

Allocated Flow Diagramming: A Structured Process and Methods for Teaching Interactive Product Prototyping in Industrial Design

Yong-Gyun Ghim*

School of Design, University of Cincinnati, Cincinnati, USA

Abstract

Background Interactive product prototyping has been widely taught in industrial design studio courses to prepare design students in dealing with dynamic interactions of using digital products. While it requires the integration of design and technology, the lack of systematic approaches of the studio curricula creates a great challenge for students who are not familiar with technology in creating interactive prototypes successfully.

Methods A theoretical framework is developed from the human-machine interaction model and the sequential flow of task analysis methods. Based on this framework, a human-machine interaction (HMI) flow diagram is proposed with which interaction elements, relationships, and flows are specified and categorized. The flow diagram is utilized to create a storyboard and an allocation flowchart subsequently. A structured design process based on these methods was applied to an interactive product design studio course in an undergraduate industrial design program.

Results Students created working prototypes of interactive products, following the structured design process consisting of an HMI flow diagram, a storyboard, and an allocation flowchart. 104 students (87.4%) among 119 succeeded in translating their ideas to functioning circuits. 74 students (62.2%) successfully integrated electronics components into working physical prototypes while 30 students (25.2%) just made the circuits work.

Conclusions The structured design process makes a transition from initial ideation to programming smooth and incremental. This helps students without previous programming experience to understand the logic flow and to develop algorithms. The HMI flow diagram is useful for analysis, concept development, and the specification of interactions, and helps students to grasp the effect of interactive features on user experience.

Keywords Interactive Product, Industrial Design, Prototyping, Task Flow, Design Education

*Corresponding author: Yong-Gyun Ghim (ghimyn@ucmail.uc.edu)

Citation: Ghim, Y. G. (2021). Allocated Flow Diagramming: A Structured Process and Methods for Teaching Interactive Product Prototyping in Industrial Design. *Archives of Design Research*, 34(4), 7-21.

<http://dx.doi.org/10.15187/adr.2021.11.34.4.7>

Received : Apr. 22. 2021 ; **Reviewed :** Jul. 20. 2021 ; **Accepted :** Jul. 28. 2021

pISSN 1226-8046 **eISSN** 2288-2987

Copyright : This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>), which permits unrestricted educational and non-commercial use, provided the original work is properly cited.

1. Introduction

Design education is undergoing a rapid change in its content and pedagogy (Mubin et al., 2017; Voute et al., 2020). As human life gets surrounded by intelligent products, such as smart devices and robots, the role of industrial designers has expanded from creating a product's physical construct with static attributes to dealing with a comprehensive product and service experience and dynamic interactions. The recent exponential growth in smart technologies and their application to everyday products further necessitates the fundamental change in industrial design education. Over the past two decades, industrial design (ID) programs in universities and design schools around the world have responded to this new technological advancement by employing various subjects that were not traditionally taught at ID programs but originate from Human-Computer Interaction (HCI) (Tung & Deng, 2004; Bud & Wang, 2017; Mubin et al., 2017), such as user-centered design, physical computing, and user interface design. The interactive product design studio is one of such attempts to combine technology and design as well as digital and physical, with students exploring new possibilities enhanced by digital products (Kolko, 2004).

But researchers and educators have observed the challenges and limitations of these attempts (Oygür Ilhan & Karapars, 2019). For example, ever-increasing needs to cover both breadth of topics and depth of skills (Weil & Mayfield, 2020), falling short of strategies and formal methods with studio courses (Wärnestål, 2016; Oygür Ilhan & Karapars, 2019), and design students' lacking motivation and knowledge of technology (Nam, 2006; Aprile & Van Der Helm, 2011; Kim, 2020) are some of them. Interactive product design involves integrating multiple elements from both sides of design and technology (Stappers et al., 2007; Aprile & Van Der Helm, 2011; Hummels et al., 2011), i.e., ideation, form-giving, programming, electronics, and physical models. While a studio course itself does not provide ample time to cover the wide spectrum of these elements thoroughly, the very nature of the ID program covering multiple topics, knowledge, and skills (Nam, 2006) further limits students' chances of working on interactive products during their undergraduate program. At the same time, studio courses lack structures and methods for teaching interactive product design (Wärnestål & Lindqvist, 2013; Oygür Ilhan & Karapars, 2019) while the HCI aspect of interactive product design requires a more robust and analytic approach to problem-solving as evidenced by various frameworks, methods and tools in HCI (Preece et al., 2002). The fact that design students normally take such studio courses without prior experiences in electronics or programming (Mikkonen, 2012) further amplifies the difficulties both students and instructors face during the course. In short, the time and depth of learning on this subject is not sufficient enough to make students fully understand interactive product design and absorb its principles and methods. Consequently, students struggle a lot with their projects, which leads to the compromised quality of their design outcomes, frustration, and superficial understanding of the subject.

