Do-It-Yourself (DIY) Assistive Technology: A Communication Board Case Study

Foad Hamidi[1], Melanie Baljko[1], Toni Kunic[1], Ray Feraday[2]
[1]*Lassonde School of Engineering, York University*
4700 Keele St., Toronto, Ontario, Canada, M3J 1P3
{fhamidi, mb, tkunic}@cse.yorku.ca
[2]*Toronto Catholic District School Board*
feradar@tcdsb.org

Abstract. Do-It-Yourself (DIY) open-source hardware allows users of technology (and/or their community members) to create their own tools and designs. In recent years, resources and components for DIY electronics have emerged that allow for the development of customized, affordable assistive technologies. These resources, which can be shared online, open doors for new ways to create and share technology and present an approach that has the potential to be more efficient, affordable, and eventually effective than traditional approaches to assistive technology development and deployment. In this paper, we present a case study of how "citizen designer" methods have been used to develop a DIY open-source Speech-Generating Device for use by children.

Keywords: Do-It-Yourself (DIY); open-source hardware; communication boards

1 Introduction

Recent years have witnessed the appearance of more accessible fabrication methods and computer-aided design software that has, in turn, brought about a proliferation of new designs for physical objects, as created both through customization of existing designs and through the development of completely new ones. Open-source hardware is able to make use of electronic components and microcontrollers that have become more readily available and increasingly more affordable. Online support communities are flourishing, providing extensive coverage of practically every aspect of the design and fabrication process for novel physical, digital objects. Barriers to necessary software are increasingly mitigated: the barrier of cost through increased availability of free or open-source packages, and the barrier of knowledge through the proliferation of more user-friendly, less-specialist software user interfaces. All of these trends are converging together, ushering in a new era of "citizen designers[1]", and subtending a new wave of technology creation that some pundits have called a "new industrial revolution" [1]. The Maker Movement (or Do-It-Yourself (DIY) Movement) loosely refers to the body of amateur and professional designers who use novel (e.g., 3D printing) and traditional (e.g., glassblowing) manufacturing methods to engage directly with every stage of the creation of their customized, small-batch designs [1].

Although some researchers and activists have identified the possibilities of these developments for assistive technology, the potential has not yet been fully explored [11]. The use of open-source hardware and software to make customizable designs is considerably easier than other approaches (e.g., trying to hack closed-box commercial off-the-shelf technologies or designing from scratch). However, different barriers exist for people who would like to adopt the DIY approach. These include knowledge barriers (having limited knowledge of programming and/or electronics), cost barriers and physical and cognitive barriers. Hurst and Kane have previously identified a need to develop tools to make "making" more accessible [10].

In this paper, we present a case study that demonstrates the application of the DIY methodology to the creation of TalkBox, an open-source, customizable Speech Generating Device (SGD). TalkBox is intended to be a more-affordable and more-easily obtainable alternative to commercial SGDs. Two prototypes were developed. The first prototype made use of a Makey Makey board for input actions, whereas the second

[1] The term "citizen designer" has been used before (e.g., [9]) to denote socially responsible designers that take into account social and human values. Here, we use the term in a different sense to refer to people who are not professional designers (i.e., don't have formal training in design and don't earn their living through design work) but similar to "citizen scientists", adopt design methods and use them in a grassroots fashion to come up with solutions to real world problems.

used a more sophisticated system configuration using capacitive touch sensors. A collaborative design methodology was utilized, and our interdisciplinary team consists of the “citizen designer” and special education specialist, and students and faculty from a computer research lab. Throughout our process, special education domain expertise, hacking and programming skills were combined.

2 Background

Conventional computer keyboards present significant accessibility barriers for individuals who do not have the required motor skills, and a variety of alternative input devices and techniques have been developed to address this barrier (e.g., single-switches, modified keyboards, eye-gaze, and speech-based input, among others) [2]. These devices provide a large variety of possibilities and address the needs of many users. And yet there are individuals who cannot easily use them. Although these devices provision for some degree of customizability, it can be difficult to change them beyond simple settings. Attempts to modify beyond these provisions (e.g., through hacking) can possibly nullify their warranty.

