# PROBLEM-BASED TEACHING OF INTERACTION DESIGN TO FRESH MINDS – LESSONS FROM INTEGRATING EXPERIENTIAL INTERACTIVE PROTOTYPING IN A COURSE ON INTERACTION DESIGN AND USER EXPERIENCE RESEARCH

**Aadjan van der Helm[1], Tomasz Jaskiewicz[1], Natalia Romero Herrera[1]**

*[1]Delft University of Technology (NETHERLANDS)*

## Abstract

It is a big challenge for second year design students to learn how to perform user research, while designing and prototyping a product service system, which involves not only screen-based, but also physical interactions. Yet, such challenges are becoming increasingly common in design practice, for which students need to be well prepared. This paper presents the teaching process, methods and techniques of the Physical Prototyping track in the Interaction and Electronics course, which introduces Interaction Design to students of the Industrial Design bachelor programme at the TUDelft. The course integrates user research, design, prototyping and evaluation activities in a single design project. Following a learning-by-doing pedagogy, the program devotes special attention to the development of prototyping competences and skills. A combination of prototyping methods and tools has been applied in the course and this paper describes them in detail. Although some shortcomings of these methods and tools can still be found, student and expert evaluations show significant success in their application in education

Keywords: Human Computer Interaction, Interaction Design, Experience Design, Prototyping, Education, Programming.

## 1 INTRODUCTION

The practice of interaction design (IxD) is concerned with design of interactive products, systems and services. In the past decades IxD has become an established discipline. While at first IxD was mostly concerned with user experiences on computer screens, in the last decades enriching user experiences through embodied interaction has become more prevalent. As a result, designed user interactions and experiences increasingly rely on product behavior. This requires an interdisciplinary and user-centered design approach, linking disciplines such as contextual research, user experience design, computer science, electronics, and communication design [1].

A suitable approach to teach product design students is Design Based Learning [2]. This approach is grounded in constructivist learning theory and involves students in projects where they iteratively reflect on hands-on experiences. However, when applying this to IxD, stronger emphasis is needed for design processes to be user-centered, utilizing quick prototyping to making designs experiential, supporting a cross-disciplinary approach and working with coaches that are experienced professionals [3-7].

In the Industrial Design bachelor programme of the TUDelft, the course Interaction and Electronics (IE) introduces students to IxD. Upto several years ago the course was mainly concerned with teaching the design of on-screen interactions. The design of embodied interactions was taught in fundamental way in a disconnected practical. The disconnection resulted in a strong bias in the minds of the students towards designing for screen-based products. The focus of this paper is to present how we transformed the IE course to better teach the skills of prototyping embodied interactions. To ultimately improve students awareness that embodied interaction can make a design more meaningful for users.

To achieve this we transformed the IE course in a two-step approach. In the first step we introduced the Physical Prototyping (PP) track as the activity to teach about embodied interaction. In the second step we changed the design brief for the IE course to include both on-screen interactions on a touch screen and embodied interaction in a separate physical device.

The main challenge in the PP track is teaching students to quickly prototype in a sketchy way. This involves the design of a user interface from sensors and actuators (electronics engineering) and the implementation of behavior in the product (software engineering) in order to be able to test it. As such the PP activity has to bring electronics and programming skills to the students.

In the next section we will briefly outline the Industrial Design bachelor to establish the skill level of the students when they enroll for the IE course. Then we will present the overall structure of the IE course. Lastly we present in more detail the Physical Prototyping track and approach we use to obtain a higher relevance towards designing for embodied interaction in the student population.

## 2 TEACHING INTERACTION DESIGN AT TUDELFT

The Industrial Design Engineering Bachelor aims to educate students in the design of products and systems with a perspective on the user central to their design process. It is a comprehensive and broad program educating the product designers of the future. The yearly influx of students is approximately 300 students The program spans 3 years with large product development courses (PD courses) and courses teaching multidisciplinary knowledge and skills [8].

| | **Quarter 1** | **Quarter 2** | **Quarter 3** | **Quarter 4** |
|---|---|---|---|---|
| **Year 1** | People and Products | Design and Experience | Business, Culture and Technology | Mechanical Engineering Design |
| | PD1-Introduction IDE | **Products in Working** | PD2-Concept Design | Research and Design |
| **Year 2** | **Product in Action** | Industrial Manufacturing | Technical product Optimization | Modeling and Simulation |
| | Strategic Product Innovation | PD3-Fuzzy Front End | **Interaction and Electronics** | PD4 Embodiment and Detail Design |
| **Year 3** | Minor | | Optional Course 1 | Bachelor Final Project |
| | | | Optional Course 2 | |

Figure 1. Overview of the IDE bachelor curriculum.

