SMSlingshot: An Expert Amateur DIY Case Study

Patrick Tobias Fischer, Eva Hornecker
University of Strathclyde
26 Richmond Street, G1 1XH, Glasgow
fischer@cis.strath.ac.uk, eva@ehornecker.de

Christian Zoellner
University of Arts Berlin (UdK)
Einsteinufer 43, 10587 Berlin
mail@christian-zoellner.com

ABSTRACT

This paper discusses the design process of VR/Urban's public tangible interface SMSlingshot, a real-time system for urban interventions on Media Façades, which we have exhibited in the last few years around the world. In this case study we investigate how the design collaboration between technologists and industrial designers contributed to the success of the urban intervention. The design process of this 'product' has many DIY aspects, with professional industrial designers and technologists becoming expert amateurs, often dealing with problems that pushed them outside of their professional comfort zone. Don't be afraid of being an amateur!

Author Keywords

Design Process, Creative Process, DIY in the wild.

ACM Classification Keywords

H5.m Information interfaces and presentation

INTRODUCTION

The 'expert amateur' [13] – in the original sense of the term, is defined as someone who loves or is fond of something – is part of our scientific persona. As HCI researchers, who traverse the disciplines of design and science, we regularly do what we love to do, to explore the boundaries between the known and the unknown. Projects that focus beyond efficiency and productivity are often realized with a very small budget. Money is traded for time, and projects acquire DIY characteristics. DIY projects are low budget [12, 16], and often involve doing something for the first time, not only within one's own domain, but also taking on other tasks and responsibilities that may arise and require learning.

The SMSlingshot project was created by a team of four (plus additional people at times) 'expert amateurs' who left their professional comfort zone to push their knowledge, each in their own way following a shared vision. To document the DIY process, which involved the creation of physical objects, software and hardware artifacts with complexities reserved for experts, we here reflect on the motivating factors that made this public interface possible. In doing this, we here focus mainly on the hardware/software design process, and not on the application context of media facades (cf. [2]).

Our project provides a case study of a longitudinal (3 year) interdisciplinary collaboration in an expert amateur DIY process. Using the Arduino [1] hardware platform as a starting point, the final hardware version created is a complex, custom embedded device. Repeated improvements were driven through 'in the wild' interventions (at various festivals exhibitions, etc.) that at the same time created hard deadlines. Part of the development process might thus be interpreted as an example of 'in-the-wild design' [3], where frequent deployments created a drive towards readiness for batch-production, requiring a high degree of robustness as well as motivating design refinement.

The Urban Intervention

SMSlingshot is a Media Façades installation situated in an urban environment, usually for around 3 hours at night. The interactive system consists of a portable wooden device in the shape of a slingshot and a rendering PC with camera and xBee receiver, which is connected to the façade display or a projector. The device is equipped with an xBee transmitter, ATmega328 microprocessor, LCD display, green laser module and batteries. The text messages are typed on a phone-sized wooden keypad which is integrated in the wooden case. After a message is typed, the user aims at a Media Façade and shoots the message at the targeted point. It will then appear as a coloured splat with the message inside. To create a smooth and magical user experience, most of the technology used is designed to recede into the background. Moreover, the projection integrates into the built environment, using natural borders of buildings (c.f. [14]), that are projected onto.

POINT OF ORIGIN AND MOTIVATION

The origins and vision of the SMSlingshot can be traced back to its precursor, the spread.gun [8], created for the Media Façade Festival 2008. The main motivation back

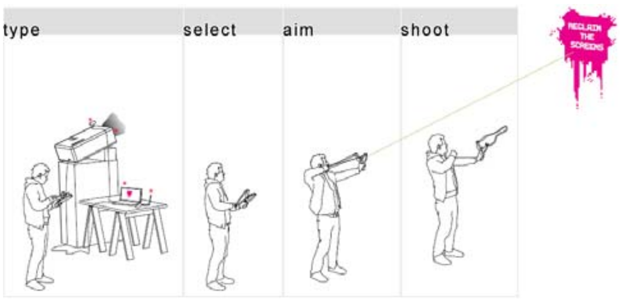

Figure 1 Interaction flow of SMSlingshot

TEI 2013, Feb 10-13, 2013, Barcelona, Spain.

then was (a) to develop something interactive for a large Media Façade and (b) to create a digital Agorá that opens up a passive medium, allows the public to 'speak up'. The maxim of the VR/Urban team became "Reclaim the Screens". Inspired by the Situationist International (cf. [5]), an arts collective in the 50s, VR/Urban employed the method of détournement. Here, the rather passive spectacle on Media Façades produced by private corporations (which usually plays back pre-produced content) is turned into an active intervention, where the people create the spectacle in front of the façade and decide on the content in situ. The spread.gun installation had similar core functionality to the SMSlingshot, but used a fixed station terminal with a touchscreen (provided by a sponsor) for typing messages and another station with a cannon-shaped device to 'shoot'.

