry as the proposed solutions did not always seem useful and practical. This introduced me to two additional tools research challenges.

First, some of the tools that I created (e.g., jQMultiTouch [24] and W3Touch [25]) were much more simple and practical than a lot of the existing user interface research which was based on more generic notions of context awareness and complex model-based approaches. While my work seemed more closely aligned with practitioner needs, it also seemed less generalizable and, to some, probably even less "research-y."

Second, some of the techniques that I created ended up being similar to what is now called "responsive web design." While I would argue that my PhD thesis pioneered many of the concepts, or at least developed them in parallel, it is difficult to hold up this claim because of articles in popular science[1] that appeared before a Master's thesis [31] was published in [22].

After my PhD thesis, I started to work on cross-device interfaces, which seemed like a natural follow-on and nice extension of my prior work on context-adaptive interfaces because techniques had so far been limited to adapting interfaces to one device at a time. As part of this research, I created a family of *XD* tools that addressed all kinds of issues around the design, development, and testing of cross-device interfaces. For example, I created XDStudio [23], a new GUI builder for visually designing distributed user interfaces for multi-device environments such as meeting rooms or classrooms, investigating simulated and cross-device authoring strategies. After XDStudio, I created tools like XDKinect [26] to enable rapid prototyping of cross-device interfaces using Kinect as an intermediator, XDSession [20] to provide new tools for developing and testing cross-device interfaces based on useful

#HCI.Tools 2017: Strategies and Best Practices for Designing, Evaluating and Sharing Technical HCI Toolkits

[1] https://alistapart.com/article/responsive-web-design

abstractions in a multi-device data session concepts, and XDBrowser to enable end-users by making the concepts of distributed interfaces so far limited to toolkits directly available in web browsers. In a first paper [19], I used XDBrowser to study what kinds of cross-device interfaces end-users would want to have given an existing single-device interface. This study led to a first set of cross-device patterns. In a second paper [18], I used XDBrowser to study how single-device interfaces can be semi-automatically transformed into cross-device interfaces based on the patterns.

OPPORTUNITIES FOR TOOLKITS RESEARCH

While most of the paper talks about challenges, let us start by highlighting a few of the opportunities for toolkits research.

First, by doing research on novel kinds of user interfaces and toolkits to support the creation of them we as researchers have an important say in what the next generation of user interfaces might be. For example, a lot of the research on multi-touch was only made possible through new technologies such as the DiamondTouch table [7] and toolkits such as DiamondSpin [30]. They formed the basis of a wide range of studies on multi-touch interaction and collaborative tabletop interfaces for many years and still continue to play a role even today.

Second, toolkits are important to push two primary aspects of research: concepts and applications.

When examining a toolkit, I look for interesting new concepts that make existing techniques significantly easier and/or faster. In practice, jQuery and Bootstrap are two of the most disruptive JavaScript and HTML/CSS toolkits we have in the web development domain. The elegance, expressiveness, and power of both found such wide adoption among developers and also researchers (jQMultiTouch [24], Weave [2]) that some of the concepts made it to the HTML5 and CSS3 standards. In research, web automation and manipulation toolkits like Chickenfoot [1] and CoScripter [14] had similar impact due to their concepts being based on rendered web pages and sloppy keywords rather than proper references to interface elements in code. A lot of the research on end-user scripting and programming by demonstration was pushed by these toolkits with Highlight [27] being an example that builds on CoScripter to enable the desktop-to-mobile adaptation based on end-user demonstration of desired interactions.

The other major question I ask about a toolkit is what kinds of new applications it enables. For example, when looking at cross-device toolkits such as Panelrama [32], Weave [2], and WatchConnect [8], one thing to notice is the increased effort to support cross-device interfaces around smartwatches. In WatchConnect, this effort does not stop with toolkit support in software. Rather, it also provides hardware support for developers to create new kinds of smartwatch sensors.

CHALLENGES FOR TOOLKITS RESEARCH

Now let us turn to some of the challenges for toolkits research. These range from practical, to technical, to methodical challenges. Another challenge is the writing of a toolkits paper itself. Despite some good pointers and recommendations from senior researchers in the field [28, 9], my own experience both as an author and as a reviewer for CHI and EICS for many years shows that there is still little agreement among researchers on what makes good systems research.