The courses Product in Working,Product in Action and Interaction and Electronics (IE) form the learning path to build skills on working with electronics and programming. Upon entry in the bachelor program students buy an Arduino starter kit that is first employed to introduce the Arduino and electronics components in Product in Working. These skills are then extended upon in Product in Action to gain skills on constructing electronics circuits with sensors and actuators. Interaction and Electronics aims to further extend skills on programming product behavior.

As can been seen in Figure 1, the IE course is taught in the third quarter of the second year.

### 2.1 Overview of Interaction and Electronics

The IE course runs for a full quarter (10 weeks) with 250 students working in teams of 5. In order to cover a wide spectrum of IxD applications, we have introduced a brief for a Personal Informatics (PI) system consisting of an interactive touch screen and a physical device. This brief involves user requirements, design and prototyping activities of both devices.

IE consists of three tracks. The main track is the Design Project (DP) that involves all activities related to designing the PI system. The two prototyping activities lead to user tests for evaluation and support communication about the concept to classmates and coaches.

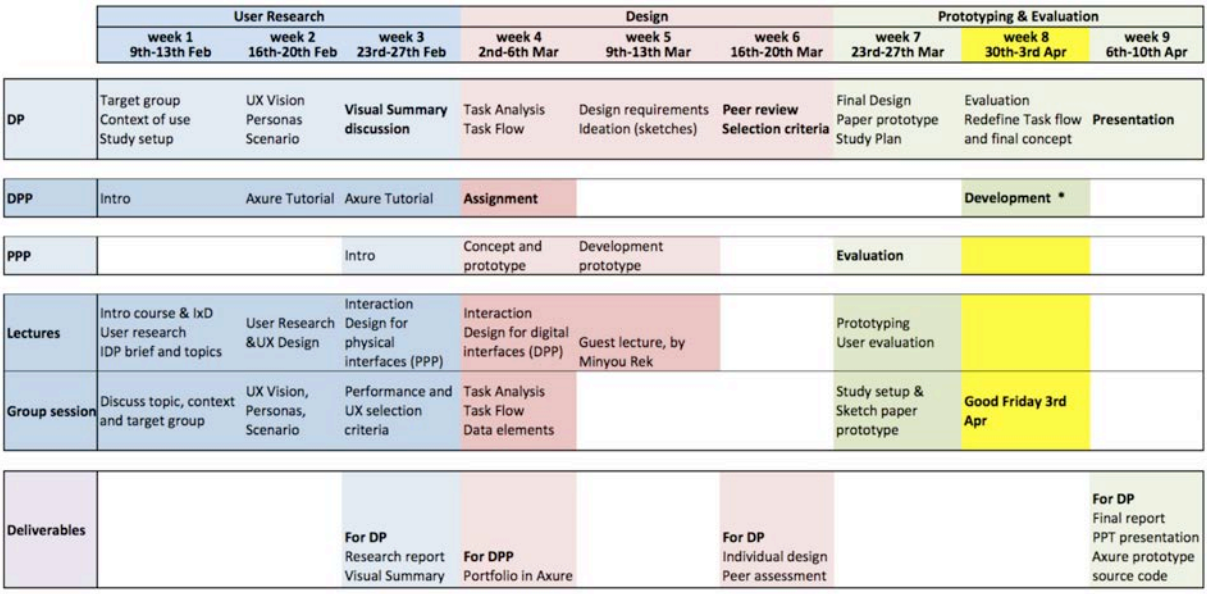
With respect to the DBL dimensions as presented [9], IE can further be characterized (see Table 2).

| | | |
|---|---|---|
| Project characteristics | Authentic | Students are assigned to studios with a theme e.g. healthcare, energy consumption awareness, mobility (roughly 80 students per theme). They have to venture into a context relevant to their theme and find opportunities for design themselves. |

| | | |
|---|---|---|
| | Open-ended | End presentations in the PP and DP tracks show multiple solutions of PI systems within a theme. Thus showing there are alternatives designs possible. |
| | Hands-on | There are multiple experiential activities scheduled in the IE course: visiting context to find design opportunity, making a prototype for the physical device, prototyping the touch screen based informative device and making movies of the envisioned use of the PI system. Working through these activities involve the students in multiple design iterations towards the concept they present at the end of the course. |
| Design Elements | The lectures give the students pointers to methodology backed up by the course syllabus. During the weeks they have 1 day per week to engage in teamwork for designing activities | |
| Teacher's role | The teams meet a DP coach every week to coach on task and process. The course manual provides clear deliverables for each week to steer the process. In the PP track the coaches involved support the students in delivering a functional physical device. | |
| Assessment | During the weekly DP track coach meetings the coaches give formative feedback. At the end of the course DP coaches prepare summative assessment. This is also discussed across the themed studios before the final grade is communicated to the students. There is no formative assessment during the PP track. At the end there is a summative grade per team that is discussed with all coaches before the grade is communicated to the students | |
| Social context | All the work in IE is team-work. The student teams are encouraged to find design opportunities in a real context of use. In addition they can return there with their prototypes to evaluate the design outcome. Eg. with the paper prototype of their design for the touch screen or the functional physical device. | |

Table 2. The DBL dimensions present in IE.