The fit of the device to the user's needs is a known factor in assistive technology abandonment, which unfortunately, occurs at a high rate (close to 30% in the US [19]). Phillips and Zhao concluded over two decades ago that one way to address the issue of abandonment is through the development of services that emphasize more consumer involvement [19]. The DIY movement offers the possibility of increased consumer involvement, since the user can be directly involved in the design and fabrication of the technology he or she will be using [11].

The use of open-source hardware is relatively new to assistive technology. There have been a small number of initiatives involving open-source software such as the ITHACA framework [18] and projects COMSPEC [14] and ACCESS [13], as well as the OATSoft open-source software repository [12]. Formally documented, open-source software, hardware, and design resources for assistive technology are beginning to become available. For instance, the specialized “Hackcess” user forum was created within the Makey Makey (an open-source hardware board) discussion board, with a stated focus on the use of this specialized electronic component in assistive technology applications (http://www.makeymakey.com/forums). Hurst and Tobias [11] presented two case studies of projects that involved DIY assistive technology hardware development. The first project involved instructors of an adaptive art class who wanted to find a means for their students to paint without using their hands. After unsatisfactory experiences with expensive consumer solutions, they decided to make their own customized drawing tools. By combining parts from solutions bought online and a face shield, they were able to come up with a more stable and comfortable solution than was otherwise available. The second case study documented the approach adopted by a member of the maker community with a focus on assistive technology. This individual, who is a retired finance professional with an engineering degree, has been independently adapting, designing and building assistive technology and disseminating the results via a website (http://workshopsolutions.com). The website currently contains more than 170 designs (both of his own and submitted by community members). His practice is a concrete instantiation of the belief that the sharing of ideas online is important and allows for people to connect with the technologies they need.

The idea of forming interdisciplinary assistive technology teams to address specific problems and customize or develop new technology is slowly becoming more common. These teams bring together diverse experiences and knowledge bases --- for instance, subject domain knowledge (e.g., engineering and computer science, speech language pathology, occupational therapy, special education teaching), and life experiences (e.g., as an assistive technology user, as a frequent interaction partner with individuals with disabilities). For example, CanAssist (http://www.canassist.ca/), an organization at the University of Victoria, BC, employs a diverse group comprising of individuals with disabilities, engineers and software developers, co-op and graduate students, and volunteers (consisting of retired engineers and other professionals). CanAssist develops technological solution for community–identified problems, oftentimes by customizing and modifying existing computer hardware and software. Another example is described by Gómez et al. (2012), where final year computer science students developed games and other interactive applications through co-design with children with cerebral palsy [7]. In our project, we take a similar approach to assistive technology development and bring together community partners and students and researchers in our lab. In addition, we also are using open-source hardware and software and make use of maker methods, such as 3D printing and rapid prototyping, to customize our designs further.

3 An Open-Source, Customizable Communication Board

Project Background: The first iteration of the interface (Figure 1, top left) was originally developed for a set of specific students in an educational setting. The users were either non-verbal or used verbal communication rarely and used various commercial Augmentative and Alternative Communication (AAC) systems with a variety of access solutions that had been provided through the available social services. For multiple and different reasons, many of the AAC system configurations had shortcomings. For instance, with the systems that employed touch screens, there were input errors caused by the user resting the side of their palm on the board while trying to touch a target with their finger and by the student's inaccuracy in targeting and calibrating pressure of touch actions (since the screens required pressure to be applied straight on or in a well-aimed swipe). With the systems that employed physical switches, the amount of pressure required for activation was too great, at least relative to the user's physical capabilities (and the force thresholds could not be calibrated). Also, the sizes of the switch were not optimally tailored to the users and could not be adjusted (some users require larger switches because of lack of hand movement precision, whereas others could have used smaller-sized switches, which would have afforded a larger number of switches and thus more selection options). For the joystick-based access solutions, the amount of hand-eye coordination required for use also presented barriers. These students, to various degrees, benefited from the assistive technology services that were formally available (from the school board and the government), but, for various reasons, were matched with technologies that were not satisfactory and were either abandoned or were at risk of abandonment. Alternative technologies were needed, but could not be obtained for multiple reasons (e.g., long waiting times for re-assessment, limitations in the repertoire of off-the-shelf technologies approved for prescription, and the high cost of working outside of the system of socialized services that is in operation in Canada).

