INTERACTIVE TECHNOLOGY DESIGN AT THE DELFT UNIVERSITY OF TECHNOLOGY – A COURSE ABOUT HOW TO DESIGN INTERACTIVE PRODUCTS

Walter A APRILE and Aadjan VAN DER HELM
Delft University of Technology, Industrial Design Engineering, Studiolab

ABSTRACT
We present the Interactive Technology Design course as a possible model of interaction design education in design schools. Interactive Technology Design is a mandatory course for the master students of the Design for Interaction programme at the Delft University of Technology. The course emphasizes a hands-on approach to interaction design and teaches the required technological skills in an applied manner. The course runs through five iteration steps of progressively longer duration, each one concluded by a prototype. The iterations produce progressively more refined prototypes, and each one is characterized by its focus on a specific aspect of the design, following the rationale that beginning designers are aided by separation of concerns and solving one set of problems at a time. The course acquires its briefs from companies or from researchers in the university. We show examples of course output, both final and intermediate, and we describe the lessons we have learned in running multiple iterations of this course.

Keywords: Interaction design, industrial design, experience design, interactive sketching, Arduino, Max5, tinkering, design

1 THE INTERACTIVE TECHNOLOGY DESIGN COURSE IN CONTEXT

The Interactive Technology Design (ITD) course is a mandatory component of the Design for Interaction (DFI) master at the Industrial Design Engineering (IDE) faculty of the Delft University of Technology. Broadly speaking, the IDE approach to design builds heavily on scientific and design research literature, user research and involvement at every step and clearly defined processes. There is no insistence on the student's development of his/her own signature "style"; on the other hand, teamwork is prevalent and adherence to rules, guidelines and deadlines is rewarded. DFI focuses on the way in which people and products or services or product/service systems interact [1] - in the remainder of this article we will make no distinction between products, services and systems: all will be referred to as "product". DFI is by necessity a multidisciplinary master that upholds experience design ([2],[3]) as a way to approach the design process.

In ITD students create concepts for interactive products that contain computer technology as a coupling layer between the controls (user interface) and the mechanism triggering some desired function (Frens 2006). This coupling layer, that we may well call "the interaction" should be included in conceptualization next to product form, controls and feedback mechanisms. In the traditional product design process[4] there are insufficient provisions to consider the design of interactive products. Alternative approaches from [5] and [3], propose a highly iterative design process that allow for concepts to grow by making experiential prototypes (also called sketches in the early stages of the design). Prototypes let the designer communicate the concept to the design team and give him insight on how well the designed features of the interactive product concept match the design brief.

2 COURSE DESIGN CHALLENGES

Technology is used and explored in multiple courses inside DFI, but it comes to front stage in the ITD. Technology education for designers poses difficulties due to the fact that designers are not intrinsically motivated by technology itself, as would be the case for engineers. Put in another way, a computer scientist is interested in the C programming language per se, or perhaps in C as opposed to LISP or Python, while a typical DFI student is interested in C as a tool for e.g. controlling the Arduino[6] prototyping platform to explore interactive objects in the home. The DFI student will, typically, treat technology like a buffet lunch and learn exactly as much as he needs of any given technology, aided in this by the availability of Q&A format online resources such as FAQs and forums. This on-demand style of learning does not, however, lead to elegant and maintainable technological solutions; we have seen students write thousand-line programs as one single large function.

Figure 1. An achieved ITD prototype engages real users in a realistic environment (in this case, the home)

The challenge is, in brief, to teach technology and design at the same time. Of course, this is not a new challenge: traces of it can be seen even in original Bauhaus education programme. We take a [7] constructionist approach, defined by Papert [8] as "a view of learning as a reconstruction rather than as a transmission of knowledge […] learning is most effective when part of an activity the learner experiences as constructing a meaningful product." We integrate the strictly constructionist approach with some elements of top-down, structured teaching for concepts we consider basic, such as voltage, resistance and abstraction hierarchy.

ITD has a typical intake of 120 students. Since individual supervision and individual projects are not practical on this scale, we form students into teams of 4 to 6.