User research follows by observing and interviewing the target users in context to find an opportunity for the design of a personal informatics system. Students continue to formulate a design vision, make profiles for their target user (Personas) and formulate an interaction scenario of the system. User and system requirements are further detailed thorough task flow diagrams and task analysis respectively. The design concept is prototyped both on the interactive touch screen and the interactive physical device and finally evaluated with potential users.



| | User Research | | | Design | | | Prototyping & Evaluation | | |
|---|---|---|---|---|---|---|---|---|---|
| | week 1<br>9th-13th Feb | week 2<br>16th-20th Feb | week 3<br>23rd-27th Feb | week 4<br>2nd-6th Mar | week 5<br>9th-13th Mar | week 6<br>16th-20th Mar | week 7<br>23rd-27th Mar | week 8<br>30th-3rd Apr | week 9<br>6th-10th Apr |
| DP | Target group<br>Context of use<br>Study setup | UX Vision<br>Personas<br>Scenario | **Visual Summary discussion** | Task Analysis<br>Task Flow | Design requirements<br>Ideation (sketches) | **Peer review<br>Selection criteria** | Final Design<br>Paper prototype<br>Study Plan | Evaluation<br>Redefine Task flow and final concept | **Presentation** |
| DPP | Intro | Axure Tutorial | Axure Tutorial | **Assignment** | | | | **Development** * | |
| PPP | | | Intro | Concept and prototype | Development prototype | | **Evaluation** | | |
| Lectures | Intro course & IxD<br>User research<br>IDP brief and topics | User Research &UX Design | Interaction Design for physical interfaces (PPP) | Interaction Design for digital interfaces (DPP) | Guest lecture, by Minyou Rek | | Prototyping<br>User evaluation | | |
| Group session | Discuss topic, context and target group | UX Vision,<br>Personas,<br>Scenario | Performance and UX selection criteria | Task Analysis<br>Task Flow<br>Data elements | | | Study setup & Sketch paper prototype | **Good Friday 3rd Apr** | |
| Deliverables | | | **For DP**<br>Research report<br>Visual Summary | **For DPP**<br>Portfolio in Axure | | **For DP**<br>Individual design<br>Peer assessment | | | **For DP**<br>Final report<br>PPT presentation<br>Axure prototype source code |

Figure 3. Detailed view of the IE course schedule.

(DP = Design Project, DPP = Digital Prototyping Practical, PP = Physical Prototyping)

The Digital Prototyping track (DPP) is an unsupervised activity. Students learn to work with Axure [10], a software tool to create prototypes of websites and apps. In the Design Project (DP) track students explore a first version of the app for their PI system through a paper prototype and after evaluation implement an improved version in Axure as an end deliverable for the course.

The Physical Prototyping (PP) track will be discussed in more detail in the next section.

## 2.2 Physical Prototyping Practical

When the interaction scenario has been defined in the DP track, the Physical Prototyping Practical starts. Students are asked to deliver an interaction scenario to start the design of an interactive physical device as part of the personal informatics system. The PP track consists of four half-day sessions (one per week) where students work in their teams guided by coaches and student assistants.

### *2.2.1 Session 1: Selection of Sensors and Actuators*

The team of students is asked to form two sub-teams: a design team and a technical team. In this first session, the design team develops a detailed interaction storyboard for the physical device [11, 12]. In parallel, the technical team explores the possibilities of the electronics given (Arduino and Grove system). The sub-teams discuss the design requirements and sensing and actuating possibilities to arrive at a selection of max 2 sensors and 2 actuators at the end of the session. Coaches support students in both activities.

In order to make the exploration of electronics manageable in the allotted time a pre-selection of components from the Grove system has been made. Exploring the collection involves taking a component, visiting the webpage for that component and uploading the example code to the Arduino microcontroller. The selected collection includes:

- Contextual sensors: light sensor or temperature sensor
- Input sensors: three way switch, tilt switch, button, rotary angle sensor
- Actuators: color LED, buzzer, vibration motor.

### *2.2.2 Session 2: Define a concept for the device*

In the second session the design team designs the interface and the embodiment of the device. They are supported in this activity by the deliverables from the DP track (user experience vision and personas). The deliverable of this team is a mockup of the device showing the position of the components in the embodiment and a sense of aesthetics appropriate for the target user and the context of use. The technology team creates a definition of the behavior of the physical device. The deliverable of this activity is an Acting Machine diagram that allows for the playback of the device behavior.

