End-user construction mechanisms for the Internet of Things

Andrea Alessandrini
DJCAD
University of Dundee
Dundee DD1 4HT
a.alessandrini@dundee.ac.uk

The prototyping process is a key phase in the design of the Internet of Things (IoT). Designing connections and communications for computational elements is a challenging part in constructing IoT prototypes. The goal of this study is to explore and describe the processes, practices, and technologies used in the construction of prototypes in practice. This article reviews interactive prototypes and interviews with practitioners. In particular I intend to describe and analyse how connections are made and which components and technologies are used. This is not intended to be an exhaustive analysis, but instead focuses on assessing the qualities of each system design in order to create an initial framework that explores an approach toward IoT construction. Finally, the implications of such a findings are discussed.

End-user. Internet of Things. Prototyping. Construction. Interactivity. Design.

1. INTRODUCTION

Internet of Things (IoT) is defined as the process of embedding computational elements into the objects and the environment. As IoT becomes more widespread and integrated into everyday life, it is important to understand the user experience with the services and devices available.

This exploratory study examines interactive prototypes and practitioner practices, with the objective to understand how heterogeneous hardware and software components are connected together by end-users in assemblies [14]. It describes and analyses how connections are made and which components and technologies are used. The study focuses on an important area in IoT: the need for simple design, construction, installation and configuration of digital artifacts from the end-user perspective. In this article, construction is defined as the process of building or assembling the IoT prototype infrastructure.

The study was driven by two specific questions: (1) how do practitioners make components communicate together? (2) which types of connections do they use?

2. INTRODUCTION AND RELATED WORKS

Several authors have described a ubiquitous computing vision, in which users are surrounded by a large number of computational and information systems. For example, according to Weiser [31], information processing will become thoroughly integrated and embedded into tiny computers distributed throughout everyday environments, interconnected and invisible. In *The Invisible Computer*, Norman [22] presented a detailed criticism of the personal computer and the vision of a future filled with information appliances. Aarts and Marzano [1] described their ambient intelligence vision, in which environments are sensitive and responsive to the presence of people. Gershenfeld [9] introduced a paradigm in which objects become active participants in information and social processes, interacting and communicating with each other and with the environment. These visions are appealing and offer enormous opportunities to users, but they might be also detrimental to users experience [16]. Already today technology can be compared to impenetrable black boxes, limiting users understanding of it, this issue will be even more critical as technology proceeds to populate more and more of our environment. A promising approach for facilitating, engaging and learning is the emergency of small-scale toolkits that offer means by which end-users constructing IoT systems are helped [7,29]. This vision is also in line with the instrumental genesis

approach: designing technologies that support human activity should include an emphasis on designing flexible, open artifacts that can be modified by end-users and adjusted for various tasks, including unanticipated tasks [2,27]. Other researchers have also called this idea: appropriation [8], reconfiguration [3], adaptation [18], serendipity [20], and hacking [25].

In cognitive science, there is an assumption that humans make sense of artifacts by interacting with them. In his embodied cognition vision, Clark [6] states that sensing, learning, thinking and feeling are based on the interactions our bodies have with the world around us. In addition, the distributed cognition perspective [12,13] asserts that the environment around us supports, organizes and informs human thought. These theories highlight that environment have a fundamental role on human cognition. More specifically on artifacts, Norman defines cognitive artifacts as those devices that maintain, display, or operate upon information in order to serve a representational function and that affect human cognitive performance [21]. Also Kirsh [15] describes how actions performed on objects can scaffold thinking and reasoning during problem solving, and highlights the role of external representation in supporting human cognitive processes. The role of artifacts properties on learning is well explored by Papert constructionism theory that focuses on the idea of learning scientific concept through the construction of computational assemblies [24]. Grounded on Papert constructionism theory, diverse didactical kits have been created to enable young children to explore and understand technology through constructing it [28]. These tools help children to obtain a deeper understanding of computation and scientific concepts.

Several researchers have described the challenges involved in constructing prototype interactive systems. For example, Myers [19] discussed the inherent difficulties in designing prototypes that contain complex interactions. Buchenau and Suri [5] described their experiences in designing a prototype interactive digital camera driven by a desktop PC. In addition, Pering [26] described the design of a prototyping application for a PDA using an assembly of custom hardware and a PC. Moreover, Hazlewood [11] described the challenges faced in designing a large-scale installation.

