Digital Bricolage: hands-on experiences with digital interaction construction

Andrea Alessandrini
University of Dundee
13 Perth Road
Dundee DD1 4HN
a.alessandrini@dundee.ac.uk

Abstract

The prototyping process of interactive systems is a key phase in the learning activities of interaction design students. Designing connections and communications for computational elements is a challenging part in constructing interactive system architectures. The goal of this study is to explore and describe the practices and technologies used in the construction of interactive system architectures in practice.

Author Keywords

Prototyping; Systems Architectures; Construction; Interactivity; Design

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction

This exploratory study examines interactive system architectures, designer tools, and practices with the objective to understand how hardware and software components are connected together by students. It describes and analyzes how connections are made and which components and technologies are used. The study was driven by three specific questions: (1) How do students make components communicate together? (2) Which types of connections do they use? (3) What issues do they encounter and how do they overcome

them?

Background

Physical computing focuses on combining physical and digital computation with methods that permit users with few skills to rapidly assemble hardware and software components to construct working devices. At the same time, this process helps them to understand and learn how the technology works [10]. In recent years, the physical computing approach has become a common method among interaction designers students for prototyping digital interactive systems. Recently, many prototyping platforms have become available that permit designers to rapidly “sketch” interactive systems in a way that was unimaginable a few years ago [3,4,8]. Readily available sets of compatible microcontrollers, sensors, actuators, and other components now allow designers to better support tinkering processes, thus stimulating their creativity. Rasberry Pi, .NET Gadgeteer, Arduino, and Phidgets are a few examples of these platforms [5,9,11].

This area of research, defined by Hartmann [6] as “opportunistic design” is stimulating for developing new tools for design; however, I believe that it is still under explored. In fact, only a few researchers have explored the practices and methods used in the development of software applications and interactive prototypes of devices and interactive systems in real contexts with the aim to create better design tools [2,7]. These researches concluded that I need to design better design tools to facilitate in those complex construction activities. Although all these studies have made great contributions, there is still little is known about how designers connect software and hardware elements to construct their interactive systems, in practice. The goal of this study is to explore how designers and researchers connect software and hardware technologies to build interactive prototypes.

Research Methodology

Fieldwork observations and interviews were used to gather data to answer the research questions. The fieldwork observation was conducted at the University of Dundee (United Kingdom). I analyzed each system to describe the challenges and issues that emerge during their development. In parallel, I conducted interviews with a total of 9 people. All interviewees Ire academics, interaction design and doctoral students. Their expertise in using the design tools was from an intermediate to advanced level. The same procedure and interview schedule was adopted for all interviews. In the interviews I asked what was the purpose of the systems was, whether there had been any problems in connecting the components, and how they solved them. I also asked interviewees to sketch a representation of their system on paper, highlighting the methods they had used to connect the different components. Transcriptions, interview notes, and interviewee sketches were later analyzed and coded. All the data from the analysis was mapped and clustered according to each discussion topic.

Examining digital bricolage

Here, I present the data I collected. This section first introduces our participants and their competences; then it reviews the systems they constructed. Next, it describes the technologies used and the reasons for their choice. Finally, it describes how the technologies are assembled together and the challenges encountered. For brevity, I only mention a subset of the analytical results and focus on commonalities within the

systems.

Interviewee A1 was a second-year PhD student in Design. He had extensive skills in hardware hacking and electrical engineering and fewer skills in web programming technologies. He was working on a system that provided visual information to the user about future asteroid Earth impact events (Fig. 1). He designed this system as part of his research theme on interactive paper design. He reported challenges required changing the voltage, the power, and the energy transmission systems (AC/DC) due to the use of electro-luminescence paper. He described his learning strategy adopted to deal with complex circuit design “especially when I use relay, which is two totally separate circuits, which keeps it simple in my head.” A6 described challenging scraping information from the internet data “to get those numbers, there are other step sites. It goes through three… It goes from this site to a PHP site, and then from there it goes into…two other steps. It goes to an XML file. It tidies it up and it makes it searchable. So it labeled date, distance, magnitude, and name, and then this file eliminated all the things we don’t want. 3, 4 are the numbers we are looking at today.”

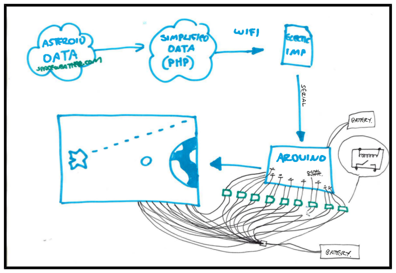

Figure 1. System architecture sketches.

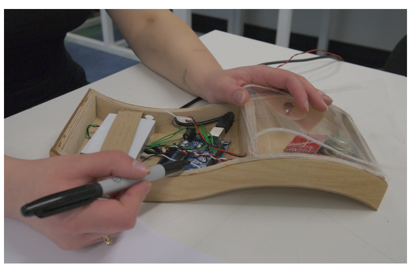

Figure 2. Prototype with multiple serial communications.