### *2.2.3 Session 3: Make a functional prototype*

The teams are asked to bring materials (boxes, coloured paper, etc.) and tools appropriate to construct a prototype. The design team works on constructing the prototype and at the same time the technical team implements the prototype behavior from the Acting Machine diagram. The sub-teams leverage each others activities to deal with the dependence on having access to the components – for building them into the prototype as well as testing the behavior.

### *2.2.4 Session 4: Evaluation and movie making*

In the last session the teams are asked to finish their prototype to an extent that they could evaluate its behavior (and the envisioned user experience). A deliverable for this session is a short movie to demonstrate the designed interactions ideally in a setting representative to their context of use. The interaction storyboard of the first session is used to make sure that all planned interactions are implemented in the prototype and are incorporated in the movie.

At the end of the session students present to each other and the staff the movie and reflect on the PP track activities.

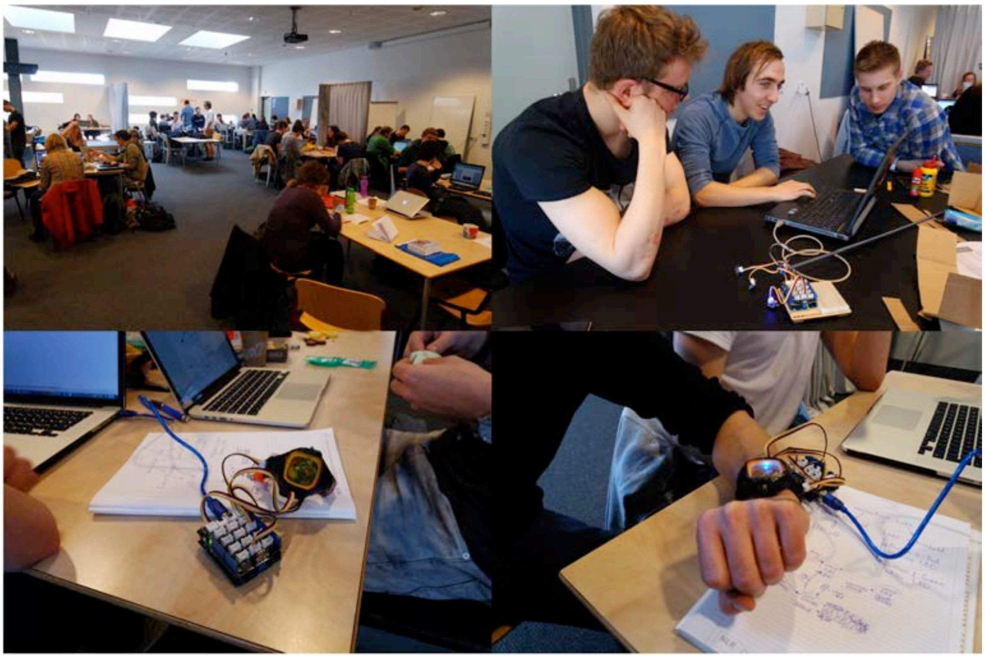


Figure 4. Impressions from the IE course. Top-left: classroom where PP track was held. Top-right: a team implementing the designed product behaviour. Bottom-left: a prototype of an interactive wearable device. Bottom-right: testing the device.

## 2.3 Choice of microcontroller and components

The field of hardware systems for the construction of a user interface from sensors and actuators is undergoing rapid development. A full discussion of this field goes beyond the scope of this paper. Systems suitable for use in the IxD process should make it possible to quickly explore components and allow for quick alterations to the prototype. Activities such as designing electronic circuits and soldering components stand in the way of this. For the PP track we make a choice for electronic components that allow for quick construction of an interface that is compatible with an Arduino.

This choice for Arduino is motivated from the technology learning path of the bachelor program leading to the IE course. But more can be said in favour of the Arduino as a platform to do prototyping in IxD. The Arduino ecology consists of an application to write code, uploading that to the microcontroller and it allows for rudimentary debugging. The Arduino API is simple and well documented from the application as well as through online resources. In addition there is a large Arduino community online that can provide assistance through forums. A multitude of websites contains examples and projects that can serve as inspiration for a new project.

In its basic configuration the Arduino should be interfaced with sensors and actuators by making small electronic circuits interfacing the input and output pins of the board. This typically involves a breadboard and lots of wires. In two ways this is not nice for designers. Firstly designers are not trained electronic engineers; the world of electronics is alien to them. Secondly, prototypes build with a circuit implemented on a breadboard are not testable in an experiential sense. Incorporating them into an embodiment is difficult, because the wires tend to pop out of the breadboard or the Arduino pins. In addition when testing the prototype the handling cause further risk of loosing integrity of the circuit. Resulting in unpredictable behaviour or even worse potential electrical shortcut damaging the components or microcontroller.