Figure 1. The first prototype (top left) consists of a computer (not shown) and a Makey Makey board connected to a hand-made foam board fitted with aluminum foil touch-switches and labeled with replaceable printed symbols. TalkBox (bottom left and in use, right) consists of a Raspberry Pi connected to a capacitive touch sensor, speakers and a hand-made foam board with aluminum foil touch switches

Design Process: The design work on the first prototype was conducted by one of the co-authors (Feraday), who is a special education teacher and maker. He has worked for many years as a special education teacher in the Greater Toronto Area (GTA), working with students with various and often multiple disabilities. In recent years, he has become interested in using maker tools and methodologies to develop custom assistive devices for his students. His interest and motivation led him to experiment with electronic prototyping tools. For the first prototype, he used a network of family and friends to bring together expertise, combined with the use of online resources, to resolve technical issues.

The first prototype made use of the Makey Makey Human Interface Device (HID) (JoyLabz, Santa Cruz, CA), which is a circuit board with 18 input ports that can be connected, via alligator clips, to any conductive object (e.g., aluminum foil, metal objects, even fruit and vegetables!) [4]. When the conductive object is touched by the user, a closed circuit is formed, which, in turn, dispatches a signal to the output USB port that emulates either a keyboard key press or a mouse click. The closed circuit is completely safe for users since the levels of electrical potential are well below harm thresholds. The HID allows essentially any conductive object to serve as an alternative to a keyboard key or mouse button. Makey Makey can be connected to any computer through USB and thus, only the most basic level of computer literacy is needed to make use of this component.

The first prototype communication board consisted of a chassis made out of polystyrene foam and cut into a rectangular shape. Arrayed along one edge of the chassis is a series of “switches” (currently, 6 are used). Any layout or configuration of the switches is possible. The switches are formed from self-adhesive aluminum duct tape, which is cut with scissors and fastened to the foam chassis such that the switches are separated from one another by (non-conductive) foam channels. Each of the switches is connected to the Makey Makey by alligator clips via a contact assembly, made from aluminum strips running from the switch to the clip flange. Additional rectangular foam boards are then cut, to form the seats for the switch labels, which are color printed and affixed by regular craft glue. These switch labels are drawn from a symbol set that is in common use among the students. The communication board was then connected to a consumer off-the-shelf (COTS) computer via USB. The computer was running the SoundPlant (http://soundplant.org) software, a shareware tool that allows the keys on the keyboard to be mapped to sound files. Through this mapping, each of the switch activations triggered the playback of the associated sound file, implementing, in effect, a basic Speech Generating Device (SGD).

A preliminary evaluation session was conducted with a female student. The student, who has multiple disabilities, including cerebral palsy, scoliosis, spina bifida and who is non-verbal, had been matched with a variety of commercially available AAC solution, but they did not work well for her. Specifically, she had difficulty with providing the pressure and precision needed to activate the switches. She was able to successfully use the current design to engage in multiple communication exchanges.

The other co-authors of this paper, during a visit to the 2013 Toronto MakerFaire (http://makerfairetoronto.com) saw Feraday’s prototype on display there. Following this initial contact possibilities for collaboration were explored and eventually a team was formed to work on a second version of the communication board (which was then dubbed “TalkBox”). There were several issues with the first prototype. First, the Makey Makey prototype required the user to touch the board’s ground wire while touching the pads. Although this was not an insurmountable obstacle (e.g., it had been shown to be possible to train users to press two pads simultaneously, where one was connected to ground), it presented a non-trivial inconvenience. Second, Makey Makey needs to be connected to a computer via USB. The computer that was used in the prototype was relatively large and inconvenient (for instance, it couldn’t be placed on the user’s wheelchair chair, nor could it be moved around too much, nor was the connective USB cable convenient in a busy classroom setting). Third, the cost of the components (the Makey Makey, combined with the cost of the COTS computer) was expected to be a problem, given fort he budgetary constraints of the school board.

To address these issues, the design shifted to the use of the Raspberry Pi single-board computer rather than a COTS computer. The SoundPlant software is not available for the Raspberry Pi’s OS (Linux), so the Scratch programming language (pre-loaded on the Raspberry Pi) was used instead to map the leads from the Makey Makey to specific sound files. Last, we explored the possibility of an alternative to the Makey Makey, using the MPR121 sensor controller (Freescale, Austin, TX), which was a less expensive option ($10 CAD compared to $50 CAD, approximate costs).