Following the completion of the spread.gun project, the team felt unsatisfied with the social interaction generated around the installation. Because the installation consisted of a fixed station, interaction spaces (see [7]) were rather small and rigid in space, people lined up, resembling a queue in front of an ATM machine, which did not encourage shared encounters among strangers. Furthermore, we had not considered that groups are commonplace in urban spaces and had designed the installation for the passers-by to arrive one-after-the-other rather than as a multi-user interface. Designing situations turned out to be much more difficult than initially thought. Only much later did we learn to drag the interaction from the screen into the space in front of it [7].

Our dissatisfaction with the spread.gun was one of the reasons for improving the concept. At this point we did not know the issues regarding how to create situations in front of a façade [7]. We felt that the design must be more flexible, guerrilla-like, smaller, portable, employing less static structures and more expressive gestures; simply put: more embodied. A design feature of the spread.gun that we particularly liked, as it supported high level goals, was a pinball trigger that bridged the physical and bodily experience with the virtual. These two aspects were the driving forces for a redesign, leading to the idea of SMSlingshot.

PROCESS

Not having to deal with a sponsor allowed the design to be more consistent and flexible. With the spead.gun, there had been a requirement to utilize a sponsor's existing city furniture that compromised the design. Nevertheless, spread.gun had allowed the team of two technologists and one designer to gain experience with situated urban interventions and Media Façades. Another designer now joined the team, and thus four core people were involved in the creation of SMSlingshot. Two took the role of design / technology lead and the other two supported them. For easier reference, we denote the roles as: D1 (Design lead, also involved in the spread.gun project), D2 (Design support and later design lead for version 3), T1 (Software developer lead, DIY hardware developer), T2 (Software development. support).

The interdisciplinary discourse around physical and digital artefacts and materials and utilization of distinct domain knowledge was essential for the development of this design. All members had professional experience in their own domain, except for D2 who was a recent graduate at the time. However, the project and the ambitions of D1 and T1 pushed their expertise to the limit, hence becoming novices in new processes and knowledge domains due to their love of their vision. These are described in the following.

Overview of Version History

Figure 2 shows an overview of the development process of form, soft- and hardware as well as the deadlines driving this process, mainly created by interventions (e.g. art festivals, street action, etc.). The diagram shows that hardware development always preceded form development. It also shows that older versions of SMSlingshot casings were often kept as a backup, hardware versions were replaced (H1) as well as kept (H2), while the software was constantly extended almost independently from the form and hardware iteration process, and was always in a productive state. In the following, we discuss the individual versions to illustrate the main development steps.

Form Development

Many aspects of form development in this project went beyond traditional industrial design. The SMSlingshot required not just the design of an object, but also of an action. Meaning is created in bodily interaction, in the kind and style of action performed (cf. [6, 11]). Shooting a message with a slingshot is not only a strong metaphor but also evokes memories and feelings of unruliness and childhood years, as well as making the action visible to others. Furthermore, the SMSlingshot consists of multiple entities: the

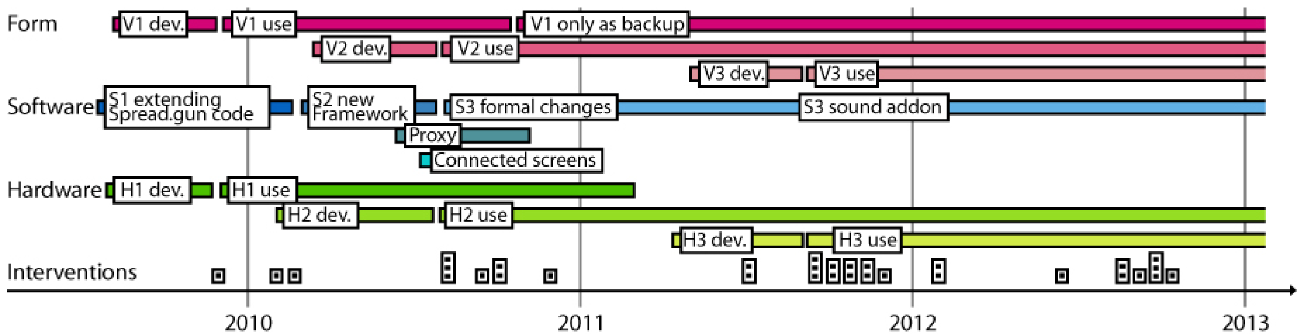

Figure 2 Overview of the agile DIY SMSlingshot development process showing phases of development and deployment milestones

Figure 3 Form development, from left to right: V1, V2, V3, then S1 splat, post produced splat, and S3 splat from real video.

slingshot case, its interface, and the renderings on the Media Façade.