For this reason, a modular sensor and actuator system is adopted. Currently the optimal choice for this course is the Grove System [13]. The Grove System consists of an Arduino shield, i.e. a small PCB that can be placed on top of the Arduino microcontroller and then offers a number of input and output connectors for a unified cable system to connect both sensors and actuators. The Grove System has a large collection of sensors and actuators. In the IE course we do not teach electronics topics beyond gaining familiarity with the above mentioned collection of sensors and actuators. The rationale for making available a limited collection is to steer for sketchy prototypes.

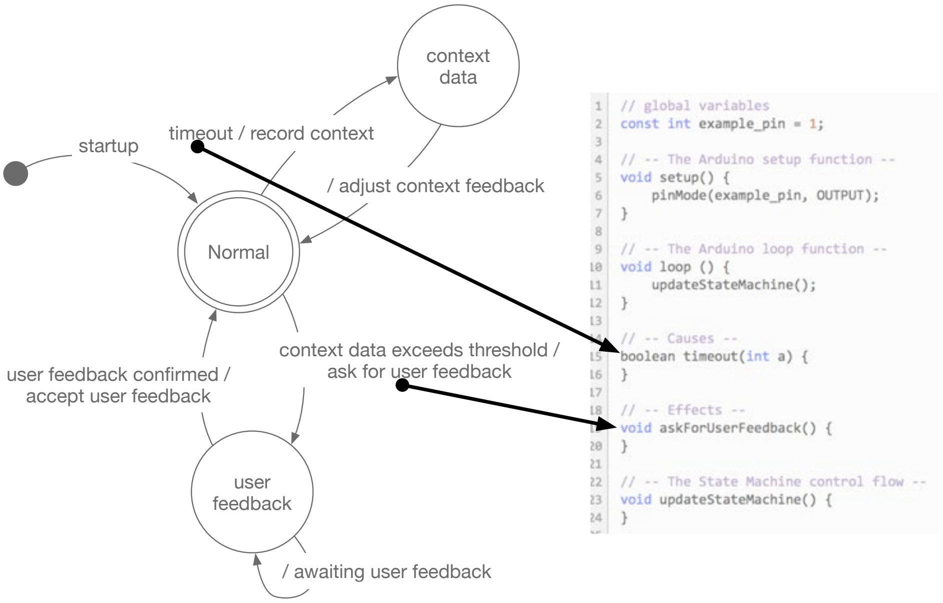
The typical modus operandi for students creating prototypes with Arduino is to search online for examples related to the hardware they want to interface and copy the found Arduino example code into the Arduino application to try it themselves. This works well if tasks are simple e.g. trying to work with a LED component. Students run into problems when they need to define their designed unique behaviour interfacing multiple components, as the program complexity increases when merging multiple examples.

## 2.4 Programming product behavior

With respect to teaching programming to fresh minds there is large body of work (for an extensive taxonomy of this field we refer to [14]), however little has been written about the specific programming needs for interaction designers [15].

We observed that teams typically delegate one or two students to do the programming, a complex and cognitive heavy task for novice programmers. Because the programmers expect to work on it for a long time they often start before the envisioned behavior has been properly decided upon. While coding they are unaware of other activities in the team and vice versa. This can negatively impact the outcome when the team forgets to update each other of changing plans during the design process. Eg. when the design team decides to change the interface because it is more appropriate to use a slider instead of a button. Or the programmers discover technical possibilities that the design team is not made aware of.

In order to support interaction designers to program behavior in the prototype we developed the Acting Machine approach [16]. The Acting Machine (see Figure 5) is derived from software engineering where the concept of state machines is often used especially in the fields of embedded computing and real time systems [17]. Our Acting Machine approach is similar to the State Machine Framework as presented in [18].


context data
startup
timeout / record context
/ adjust context feedback
Normal
context data exceeds threshold / ask for user feedback
user feedback confirmed / accept user feedback
user feedback
/ awaiting user feedback

```
// global variables
const int example_pin = 1;

// -- The Arduino setup function --
void setup() {
    pinMode(example_pin, OUTPUT);
}

// -- The Arduino loop function --
void loop () {
    updateStateMachine();
}

// -- Causes --
boolean timeout(int a) {
}

// -- Effects --
void askForUserFeedback() {
}

// -- The State Machine control flow --
void updateStateMachine() {
}
```

Figure 5. An Acting machine diagram and an excerpt of the generated Arduino code. The arrows show how conditions and actions in the diagram relate to their corresponding functions.