In recent years physical computing has become a common method among designers and researchers for prototyping interactive systems (i.e. IoT). It focuses on combining physical and digital computation with methods that permit users with

little skill to rapidly construct working assemblies, and at the same time earn understanding of how the technology works [23]. Raspberry Pi and Arduino are a few examples of these construction platforms [17,30].

However in recent years, the need has emerged for deepening our understanding on how people connect software and hardware elements to construct interactive assemblies. Only a very few studies are available that explain the process of prototyping, practice and the choice of hardware-software from the practitioners perspective. A work by Hartmann and colleagues [10] has explored the practices and hacking methods used in the development of interactive prototypes of devices and interactive systems. In the same prospective, Brandt and colleagues [4] studied mashup practices of professionals in the development of software applications.

In spite of the important contributions of researchers described earlier, today, little is known about how people connect software and hardware elements to construct their interactive assemblies in practice. To meet this challenge, the study examines five interactive prototypes projects and conducts five practitioner interviews with the aim of identifying more detailed insights about how digital connections are made and how interactive prototypes are constructed in practice.

3. METHOD

Two methodologies were used to answer the research questions. I conducted Strengths, Weaknesses, Opportunities and Threats (or SWOT) analysis of five prototypes conducted at the Interaction Design Laboratory at the University of Siena (Italy) from 2007 to 2011. The aims of these projects were to design and prototype Internet of Things (IoT) devices and interactive systems. I analysed project architectures and highlighted the strengths and weaknesses associated with their construction. I identified the different architecture types and the challenges involved in the construction of prototypes and devices.

In parallel, I conducted semi-structured interviews with five professional practitioners (2010-2011) with the specific aim of understanding how connections are made by interviewees in making interactive prototypes. Participants were chosen randomly from a designers and researcher practitioners network. The same face-to-face procedure and interview schedule was adopted for all interviews. Data from the analysis was mapped and clustered according to each discussion topic.

Figure 1: Prototype systems –clockwise from upper left– for (A) sensing aerobic and anaerobic activity; (B) information appliance; (C) encouraging young children to listen to music; (D) extending the role of artificial lighting in the user experience; and (E) facilitating socialization in public spaces among local citizens.

4. RESULTS

4.1 Project Cases

In this section, I present the SWOT analysis of interactive prototypes designed at the Interaction Design Laboratory at the University of Siena (Italy). (2009-2011) The aims of these projects were to design IoT and interactive systems.

Prototype 1 was the first prototype of a research project aimed at creating a system to allow users to obtain information about their aerobic and anaerobic activities (e.g. walking, running and climbing stairs). Two units were created in this system: one for the wrist and another for the ankle. The slave unit comprised an Arduino Nano 3.0, accelerometer, a gyroscope and a radio module, while the master unit was identical to the slave with plus a MicroSD module. The slave unit sensors sent data through a radio frequency module to the master unit, which logged the data coming from the slave unit sensors and data detected synchronously by the master unit's on-board sensors. In addition to the wireless receiver, the master unit also contained a memory card that was used to log the received data. All components were connected in place on small breadboards using jumper wires (Figure1A).

Prototype 2 was the result of a workshop organized for the prototyping of an IoT device designed to provide information on energy consumption. The prototype was shaped to resemble a plant. At the base of the vase was embedded a high power RGB LED strip. This was attached to a custom RGB

amplifier shield attached in turn to the PWM inputs of an Arduino board running custom software that controlled predefined luminous patterns. Designers addressed the problem of obtaining the perfect dosage of red, green and blue values to create white light by adding a potentiometer that could be controlled manually to obtain the correct values (Figure1B).

Prototype 3 was constructed within a research project aimed at creating a mobile platform for music listening. Researchers used a system composed of a computer running a custom Max/MSP (www.cycling74.com) patch connected, through Bluetooth, to a Wii Remote sensor (Figure1C). This patch permitted the Wiimote device to be easily connected to the computer via Bluetooth, while also making it possible to recognize and connect multiple Wiimote devices through their identification (ID) address (i.e. 00:1 F: 32:95: EF: B0).