This paper seeks a way of helping students better navigate through the challenges they will face in the interactive product design studio, particularly with grasping the underlying principles needed for developing interactive design concepts and translating the concepts to

programming. As the lack of a structured process for these stages makes design students' dealing with technologies harder (Mikkonen, 2012), streamlining the process with analytic methods will clarify the thought process and provide a helpful base for programming. Accordingly, we will examine interaction models and task analysis methods from HCI and flowcharts from computer science, then propose a structured design process that can be easily applied to the workflow of interactive prototyping projects in industrial design studio courses. This approach is further examined through its application to an interactive product design studio course in a university in the United States. While the range of interactive products is quite broad, this study limits the scope of interactive products to tangible and physical objects with embedded sensory or tactile interfaces for relevance to ID programs. Software applications and connected devices are excluded so that the core principles of interaction can be addressed effectively to novice students in a limited time.

2. Interactive Product Design: Process and Methods

The focus of the interactive product design studio is not to be laid on teaching technical skills like programming and electronics, but rather on letting students understand the principles of human-machine interaction and apply them to solving design problems. Without a clear process and structured methods that are weaved into a design process, there lies a danger of letting students struggle during the project, easy to get lost in the details of technical issues (Koskinen & Mikkonen, 2014). In this section, a theoretical framework is examined, then a design process and the subsequent methods are further discussed.

2. 1. Theoretical framework of designing interactive products

Interactive products have two distinctive characteristics in comparison to static products: human-machine interaction and a temporal sequence. For example, users give inputs to products through their physical actions and interpret the products' operations by mental processes. Interactive products perceive inputs through sensors, whether they are from users or by environmental changes, and respond through actuators or emitters. This exchange of action and communication happens over the course of time in a sequential manner to achieve an intended goal of the product's usage. Based on these two aspects, a framework can be constructed to aid design students with understanding and designing interactive products.

First, an interaction system can be modeled by examining actors and underlying components. Interaction means "the relation in use between a product and its user mediated by an interface" (Frens, 2006). This indicates human (user) and machine (product) as two separate actors involved in the use of interactive products. Thanks to the development of sensor technologies and algorithms, machines are becoming more capable of interpreting human actions and communications as well as perceiving and responding to their surrounding environment. The machine is an active participant in interactions that both actors should be examined distinctively and equally. The human side in human-machine interaction consists of three parts, the *perceptual* system, the *cognitive* system, and the *motor* system (Card et al., 1983), and each of them corresponds to sensors, processors, and actuators on the machine

side respectively. Dumas et al. (2009), in their multimodal man-machine interaction model, illustrate how information flows from human to machine (or vice versa) through the four sequential states of *decision (computation)*, *action*, *perception*, and *interpretation*. Here, *action* from one side is transmitted to *perception* on the other side through an interface, and *cognition* from Card et al. is separated into *decision* and *interpretation*. However, for an easy application of the human-machine system model to studio courses, so that students are not burdened with the complexity of the model, this paper simplifies the model to consist of the mental processes (*perception* and *cognition*) and the physical actions (*action*) for both human and machine sides. For the convenience of communication to students, the mental processes and the physical actions are replaced with *cognition* and *action* for the human side, and *sensing* and *operation* for the machine side respectively (Figure 1).

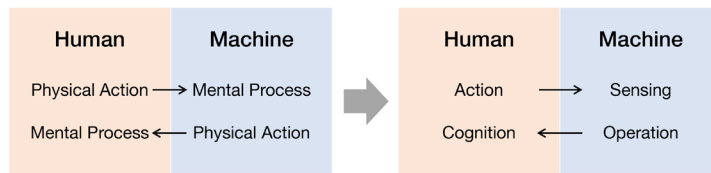


Figure 1 Human-machine interaction model

Second, interactivity entails changes of actions and states as well as a continuous flow of information between human, machine, and the environment. The user undertakes a sequence of activities to achieve a goal with using the product while the product continues to respond to the user's actions and outside conditions. So, the whole interaction should be examined from the perspective of time sequence rather than a static viewpoint. Task analysis is a tool to identify the activities of a system, more specifically cognitive processes and physical actions, by decomposing them into a sequence of smaller steps and decisions. (Hackos & Redish, 1998; Preece et al., 2002; Norman, 2016). Hence, task analysis methods, such as task flow diagrams and allocation charts (Maguire, 1998) can guide the designers to understand the current situation, assess problem areas, and design a new interface and system.

So, categorizing interaction components into the mental processes and the physical actions under each of the human and the machine side, then arranging them in a sequence will help students not only with understanding the hidden mechanism below the interface surface but also with developing ideas, then translating them into programming.