Teams are encouraged to first define the behavior of the interactive product in an Acting Machine diagram on paper. The diagram can be verified by using a (physical) pawn to resemble the current state and move that through all defined states upon acting out the user input and the resulting product behavior (this resembles playing a board game). When an error or incomplete behavior is detected the diagram on paper can readily be modified. By first defining the behavior in a diagram on paper, students can quickly iterate to reach the envisioned behavior.

Once the Acting Machine diagram is verified for the expected variation of user inputs, the programming code to implement the behavior can easily be obtained through applying standard methods from state machine theory. A full explanation goes beyond the scope of this paper, it is important to highlight that the Acting Machine diagram actually has all the control logic embedded in the diagram. We provide an application called Machino to our students that takes a diagram and transcribes that to code for the Arduino but the transcribing can also be done manually.

Our method differs form the State Machine Framework in two ways. Firstly, we take the Acting Machine to be a Mealy type state machine: a state machine where conditions and actions are specified explicitly on the transition arrows moving from state to state. Secondly, we transcribe the Acting Machine diagram directly to Arduino code. The resulting code has three parts.

- **Conditions**: Upon user input a change in the sensor signal might cause the Acting machine to activate a transition. Each condition is separately modeled as a function in the resulting code. Typically in the condition function the programmer implements checks for button presses or sensor signals passing a threshold.
- **Actions**: When a condition evaluates true an actuator might be controlled to show the effect of user input. Each defined action is also modeled as a unique function in the resulting code. Typically in the action function the programmer implements feedback of the device e.g. turning a LED or a vibration motor on.
- **State machine logic**: The control flow logic of the behavior moving from state to state over time is encapsulated in one function with a switch statement. The programmer typically does not have to change this code.

To deliver a working prototype with the envisioned product behavior implement, the remaining task is to write short code snippets for implementing the conditions and actions i.e. for interfacing the sensors and actuators. Thus the complexity is reduced to simple piecemeal searching for example Arduino code to interface a single component and copy/pasting that in the right function for condition or action (See Figure 5). A task most novice programmers can easily manage.

# 3 RESULTS

We first present our observations about the transformation process to bring the IE course to the form described in this paper. Subsequently we discuss the outcome of running the IE course for the first time in the year 2015.

## 3.1 About the transformation process

Since the inception of the IE course in the bachelor programme, the skill of programming product behavior was taught in a fundamental approach working with small practical assignments to interface more complex components such as LED segment displays and LED-drivers. While this approach in itself delivered skills in the students we observed the students did not often apply these skills to prototyping product behavior after the IE course was followed.

In a three-year transition period we aimed to transform the IE course to improve the adoption to prototype embodied interaction. First we introduced the Physical Prototyping track to explore how with modular hardware we could teach about prototyping product behavior. We created a kit for a simple product for photographers, a slave flash that can be triggered to flash for off-camera lighting effects. Students were asked to customize the interface and embodiment of this device. Taught in this form we reached learning goals about prototyping embodied interaction. We saw many interesting and creative prototypes. However, there was still no connection to the main design brief in the DP track, which at this time was still solely centered around designing on-screen interfaces.

To obtain true integration we redesigned all tracks in IE around the design brief of a PI system consisting of a touch screen application and a physical device with possibilities to combine teaching about screen interaction and embodied interaction.

## 3.2 Analysis of the course survey

The course has been evaluated before and after the transformation in a survey among participating students. 77% of students (169 persons) responded in 2014 and 80% (197 persons) in 2015. Additionally, feedback from coaches guiding the students has been collected in both cases. In the survey the PP track was evaluated using several statements responded to using a Likert scale. As seen in figure 6, in 2014 the statement “the (physical prototyping) practical contributes to the content of the course” scored 2.77 on the Likert scale, with 17,2% of students marking the “fully disagree” option and 27,8% marking “disagree". In 2015 the same question received a score of 3,5, with 45,7% of the answers indicating agreement with the statement.

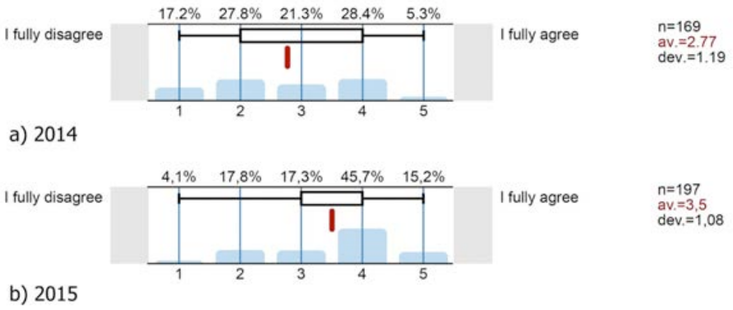


Figure 6. Responses to the statement "the (physical prototyping) practical contributes to the content of the course" show a substantial improvement of the way in which the physical prototype supports the DP design assignment from 2014 to 2015

Another statement "the instructor presents the material in a clear and structured way", has also seen a visible improvement from 2014 to 2015, with the score rising from 2,95 to 3,25, as shown in figure 7. Other statements were less relevant for the transformation of the course and have seen only slight improvements.