3 COURSE STRUCTURE

The course lasts a whole semester, one day per week for twenty calendar weeks. We take in briefs from a variety of sources (that we call *brief holders*). Briefs originating from outside the Delft University of Technology may come from NGOs (such as the one we have previously described in [7]), government agencies and private companies: internally sourced briefs come from faculty and doctoral students. Internal briefs are research oriented, and produce work that can be the basis for a publication. We aim to have five to seven briefs that are distributed among all teams. Each brief engages two to five student teams. We develop the brief together with the brief holder who may contribute some money to partially cover prototyping material and expenses. The key contribution of the brief holder, however, is feedback to the student teams during and after the course. One brief we received from a company (Philips) asked students to explore the very broad concept of "hospitality" in the context of lighting. A very different one came from the TNO, the Netherlands Organization for Applied Scientific Research: the students were asked to design an intervention in a public space that would promote physical activities among young people. Lastly, a research brief originating from our own lab asked the students to design a toolkit for interactive design, designed by designers for designers, as opposed to designed by engineers for designer. This brief resulted in a paper, [9], presented at the Tangible and Embedded Interaction conference. After a kick-off presentation, the students are distributed into teams, and each team receives its brief. The course consists of five design iterations of increasing length, each one producing a prototype with a different focus. The five iterations can be summarized under the form of assignments given to the students

1. Hack: capture one aspect of the solution in a prototype made from waste and recycled material
2. Autonomy: make a prototype that does at least one thing on its own
3. Nut cracking: show that you can solve the hardest technological challenge in your solution
4. Users: perform an informal, inspirational user test and redesign based on results
5. Integration: make and display in an exhibition an experiential interactive prototype

The steps have been designed with the objective of arriving at remarkable concept and interactive prototypes at the end, while avoiding a set of pitfalls that we have observed over the years.

3.1 Hack

The hack iteration aims at getting the creative juices of the students flowing. Its implicit message is also that a prototype is limited and focused on some specific aspect, constrained by resources and easily disposed of. The students are provided with a large collection of found objects, to be used to "design and make something from found objects and existing technology, incorporating one concern from your design brief. The something may be a game or a toy or even something useful."

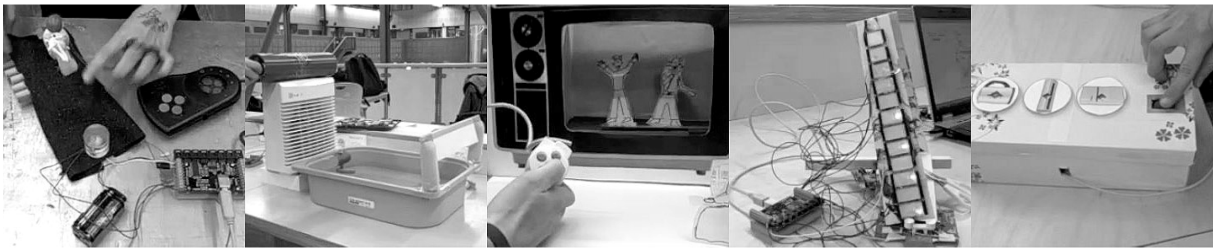

Figure 2. Hack prototypes made in just a few hours. Students are encouraged to be as rough as possible.

This first iteration should incorporate at least some of the brief's main concerns. The time allotted is just four hours, an intentionally tight timeframe. In a ha-ha-only-serious way, students are forbidden to brainstorm. We aim for a rush of energy and improvisation where all the energy is dedicated to finding the right hacking material and hacking it into something that fits the brief. Students also make a simple one minute long video that will be shared among all teams.

Figure 3. This successful hack exploits a cut-up plastic cup both as a spring and as a light diffuser

In the design critique we give at this stage we aim to expose overly complex or non-realistic concepts and at giving students a first taste of the technologies we propose.

3.2 Autonomy

In the autonomy iteration students are asked to make a prototype of their solution that exhibits a measure of autonomous behaviour, described as "doing, on its own and under software control, something fundamental to its intended interaction with the user". We encourage students to use the Wizard of Oz technique for some parts of the prototype behaviour. We suggest roles that the students may take up inside the group, such as Manager, Reporter and Technologist. This creates an additional socially-driven motivation [10] for the individual student. Students are encouraged to try on different roles and explore tools that are new to them..

3.3 Nutcracking

The name of this iteration comes from the old adage that suggest cracking the hardest nut first. Teams are asked to identify the most difficult technological challenge in their concept (which, at this point, should already be reasonably defined) and build a prototype that demonstrates their ability to solve that challenge. If the nut is not cracked, students are strongly advised to rethink their concept.

In our experience, two of the ways that interactive technology projects can go wrong is that the designers either delay the nutcracking until the very end, which produces a prototype that falls short of the concept, or choose nuts that are too tough to crack, which instead produces a technology demonstrator. Of course a technology demonstrator is a very respectable result for a course in engineering! But in our view of design education, technology is a means to an end, not an end in itself. As an additional deliverable, we ask students to produce two A5 postcards that describe respectively their concept and the technological architecture of the system they are envisioning.