2. 2. Design Process

Design studio courses should provide carefully planned “right progression steps” to keep students motivated (Wärnestål, 2016). In computer science, a systematic approach to programming based on a clear procedure and structured flowcharts is discussed to deal with teaching programming to students (Scanlan, 1989; Crews & Ziegler, 1998). In particular, Crews and Ziegler lay out a five-step procedure to programming with 1) problem analysis, 2) solution planning, 3) algorithm construction, 4) implementation, and 5) testing and debugging. Here, algorithm construction involves constructing flowcharts. This procedure overlaps with the design process normally used in interactive prototyping. For example, Tung and Deng (2004) propose a procedure for interactive product design by integrating the three

interaction design areas of information, interactive, and sensorial design into the industrial design process of analysis, creation, and execution respectively. Steps 3 to 5 from Crews and Ziegler (1998) can be categorized under the execution, while each of steps 1 and 2 directly matches the analysis and the creation phase respectively. During the analysis phase, a task flow (here the term *flowchart* from Tung and Deng is replaced with *task flow* to prevent confusion with a flowchart used for programming) is used for the analysis of the product and its target users. This task flow is converted to a storyboard during the creation phase, with which designers visualize their ideas and predict user activities. The execution phase involves producing physical prototypes to check physical properties and screen prototypes to simulate user interaction. Here, “the operational logic and path” is emphasized in designing interactive products, so the task flow and the storyboard are mentioned as important tools of the design process. However, when moving from storyboarding to prototyping, this analytical approach based on a sequential flow is not further elaborated, lacking an explanation of the step for creating a working digital prototype. On the other hand, Mikkonen (2012) explores using flowchart-based cards to help develop working prototypes and ease communication between designers and engineers. Through the analysis of student projects, it is argued that flowcharts provide a visual and structured means to develop interaction ideas and translate them to code for interactive prototyping, which is beneficial to design students without prior knowledge in programming. Both studies from Tung and Deng (2004) and Mikkonen (2012) can complement each other for a structured design process with visual tools designated for each step. The design process and its ensuing tasks and tools discussed so far are summarized in Table 1. As the focus of this paper is on a structured process for developing ideas and translating them to programming, the discovery phase before analysis as well as physical prototyping and the refinement phase (Wärnestål, 2016) are left out in Table 1.

Table 1 Design process and tools for interactive product design

Phase	Analysis	Creation	Execution
<i>Design tasks</i>	Product, user analysis Problem definition	Ideation Visualization and prediction	Prototyping Documentation
<i>Programming tasks</i>	Problem analysis	Solution planning	Algorithm construction Algorithm implementation Testing and debugging
<i>Tools</i>	Task flow	Storyboard	Flowchart

2. 3. Design Tools

As mentioned in the previous section, the operation of an interactive product can be analyzed from interactions between human and machine, with each side further divided into mental processes and physical actions. A sequential flow of user actions and machine operations also plays an important role in the analysis and design of an interactive system. Based on this interaction framework as well as the design process shown in Table 1, this paper develops a three-step structured process for interactive product prototyping: 1) analysis and ideation through a human-machine interaction flow diagram, 2) storyboarding for visualization and concept development, and 3) flowcharting for programming. Figure 2 illustrates this process overlaid on the overall design process of the studio (Wärnestål, 2016). Each step is further explained below in detail.

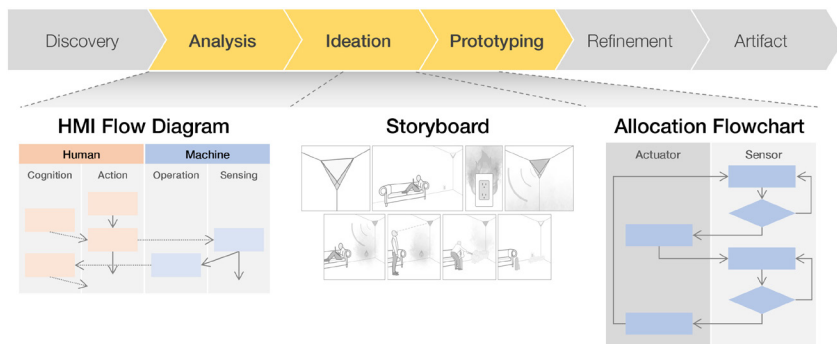


Figure 2 A structured process for interactive product prototyping

2. 3. 1. Human–Machine Interaction Flow Diagram

The first step is constructing a human-machine interaction (HMI) flow diagram. An easy start is creating a simple activity diagram about using a reference product without dividing it into actors or process parts. By analyzing the diagram, students can point out pain points or opportunities. Based on their initial insights, students explore ideas through iterations of sketching. Then the initial diagram is expanded to a more detailed interaction flow diagram, now separating human and machine parts, decomposing tasks into more detailed sub-tasks, and assigning them under cognition, action, operation, and sensing with blocks. All the blocks must be arranged in a temporal sequence and positioned in relation to corresponding blocks in other columns (Figure 3). Contrary to task flow diagrams used in other studies (Tung & Deng, 2004), the HMI flow diagram looks into specific elements of interaction by combining the human-machine interaction model and the task allocation chart from Maguire (1998). This diagramming helps students get a clear understanding of the interaction sequence and necessary electronic components that otherwise remain uncertain until after starting to work on electronics, which may cause reworking and delays.