Prototype 4 was the first one of a research project aimed at creating a platform to extend user experience of artificial lighting. The most challenging construction issue of this prototype was that hardware architecture was unable to detect changes in the custom operating system. This issue was caused by the initialization hardware configuration file being locked in the flash memory. Changing the file in the memory required access to a serial port used permanently by another component. Thus, it was necessary to switch the hardware patch configuration to access resources for changing and updating the functionality of the operating system. The ad-hoc hardware patch enabled temporary deactivation of the serial

communication between the main layer and the low layer in order to enable access to the configuration file for changing it (Figure1D).

Prototype 5 was a prototype constructed during an academic research project aimed at facilitating socialization in public spaces among local citizens. Researchers worked on a system to provide access to online audio content (e.g. radio and podcast) to users in a public park through gesture-based interactions. To capture users' gestures, the interactive prototype had three ultrasonic sensors attached to the Arduino board running custom software and connected through a USB to a Linux server. The server ran custom Java Player software that allowed the playback of audio content, aggregated daily on a custom site, by interpreting data streams from Arduino containing the predefined patterns that constitute command gestures. (Figure1E).

4.2 Semi-structured Interviews

Semi-structured interviews provide a wide-ranging and in-depth exploration of the research domain. The specific aim of the interviews was to explore how practitioners build interactive prototypes in practice. In particular I focused to analyse how connections was made and which components and technologies were used. Each interviewee described an interactive system he was constructing. The interviewees didn't participate in the design of the five prototypes described earlier.

Interviewee A1 was an expert electronic technician with a passion for music. He worked on a project aimed at recreating certain specific sounds through the reconstruction of modular analogue music synthesizers, based on seventies ('70) techniques and documentations. He built the system using several custom modules. The synth required a keyboard to generate input voltage to the system. For this, he used a MIDI (www.midi.org), saying "The keyboard is digital [MIDI] for convenience to simplify, I took a commercial keyboard with a MIDI output when I press this key, it doesn't return a voltage, but instead gives me a MIDI message." The keyboard communicated with the analogue system through a MIDI to control voltage (CV) converter (www.paia.com/midi2cv.asp), which he clarified as "basically a little box that does everything itself. MIDI is the standard, today everything runs on MIDI."

Interviewee A2 was a professional who managed his own interaction design studio. He worked on a system for live clubbing events designed to create a strong "lighting bath" experience for participants. He explained "If you have few days, the strategy is to use only the components available from existing devices and find another way to put them together that transforms them." He continued "Once, I used to be more ambitious and thought I could do it

myself, but the devices have become more fragile. Robustness is important, it is important that it works when you use it." He made an instrument that was "connected to the light boxes through a DMX (Digital MultipleX) layer." He used a DMX to Ethernet convert. "I send that through the network (using an OSC protocol) to the instrument; from here, I always send signals via UDP to the little DMX box that 'talks' with the objects."

Interviewee A3 was an independent media artist and university lecturer. He worked on an interactive system for live performance aimed to enable the performer to interact through his body with projected visuals and lighting system. The interactive system comprised a camera above the stage, connected to a computer running custom software to manage and analyse the stream of data coming from the camera. He started developing the system "through an enjoyable and stimulating process," reaching the "system core, the engine." He continued "when I start to develop the core of a system, usually this phase is the more playful and the more enjoyable and exciting phase, because you start with one idea and then you realize another." He said "The idea is to work in a modular way, not using software that makes everything but instead, using small connectable subsystems." He concludes, "I always use audio Cannon XLR connectors for transmitting a current of 12 volts. For example, for the cameras, I added an XLR connector. If I need a extra cable, I can always find that kind of audio cable in a theatre."

Interviewee A4 ran his own interaction design studio and was a university lecturer. He worked on a system that aimed to enable users visiting art exhibition to navigate through large multimedia catalogues using simple hand gestures. His system shaped like a table had a screen to visualized information and a camera to capture gestures connected to a computer running custom software. The system "was made using computer vision software that we wrote ourselves. There is a camera, two-capacitor sensors a projection and a system". "Downgrading the system from a more complex one was easy." He stated "in the project the main problem was the performance. The camera takes a picture every 20 seconds, and this code worked in Adobe Flash, which has very poor performance."