The form development of V1 commenced with the general question: "What is a suitable device to throw colour splats with messages in it?" in conjunction with an image research of slingshots and similar devices, avoiding to limit the solution space too early. This revealed a variety of existing forms and strange devices, which served as inspiration and verified that our idea was novel. A quick photo mock up of a Siemens S35 mobile screwed onto a slingshot helped designer D1 to visualize the idea, while T1 had been inspired by the spread.gun's pinball trigger. D2 then was tasked to explore how such a throwing device could look like, using blue foam and shaping tools. Among these creations were models exploring slingshot-like forms (Fig. 4 left), but also fantasy devices based on a throwing or hurling metaphor. Additionally, a variety of sketches were created to explore the solution space. T1 experimented at this stage with text entry mechanisms in close collaboration with D2. Typing via a rotary encoder versus typing on a keypad was heavily discussed and explored on a working prototype using Arduino. The main reasons for choosing a keypad solution were simplicity, symbolism, speed of typing and the fact that people are used to keypads and thus do not need to figure out how to create a message (which might make people hesitant from using the slingshot and distract from the core vision of composing and shooting personal statements).

The design process of the next version, V2, was less straight forward. D1 (lead designer) found that it pushed the boundaries of conventional practice. The reason for this iteration after such a major milestone (a design competition) was to address the unsatisfactory aspects of V1, as well as a deadline set by a funded exhibition. The main issue for designers D1 and D2 was the simplistic, symmetric shape of the device. D1 felt it looked "cheap and unsmart, too glossy". The design objective for V1 was to be highly symbolic within the form constraints of a slingshot. For V2, a natural, less arbitrary form was desired. New sketches were created, most of them 1:1 to relate size to the hand and get the components right. However, this did not result in a satisfying solution. Most sketches still felt too 2D and symmetrical. Designer D1 argued that we had to go back to the start, to understand what it means to be a child, play in the woods and build your own slingshot. Thus, D2 ventured into the woods and collected an impressive variety of branches. From this exercise an understanding of how natural branches grow evolved and a similar manual modelling process started as earlier with blue foam (Fig. 4 left). Branches were modified to explore how technical components can be merged with/into the natural wood (Fig. 4 right). The disparity between technology and nature became apparent. We eventually scanned suitable branches in 3D, imported them into 3D Coat, modified the shape, and then used Rhino 3D to add cavities for the technical components, and generate code for the milling machine. The machined piece was manually adjusted (material added/removed where the branch looked artificial and added to make space for the electronics) and 3D scanned again. This process was iterated three times until the slingshot looked natural *and* could contain all technical components.

At this point T1 (technology lead) would have been satisfied to improve the V1 to make it more robust, while designer D1 felt unconfident and went on with V2. Without D1's design ambition that nurtured his dissatisfaction with version V1, the advanced design of the V2 would not have emerged. As none of the designers in the team had done this kind of work before, they also had to resort to working in DIY-style mode, including devising a novel design process.

For the version V3 design iteration, designer D2 took over the design lead, as D1 was not interested in the aspects focused on. D2 mainly fine-tuned version V2, focusing on improving robustness (encouraged by technology lead T1) and the creation of a more organic look by having wooden keys made from the same block of material. Also the final version V3 now comes in three different shape variations, as a set of three. Moreover, the shape of the slingshot had to be adjusted to fit new hardware components.

The brief of a 'more organic, less symmetrical' style for the V2 also had an impact on the design of the splat visuals, which initially had been in a pixellated 8-bit style (Fig. 3). T2 had tried to create organic splats via programming experiments, without satisfactory results. T1 realized that, just as we studied branches, one has to study real splats to create

Figure 4 Form finding V1 (left) vs. V2 (right)

a natural splat. Thus, D2 was sent out to fill balloons with paint and film them being thrown against a wall. A VJ friend post produced these splats, so that T2 could use these for real-time graphics (Fig. 3). To reduce visual repetition, colour drips were coded using random parameters.