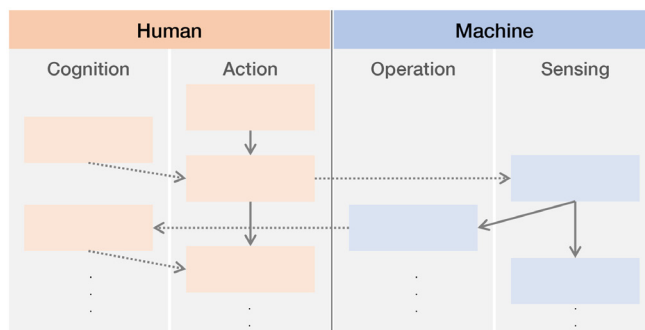


Figure 3 Human-machine interaction (HMI) flow diagram

2. 3. 2. Storyboard

A storyboard is an efficient and effective tool to explore, clarify, and communicate ideas in the early stage of design development (Budd & Wang, 2017) and helps design students better understand the context of interaction and discover opportunities (Tung & Deng, 2004). Idea sketches from conventional ID studios capture product attributes at a certain moment, failing to communicate the context and sequence of use and interaction. On the other hand, storyboards illustrate “the new design in the context of users, tasks, and environments”

(Hackos & Redish, 1998). They are effective at sequencing interactions and clarifying the interfaces and their effects. Major interaction steps are to be selected from the HMI flow diagram and translated to a storyboard. As the interaction sequence has been already defined from the HMI flow diagram, students can move from one step to the next easily. This procedure also enables students to specify the types of interface components and their effects through visualization. Figure 4 illustrates the procedure of creating a storyboard by selecting major steps from the HMI flow diagram and translating them into visualized scenes.

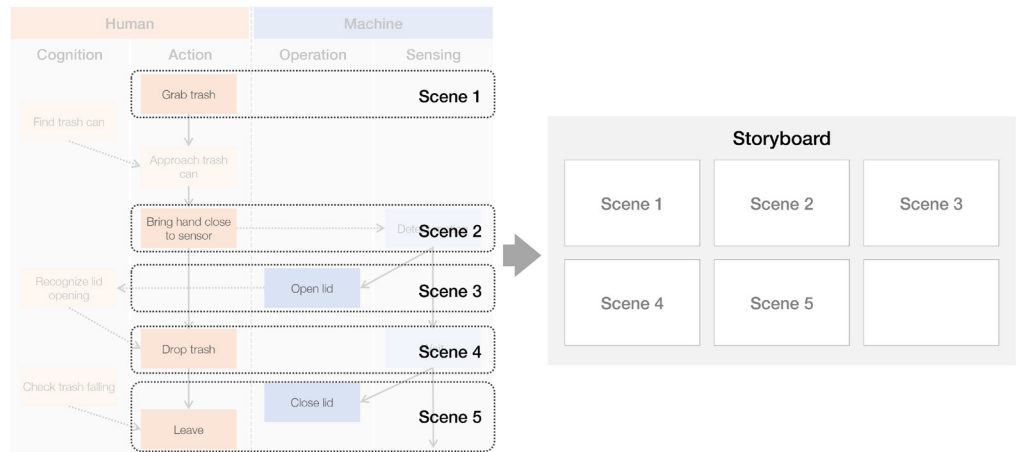


Figure 4 Storyboard from the HMI flow diagram

2. 3. 3. Allocation Flowchart

Flowcharts have been widely used in teaching programming and algorithms to computer science students (Xinogalos, 2013). Thanks to their visual clarity and minimal use of syntax, flowcharts have been an effective tool to novice programmers (Crews & Ziegler, 1998), and the same can be applied to design students with their interactive product prototyping for defining the logic flow of algorithms and specifying a product’s inputs, outputs, and processes. A flowchart’s basic form consists of functional boxes for operations to be carried out, predicative diamonds for decision making, and arrows to specify logic flows from one point to the next (Böhm & Jacopini, 1966; Mikkonen, 2012). Contrary to conventional flowcharts, this paper proposes an allocation flowchart based on the task allocation chart (Maguire, 1998), where the flowchart is divided into two parts, each part representing *operation* and *sensing* of the machine from the HMI flow diagram described earlier. For design students who lack programming experience, this allocation approach allows an easy translation from the interaction flow to the flowchart and to programming. First, after removing the human part from the HMI flow diagram, each column of the machine part is converted to the same corresponding column of algorithms: the operation column is converted to the actuator and the sensing to the sensor. So, the flowchart’s overall spatial layout remains the same as the HMI flow diagram, which further eases the mental complexity of building an algorithm. Second, blocks from the HMI flow diagram are converted to boxes and diamonds with specific functions and variables suitable for the machine’s understanding, and arrows are modified to fit the flowchart convention (Figure 5).