Staying Ahead of the Game

Toolkits research should always attempt to stay ahead of the game. I have seen many "good" papers rejected because they either did not significantly push the concepts, the applications, or both parts. It seems harder to "sell" a toolkit that tackles an old problem, even though it might do it very well, than a toolkit that tackles a new problem, even though it might just be scratching the surface. So one way to alleviate shortcomings in toolkit design can be targeting cutting-edge interaction technologies. For example, I would say that the earlier generation of multi-touch toolkits did not innovate with concepts, but it enabled new applications. In the later generation (e.g., Proton [12]), this shifted towards new concepts that essentially enabled very similar applications, but did so in much more innovative ways. This was quite similar with multimodal and multi-device toolkits. After crowdsourcing, it is currently 3D printing and fabrication that receive a lot of interest in systems research. Note that many of these technologies were not novel at the time; rather, we speak of the multitouch and 3D printing *re*volution. Interestingly, although IoT definitely received a big push in industry, in toolkits research this was not so much the case. The researchers that I know worked on IoT toolkits (e.g., fabryq [16], Bluewave [6]) were given a hard time making the unique challenges clear given that a lot of the problems seemed to have already been addressed by prior multi-device research. Given the proliferation of new VR/AR consumer devices, it will be interesting to see whether there will be another wave of VR/AR toolkits, perhaps focused on wearable devices, after the success of projection-based toolkits such as RoomAlive [10]. In any case, support for blending the physical and the digital design worlds will become more important in the future. Again, WatchConnect [8] is a good example here as it supports both software and hardware interface prototyping in one toolkit.

Balancing Toolkit Practicality and Generalizability

This goes back to what I said earlier about practical vs. generic solutions. The literature on model-driven user interface research is full of comprehensive approaches based on complex models and multi-level abstractions. For example, MARIA [29] is a versatile and powerful model-based framework that was created based on many years of research. Yet, the process required to define interfaces and the kinds of interfaces that can be generated in the end often seem neither practical nor complex. My stance on this is that less is more. It is okay if a particular proposal does not provide full-fledged support as long as the design rationale is sound and limitations are clearly articulated. I find an elegant solution for a well-scoped interface problem is more likely to generate concrete results and hence gain traction as long as it improves, rather than trying to replace, existing workflows. As an inspiring example, I would like to mention the case of Adobe Lightroom here [11], where studies with professional and serious amateur photographers provided unique insights into their existing patchwork processes and how to best provide a solution that integrates well with Adobe Photoshop.

Designing for the Next Generation of Designers

Another common pitfall with toolkits research is not clearly identifying the users. This was not so much a problem some years ago when the distinction between users and developers was clearer, but given that users nowadays often are both consumers and producers thanks to enabling tools, the line becomes fuzzier. In the research on end-user programming, the term "end-user" was commonly used to refer to non-technical users as opposed to developers with programming skill. It can help to put the research into the appropriate context by citing relevant research in that domain (e.g., from [4]), but it is better to make it explicit by clearly stating the assumed skill of target users and ideally include studies that help identify the needs of those users. This is something that I think was quite well done in Snap-to-It [5]. It goes without saying that the expectation will be that it is also those kinds of users that will be recruited for testing a toolkit as part of the evaluation. For many of my cross-device systems, I had to explain and justify why I studied with participants that only had experience creating mobile and responsive interfaces rather than "real" cross-device developers. It took some effort to convince reviewers that this generation of developers does not exist yet, as the solutions so far are often still research prototypes and it will take some time before they mature and are picked up.

Dealing with the Proliferation of New Toolkits

In some of the recently booming areas such as multi-touch or cross-device interfaces, a large number of toolkits were created and documented in the literature. In particular, in the cross-device domain, many of them almost seem to have been developed in parallel, without actually citing or building on top of each other. I remember presenting in the CHI 2014 session on multi-device interfaces and all of us were surprised to see that we worked on toolkits pushing similar ideas and developing many of the same features. I was surprised to see the sheer number of cross-device toolkits that came out in 2014 and 2015. I would say that in some areas the "market" is saturated and any new attempt to publish a toolkit may just be turned down as "yet another toolkit." This does not mean that there is no more need for new systems research, but it will become increasingly difficult for a new toolkit to be significantly different in the concepts that it proposes or applications that it enables. Exceptions include the Weave and WatchConnect toolkits mentioned earlier, where jQuery-like device selection techniques, storyboard generation from cross-device code [3], and support for hardware prototyping of smartwatch interfaces added significant research value.