The Raspberry Pi computer is a low-cost single-board computer developed in 2012 by the Raspberry Pi Foundation (http://www.raspberrypi.org) in the UK to promote programming and computer science education in the classroom. Despite its recent appearance, Raspberry Pi has received a lot of attention and a vibrant community is already formed around its use [3]. Although small (credit-card sized), Raspberry Pi is a full-fledged computer with processing power sufficient to perform speech synthesis and even high-definition video processing. Because of its size and small power consumption, it is used extensively for embedded and physical computing projects.

The MPR121 sensor controller provides a capacitor touch sensor that accepts human body capacitance as input and is activated when a hand or finger touches a pad connected to it. The capacitor touch sensor does not require a simultaneous connection to a ground wire. The Raspberry Pi and touch sensor combination is small enough to be connected to a wheelchair, blackboard or even clothing and can be powered with batteries; the combination of Raspberry Pi and touch sensor costs significantly less than a computer connected to the Makey Makey. In order to capitalize on these advantages, we need to develop a software interface to connect the touch sensor and the Raspberry Pi; further software was developed to actually implement the mapping from input event to communication board behaviors. Last, an additional software module was developed to provide the capability to configure this mapping (e.g., the ability to define multiple mappings between different input actions and sound files, and defining the trigger to switch among multiple mappings). This work unfolded over 6 weeks, with weekly design meetings, and iterative design methodology, and frequent modifications. A GitHub project was created (http://hrairhlessil.github.io/TalkBox/), which provided software versioning, issue tracking, and open-source deployment.

The TalkBox prototype is shown in Figure 1 (bottom left). TalkBox consists of a Raspberry Pi board connected to a MPR121 capacitive touch sensor, an inexpensive battery power source for the Raspberry Pi, a battery-powered USB speaker and a wireless mouse. The setup is connected to a polystyrene foam chassis similar to the one used by the first prototype but with smaller aluminum tape switches and shorter wires for connecting the switches to the MPR121 pin sensor connections. Initial experiments showed that the use of alligator clips and large touch pads did not work well with the capacitive touch sensors (many false positive and false negative activations registered). The core software runs as a daemon, which initializes by loading the required sound files. The prototype uses 6 pads, each of which is mapped to a sound file (and is signified by an image placed above the pad). Whenever a pad is touched, the corresponding sound file is played back on the attached speaker. Thus, TalkBox, in effect, implements an open-source SGD.

In the current version both 6 and 12 switch variants were developed (12 pins are available in the MPR121 sensor). The 6-pad variant keeps the interface relatively compact. To expand the repertoire of words and phrases, a scheme was used whereby they are arranged into categories, and the user can switch among different categories. The software module is used to configure the set of categories, the words and phrases in each category, and the sound files that each word or phrase is mapped to. The user can switch between categories by pressing the mouse button (each button press selects the next category in the category sequence). This method of category switching was chosen because currently the teacher makes the change. Each category also has a corresponding foam-core strip that has the images affixed for each word or phrase in the category, which can be placed on the chassis of the interface by the teacher when each category is changed. It is possible that the user could change the category as he or she wants (by assigning the "change category" function to one of the pads), although some further modification would be needed (e.g., some sort of digital display so that the image label for the switches could be dynamically updated, or a scheme whereby the switch labels are left out).

The current words were selected specifically for the user for whom TalkBox is being designed. The set includes some common words for use in the classroom setting, as well as words and phrases for an activity done in the specific context (i.e., school and special education class) in which the current prototype is going to be used. The activity, which involves collecting attendance from teachers, requires the user to go through a series of questions and responses. TalkBox is thus serving as both the script and a visual checklist for that activity. It assists both in communication and in helping to focus on, and better understand the task at hand. For each of the identified words and phrases, two variants of the digitized speech files were recorded (one male and one female young adult voice), so that even this early version of TalkBox could offer that degree of user tailoring.