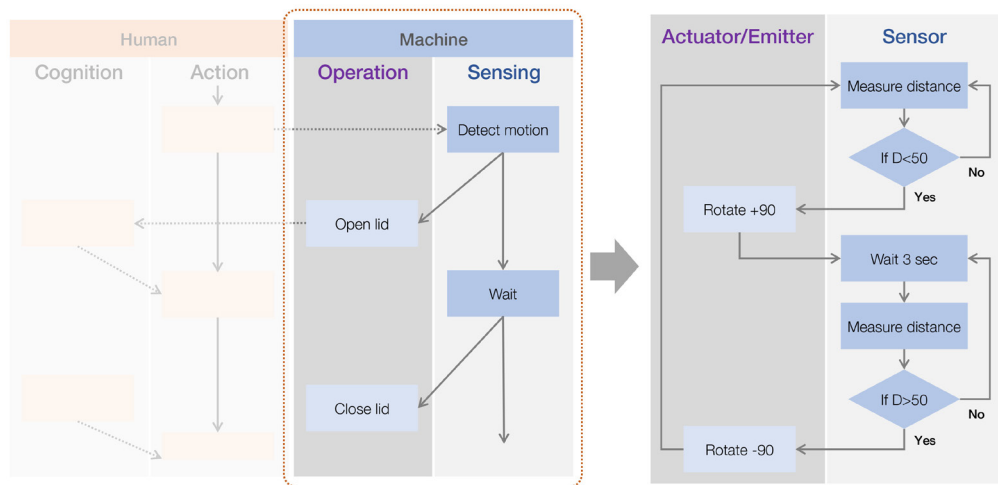


Figure 5 Allocation flowchart from the HMI flow diagram

3. Project Intelligence

Since 2019, the industrial design program at the University of Cincinnati's School of Design has been running its Fall design studio for sophomores as a collection of three five-week intensive modules. Sections of twenty students rotate through three different projects of their choice throughout the semester, dealing with a variety of design issues and learning new methods and skills for each project. One of the modules is Project Intelligence where students get introduced to HCI and work on an interaction design project for the first time during their five-year industrial design program. Students are asked to create a new user experience of simple everyday objects by incorporating interactive technology into physical objects. This new experience is materialized and communicated through a physical interactive prototype in which an Arduino board with sensors and actuators is integrated into desirable form and materials.

Because of the short project period, the focus is placed on ideation and implementation through physical prototyping rather than heavy contextual research. After brief analyses of the current products, relevant technology, and user tasks, students spend the first two weeks on initial idea exploration and iteration. Students quickly move from the combination of their first-hand experience of reference products and cognitive walkthroughs to simple activity diagrams with which the workflow is described and analyzed, then followed by idea sketches. Initial ideas are further specified through HMI flow diagrams, and storyboards are extracted from the diagrams. By the end of the second week, most of the students get their design concepts clarified concerning the main functionality, overall appearance, and interaction. Figures 6 to 8 show an example of the HMI flow diagram, the storyboard, and the allocation chart respectively from a student's project of designing a night light. In particular, Figure 6 shows how the student extracted pain points from the task analysis of the current situation (left) and then specified a new interaction flow to solve the problem by drawing the HMI flow diagram (right).

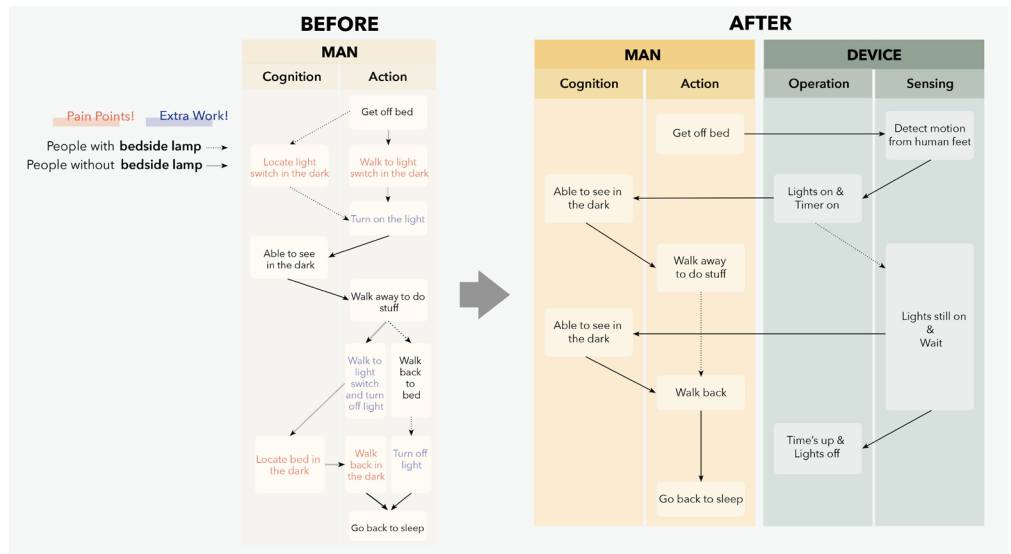


Figure 6 HMI flow diagram: a night light



Figure 7 Storyboard: a night light

From the third week, students start building prototypes while further specifying the interaction flow, form factors, usability, and electronic components: an allocation flowchart is created based on the previous HMI flow diagram (Figure 8); a rough prototype is built from easily accessible materials like foam core, cardboard, or clay to check the form factor and basic usability; an initial CAD model is built with internal components placed inside the model. At the end of the third week, students start working on electronics and programming.