Releasing Toolkits to Facilitate Toolkits Research

Last, I wanted to raise the issue that most toolkits research I know ended with the release of the toolkit for download. This is necessary but not sufficient. It is necessary to enable others to try out toolkits and do comparative evaluations, which is often asked by reviewers despite the fact that many previous systems are not actually available and only "exist" in research papers. It is not sufficient, however, because in most cases this is where the toolkits research actually begins. To truly understand the capabilities and value proposition of a toolkit, it is important to study how it is used by others than the toolkit authors and a few study participants. The true value of Chickenfoot and CoScripter was revealed when others started to adopt the ideas and build on top of those tools. Unfortunately, in the multi-device domain, there are no such leading examples. Perhaps tools like Webstrates [13] and XDBrowser [19] could grow into that role as they recently enabled some workshop activities at CHI and EICS. But we need more hands-on and discussion-heavy workshops like XDUI, Cross-Surface and now #HCI.Tools to foster discussion around those issues.

WORKSHOP CONTRIBUTION AND ACTIVITIES

In summary, I am excited about the #HCI.Tools workshop to be held at CHI 2017. I would like to contribute my experience and knowledge in the form of discussion or presentation of selected research prototypes from my *XD* tools research.

Moreover, I would like to propose three types of activities that I think would benefit #HCI.Tools workshop participants.

First, I have previously organized mock program committees reviewing systems papers and run reading groups analyzing examples of "good" toolkits papers. This would require some prep work of participants, but could be limited to one paper, e.g., WatchConnect [8] and a framework such as Olsen's [28].

Second, I think it will be interesting to look at how toolkits research has evolved over time both in terms of design and evaluation. An activity that extracts best practices and guidelines from a larger corpus of papers could be based on group work reviewing selected genres of toolkit papers, e.g., on multi-touch or cross-device interfaces, or time, e.g., before and after Olsen's framework appeared at UIST 2007.

Third, let us discuss new trends in toolkits research such as blending the digital and the physical, and again the impact on both design and evaluation. Interesting examples can be found in the research on cross-device and proxemic interaction, including WatchConnect and Proximity Toolkit [15].

ABOUT THE AUTHOR

Michael Nebeling is an Assistant Professor at the University of Michigan School of Information. He investigates new methods, tools and technologies that enable users to interact with information in more natural and powerful ways, and also make it easier for designers to create more usable and effective user interfaces. As part of his research, he has created many systems to support the design and evaluation of rich, context-aware and adaptive, cross-device, multi-touch and multi-modal gesture and speech interfaces. He is committed to promoting engineering research within the HCI community. He has been an Associate Chair for the CHI Technology, Systems and Engineering subcommittee for CHI 2014-2016. He was EICS 2015 Papers co-chair and EICS 2014 Late-Breaking Results co-chair. He has been a member of the steering committee and Senior PC for EICS since 2016.

REFERENCES

1. Bolin, M., Webber, M., Rha, P., Wilson, T., and Miller, R. C. Automation and Customization of Rendered Web Pages. In *Proc. UIST* (2005).

2. Chi, P. P., and Li, Y. Weave: Scripting Cross-Device Wearable Interaction. In *Proc. CHI* (2015).

3. Chi, P. P., Li, Y., and Hartmann, B. Enhancing cross-device interaction scripting with interactive illustrations. In *Proc. CHI* (2016).

4. Cypher, A., Dontcheva, M., Lau, T., and Nichols, J. *No Code Required: Giving Users Tools to Transform the Web*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2010.

5. de Freitas, A. A., Nebeling, M., Chen, X. A., Yang, J., Ranithangam, A. S. K. K., and Dey, A. K. Snap-to-it: A user-inspired platform for opportunistic device interactions. In *Proc. CHI* (2016).

6. de Freitas, A. A., Nebeling, M., Ranithangam, A. S. K. K., Yang, J., and Dey, A. K. Bluewave: Enabling Opportunistic Context Sharing via Bluetooth Device Names. In *Proc. EICS* (2016).

7. Dietz, P. H., and Leigh, D. Diamondtouch: a multi-user touch technology. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology, UIST 2001, Disney's BoardWalk Inn Resort, Walt Disney World, Orlando, Florida, USA, November 11-14, 2001* (2001).

8. Houben, S., and Marquardt, N. Watchconnect: A toolkit for prototyping smartwatch-centric cross-device applications. In *Proc. CHI* (2015).

