
Tools for Prototyping of Organic UIs

Reto Wettach

University of Applied Sciences
Potsdam
Pappellallee 8-9
14469 Potsdam, Germany
wettach@fh-potsdam.de

Abstract

Making technologies accessible for non-experts facilitates an explosion of innovation in areas related to that technology. We suggest two central requirements for those technologies, based on earlier introductions of technologies for non-experts: DTP and Processing. Then by analyzing the development process of our own Organic UI project we show that current tools are neither simplified enough nor sufficient for the creative exploration of Organic UIs by non-engineers. We then briefly introduce our own research into simplifying some of the technologies necessary to prototype Organic UI.

Last but not least we are starting a conversation about what other tools are necessary and how they need to be designed to facilitate further innovation and development in the area of Organic UIs.

Technology and Creativity

Historically there has always been an increase in diversity of creative exploration within a given technology once this technology became accessible to non-experts, for example, photography, desktop publishing or the internet. Before this step, these technologies were usable only to experts, after a long and painful learning process.

What are the requirements for designing the supporting technology, which finally initiates such an explosion of

innovation? And what can be learned from the outcome of those previous explosions of innovation for the development of future tools for non-experts?

A discussion of these questions is based on two tools for non-experts which had a big impact on creativity:

- DTP (desk top publishing) tools and production technologies.
- Processing [1], a development environment “to program images, animation, and interactions.”

These two examples have been chosen due to the following reasons:

The simplification of the design and production process through the introduction of DTP led to a wave of innovation which is not only visible in the increasing diversity and quality of design works [2], but also in the dramatic economic impact on the publishing industry: prior to the introduction of DTP, “innovation was infrequent, costly and producer-led.” [3]

Similarly, the impact of Processing has been huge and was therefore awarded with a Golden Nika in 2005 [4]. Today, designers are programming their own games, their own dynamic interfaces and even developing innovative usages beyond Processing’s initial scope: Jeff Han, for example, used Processing in the development of his first multi-touch explorations.

Requirements for Tools to Unleash Creativity [5]:

1. Low barrier of entry for the creative community
Prior to the introduction of DTP tools, the designer was limited in two fundamental ways in his or her design work:
The workflow of designing and printing a page laid-out

with text and images was complex and involved various specialized professionals as typesetters and printers. Furthermore the process itself was complex and abstract: “Before DTP graphic designers could not see the layout of an actual page of text with colour artwork included.” [6]

Similarly, the exploration of visual expression through code required, prior to the introduction of Processing, a huge learning effort for non-engineers. To set up an IDE, import the right libraries, have the right hardware setting etc. was not possible without a technical background.

At this point, it is interesting to have a look at one of the predecessors of Processing: “Design by Numbers” (DBN)[7, 8]. Processing improved on DBN in two major ways:

- A. Direct access to all elements possibly involved in this area of technology: DNB proved to be a too abstract representation of the technology itself, which ultimately wouldn't allow the user to dive deeper into the technology. As Processing is based on Java, it allows its experienced users to seamlessly move on.
- B. Open access to all required information: Maeda put all examples and references into a book, which cost \$34. By contrast, Processing implemented an open-source approach to code generated with Processing as well as to Processing itself.

The concept of sharing information and experiences as implemented in Processing leads to the second requirement:

2. Self-determined learning

When working with tools there is always a need to improve one’s skills with these tools. However, with

tools for non-experts that provide for self-determined learning, the creative user is enabled to determine the direction to extend his or her personal skills based on his or her own experiences.

In using DTP tools, the user can recognize his lack of abilities to create certain designs and is then also able to identify and learn new features or new plug-ins.

Similarly, the designer who works with Processing can improve his skills based on his own determination: he can check for interesting samples and explore their code [9] or he can investigate a large selection of libraries.

In both examples, the internet had a huge impact on self-determined learning, not only through information provided by the developers, but also through the exchange within the creative community.

Tools for Prototyping Organic UIs

By analyzing the development process of the “Dynamic Knobs” project [10] we can perform an initial exploration of the tools required for prototyping Organic UIs. “Dynamic Knobs” is a concept for Organic UIs on mobile phones, based on shape-shifting to indicate changes in the status of the device.