Interviewee A5 was a young designer who ran his own interaction design company. He worked on an installation to control small flying machines using a physical controller. He explained "The webcam is connected to a computer board via a USB connected to a maxduino (which connects MAX to Arduino), and this is connected to the prototyping board that connects to all of the 220-volt fans."

He continued "I used Arduino because I know it, and it's very simple: a micro with inputs and

outputs, and a lot of ready-to-use electronics, shields, many modules and a lot of documentation". He continued "disassembling the fan help me to understand the circuit and connect it to the Arduino".

These projects demonstrate the opportunities and challenges associated with the design of IoT devices and interactive systems. All projects included many heterogeneous components that required several standard or ad-hoc converters produced to enable communication among them.

5. DISCUSSION

The goal of this study was to explore and analyse how practitioners build interactive prototypes in practice. In particular I sought to describe and analyse how connections was made and which components and technologies were used.

All projects analysed and all subjects involved in the study adopted a digital bricolage approach, in which the activities of tinkering and patching guided them in constructing and deconstructing their systems. This approach enables them to think about their design objectives through the components, it gives users a cognitive space to experiment with components. This digital inquiring material helps them to understand complex dynamic flows by experimenting, obtaining results, and refining and retuning them, always focusing on their final goals. As described by Kirsh [15] external representation scaffold user's cognitive processes, and the role of materials in supporting and anchoring thinking is well described by Hutchins [13]

The prototyping of interactive systems isn't strictly anchored to the rigid distinction found in the traditional hardware and software design. Tools used in this context require an approach different from engineering. In design it is fundamental to soften technology to make it suitable for quick, imaginative solutions, and easy to shape the user's experience. The building of these systems is fluid and permits the user to control the hardware–software balance. Users might choose easily between hardware or software to obtain particular behaviours. These permit a more fluid user control to the hardware–software balance according to their knowledge, objectives and context. For

8. REFERENCES

1. Aarts, E.H.L. and Marzano, S. *The new everyday: Views on ambient intelligence*. 010 Publishers, 2003.
2. Beguin, P. and Rabardel, P. Designing for instrument-mediated activity. *Scandinavian Journal of Information Systems* 12, (2000), 173–190.

example, users can use hardware components to obtain results that are more difficult to obtain using a software component alone and vice versa. Moreover the communication model between components differs from the classical communication model. For example, in certain situations specific data is required to activate a connection or establish a communication session. In this study, there has been seen a more fluid and dynamic communication model, in which data and information can be conveyed from both hardware and software. For example, if a specific data type is required to establish a communication, then a hardware component may be used instead of a software command and vice versa.

Each project provides an opportunity to describe challenges and issues that emerge during the ideation, development, and testing of systems, as well as their behaviour. I focused on describing, analysing and categorizing the different types of connections between heterogeneous devices and components in order to describe the challenges encountered and the solutions developed to overcome them. Although the solutions are not applicable across projects, common challenges were experienced across multiple projects. The analysis is not intended to be exhaustive and complete, but instead focuses on the qualities that each system contributes to the creation of an initial framework to explore construction systems that support a fluid continuum between software and hardware.

6. CONCLUSIONS AND FURTHER WORK

These projects highlighted the opportunities, challenges and limitations associated with the design of interactive systems. As an exploratory study effort to develop a preliminary framework the analysis is not intended to be exhaustive and complete, but instead focuses on the qualities that each projects bring to the initial framework. More explorations along the dimensions discussed above are planned in the next future.

7. ACKNOWLEDGEMENT

I would like to thank the designers, researchers and practitioners involved in this study.