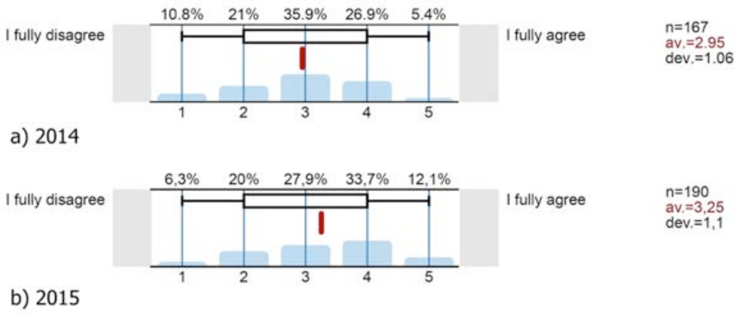


Figure 7. Responses to the statement "the instructor presents the material in a clear and structured way" show that documentation, support material, and (indirectly) prototyping tools were easier to use for the students in 2015.

## 3.3 Connecting the PP track to the DP design brief

The additional comments in the survey of 2015 provided by some of the students indicated concern about the link between the PP track and the DP design brief in both 2014 and 2015. However, there was a noticeable difference regarding the merit of those comments. In 2014 many comments questioned the sole purpose of the PP track in the course in general, while in 2015 they were pointing out shortcomings in the way that the PP track was integrated with the main assignments. Several comments indicated that communication between PP track coaches and DP track coaches was lacking.

In 2014 most coaches in the DP track were not aware of the work done in the PP track, as the PP assignment was very loosely connected to the main design assignment. The verbal comments from the coaches evaluating the student work in 2015 were generally positive. Some coaches struggled with understanding the new PI design brief, and still lacked insights into PP track activities. However, coaches saw an increase in the overall quality and complexity of concepts. Especially in the experiential qualities that the physical prototypes have added to the end results. The majority of coaches suggested further integration of the PP track with the DP track, which emphasizes that the qualities of the prototypes and their purpose in the design process were acknowledged and appreciated.

## 3.4 Issues related to designing the physical device

Additionally, other comments in the survey 2015 also indicated that the choice of electronic components was too limiting and that course materials clarity was not helping to build satisfactory prototypes. This shows a significant change, since rather than questioning the very purpose of prototyping, students were expressing frustration (sometimes oriented at course teachers) with not being able to prototype well enough for it to be useful for them.

Students have also encountered problems with using the Grove System. Many students had to compromise their choice of sensors or actuators due to limited availability of components. This has caused some frustration, even though it should not have coincided with the learning goal of the PP track, which was to learn prototyping, not necessarily fully matching the DP track brief.

Some teams had designed physical interaction in systems embedded in an environment rather than an object. This resulted in prototypes that were built as scale models that made it impossible to test the product experience.

Lastly, some teams took the PP track as the final product design cycle. Their aim was to make the physical device a viable manufacturing option, In this sense they misunderstood the purpose of the prototyping activity, which was more directed towards generating insight into the functioning of a PI system as a combination of a data rich display and a data collecting device.

## 3.5 Using the Acting Machine but not coding a state machine

Analyzing the material delivered by the PP students in 2015, 29 teams out of 46 were capable of delivering an Acting Machine diagram, while only 12 out of 46 teams actually coded according to the state machine paradigm. This shows that the majority of students did not understand how to use the Acting Machine approach to support the entire transition from diagram to code, which clearly indicates that improvements are needed. We see three possible explanations.

Firstly, some teams had a team member capable of implementing the behavior of the physical device. In these teams there was a tendency to produce the Acting Machine diagram as an afterthought. The diagram did not have the effect of being a roadmap into the code that often times did implement similar product behavior.

Secondly, the coaches involved with the PP track were not all able to help the teams with translating described behavior into Acting Machine diagrams. This resulted into teams skipping the diagramming step and moving straight to coding the behavior. This sometimes caused problems down the line when a novice programming team member had spend hours on making it partially working but found it impossible to implement all of the required product behavior.

Lastly, in the DP track teams are required to do a task analysis and draw a task flow diagram for the interaction with the touch screen display. We observed some hybrid forms of task flow diagram and Acting Machine diagrams. However, if the Acting Machine diagram does not adhere to the Mealy form of state machines (passive states and transitions with both conditions and actions explicitly defined), the Arduino code cannot readily be generated.