Two technologies were used in the development process:

- A. Microcontrollers including actuators and sensors
- B. Mechanisms and mechanical parts

Prototyping of electronics become more and more accessible to non-experts over last couple of years, due to the work of various arts and media and technology

programs in many universities [11] and due to the introduction of low-cost, stable and easy-to-use microcontrollers such as Arduino [12].

However, the basic concept of prototyping with a microcontroller and a breadboard, as most designers/artists do it, has certain limitations: the resulting prototypes are fragile, unreliable, and don't lend themselves easily to further miniaturization or replication. These limitations are being addressed by the author in a research project called Fritzing [13], an open-source Electronic Design Automation (EDA) to assist non-engineers in taking the step from physical prototyping to actual product.

The access to mechanisms and mechanical parts however is quite complicated. For our prototypes, we were using Lego, which is inflexible, quite large in size and difficult to integrate in non-Lego environments. Therefore we propose to look into making expertise from the mechanical engineering community accessible to the non-engineering community. Due to the wide distribution of rapid prototyping machines such an approach is absolutely possible.

Question for Conversation

What kind of prototyping tools are needed to ensure that the design community starts exploring the area of Organic UIs?

How do these tools need to be designed and integrated into the community to make sure that they are going to be used? As discussed previously, easy access to continuous learning is one important factor. As organic UIs are require knowledge in so many technical fields, I am wondering how the barrier to entry can be kept low and self-determined learning can be facilitated.

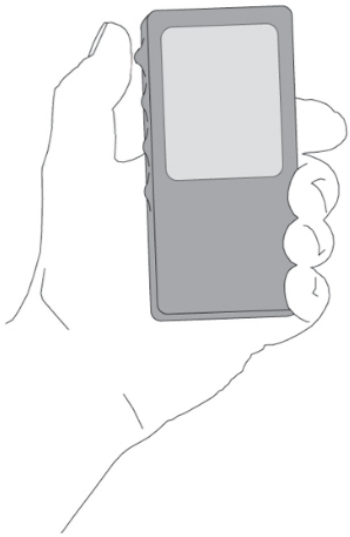


Fig. 1: Dynamic Knobs: Checking for missed calls through feeling the extensions on the phone's side

Acknowledgements

I thank Tom Igoe for his input at the "Friends of Fritzing"-Summit 2008 in Potsdam: his contribution heavily influenced the ideas in this extended abstract. Furthermore I thank Jonathan Cohen and André Knörig for discussing and reviewing my thoughts. Last but not least I am thanking the entire Fritzing-team for their great passion when developing this tool.

References

- [1] Fry, B., Reas, C., Processing, 2001, <http://processing.org/>.
- [2] King, E., New Faces: Type Design in the First Decade of Device Independent Digital (1987-1997), <http://www.typotheque.com>, 1999
- [3] Cox, H., Mowatt, S., Technological change and innovation in consumer magazine publishing - a UK-based study, eprints.worc.ac.uk/229/1/TASM_preprint.pdf, 2007
- [4] Golden Nica Net Vision, Prix Ars Electronica 2005
- [5] *There has been a lot of academic work about this topic. For the workshop I am reducing the discussion to two points. Further reading recommendation:* Knörig, A., "Design Tools Design", University of Applied Sciences Potsdam, 2008
- [6] MacNeill, T., Technology and Design: VANCOUVER MAGAZINE before and after Desktop Publishing, Simon Fraser University, 2005
- [7] Maeda, J., Design by Numbers, <http://dbn.media.mit.edu/>, 1999,
- [8] Flash (<http://www.adobe.com/products/flash/>) and Director (<http://www.adobe.com/products/director/>) are not mentioned here, because they had similar limitations as DBN
- [9] Processing [1] has an Feature called "Export", which exports a Java Applet and creates and embeds it into an HTML file - including the source code.
- [10] Hemmert, F., Joost, G., Knörig, A., Wettach, R., Dynamic knobs: shape change as a means of interaction on a mobile phone, Proc. CHI08, 2008, 2309-2314
- [11] Igoe, T., Physical Computing, <http://www.tigoe.net/pcomp/index.shtml>
- [12] Banzi, M., Cuartielles, D., Igoe, T., Mellis, Martino, G., Arduino, 2004, <http://arduino.cc/>
- [13] Wettach, R., et. al., Fritzing, 2007, <http://www.fritzing.org/>