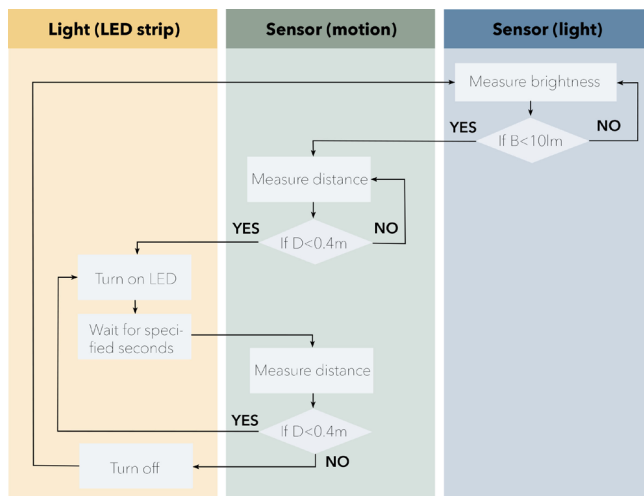


Figure 8 Allocation flowchart: a night light

The fourth week involves testing and refinement dealing with multiple prototyping aspects concurrently. Students continue to work with Arduino, checking whether their selected sensors and actuators offer the desired interactions and effects. Students come across various issues at this point and they have to figure out additional materials and/or methods to implement their concepts to working prototypes. Based on the latest findings and changes, they refine the CAD models to be ready for 3D printing and adjust the electronics. During the final week, students put together all the components into working prototypes, present them to class, and document the whole process. Figure 9 shows a student's prototyping process.

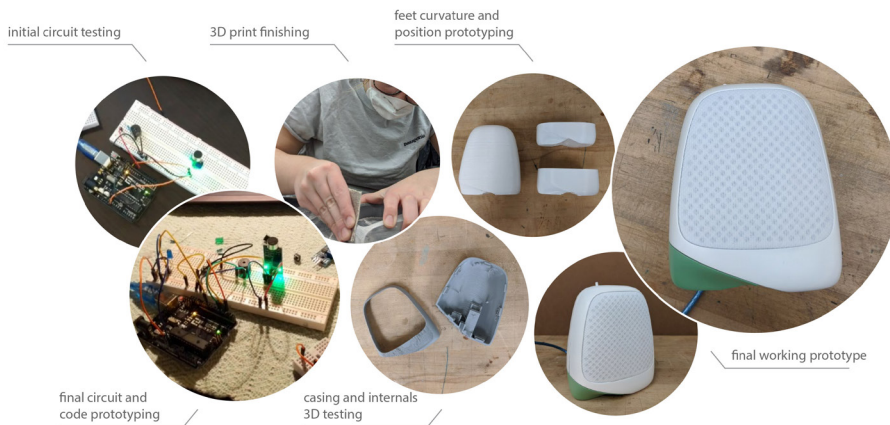


Figure 9 Baby soothing device – final prototype and the prototyping process

4. Results

A total of 119 students participated in the Project Intelligence studio from 2019 to 2020. Students developed interactive design concepts and created functioning physical prototypes following the design process described in the previous section. The range of design

outcomes from the studio covers a wide spectrum, from practical products for everyday use (e.g., a smart dog bowl, a self-watering planter, a smart trash can, etc.) to informative or communication devices (e.g., an informative food container; a bike lock alarm; a wrist band to relieve anxiety, etc.), and to toys for children’s emotional development. Table 2 shows the results showing the number and percentage of students who either completed making functional prototypes or made only the electronic parts work without succeeding in integration. There is no significant difference between the year 2019 and 2020, so it is safe to combine the results from both years. Among 119 students, 74 students (62.2%) successfully integrated digital components into functioning physical prototypes and 30 students (25.2%) managed to show only the electronic parts work. When combined, a total of 104 students (87.4%) succeeded in translating their ideas to functioning code and electronics.

Table 2 Result of students’ prototyping accomplishment

	Studio enrollment	(A) Students who completed working prototypes	(B) Students who made only circuits work	(A+B)
Year 2019	62	38 (61.3%)	15 (24.2%)	53 (85.5%)
Year 2020	57	36 (63.2%)	15 (26.3%)	51 (89.5%)
Total	119	74 (62.2%)	30 (25.2%)	104 (87.4%)

In general, most of the students followed this flow-based process well. But there were a few students who either skipped the HMI flow diagram or did not draw it properly: 5 students among 57 in 2020. The same was observed with the allocation flowchart, with a slight increase in numbers: 8 students in 2020. Even among the students who completed both, the level of accuracy and details with both assignments varied. This may be due to design students’ unfamiliarity with this analytic approach or feeling burdened with extra steps in the design process, especially when considering the short project period of five weeks. The real reasons remain unclear in this study and it requires further investigation.

It was found that the HMI flow diagram helped students analyze user experience, develop concepts, and configure specific interactions. In addition, students got a clear picture of how an interactive product affects user experience by comparing the current flow diagram and the new one. This can be evidenced in a student’s work (Figure 10) where the student, using the HMI flow diagram, analyzed the user interaction of an existing system, then diagrammed a new interactive system and built an interactive prototype accordingly.

How does automating the watering change the user interaction?

