# “Bring-your-own-app?!” – Why apps hinder us from achieving true cross-device BYOD interaction

**Hans-Christian Jetter**
University of Applied Sciences
Upper Austria
Hagenberg, Austria
hans-christian.jetter@fh-hagenberg.at

Workshop Position Paper submitted to

Second International Workshop on Interacting with Multi-Device ecologies "in the wild" at CHI 2016

## Abstract

I believe that our community is widely ignoring a fundamental challenge that stands between our ambitious visions for cross-device interaction and what we actually achieve when deploying our prototypes in the real world. The problem is that we still “think” in apps and design BYOD prototypes as apps for a few selected tasks and for clearly defined combinations and configurations of devices. We therefore support only a tiny fraction of the wealth of possible BYOD usages, device combinations, and collaboration styles. To build and observe BYOD technologies that fundamentally change how we interact with computing systems, we have to move beyond the concept of single BYOD apps and find ways to make our prototypes more *adaptable* and *interoperable* so that they support unanticipated and fundamentally new usage patterns in the wild.

## Author Keywords

Cross-Device Interaction; Bring-your-own-device; Adaptivity; Adaptability; Interoperability; Instrumental Interaction; Object-Oriented User Interfaces; Webstrates.

## Introduction

Most likely all participants of this workshop will agree that cross-device interaction promises a fascinating new way of using our increasingly diverse device ecosystems for solving real-world problems. Many of us share the vision of a world in which users can rapidly shape "symphonies" [4] or "communities" [6] of devices that feel like one seamless natural UI for cross-device applications. We hope to achieve this not only for single users but also for multiple users. As a result, our community has started to explore bring-your-own-device (BYOD) scenarios in which users join their personal devices to create a shared community of devices for collaboration (e.g. [14]).

Overall, HCI research has made great progress in this field. I will illustrate this by shamelessly using two examples from my own work: My work on ZOIL at the University of Konstanz [7] explored how the mobile and stationary devices inside a physical interactive space (e.g. tabletops, data walls, tablets, PCs) could be combined for multi-user sensemaking in a shared visual workspace. In subsequent work at UCL, we worked together to create a more lightweight and portable cross-device technology. The result was HuddleLamp [14] that enables users to combine off-the-shelf mobile devices for spontaneous collaboration simply by putting them under a desk lamp.

## Lessons from ZOIL and HuddleLamp

Naturally both results are not perfect. For example, applications built with ZOIL are not easy to deploy in the wild because they can be installed only on Windows devices and need network connections that are often blocked by firewalls in real-world settings. As a result, we decided that HuddleLamp applications should become HTML5 browser applications to overcome the problem of incompatible devices or operating systems and eliminating the need for local installation or configuration of apps. Moreover, all communication should happen using web sockets, so that firewalls become much less problematic. This strategy proved to be very successful.

ZOIL also does not support BYOD scenarios: There is no possibility to easily detect the presence of a new device and automatically connect it without manually configuring network addresses and ports. In HuddleLamp this is much easier by simply opening a web page (e.g. by scanning a QR code) that will briefly flash a marker on the screen to identify the device and establish a connection without any manual setup.

A further problem is that ZOIL does not track spatial positions of devices, so that interactions or object transfers between devices require choosing device IDs from lists or placing objects in shared locations in a visual workspace. This feels much less fluid and more difficult than with other cross-device systems that extensively make use of inter-device spatial relations or *proxemics* [3]. I repeatedly discussed this important role of physical space and gestures for cross-device interaction in my work. Two examples are a workshop paper [9] at CHI 2014 and a resulting full paper at CHI 2015 [15]. As a consequence, already the very first ideas for HuddleLamp were centered around spatially-aware interactions [6,8]. However, all these interactions can only be detected inside the field of view of HuddleLamp's camera system. This is why Jin et al.'s recent work for sensing device locations without external hardware or device modifications is a very important step forward [10].

## The really hard problems with BYOD apps

I believe that all above lessons and challenges have something in common: They will be solved within a few years. It is fair to assume that we will soon be able to detect, identify, and connect multiple devices of different types and sizes and to track their positions and the gestures between them. We also already know a lot about how to design such gestures, so what is the real challenge for BYOD in the wild?

In my opinion, there is a widely underrated challenge that stands between our ambitious visions for cross-device interaction and what we currently achieve when deploying our prototypes in the real-world. I believe that this challenge is only seldomly addressed in research yet, because it sits between the traditional research topics of HCI (e.g., gesture design, user studies, new sensors & algorithms) and software engineering (e.g., software architectures, distributed systems, standards for interoperability). Furthermore, it focuses on a concept that we are so familiar with that we find it "natural" and hardly recognize it as a deliberate design choice that has been made for us decades ago and that we need to challenge: the concept of packaging and distributing computing functionality as *applications* or *apps*.