Analysis: The two prototypes of the open-source communication board have several benefits over existing alternatives:

- **Ease of deployment:** From the very beginning, a great motivation was to share the design in such a way that other people can recreate it themselves. To this end, documentation and instructions on how a version could be made were posted online [6]. For the second prototype, TalkBox, schematics of the electronic components, the developed software code, instructions on how to assemble the hardware and load the software, as well as, small libraries of original voice samples that can be used free of charge for the speech synthesizer are posted on the GitHub repository (http://hrairhlessil.github.io/TalkBox/).
- **Customizability:** Both the size and number of switches are customizable within constraints. For the first prototype, the size of the pads is quite flexible and provided a conductive material is used, touch activations are reliably detected. For TalkBox, if the pads get larger than the size currently

specified, touch detection becomes less consistent. We are currently exploring the range of sizes with which the interface works reliably. For both prototypes, we have implemented 6 pads but many more pads can be used in the future (up to 18 for the first prototype and up to12 for each sensor connected to TalkBox). On the software side, for the first prototype, any software program that uses the keyboard as input can be used. For TalkBox, we are currently developing a software interface that allows it to be used to interact with any software program that uses the keyboard as an option.

- **Cost:** Compared to the cost of a conventional communication board, the cost of the open-source communication board is small. For the first prototype, the costs include the Makey Makey board ($50 CAD), plus the cost of the computer (to which the Makey Makey is connected, via USB). The other parts and materials cost less than $10. For TalkBox, the costs include a Raspberry Pi ($40 CAD), a capacitive touch sensor ($10 CAD), the power supply, a speaker, and other materials. Depending on the use scenario and context, various components (such as a monitor or Wi-Fi module) can be added to Raspberry Pi for additional functionality. (All costs are approximate and depend on USD-CAD exchange rate).
- **Functional visibility:** With technologies becoming more complicated, it is common to hide the functionality of new devices. One of the appeals of this design is that the interface's underlying mechanism is easily visible and understandable. Design visibility provides the potential for the user or other stakeholders to understand how the system works, be less intimated about attempting to customize or fix it, and to learn about interfaces and electronics through hands-on use [17]. In the future, we aim to capitalize on this aspect of the design by using the device as an educational tool and to explore the possibility of using it to teach digital design to students with disabilities.
- **Durability:** Some students do not have good control over the pressure they exert on the communication board, resulting in heavy use. Although many commercial AAC boards are made sturdy and durable, switch failure due to heavy use does occur and swapping out hardware components can be difficult and expensive. It is easy and inexpensive to replace any of the components of TalkBox (e.g., foam board chassis, aluminum foil switches, connector assemblies). Moreover, other types of materials can easily be substituted, as the situation warrants.

4 Do-It-Yourself Assistive Technologies: A Promising Future

The current project is an example of a DIY grassroots solution to the immediate need for assistive technology to facilitate communication and self-expression in a classroom setting. It is also an example of academia and community collaboration. The benefits of this mode of research have been discussed at length in the Knowledge Mobilization community [21], and have been noted by us and others in prior work [7, 8]. Through our experiences, we have come to believe that bringing together community partners, researchers, students, and users has great potential for the development of relevant projects that incorporate key insights rooted in concrete domain expertise. In our experience the collaboration was beneficial to all parties. For the students, this project was an excellent learning experience and an example of how one can craft one's education by working on projects one deems as meaningful. For the community partner, the collaboration provides sustainable and stable access to complementary programming and analysis skills, as well as, a point of interface to the research community. For the academic researchers, the collaboration provided valuable contact with the community and a meaningful design domain. Needless to say, for all parties involved many of these motivations coincided. More importantly, for all the parties, this was a volunteering-based self-motivated endeavor (the project, as of yet, is unfunded) and the real reward was in conducting the project itself: exercising creativity and problem-solving skills in a collaborative atmosphere towards the higher goal of helping people with disabilities communicate through technology.

The emphasis of the maker and DIY movement is on hands-on creation activities. We believe it is important to balance this hands-on and experiential aspect with reflection. Academia can play a contributing role in the Maker Movement, for instance, through analysis and exploration of underlying theoretical frameworks and through increasing awareness and critical reflection on the hidden assumptions, ideologies, values and potentialities of novel artifacts and designs. This blend can be found in the current project. We recruited the maker methods of combining low-tech prototyping with open-source hardware, but we also developed clear user scenarios and recognized the values embedded in the design.