Technology Development

Software Development

The JAVA software development process had only minor DIY aspects. Software developers T1 and T2 worked together remotely via SVN and Skype, and reused graphics rendering code from the spread.gun. The main DIY support consisted of Open Source libraries. The overall system also utilized a third party Open Source software called Lasertag from the Graffiti Research Lab, NY [9] to track the laser pointer for splat positioning. This was proven to work outdoors for up to 50m, providing confidence for the V1 design that tracking via low-cost camera is feasible. Extensions were made along the demands of particular exhibitions and research interest. These included a mode that connected Liverpool and Berlin, where people could ‘shoot’ messages from one city to the other, background images or text to focus the content of messages and scrolling text, and splashing sounds (S3) when coloured splats ‘hit’ the wall.

Along the entire software development process one major framework refactoring (S2) was made to make the code easier to extend. Flexibility is also a characteristic of the overall system. In-the-wild setups require mounting system components such as a tracking camera, rendering PC and xBee receiver at varying distances of each other, depending on the setting. Data connections ranged from xBee, Ethernet, GSM, WiFi and USB to Ethernet converters. The system was also adapted to work with low-res LED façades. All of this was a black box for the designers in the team, but clear for the technologist T1 due to his background.

Hardware Development

The hardware is a prime example of the expert amateur being pushed into a novice amateur role. The version differences in PCB development show how skills increased, from prototyping boards (Fig. 5 middle) to near-industrial manufacturing methods such as reflow soldering being used (Fig. 6). The hardware design was solely done by T1, a software developer with limited hardware development skills at the start. While the spread.gun design had solved most interface requirements in software, the mobile slingshot device demanded wireless embedded hardware that required more advanced electrical engineering skills. T1’s approach to hard- and firm-ware design was heavily supported by Open Source information related to Arduino. IC datasheets, circuit schematics, reference designs, library source code and example code helped him to understand how PCB’s are designed. DIY projects in general focus on cost efficiency and are creative in filling knowledge gaps and sourcing components. The early adoption of specific hardware components creates trends in the DIY community. In the SMSlingshot project, the display used is from a Siemens S65 mobile (one of the first mobiles with colour LCD’s, released 2004) 176x132 pixel LCD display (LS020, LPH88 or L2F50). These were mass produced, widely available from mobile phone repair shops, and thus cheap. However, the datasheets were confidential for a long time, and pins and protocols had to be hacked for DIY. Tracing the usage of these displays, we found the first working code and circuits on mikrokontroller.net posted in 2005. Later, the display appeared as an Arduino shield made by Watterott released in 2009. A hacker ethics [16] helps DIY projects by storing usable information on the web and is even exploited by some companies in commercial products, which then helps other amateurs to dismantle the product and extract the needed functions, components and knowledge.

DIY Circuit Development Process

For DIY electrical circuit development the possibility to dismantle existing circuits and learn from them is essential. In contrast to analog circuits, digital technology often allows for plug-and-play development. The steps for the first hardware iteration (H1) were to first get the individual components (display, keypad, xBee, trigger switch, laser module) to work, and then to have all components working together, which can be difficult when I/O ports and timer get sparse. Because the size of an Arduino in combination with the display shield would have compromised the design, we reduced the Arduino to a so-called RBBB (Really Bare Bone Board) and used wires to connect everything together. This resulted in a mess of wires, but it was the fastest method to get a first working slingshot that was used for our first intervention. Similar to D1’s dissatisfaction with the V1 form solution, T1 wanted to develop a single board PCB solution. Having never done this before, T1 first pro-

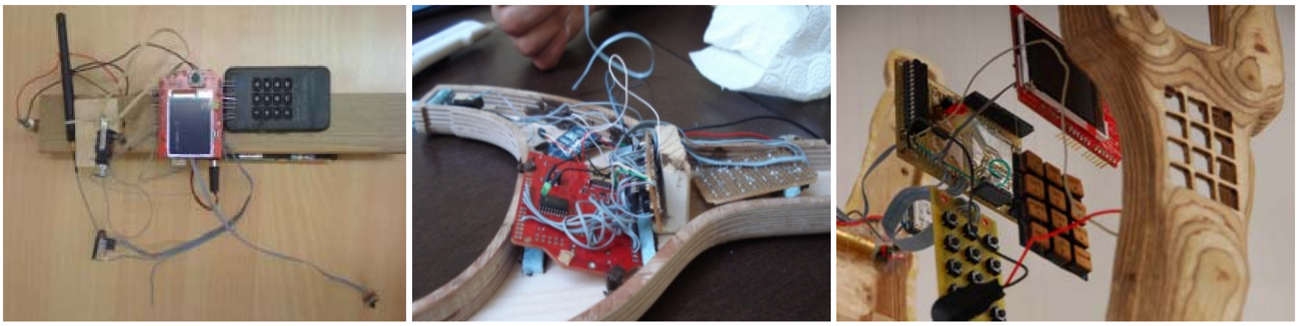

Figure 5 From left to right: Early functional prototype, the V1 with H1 hardware, and the V2 with the H2 double PCB solution.

duced a sandwich solution for H2 (Fig. 5 right) to reduce the risk of failure. This translated the working RBBB wire circuit to a PCB, leaving the display shield intact. At this point an experimental circuit was added to implement two different voltage supplies (5V and 3V3) on one board which proved problematic. For the last iteration H3 completely switched to a 3V3 design, with all components integrated on a coated PCB.