3. Bødker, S. Design for reconfiguration. *Proceedings of the 26th annual ACM international conference on Design of communication*, ACM New York, NY, USA (2008), 263–264.
4. Brandt, J., Guo, P.J., Lewenstein, J., Dontcheva, M., and Klemmer, S.R. Two studies of opportunistic programming: interleaving web foraging, learning, and writing code. *Proceedings of the 27th international*

- conference on Human factors in computing systems, ACM (2009), 1589–1598.
5. Buchenau, M. and Suri, J.F. Experience prototyping. *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, ACM (2000), 424–433.
6. Clark, A. *Supersizing the mind: Embodiment, action, and cognitive extension*. Oxford University Press, USA, 2008.
7. Cvijikj, I.P. and Michahelles, F. The Toolkit Approach for End-user Participation in the Internet of Things. *Architecting the Internet of Things*, (2011), 65–96.
8. Dix, A. Designing for appropriation. *Proceedings of the 21st British HCI Group Annual Conference on HCI 2008: People and Computers XXI: HCI... but not as we know it- Volume 2*, British Computer Society Swinton, UK, UK (2007), 27–30.
9. Gershenfeld, N., Krikorian, R., and Cohen, D. The Internet of Things. *Scientific American* 291, 4 (2004), 76–81.
10. 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.
11. Hazlewood, W.R., Dalton, N., Marshall, P., Rogers, Y., and Hertrich, S. Bricolage and consultation: addressing new design challenges when building large-scale installations. *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, ACM (2010), 380–389.
12. Hollan, J., Hutchins, E., and Kirsh, D. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction (TOCHI)* 7, 2 (2000), 196.
13. Hutchins, E. Material anchors for conceptual blends. *Journal of pragmatics* 37, 10 (2005), 1555–1577.
14. Ingstrup, M. and Hansen, K.M. Palpable assemblies: Dynamic service composition for ubiquitous computing. *Proceedings of Software Engineering and Knowledge Engineering (SEKE)*, Citeseer (2005).
15. Kirsh, D. Thinking with external representations. *AI & society* 25, 4 (2010), 441–454.
16. Kyng, M., Nielsen, E.T., and Kristensen, M. Challenges in designing interactive systems for emergency response. *Proceedings of the 6th conference on Designing Interactive systems*, ACM (2006), 310.
17. Mellis, D., Banzi, M., Cuartielles, D., and Igoe, T. Arduino: An Open Electronic Prototyping Platform. *CHI: ACM Conference on Human Factors in Computing Systems*, (2007).
18. Moran, T.P. Everyday adaptive design. *Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques*, ACM (2002), 13–14.
19. Myers, B., Park, S.Y., Nakano, Y., Mueller, G., and Ko, A. How designers design and program interactive behaviors. *Visual Languages and Human-Centric Computing, 2008. VL/HCC 2008. IEEE Symposium on*, IEEE (2008), 177–184.
20. Newman, M.W., Sedivy, J.Z., Neuwirth, C.M., et al. Designing for serendipity: supporting end-user configuration of ubiquitous computing environments. *Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques*, ACM (2002), 156.
21. Norman, D.A. *Cognitive artifacts*. Department of Cognitive Science, University of California, San Diego, 1990.
22. Norman, D.A. *The invisible computer*. MIT press Cambridge, MA, 1998.
23. O’Sullivan, D. and Igoe, T. *Physical computing: sensing and controlling the physical world with computers*. Course Technology, 2004.
24. Papert, S. *Mindstorms*. Basic Books New York, 1980.
25. Paradiso, J.A., Heidemann, J., and Zimmerman, T.G. Hacking is pervasive. *Pervasive Computing, IEEE* 7, 3 (2008), 13–15.
26. Perring, C. Interaction design prototyping of communicator devices: Towards meeting the hardware-software challenge. *interactions* 9, 6 (2002), 36–46.
27. Rabardel, P. and Bourmaud, G. From computer to instrument system: a developmental perspective. *Interacting with Computers* 15, 5 (2003), 665–691.
28. Resnick, M. and Silverman, B. Some reflections on designing construction kits for kids. *Proceedings of the 2005 conference on Interaction design and children*, ACM (2005), 117–122.
29. Rogers, Y. Moving on from Weiser’s vision of calm computing: Engaging UbiComp experiences. *Lecture Notes in Computer Science* 4206, (2006), 404.
30. Villar, N., Scott, J., Hodges, S., Hammil, K., and Miller, C. .NET Gadgeteer: A Platform for Custom Devices. In J. Kay, P. Lukowicz, H. Tokuda, P. Olivier and A. Krüger, eds., *Pervasive Computing*. Springer Berlin Heidelberg, 2012, 216–233.
31. Weiser, M. Ubiquitous computing. *IEEE Computer* 26, 10 (1993), 71–72.