Another goal is to create new ways for people with disabilities to come up with ideas, and to design and to make the technologies they themselves and other people with disabilities would use is another one of our goals. Our conjecture is that the TalkBox system can be assembled by youth with mild cognitive disabilities, and we plan to investigate the fabrication process as a potential opportunity for paid employment. In addition, we see the Scratch programming language (http://scratch.mit.edu), which by virtue of being distributed with the Raspberry Pi Linux operating system, is already built into the TalkBox device, as playing a role in the teaching of programming to students who are presently not offered that learning opportunity. The original idea behind developing the Raspberry Pi computers was to incorporate programming into the school curriculum. In the UK where the Raspberry Pi was originally developed, the year 2014 was dubbed "the Year of the Code" (http://www.yearofcode.org) to reflect a recognition of the importance of programming, digital design and computer literacy in the future of education. Some critics have found the notion of the Year of Code immature and the technology not ready to catch up with the concept (e.g., [15]), partly because of the technical and conceptual background that is needed in the classroom to actually make the idea happen. With respect to students with disabilities, accessibility issues are still present when it comes to using new technologies such as the Raspberry Pi. Having said that the idea of tinkering and making computer programs and digital designs as effective ways of learning is much older and has been in practice in small scale since the 70s (e.g., in the work of Constructionists such as Seymour Pappert, among others [16]). So perhaps, developing interfaces that make these new promising hardware more accessible for everyone is a step towards the goal of using digital design (inclusively) in education. Towards this end, we have experimented with teaching the basics of programming with Scratch to special needs students, using the Raspberry Pi and resources at websites such as, Hour of Code (http://code.org/learn) (This is somewhat hampered by the fact that Scratch 2 is currently unavailable on the Raspberry Pi). The possibility of putting the "human-in-the-loop" is an important aspect of our system design (i.e. involving students in helping to construct and/or program elements of the device they will be using). Thus, the design itself builds in empowerment and agency, and breaks down the "provider" and "receiver" roles.

Our approach to assistive technology has another aim, as well, and that is making assistive technology solutions available to international communities. Due to their low-cost and open-source designs (and thus reproducibility), new revolutionary design ideas that use low-tech DIY and open-source hardware are already appearing around the world (e.g., the freely available Disabled Village Children [22] and the Robohand project: http://robohand.net/). Our vision is to expand these possibilities by making digital Assistive Technology designs available to more people in developing countries, through online publishing of open-source hardware ideas. This will also provide a space for innovator in these countries to share their ideas and vision, and will dialogue between interested stakeholders to create mutually beneficial relationships.

5 Conclusion and Future Work

New "citizen designer" tools and methods provide great potential for the development and deployment of novel, low-cost assistive technologies. We have presented two prototypes of an open-source communication board that was developed using these methods. The board is provided as an alternative to commercial proprietary design components that are often hard to modify. An important goal for this design is its availability to other potential "citizen designers", who can then build and modify it in whatever way they feel would benefit the end user. We have also discussed ways to foster similar projects by linking in-the-field special education teachers and caregivers with novel ideas for assistive technology with students: an exchange potentially beneficial to both parties, as well as, more importantly, to users with disabilities. Finally, we believe we have taken a step towards developing accessible interfaces to making tools, such as the Raspberry Pi and Arduino that are deemed important components of future education.

The first step in our future plan is to evaluate the interface with more users with disabilities. This will provide us with insights into the potentials and shortcomings of the design and help us refine it in future iterations. While the current interface was informed by many years of experience working with people with disabilities and by taking into account first hand information on the needs of specific users with disabilities, having a working prototype allows us to communicate and explore design alternatives more effectively with the users of our system and follow a Participatory Design methodology to refine and extend the interface, a method that has been found to be effective in previous research [5]. We plan to make another version of TalkBox that is not limited to synthesizing sound and where touching the pads correspond to general input actions that can control a variety of applications and programs on the Raspberry Pi. This will allow us to have an accessible interface to the Raspberry Pi, a possible step towards helping students with disabilities exercise the learning potential of the Raspberry Pi. Finally, although we have detailed instructions and code on the project website, we want to come up with a process to assemble and customize the system that is accessible to students with disabilities. Currently, we are examining ways to present TalkBox and other variations through an affordable, easy-to-assemble and customizable kit (possibly with some 3D printed components).