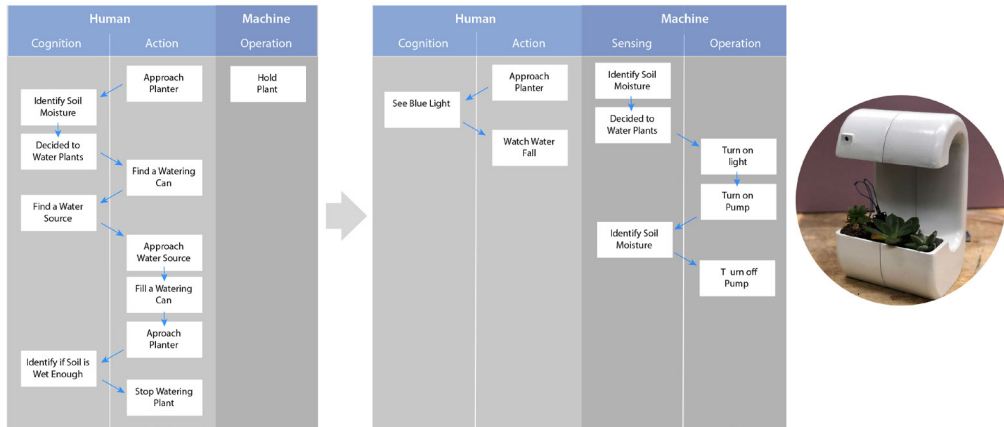


Figure 10 HMI flow diagram: comparison of manual and self-watering planter

This HMI flow diagram also functions as a blueprint that enables a smoother transition to the next steps. As interaction components, tasks, and relationships are clearly defined and distributed on this diagram, design concepts can be quickly visualized through a storyboard and then refined and translated into an allocation flowchart that bridges the interaction flow and the code. A student's work progress shown in Figures 6 to 8 exemplifies this. The storyboard (Figure 7) is a helpful tool to clarify the interaction sequence and specify the interface types and effects. In particular, allocating interaction tasks to the four process parts of human and machine helps clarify the hidden dynamics of interactive products better, in comparison to previous studies mentioned in section 2, with which this allocation method was not explicitly utilized in studio curricula.

5. Conclusions and Discussions

In this paper, we examined a structured process to guide design students without prior programming experience in interactive product prototyping. Previous studies confirm that the lack of systematic approaches of the studio curricula has posed a great challenge to students and instructors (Mikkonen, 2012; Wärnestål & Lindqvist, 2013; Oygür Ilhan & Karapars, 2019). Choi and Jung (2016) also argued that the challenge is how to integrate “technology-based knowledge” and “user-centered knowledge” in a systematic way. As an attempt to provide an answer to those challenges, this paper proposed a structured design process for interactive prototyping that consists of sequential development of the HMI flow diagram, a storyboard, and the allocation flowchart. This process makes a transition from initial ideation to programming smooth and incremental, so that design students can grasp the logic flow and develop algorithms more easily. Also, its allocated flow diagramming methods help students analyze the user experience and specify interactions. Thanks to this incremental and systematic process, it was found that students without prior programming experience can build prototypes successfully in a short period of time. This also serves as an example of integrating design and technology by enabling algorithms to be developed

gradually from the initial interaction flow diagram instead of jumping from idea sketches to programming.

However, it was observed that there are certain cases where this interaction model needs improvements. When actors other than human and machine are involved in interactions, such as pets or the surrounding environment, the model needs to be expanded or modified to include those actors. Also, when the interface is multimodal, dividing the machine part of the allocation flowchart according to the number of input and output types will further improve its effectiveness. For example, Figure 8 shows how the student spontaneously added a column for the additional sensor in the allocation chart. Finally, internet connectivity was intentionally left out in this paper but it is necessary to update the model and methods to accommodate the connectivity in the future, especially when considering the growing number of IoT devices in our lives.

There are multiple factors affecting successful interactive product prototyping besides the tools discussed in this paper. For example, the utility of digital circuit prototyping and coding tools like TinkerCAD and how these tools can be integrated into the proposed design process is not discussed here. As more advanced software tools for prototyping are becoming available, the level of difficulty with implementation may decrease. Developing a software application to merge the proposed design process and a digital prototyping tool will further streamline the prototyping process. Physical attributes are also very important factors of prototyping, but they are mostly left out in this paper. Detailed physical functionality of sensors and actuators, materiality and its effects on interactivity, assembly and integration are some of the challenges the students had to deal with during their projects. Table 2 shows that about a quarter of the students failed to integrate electronics into functioning physical prototypes even though they succeeded in making circuits work. This implies that the future study should include physicality and integration, and how the proposed process can be expanded to cover these aspects. Finally, another limitation of this study is the lack of controlled study about the true effectiveness of the process and methods proposed here. For a more accurate evaluation, there is a need to separate students into two groups based on whether or not to apply the structured process and to compare the students' achievements between the two groups. At the same time, investigating students' level of understanding and acceptance of each method through close observation will provide a better understanding of the proposed methods.