Interviewee A5 was a third-year undergraduate interaction design student. She had adequate skills in hardware hacking and programming. She and her colleagues designed a tangible interface system for children for interacting with audible online news. Her system comprises a microcontroller, an RFID reader, RFID tokens, an LED system for feedback, speakers, and a laptop. In her system, the communications between components were made through serial communication. She reported that learning to connect those components together was the most challenging part of the project.

She said that “When you upload the Arduino software, you need to take this [Rx RFID] wire out (Fig. 2). The RFID’s Rx connection blocks Arduino programming.” Furthermore, she learned strategies to simplify the communication between the RFID, microcontroller, and Processing. She said “The RFID reader is sent in BYTES. Arduino translates this in HEX values that I think has 10 characters. I’m always sending the last two [characters] to Processing”.

Interviewee A7 was a third-year undergraduate interaction design student. He was skilled in hardware hacking and software programming. He and his colleagues were designing a system for farmers that provides an auditory awareness about the real-time milk stock price (Fig. 3). His system comprises a microcontroller, a water pump, and a sensor for perceiving a user approaching the system. His project required learning web data-scraping techniques. As he described, “Processing processes the web page to find the exact number; five lines of code that just split into smaller and smaller text until we get the number we need.” A7 found challenging exploiting a specific library to facilitate the communication between the microcontroller and the Processing software running on the laptop, “You can listen to the Arduino trough Processing, but this was the first thing we tried. Unfortunately, you can’t have an Arduino library in Processing. This with the capacitive sensor that uses its own library.” A7 reported that he learns how to enable communication between components, “We have 0.05. Each time we multiply we have a whole number like 500, and we store this as an integer instead of a float, and float doesn’t go across the serial.”

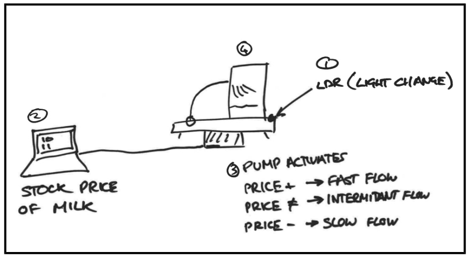

Figure 3: System architecture sketches.

Discussion

This research investigates the technologies, practices, and challenges of using design tools to support design students in the construction of interactive systems. The study try to be a first attempt to provide suggestions for the design of next design tools. Design tools require an approach different from engineering. In design it is fundamental to soften technology to make it quick, imaginative, and easy in order to shape the user's experience [1]. All participants involved in the study adopted a digital bricolage approach in which the activities of tinkering and patching guided them in constructing their systems. Digital bricolage is characterized by tinkering processes. Design tools must permit fluid control of the hardware–software balance. Designers must be free to choose easily between hardware or software to obtain a particular behavior of their system. This control permits a more fluid control to the hardware–software balance according to user knowledge, objectives, and context. Design tools must permit a progressive and dynamic transition between components and behavior arrangements. Design tools must permit the transition between a centralized solution to a distributed solution enabling duplication and separation of components and behaviors dynamically according to system requirements. This control permits a more fluid control from central to distributed balance and vice versa according to designer skills, objective, development phase, and contexts.

References

1. Alessandrini, A. End-user construction mechanisms for the Internet of Things. *Proceedings of the 27th British HCI Conference, (2013).*
2. Brandt, J., Guo, P.J., Lewenstein, J., Klemmer, S.R., and Dontcheva, M. Writing Code to Prototype, Ideate, and Discover. *Software, IEEE 26*, 5 (2009), 18–24.
3. Buechley, L. and Hill, B.M. LilyPad in the wild: how hardware's long tail is supporting new engineering and design communities. *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, ACM (2010), 199–207.
4. Buxton, B. *Sketching User Experiences: Getting the Design Right and the Right Design: Getting the Design Right and the Right Design*. Morgan Kaufmann, 2010.
5. Greenberg, S. and Fitchett, C. Phidgets: Incorporating physical devices into the interface. *Proc. UIST*, Citeseer (2001), 209–218.
6. Hartmann, B., Doorley, S., and Klemmer, S.R. Hacking, Mashing, Gluing: A Study of Opportunistic Design and Development. *Pervasive Computing 7*, 3 (2006), 46–54.
7. Hartmann, B., Doorley, S., and Klemmer, S.R. Hacking, mashing, gluing: Understanding opportunistic design. *Pervasive Computing, IEEE 7*, 3 (2008), 46–54.
8. Kuniavsky, M. *Smart Things: Ubiquitous Computing User Experience Design: Ubiquitous Computing User Experience Design*. Elsevier, 2010.
9. Mellis, D., Banzi, M., Cuartielles, D., and Igoe, T. Arduino: An open electronic prototyping platform. *Proc. CHI*, (2007).
10. O'Sullivan, D. and Igoe, T. *Physical computing: sensing and controlling the physical world with computers*. Cengage Learning, 2004.
11. Villar, N., Scott, J., Hodges, S., Hammil, K., and Miller, C. . NET gadgeteer: a platform for custom devices. In *Pervasive Computing*. Springer, 2012, 216–233.