3.4 Users

This iteration is not a formal user test, but rather "discount" usability engineering [9]. We do not aim at scientific validation of the design, but rather at finding the sorest in the design or in its implementation. We suggest that the students ask themselves these questions: does your design provide clear interaction cues to the user? Are the actions and goals appreciated by the users? Does your design keep the users interested? Does your design work with multiple users?

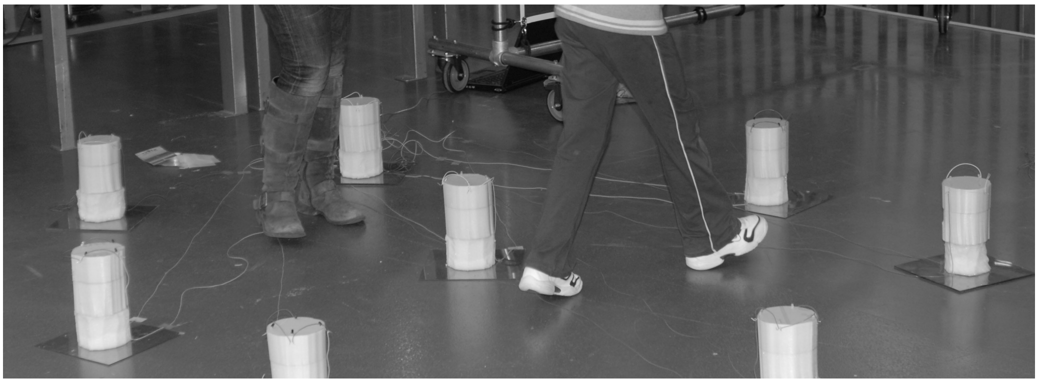

Figure 4. This users iteration prototype is functional. The materials and the finish are still missing.

In a somewhat lateral way, the user test is also a chance for the brief holder to see an approximation of the final output. At the end of the "Users" iteration, the concept should be locked in.

3.5 Integration

The objective of this final step is to produce an interactive prototype that is exhibited and used during the final day of ITD. We organize a public event and invite hundreds of visitors from inside and outside the university. The public nature of the event means that the prototype must be made understandable to someone who has no familiarity with the brief, who is not a designer and who is willing to dedicate at most five minutes to it. This, in turn, makes exhibition design and the construction of a believable setting for the prototype a very important task inside this step.

Figure 5. The final prototype should be interactive and functional for the duration of the exhibition

Students start this step by developing a final prototype plan. This is a detailed description of the prototype making process that outlines what materials are going to be purchased, what services (such as laser cutting) need to be procured and what type of assistance is required from the ID-Studiolab and ITD staff. The plan must also include a timeline with milestones, a SWOT analysis of the project team and detailed information about how the team is planning to document its work. Students need to plan and produce artefacts like photographs, renderings and videos so that they can refer to their work in the future.

This final Integration step in the ITD course has multiple outcomes: the students get raw direct feedback from the users after going through the process of making their ideas understandable. The staff and the brief holders have an opportunity for networking, plus the reaction of the visitors act as an additional element for choosing projects to take further.

4 TECHNOLOGICAL PLATFORMS

Toolkits have a strong influence on creative output [11], and we have discussed elsewhere [9] design work towards an interactive prototyping platform that fits the specific needs of interaction designers. Students are exposed to both Arduino and Max5: we advise them to use Max5 for quick interaction sketches that can be tethered to a PC, and to use Arduino for embedding into portable objects.

Ambitious students progress to the use of local wireless network technologies like ZigBee [11].

We have to take into account that our students have very basic skills in electronics and software engineering; moreover, the focus of the course is design rather than engineering technology. To address these issues we provide two initial technology toolkits.

4.1 Max5 with Phidgets

Max5 is a visual programming development environment with many functionalities, eg. sensor and actuator control, video and audio processing and 3D capabilities. Its visual programming nature makes it relatively easy for a non-programmer to get started. The rich set of tools at the disposal of a user makes it possible to include many advanced technologies in the developing concept. Max5 can be accompanied by the Phidgets[12] collection of sensors and actuators. The combination of Max5 plus Phidgets is very effective at lowering the entrance barrier to prototyping interactive products: in our experience, students start making simple interactive sketches within hours of being introduced to the environment. For the effective introduction of Max5, we recommend that the course coordinator should build beforehand a library of Phidgets components and cabling. One common desire of industrial designers particularly in the Hack phase, is to control existing hardware. Small loads can be powered directly by Phidgets, but for bigger loads we advise the use of opto-isolators. It goes without saying that any activity that involves dangerous currents or voltages should be closely supervised and all prototypes should be carefully inspected by staff familiar with electrical safety.