References

1. Aprile, W. A., & Van Der Helm, A. (2011). Interactive technology design at the Delft University of Technology – a course about how to design interactive products. *Proceeding of the 13th International Conference on Engineering and Product Design Education* (553–558), London, UK.
2. Böhm, C., & Jacopini, G. (1966). Flow diagrams, Turing machines and languages with only two formation rules. *Communications of the ACM*, 9(5), 366–371.
3. Budd, J., & Wang, W. (2017). Industrial design education: Taming technology to enhance user experience. *Archives of Design Research*, 30(3), 17–27. doi:/10.15187/adr.2017.08.30.3.17
4. Card, S. K., Newell, A., & Moran, T. P. (1983). *The Psychology of Human–Computer Interaction*. Hillsdale: Lawrence Erlbaum.

5. Choi, J. M., & Jung, E. (2016). User experience (UX) design curriculum bridging physical and screen interaction. *Journal of Korea Design Forum*, 21(1), 155–162.
6. Crews, T., & Ziegler, U. (1998). The flowchart interpreter for introductory programming courses. *Proceeding of FIE '98 Conference* (307–312), Tempe, AZ.
7. Dumas, B., Lalanne, D., & Oviatt, S. (2009). Multimodal interfaces: A survey of principles, models and frameworks. *Human Machine Interaction* (3–26), Berlin: Springer.
8. Frens, J. W. (2006). *Designing for rich interaction: Integrating form, interaction and function* (Doctoral dissertation). Eindhoven University of Technology, Eindhoven, the Netherlands.
9. Hackos, J. T., & Redish, J. (1998). *User and task analysis for interface design*. New York, NY: Wiley.
10. Hummels, C., Vinke, D., Frens, J., & Hu, J. (2011). Competency-centered education for designing interactive and intelligent products. *Creation and Design*, 13(2), 4–17.
11. Kim, H. (2020). Learning from two types of class projects in interactive physical prototyping: comparison between technology-driven and experience-driven project results. *Archives of Design Research*, 33(3), 75–86. doi:/10.15187/ adr.2020.08.33.3.75
12. Kolko, J. (2004). Mixing disciplines in anticipation of convergence: a curriculum for teaching interaction design to industrial designers. *Interactions*, 11(4), 18–23.
13. Koskinen, I., & Mikkonen, J. (2014). Interactive prototyping: teaching interaction to industrial designers. *Proceeding of DesignED Asia Conference*, Hong Kong.
14. Maguire, M. C. (1998). *User-Centred Requirements Handbook*. EC Telematics Applications Programme, Project TE 2010 RESPECT (Requirements Engineering and Specification in Telematics). WP5 Deliverable D5.3.
15. Mikkonen, J. (2012). Flowcards – a communication tool. *Proceeding of DRS 2012* (1215–1229), Bangkok, Thailand.
16. Mubin, O., Novoa, M., & Al Mahmud, A. (2017). Infusing technology driven design thinking in industrial design education: a case study. *Interactive Technology and Smart Education*, 14(3), 216–229. doi:/10.1108/ITSE-01-2017-0008
17. Nam, T. J. (2006). Educational framework for interactive product prototyping. *Journal of Korean Society of Design Science*, 19(3), 93–104.
18. Norman, D. A. (2016). *Living with complexity*. Cambridge, MA: MIT press.
19. Oygür İlhan, I., & Karapars, Z. (2019). Industrial design education in the age of digital products. *The Design Journal*, 22(sup1), 1973–1982. doi:10.1080/14606925.2019.1594922
20. Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction design: beyond human-computer interaction*. New York, NY: John Wiley & Sons.
21. Scanlan, D. A. (1989). Structured flowcharts outperform pseudocode: an experimental comparison. *IEEE Software*, 6(5), 28–36.
22. Stappers, P. J., Hekkert, P., & Keyson, D. (2007). Design for interaction: consolidating the user-centred focus in industrial design engineering. *Proceeding of the 9th International Conference on Engineering and Product Design Education* (69–74), Newcastle, UK.
23. Tung, F. W., & Deng, Y. S. (2004). A study on integrating interaction design into industrial design processes. *Proceeding of the 6th Asian Design International Conference*, Tsukuba, Japan.
24. Voûte, E., Stappers, P. J., Giaccardi, E., Mooij, S., & van Boeijen, A. (2020). Innovating a large design education program at a university of technology. *She Ji: The Journal of Design, Economics, and Innovation*, 6(1), 50–66. doi:10.1016/j.sheji.2019.12.001
25. Wärnestål, P. (2016). Formal learning sequences and progression in the studio: a framework for digital design education. *Journal of Information Technology Education: Innovations in Practice*, 15(1), 35–52. doi:10.28945/3406
26. Wärnestål, P., & Lindqvist, M. (2013). Designerly ways of teaching and learning: a course structure for interaction design. *Journal of Learning in Higher Education*, 9(1), 179–188.
27. Weil, D., & Mayfield, M. (2020). Tomorrow's critical design competencies: Building a course system for 21st century designers. *She Ji: The Journal of Design, Economics, and Innovation*, 6(2), 157–169. doi:10.1016/j.sheji.2020.03.001

28. Xinogalos, S. (2013). Using flowchart-based programming environments for simplifying programming and software engineering processes. *Proceeding of 2013 IEEE Global Engineering Education Conference* (1313-1322), Berlin, Germany.