Another example of the amateur expert development process was the integration of the power supply. For the V1, T1 was confronted with the problem that some components run at 3V3 and others need 5V. Having only novice skills at that time and under time pressure, two separate sets of batteries were used as power supply. Several interim solutions were then experimented with, including switching to 3V3 for all parts, which worked as long as a high voltage supply was used. The integration of rechargeable batteries generated a new problem: current capacity. Power supply design requires an understanding of the dynamic curves of discharge and recharge time for batteries. After much experimentation and investigation of data sheets, a one cell (3V7) LiPo power solution that can be recharged via 5V USB was used for live shows, but this discharged too quickly and then stopped working. This left us frequently recharging and reverting to old slingshots as a backup. Finally, an expert solution was created using a Buck-Boost IC solution (Fig. 6). The drawback was that a more costly manufacturing process (reflow soldering) was needed.

The first example shows that the DIY community and available information around the Arduino platform is sufficient to advance from being a user of available components to creating custom solutions. Through reengineering of existing designs, T1 learned best practices in circuit design that helped the team to create a custom solution for a desired form of the SMSlingshot. The second example shows how T1 learned to create his own electronic designs, but then touched the boundary of miniaturization where manual soldering skills are not enough anymore.

Creating for the wild

For public interfaces, robustness is a big issue. Hence we were surprised that our agile development process worked even though we had to address three main problem areas simultaneously: form, hardware and software. Whereas software development has established agile methods [4], form and hardware development may fail in such a process as, for instance, certain parts don't always fit and electronic characteristics change because of environmental influences. An added complication arose because the team, over time, relocated to four cities in two countries, which often meant that we could only put all the components together and test them just before an upcoming event. This is less of a problem in software development, but creates problems when physical artefacts are involved.

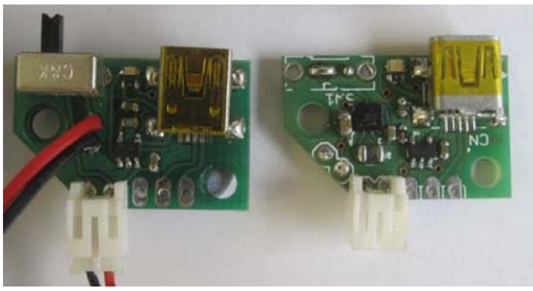

Figure 6 Expert solutions: Charger (left), charger incl. buck-boost power management IC reflow hand soldered (right).

Yet exhibiting live, from the early stages on, contributed to increased robustness in two ways. It established common knowledge among the team of how to do the live setup and how to react if something stopped working. In this case, each team member needed to be able to diagnose and solve problems fast, either by using a backup slingshot (old versions) or repairing the device during live exhibition. Furthermore, designing for the 'wild' also drove the hardware and form development process because of the diversity of environmental conditions. For example, it turned out that a green laser usually does not work below 10° C. At one of our interventions (Liverpool) for unidentified reasons, the radio transmission was considerably shorter (5m instead of 50m) than in all other environments tested. Fortunately, we had a test screening at the location, originally meant for testing the connected screens mode. We solved this by routing xBee messages via a proxy over Wifi to the rendering PC. Our experiences here mirror those described for UbiComp in the Wild(erness) [10], where technical issues only emerged when testing in the actual environment and had to take account of e.g. weather conditions.

In-the-wild testing and development was not only necessary for technical details, but also for design aspects. For example, the typography and colours used for the splats had to be tested on-site, as lighting levels, backgrounds, and the type of projector influenced the resulting visuals. This also resulted in compromising our design ideal (a stencil-like font that resembles authentic graffiti) for readability.

Our experiences with each deployment, seeing the slingshot being handed from one participant to the next, added motivation for re-design when we were unsatisfied with aspects of the design, and provided indications of which aspects of the design worked. This could sometimes be unexpected. For example, the slightly oversized shape, resulting from having to house the electronic components, turned out to contribute positively to the user experience and strengthen the kinds of associations we desired. One person said: *"It's great that the slingshot is so big. It brings me back to childhood due to the relative scale of my hand. I feel small again now."* Without feedback like this, we might have attempted to miniaturize the device further.