Why are apps problematic for cross-device interaction and BYOD? First of all, they exacerbate the problem of *adaptability*. Monolithic walled apps are inherently bad at adapting to sudden changes in context, e.g. in the number and kind of present devices. As I discuss in [4], such changes will happen permanently and it is impossible to enumerate the set of contextual states that may exist. Therefore the traditional idea of designing an app for a clearly defined number and combination of devices and trying to predict all possible states cannot keep up with the complexity of real-world users and usage. If "bring-your-own-device" (BYOD) should not mean "bring-your-own-device *(as long as your task is T, you are using app A, your device is a phone running operating system X and browser Y, has a screen size of S, and there are only between N and M other devices involved)*" we must find more flexible and adaptable ways of providing functionality such as *commands*, *objects*, or *instruments*.

Second, there is the problem of missing *interoperability* between apps: Even apps that serve a very similar purpose, (e.g. different apps for taking notes with a stylus, different apps for visualizing data in bar charts) cannot talk to each other in BYOD settings. In the best case, they share a file format, so that data can be exchanged between them via cloud services by manually storing and opening files on different devices. But this is far from the seamless real-time collaboration across devices that we intend to realize. We must find alternatives that enables user to flexibly connect or combine functionality across devices, even in ways that were not anticipated by the developers but successfully emerge from usage in the wild.

If we keep on thinking about BYOD in terms of a single app with companion devices that can only be used according to the plan of the app's designer, we only scratch the surface of what could be achieved. But introducing new ideas how multiple devices could provide, share, distribute, and combines functionality and content across different devices in unanticipated ways could truly revolutionize the way how we interact with computing.

## Alternatives to apps

The good news is that there are some alternatives out there. Researchers & practitioners have proposed a number of approaches for replacing the application-centric model with alternatives that are more flexible and arguably closer to the way we work and think in the real word, e.g. the *object-oriented user interfaces* (OOUIs) of the early 1990s [2], the *instrumental interaction* of the early 2000s [1], and its more recent incarnations as *VIGO* [11] or *Webstrates* [12]. Therefore I believe that we should use the current shift from the single-device to the multi-device era as an opportunity to critically reflect about the role that monolithic apps or applications should have in future and if alternatives such as *objects*, *instruments*, or *webstrates* would not meet the requirements of true BYOD and cross-device computing much more.

## References


1. Beaudouin-Lafon, M. 2000. Instrumental interaction: an interaction model for designing post-WIMP user interfaces. In Proc *CHI '00*. 446-453.
2. Collins, D. 1995. *Designing Object-Oriented User Interfaces*. Benjamin/Cummings.
3. Greenberg, S., Marquardt, N., Ballendat, T., Diaz-Marino, R. and Wang, M. 2011. Proxemic interactions: the new ubicomp? *interactions*. 18, 1 (Jan. 2011), 42-50.
4. Hamilton, P. and Wigdor, D.J. 2014. Conductor: enabling and understanding cross-device interaction. In *Proc. CHI '14*, 2773–2782.
5. Jetter, H.-C., Zöllner, M., Gerken, J. and Reiterer, H. 2012. Design and Implementation of Post-WIMP Distributed User Interfaces with ZOIL. *International Journal of Human-Computer Interaction*. 28, 11 (2012), 737–747.
6. Jetter, H.-C. and Reiterer, H. 2013. Self-Organizing User Interfaces: Envisioning the Future of Ubicomp UIs. *Workshop "Blended Interaction" (CHI '13)*.
7. Jetter, H.-C. 2013. *Design and Implementation of Post-WIMP Interactive Spaces with the ZOIL Paradigm.* PhD Thesis, University of Konstanz.
8. Jetter, H.-C. 2013. Visual and Functional Adaptation in Ad-hoc Communities of Devices. *Workshop on Visual Adaptation of Interfaces (In conjunction with ITS '13).*
9. Jetter, H.-C. 2014. A Cognitive Perspective on Gestures, Manipulations, and Space in Future Multi-Device Interaction. *Workshop "Gesture-based Interaction Design" (In conjunction with CHI '14).*
10. Jin, H., Holz, C. and Hornbæk, K. 2015. Tracko: Ad-hoc Mobile 3D Tracking Using Bluetooth Low Energy and Inaudible Signals for Cross-Device Interaction. In *Proc UIST '15*. 147-156.
11. Klokmose, C. N. and Beaudouin-Lafon, M. 2009. VIGO: instrumental interaction in multi-surface environments. In *Proc CHI '09*, 869-878.
12. Klokmose, C. N., Eagan, J. R., Baader, S., Mackay, W., Beaudouin-Lafon, M. 2015. *Webstrates*: Shareable Dynamic Media. In *Proc UIST '15*. 280-290.
13. Marquardt, N., Hinckley, K. and Greenberg, S. 2012. Cross-device interaction via micro-mobility and f-formations. In *Proc UIST '12.*
14. Rädle, R., Jetter, H.-C., Marquardt, N., Reiterer, H. and Rogers, Y. 2014. HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration. In *Proc. ITS '14*, 45–54.
15. Rädle, R., Jetter, H.-C., Schreiner, M., Lu, Z., Reiterer, H. and Rogers, Y. 2015. Spatially-aware or spatially-agnostic? Elicitation and Evaluation of User-Defined Cross-Device Interactions. In *Proc. CHI '15,* 3913-3922.