# 4 LESSONS LEARNED

We have succeeded in improving the integration of the PP track and the DP track by reducing the complexity of the programming through state machines, reusable examples and modular electronics. Generally, students realized the potential experiential interactive prototypes have for embodied interaction design.

Many students still struggle with the above, and clearly further improvements are needed a) Acting Machines shouldn't become too complex b) Arduino and the Grove System have limitations c) there are some bugs or unsupported features (e.g. Machino is currently not running on Windows) that create barriers for students d) these limitations can be better communicated and ambitious students can be pointed to ways to overcome them.

We thought reducing the number of available Grove System components would help, but it was too constraining for students.

Generally, coaches support stronger involvement of interactive prototypes in the DP track.

The lessons learned show us that the direction is good. Tools and methods are working, although they need some improvements. However, more integration of the PP track and the DP track will benefit the prototyping learning objectives.

In order to establish that we reached the goal of delivering prototyping of embodied interactions as a skill to students, we need to find ways to determine if students are more readily resorting to prototyping in activities after they completed the IE course.

## 5 ACKNOWLEDGEMENTS

Teaching at the TUDelft is a team effort. We are indebted to many colleagues helping us to define, organize and deliver the course material. But we would like to acknowledge the students first. Students who enthusiastically absorb what we prepare and make the results shine. With respect to the theory about prototyping for design and the concept of the Acting Machine we acknowledge Pieter Jan Stappers who has been advocating state machines in so many designerly ways. Patrick Pijnappel who developed Machino. Lastly we acknowledge our colleagues Martin Havranek, Martin Verwaal and Adrie Kooijman and the student assistants who have been involved with the IE course in the past years for their valuable input and effort to deliver the skills of experiential prototyping with Arduino.

## REFERENCES

[1] Saffer, D., Designing for interaction: creating innovative applications and devices. 2009: New Riders.

[2] Puente, S.M.G., M. van Eijck, and W. Jochems, A sampled literature review of design-based learning approaches: a search for key characteristics. International Journal of Technology and Design Education, 2013. **23**(3): p. 717-732.

[3] Wang, S.J., P. Moriarty, and S.-Z. Wu, Tangible Interaction Design: Preparing Future Designers for the Needs of Industrial Innovation. Procedia Technology, 2015. **20**: p. 162-169.

[4] Qian, Z.C., S. Visser, and Y.V. Chen. Integrating user experience research into industrial design education: interaction design program in Purdue. in NCIIA 15th conference. 2011.

[5] Sas, C. Learning approaches for teaching interaction design. in HCI educators workshop. 2006.

[6] Klemmer, S.R., B. Verplank, and W. Ju. Teaching embodied interaction design practice. in Proceedings of the 2005 conference on Designing for User eXperience. 2005. AIGA: American Institute of Graphic Arts.

[7] Buxton, B., Sketching user experiences: getting the design right and the right design: getting the design right and the right design. 2010: Morgan Kaufmann.

[8] Roozenburg, N., E. van Breemen, and S. Mooy, A competency-directed curriculum for industrial design engineering. A Grand Day Out: Empathic Approaches to Design, 2008.

[9] Gómez Puente, S.M., M. van Eijck, and W. Jochems, Professional development for design-based learning in engineering education: a case study. European Journal of Engineering Education, 2015. **40**(1): p. 14-31.

[10] Axure, R., Interactive wireframe software and mockup tool. 2012.

[11] Van der Lelie, C., The value of storyboards in the product design process. Personal and ubiquitous computing, 2006. **10**(2-3): p. 159-162.

[12] Van der Lugt, R., C. Postma, and P.J. Stappers, Photoboarding: Exploring service interactions with acting-out and storyboarding. Touchpoint: the journal of service design, 4 (2) 2012, 2012.

[13] SeeedStudio. The Grove System - a modular system for assembling electronics. 2015 [cited 2015; Available from: http://www.seeedstudio.com/wiki/Grove_System.

[14] Kelleher, C. and R. Pausch, Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. ACM Computing Surveys (CSUR), 2005. **37**(2): p. 83-137.

[15] Paauwe, R., W. Aprile, and A. Van der Helm, Teaching Interaction Designers To Sketch In Software. EDULEARN12 Proceedings, 2012: p. 1398-1405.

[16] Pijnappel, P.I.R., Programming for Interactive Prototypes, in M.Sc. thesis, D.U.o. Technology, Editor. 2013: Delft.

[17] Wagner, F., et al., Modeling software with finite state machines: a practical approach. 2006: CRC Press.

[18] himbleby, H., Press on: principles of interaction programming. 2010: The MIT Press.