Teamwork and Interdisciplinarity

The interdisciplinary process was mostly driven by design and technology leaders D1 and T1, each pushing beyond their own knowledge boundary. While software developers

T1 and T2 were accustomed to objects that can change their activity, it was fascinating for D1 to see how functions can act like material. E.g. the ‘mysterious fact’ that a physical button in combination with a microcontroller can be used for switching ‘on’ when pushed or released enhanced D1’s understanding of digital material. For D1, it was somewhat confusing, but also very appealing that form (such as the state of a button) does not have to follow function, but form can be programmed by a microprocessor. Both designers and technologists thus learned more about how the other works and thinks, and together were able to do more than when alone.

Throughout the development process D1 was trying to learn basic Arduino skills. This provided a shared communication base and a general understanding of the practical difficulties in connecting components, where even an expert often has to experiment and muddle through. Although mutual appreciation of each others’ work was supported via the Arduino platform, an assessment of the others’ workload was almost impossible. Moreover, in a DIY learning process, time management and estimates of how long things take (that are done for the first time) are generally difficult. This can create conflict. Other areas of conflicts arose when domain experts were tasked to solve a problem they were not interested in, to redo something only slightly differently, or something that required them to compromise on their design ideals. For example, D1 and D2 were asked to implement changes to the slingshot in order to improve its robustness, such as securing a wooden trigger button from falling off and raising the keypad buttons in the V3 version, so they do not get stuck under the casing. Yet increased robustness was essential for the exhibition history of the SMSlingshot, allowing us over time to focus more on observing and evaluating its use (see [7]), thus indirectly contributing to design knowledge.

Some design decisions remained solely in one domain, while others went across domain boundaries or were influenced by the collaboration. Decisions by the designers that had a clear impact on the technology side were D1’s suggestion for the organic splats (for V2) and adding sound for V3 as well as the request for Mac support. A couple of decisions were driven by prior experience with similar components. For example, while positioning the laser off-axis would have made better use of the space inside the casing and was thus desirable from a practical design standpoint, T1 knew that this would make aiming more difficult and was thus not desirable for interaction design.

The Challenges of Distributed Collaboration

For the success of the project, it was essential that in the early conceptual phase D1, D2, and T1 were in the same location, as regular discussions were needed about the main form, interaction principle, and technology. This created a common vision, while common trust had already been established through the prior collaboration on the spread.gun project. Once the team distributed over four cities in two countries, opportunities to meet in-person were rare. Besides of a common vision, having a clear task allocation was useful, as well as creative improvisation when meeting for a live intervention and putting everything together.

But the project also occasionally suffered from the lack of opportunity for close collaboration. This was especially a problem if interdisciplinary problem solving was required. The slingshot’s trigger button for releasing the virtual splat exemplifies this. The issue around the trigger button shows the boundary of domain responsibility between hardware and form design. T1 had thought that the job was done once the button was soldered to the hardware. For D1, it was finished once the case had a hole for it and a type of elastic band was chosen to trigger it. As an interim solution (for V2) a piece of wood was glued to the metal trigger of the microswitch to integrate better with the slingshot, but this failed in crucial live situations. For the V3, T1 brought up the issue again, resulting in better robustness of the button. In retrospect, both domain experts should have taken more time to sit around a table and sketch alternatives. Combining the technologist’s knowledge about the existing form factors of buttons with the designers’ knowledge of how to work with wood could have resulted in a smarter solution, rather than iterating and varying one trigger principle.

PRACTICAL RECOMMENDATIONS

It has been known [10] that deploying in the wild has special requirements and that the technology needs to be tested again and again, in isolation and as part of the overall system. Especially in DIY real-time projects like ours, where skills vary from novice to expert and are only developed over the course of the design process, the risk of failure is high. In the following, we discuss some of the practical lessons learned that might be useful for others.

To start with, we have experienced that sticking to a standard or available readymade solution is a good idea when deploying in the wild. We did not do this, and in hindsight we should have invested more time in researching available components instead of building our own solutions. Another useful strategy in DIY is copying available solutions and decomposing them. This is not only advancing the practical work, but also provides examples to learn from.

Some of our recommendations are inspired by professional and industrial practice. We recommend buying and building physical things at least three times. In software, it is a known procedure to always have three versions: a development system, a test system, and a productive system. The first is constantly under development, with bugs being removed and new features or improvements added. The productive system is deployed (in our case: during live shows), and bug fixes go directly into this version. The test system is constantly available for the entire team for testing, evaluation, not interfering with deployment or development, and in our case was useful as a backup in case the deployment hardware broke or batteries failed. For similar reasons, one should purchase a higher number of pieces than are actually needed when sourcing electronic parts. Also, for distributed work it is useful if e.g. the form designer has the required components at hand to assess their size and depth.