6 Acknowledgements

We would like to thank Max and Christine Feraday for recording the words and phrases used with their voices. Sincere appreciation to the Toronto Catholic District School Board and TSU/OECTA for their support, especially Frank Piddisi, Peter Stachiw, Susan Menary and Thérèse MacNeil.

7 References

1. Anderson, C. 2012. *Makers: The New Industrial Revolution.* Crown Business.
2. Beukelman, D.R., and Mirenda, P. 1998. *Augmentative and alternative communication: Management of severe communication disorders in children and adults* (2nd Ed.). Paul H. Brooks Publishing Co, Baltimore, MD.
3. Bridgewater, A. 2012. Community strength blossoms for Raspberry Pi. In *ComputerWeekly.com.* http://www.computerweekly.com/blogs/open-source-insider/2012/03/community-strength-blossoms-for-raspberry-pi.html
4. Collective, B. S. M., and Shaw, D. 2012. Makey Makey: improvising tangible and nature-based user interfaces. In *Proc. of TEI'12*, 367-370.
5. Dawe, M. 2007. Let me show you what I want: engaging individuals with cognitive disabilities and their families in design. In *Proc. of CHI'07*, 2177-2182.
6. Feraday, R. and MacNeil, T. 2013. http://www.sombrepeacock.com/a-role-for-the-makey-makey-in-augmentative-communication/
7. Gómez, I. M., Cabrera, R., Ojeda, J., García, P., Molina, A. J., Rivera, O., and Esteban, A. M. 2012. One way of bringing final year computer science student world to the world of children with cerebral palsy: a case study. In *Proc. of ICCHP'12,* 436-442.
8. Hamidi, F., Baljko, M., Livingston, N., Spalteholz, L. 2010. CanSpeak: A Customizable Speech Interface for People with Dysarthric Speech. In *Proc. of ICCHP'10*, 605-612.
9. Heller, S., and Vienne, V. (Eds.). 2003. *Citizen designer: perspectives on design responsibility*. Skyhorse Publishing Inc.
10. Hurst, A. and Kane, S. 2013. Making "making" accessible. In *Proc. of IDC '13*. ACM, New York, NY, 635-638.
11. Hurst, A. and Tobias, J. 2011. Empowering individuals with do-it-yourself assistive technology. In *Proc. of Assets'11*, 11-18.
12. Judge, S., Lysley, A., Walsh, J., Judson, A., and Druce, S. 2006. OATS - Open source assistive technology software—a way forward. In *Proc. of AAC'06*, ISAAC Press.
13. Kouroupetroglou, G. and Pino, A. 2001. ULYSSES: A Framework for Incorporating Multi-Vendor Components in Interpersonal Communication Applications. In *Proc. of AAATE'01*, 55–59.
14. Lund'alv, M., Lysley, A., Head, P. and Hekstra, D. 1999. ComLink, an open and component based development environment for communication aids. In *Proc. of AAATE'99*, 174–179.
15. Naghton, J. 2014. Year of Code already needs a reboot. In *the Guardian*, http://www.theguardian.com/technology/2014/feb/15/year-of-code-needs-reboot-teachers
16. Papert, S. 1980. *Mindstorms: Children, computers, and powerful ideas.* Basic Books, Inc.
17. Perner-Wilson, H., Buechley, L., and Satomi, M. 2011. Handcrafting textile interfaces from a kit-of-no-parts. In *Proc. of TEI'11*, 61-68.
18. Pino, A., and Kouroupetroglou, G. 2010. ITHACA: An Open Source Framework for Building Component-Based Augmentative and Alternative Communication Applications. In *TACCESS*, 2(4), 14.
19. Phillips, B. and Zhao, H. 1993. Predictors of assistive technology abandonment. In *Assistive Technology.* Taylor & Francis. 5(1), 36–45.
20. Tanenbaum, J. G., Williams, A. M., Desjardins, A., and Tanenbaum, K. 2013. Democratizing technology: pleasure, utility and expressiveness in DIY and maker practice. In *Proc. of CHI'13,* 2603-2612.
21. van de Ven, A. H. and Johnson, P. 2006. Knowledge for theory and practice. In *Academy of Management Review*, 31(4), 802–821.
22. Werner, D. 1988. *Disabled village children*. Hesperian Foundation Palo Alto, CA.