4.2 Arduino

Arduino[6] is an open hardware microcontroller platform designed with non-technologists in mind. It comes with an open source development environment that puts a simple face on a relatively complex and unfriendly tool chain. The platform has been around for some years now; a large on-line community supports a budding Arduino developer with example projects and custom hardware add-ons that can be easily purchased as a kit or preassembled part. The development is done in C, a classic procedural language from the Sixties, very efficient on limited hardware but harder than contemporary languages like JavaScript or Python. Arduino boards are cheap enough that they can be loaned out in the tens. The same hardware used to connect Phidgets to bigger loads can be used for Arduino.

If the concept goes in a direction where portability is an important factor, we steer students towards Arduino. Designs that involve audio and video processing fit Max quite naturally. Moreover, students who have the technological ability to explore different platform (such as Python or OpenFrameworks) are encouraged to do so, with the proviso that the staff does not guarantee support.

4.3 Other platform elements

Arduino and Max5 material gets distributed to the students group who, moreover, have access to a small electronics laboratory with a helpful staff and the general prototyping facilities of the Industrial Design Engineering faculty, including metalworking and woodworking labs. Rapid prototyping and

CNC machining grow more important every year. These technologies, together with CAD, allow the students to build structural components and believable enclosures for their projects.

5 LEARNING OUTCOMES

We have mentioned that the objective of ITD is to teach students how to create concepts in a group for interactive products manifested as interactive experiential prototypes. This objective places technology in a clear ancillary role. The course structure, based on groupwork, tends to obscure individual contributions and, in particular, makes it impossible to evaluate the degree to which students have learned prototyping technologies. In learning terms, the consequence is that a student may go through ITD without ever writing a line of code or touching a soldering iron, if she so wishes. This is not an inadvertent consequence: placing objective requirements on technology learning would shift the focus away from "learning to design" to "learning techniques". While the course has a big space for techniques of all sorts, something that we personally delight in, it remains a design course in a faculty of Design.

6 CONCLUSIONS

We have presented the ITD model of interactive technology education for designers. We have seen this aggressively prototyping oriented approach produce good designs that have resulted in publications and products. We believe that this same approach could be used for other adjacent domains that do not have a physical base like web application design or mobile application development.

REFERENCES

[1] Stappers, P., P. Hekkert, and D. Keyson, *Design for interaction: consolidating the user-centered design focus in industrial design engineering*, in *International Conference On Engineering and Product Design Education*. 2007: Northhumbria University, Newcastle upon Tyne, United Kingdom. p. 69-74.

[2] Schifferstein, H.N.J. and P. Hekkert, *Product experience*. 2007: Elsevier Science Ltd.

[3] Buxton, W., *Sketching user experience – Getting the design right and the right design*. 2007, San Francisco, CA: Morgan Kaufmann.

[4] Muller, W., *Order and meaning in design*. 2001: Purdue University Press.

[5] Frens, J.W. and C. Overbeeke, *Designing for rich interaction: integrating form, interaction, and function*. 2006.

[6] Mellis, D.A., et al. *Arduino: An Open Electronics Prototyping Platform*. in *CHI 2007*. 2007. San Josè, USA: CHI.

[7] van der Helm, A., W. Aprile, and D. Keyson, *Experience Design for Interactive Products: Designing Technology Augmented Urban Playgrounds for Girls*. PSYCHNOLOGY JOURNAL, 2008. **6**(2): p. 173-188.

[8] Papert, S., *Constructionism: A new opportunity for elementary science education*. 1986: Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group.

[9] Aprile, W. and A. van der Helm, *Designer Driven Interaction Toolkits*, in *TEI'11, Work-in-Progress Workshop*, E. Yi-Luen Do, M.D. Gross, and I. Oakley, Editors. 2011: Madeira, Portugal. p. 91-96.

[10] Woolfolk, A., P.H. Winne, and N.E. Perry, *Social cognitive and constructivist views of learning.*, in *Educational Psychology*. 2009, Pearson Canda: Toronto. p. 329-370.

[11] Greenberg, S., *Toolkits and interface creativity*. Multimedia Tools and Applications, 2007. **32**(2): p. 139-159.

[12] Marquardt, N. and S. Greenberg, *Shared Phidgets: A Toolkit for Rapidly Prototyping Distributed Physical User Interfaces*. Proceedings of Tangible and Embedded Interaction (TEI), 2007: p. 13-20.