Moreover, to simplify repairs and repurchasing, the list of parts needed should be listed with links to sources, prices, and product code. We also recommend proper versioning, using a document that logs system changes and names the different prototypes. This allows each team member to reference these more easily when repairs have to be done or when discussing alternative technical solutions. These are practices that are easily forgotten and seem tedious, but become relevant in long-term, complex projects.

Developing a non-tethered device that works for extended periods of time differs in many ways from the kinds of functional prototypes and proofs of concepts that typically are developed in design degree programs. Regarding electronic technology, on a very practical level we learned the hard way that one should switch as early as possible to the actual power supply configuration, as changes here have knock-on effects on the overall design, from electronic design, over the code, up to form shaping (space for batteries).

Regarding form giving, we found that the main problem during early phases was having space for the cables (CAD programs are not good in modeling them). Ideally, the casing should be rather spacious, and can be tightened once hardware components are decided upon. But with every change of the technical solution, we needed to adapt the inside of the slingshot to create space again. The shape change required by the new rechargeable battery unfortunately does not allow for the standard battery pack any more, making the slingshot reliant on one set of batteries. Ideally, one should develop the casing so as to fit different versions of the hardware, allowing for a backup solution.

SUCCESS AND DILEMMA

The redesign of the spread.gun into the SMSlingshot created a stronger image that better communicated the power difference between corporations and real estate owners who have the control over Media Façades and the normal person on the street. The romantic image of David versus Goliath was a welcome analogy that could be drawn between the user of the SMSlingshot and the Façade owner. Furthermore, the slingshot carried associations of child's play. Over the course of more than 3 years, we have perfected both the physical design of the slingshot and its technical working, repeatedly iterating various aspects.

With the SMSlingshot almost all design decisions were carefully considered and improvements integrated gradually. We consider it a sign of success that a number of copy versions of the SMSlingshot have emerged. These often remove design aspects that we consider part of what makes our project a refined 'product'. Fig. 7 (right) shows a DIY production from a Palestinian student group that liked our vision of free speech and reclaiming public space. Other copies (un-authorized) were generated by agencies that used it in a brand campaign. Both copies reverted design decisions we had identified very early, probably to simplify construction. E.g. they used simple, non-organic forms, employed a touchscreen (Fig. 7 left) as keypad or a standard plastic keypad (Fig. 7 middle).

This shows how the spirit of Open Source can create a dilemma. On the one hand we want to support other DIY amateurs recreating the project by providing CAD files and by making the software available, so people that share our vision can extend and improve it. On the other hand, this also makes it even easier for agencies to misuse our vision commercially. Media artists have always had this problem, aspects of their work (which often is meant to comment on and criticize society) being copied without permission.

DISCUSSION AND CONCLUSION

We have provided a longitudinal case study of a development process that employed a design-in-the-wild approach. This multi-faceted development process of form, soft- and hardware was inspired without a specific user need in mind. Typical DIY projects are motivated by: Expression of the self and creativity, learning new skills, creating things one cannot buy, solving problems and challenging oneself, saving money [12], and to experiment with new technology possibilities that can change and even disrupt behavior [15]. However, the process presented here was infused with expert knowledge, while simultaneously members of the team over time transitioned from novice to expert, walking the line of novice and expert amateur, with growing expertise allowing for quicker re-development.

Different from most media arts projects, we used Arduino not just as a simple interface or sensor node to transfer data to a PC, but created a complex embedded device that required learning electronics at a component level, rather than using pre-made modules. Our case study shows that the Arduino platform supports novice amateurs in learning by doing up to a point where the prototype becomes product-near. Fast 'in the wild' deployments supported this drive towards product-nearness, because high degrees of robustness were demanded from the environment. Furthermore, this case showed that interdisciplinary DIY projects delegate tasks naturally to different persons, but at the same time design decisions can be made by the technologist and

Figure 7 (left) Smart phone used as readymade technology, (middle) obvious space problems compromising design, (right) authorized student project copy.

technical decisions may originate from the designer's desires. A problem then might be that these tasks may feel boring for the other half of the team, do not challenge them, or they don't see the benefit of investing the effort. Sometimes, new suggestions can spark 'sweet spots' that people are happy to pursue, as in our case where D2 suggested using rechargeable batteries, or when T1 suggested exploring how real splats look like.