9. Hudson, S. E., and Mankoff, J. *Concepts, Values, and Methods for Technical Human–Computer Interaction Research*. Springer New York, New York, NY, 2014, 69–93.

10. Jones, B. R., Sodhi, R., Murdock, M., Mehra, R., Benko, H., Wilson, A., Ofek, E., MacIntyre, B., Raghuvanshi, N., and Shapira, L. Roomalive: magical experiences enabled by scalable, adaptive projector-camera units. In *Proc. UIST* (2014).

11. Kim, G. K. Early Research Strategies in Context: Adobe Photoshop Lightroom. In *Proc. CHI EA* (2007).

12. Kin, K., Hartmann, B., DeRose, T., and Agrawala, M. Proton: multitouch gestures as regular expressions. In *Proc. CHI* (2012).

13. Klokmose, C. N., Eagan, J. R., Baader, S., Mackay, W. E., and Beaudouin-Lafon, M. *Webstrates*: Shareable dynamic media. In *Proc. UIST* (2015).

14. Leshed, G., Haber, E. M., Matthews, T., and Lau, T. A. CoScripter: Automating & Sharing How-To Knowledge in the Enterprise. In *Proc. CHI* (2008).

15. Marquardt, N., Diaz-Marino, R., Boring, S., and Greenberg, S. The Proximity Toolkit: Prototyping Proxemic Interactions in Ubiquitous Computing Ecologies. In *Proc. UIST* (2011).

16. McGrath, W., Etemadi, M., Roy, S., and Hartmann, B. fabryq: using phones as gateways to prototype internet of things applications using web scripting. In *Proc. EICS* (2015).

17. Nebeling, M. *Lightweight Informed Adaptation: Methods and Tools for Responsive Design and Development of Very Flexible, Highly Adaptive Web Interfaces*. PhD thesis, ETH Zurich, 2012.

18. Nebeling, M. XDBrowser 2.0: Semi-Automatic Generation of Cross-Device Interfaces. In *Proc. CHI* (2017).

19. Nebeling, M., and Dey, A. K. XDBrowser: User-Defined Cross-Device Web Page Designs. In *Proc. CHI* (2016).

20. Nebeling, M., Husmann, M., Zimmerli, C., Valente, G., and Norrie, M. C. XDSession: Integrated Development and Testing of Cross-Device Applications. In *Proc. EICS* (2015).

21. Nebeling, M., Matulic, F., and Norrie, M. C. Metrics for the Evaluation of News Site Content Layout in Large-Screen Contexts. In *Proc. CHI* (2011).

22. Nebeling, M., Matulic, F., Streit, L., and Norrie, M. C. Adaptive Layout Template for Effective Web Content Presentation in Large-Screen Contexts. In *Proc. DocEng* (2011).

23. Nebeling, M., Mintsi, T., Husmann, M., and Norrie, M. C. Interactive Development of Cross-Device User Interfaces. In *Proc. CHI* (2014).

24. Nebeling, M., and Norrie, M. C. jQMultiTouch: Lightweight Toolkit and Development Framework for Multi-touch/Multi-device Web Interfaces. In *Proc. EICS* (2012).

25. Nebeling, M., Speicher, M., and Norrie, M. C. W3Touch: Metrics-based Web Page Adaptation for Touch. In *Proc. CHI* (2013).

26. Nebeling, M., Teunissen, E., and Norrie, M. H. M. C. XDKinect: Development Framework for Cross-Device Interaction using Kinect. In *Proc. EICS* (2014).

27. Nichols, J., Hua, Z., and Barton, J. Highlight: A System for Creating and Deploying Mobile Web Applications. In *Proc. UIST* (2008).

28. Olsen Jr., D. R. Evaluating User Interface Systems Research. In *Proc. UIST* (2007).

29. Paternò, F., Santoro, C., and Spano, L. MARIA: A Universal, Declarative, Multiple Abstraction-Level Language for Service-Oriented Applications in Ubiquitous Environments. *TOCHI 16*, 4 (2009).

30. Shen, C., Vernier, F., Forlines, C., and Ringel, M. Diamondspin: An extensible toolkit for around-the-table interaction. In *Proc. CHI* (2004).

31. Streit, L. Investigating Web Site Adaptation to Large Screens. Master's thesis, ETH Zurich, DOI: 10.3929/ethz-a-006250434, 2010.

32. Yang, J., and Wigdor, D. Panelrama: Enabling Easy Specification of Cross-Device Web Applications. In *Proc. CHI* (2014).