While distributed development is not an ideal case, our project illustrates that it is possible, provided time is invested early-on for an intense face-to-face ideation and practical work phase that generates a shared vision and creates trust. Our experiences also demonstrate the limits of distributed working, as it is important to identify whether a problem (which may become apparent at a later stage) crosses disciplinary boundaries, as solving this may again require a phase of collocated work.

In DIY projects there are always makeshift solutions due to a lack of knowledge, time, money, available parts, etc. This is especially important when coping with early and frequent deployments in the wild, which often require pragmatic solutions. From a science or engineering perspective, this might be considered unsatisfactory, even unprofessional. But the ability to improvise and work around problems constitutes valuable knowledge and experience that is essential when the situation might change anyway with each new 'wild' deployment. Knowing how to adapt and modify an urban setting in which the technology is not going to work in order to create a workable state is a skill all members of the team acquired that will help them to consider influences of urban spaces in future designs. This includes working around problems for now and becoming an expert later. But where does expertise start and the novice end in an amateur process? We have seen here that members can move between these stages. We need to point out that there are hard boundaries, where amateurs cannot push further on. These include professional manufacturing processes that require expensive equipment. In the industrial design domain these processes tend to open up with new rapid prototyping techniques offered as services. However, in hardware most amateur projects are blocked when patents or miniaturization to the silicon level are approached. In this project, the natural boundary was reached when switching to manual reflow soldering required specific manual skills and tools. Of course these tasks can be outsourced, but if money is short a DIY project is left to improvisation.

The vision that was created at a very early stage and is also embedded in the team's name "VR/Urban", fostered identification and self expression throughout the project. This resulted in pushing personal knowledge, skill, motivation, and boundaries in the technological as well as the design domain that made the final product the novelty it became.

In this case study we have illustrated how designing for the wild, and partially in the wild, drove our design process concerning the concept design, visual and shape as well as technological design and made us push boundaries, becoming expert amateurs. From our experience, we give a range of recommendations for others embarking on similar projects, ranging from the technical, the organizational, to the processual. Similar to [10] we found that flexibility and responsiveness of the design and deployment process were important in order to respond to new requirements emerging from in-the-wild deployment in ever changing settings. Don't be afraid of being an amateur.

ACKNOWLEDGMENTS

We thank Sebastian Piatza, Thilo Hoffmann, and early stage supporters and curators Susa Pop, Miriam Struppek.

REFERENCES

[1] Arduino. http://www.arduino.cc, 2012.

[2] Böhmer, M., et al. The Mighty Un-touchables – Creating Playful Engagement on Media Façades. Proc. of Mobile HCI 2011, ACM, 605 - 610.

[3] Chamberlain, A., et al. Research in the wild: understanding 'in the wild' approaches to design and development. Proc. of DIS 2012, ACM, 795 - 796.

[4] Cohn, M. *Succeeding with Agile: Software Development Using Scrum*. Addison Wesley, 2009.

[5] Debord, G. *Society of the Spectacle*. Rebel Press, 1992.

[6] Djajadiningrat, T., et al. Tangible products: redressing the balance between appearance and action. *PUC*, 8(5), 294 - 309 (2004).

[7] Fischer, P. T. and Hornecker, E. Urban HCI: Spatial Aspects in the Design of Shared Encounters for Media Facades. Proc. of CHI'12 2012, ACM, 307 - 316.

[8] Fischer, P. T., et al. VR/Urban: Spread.gun - Design Process and Challenges in Developing a Shared Encounter for Media Façades. Proc. of British HCI 2010, ACM, 289 - 298

[9] Graffiti Research Lab. http://graffitiresearchlab.com/, 2012.

[10] Harris, E., et al. From snark to park: lessons learnt moving pervasive experiences from indoors to outdoors. Proc. of OzCHI'04 2004, Australian Computer Society, Inc.,

[11] Hummels, C., et al. Move to get moved: a search for methods, tools and knowledge. *PUC*, 11(8), 677 - 690 (2007).

[12] Kuznetsov, S. and Paulos, E. Rise of the Expert Amateur: DIY Projects, Communities, and Culture. Proc. of NordiChi 2010, ACM, 295 - 304

[13] Paulos, E. 2012. *You amateur!* Interactions 19, 1, 52 - 57.

[14] Pinhanez, C. and Podlaseck, M. To Frame or Not to Frame: The Role and Design of Frameless Displays in Ubiquitous Applications. Proc. of UbiComp 2005, Springer, 340-357.

[15] Rogers, Y. 2011. *Interaction Design Gone Wild: Striving for Wild Theory*. interactions 18, 4, 58 - 62.

[16] Williams, A., et al. 2012. *Research with a Hacker Ethos: What DIY Means for Tangible Interaction Research*. interactions 19, 2, 14 - 19.