

# SUPPORTIVE TOOLS FOR COLLABORATIVE PROTOTYPING

WILLEM HORST  
SPIRE CENTRE, MADS CLAUSEN INSTITUTE,  
UNIVERSITY OF SOUTHERN DENMARK  
WILLEM@MCI.SDU.DK

## ABSTRACT

The development of interactive products requires the integration of different disciplines, such as interaction design, design engineering, marketing and R&D. This paper explores how these disciplines can be involved in the prototyping process by introducing a set of tools. In the literature, various tools and toolkits are described that support interaction designers in the design and modification of prototypes in the early stages of a project. Although these make prototyping easier for interaction designers, it remains challenging to involve other disciplines in a *collaborative prototyping* process.

In this design case I describe a set of tailor-made tools that were designed to support the collaborative development of an interactive prototype in an industrial setting. I demonstrate how these tools supported collaboration and communication across functional units, and allowed different stakeholders to make concrete design contributions. I propose that investing in the development of such supportive tools is beneficial to product development, as they allow different stakeholders to use the prototype as a development tool, facilitate cross-functional

collaboration and enable appropriation and repurposing of the prototype across different departments.

## INTRODUCTION

The development of interactive products requires the integration of different disciplines, such as interaction design, design engineering, marketing and R&D. Interaction designers play a central role in defining the user interface of the product and prototyping is an important activity in this process. It is not only the outcome, i.e. the prototype, but also the process of prototyping that is relevant for development. As designers make the prototype they engage in what Klemmer et al. (2006) describe as “thinking through prototyping”. The iterative prototyping process is a reflective learning process, where the prototyper develops a deep understanding of the implications of both big and small design decisions on the use experience.

Although the importance of prototyping in the design of interactive products is well understood, making interactive prototypes is not trivial. Prototyping requires some technical expertise and can be costly. Depending on the complexity of the product and the fidelity of the prototype, the expertise required could be programming, electronics, graphic design and/or hardware integration; and the costs are related to the time it takes to make them and components necessary to assemble them. In the next section I briefly review two approaches in the literature that support designers to make prototypes. Then I describe my involvement as an interaction design consultant in the development of a programmable radiator thermostat in an industrial setting where I designed tools to support members of a cross-functional team to develop a prototype. I describe the supportive tools and their relation to the prototype itself and describe how they supported cross-functional collaboration in the prototyping process. Finally I present my conclusions and discuss opportunities for future research.

## SUPPORTING PROTOTYPING BY DESIGNERS

Making interactive prototypes is an important design activity, but can be difficult and expensive. In the literature there are two main approaches that support designers in making prototypes by making it easier and less costly to develop them.

One approach to overcome the high cost and expertise required to prototype is to lower the fidelity of the prototype, for example through paper prototyping (Rettig 1994), using cardboard mock-ups (Ehn & Kyng 1991), or PowerPoint-based prototypes. This lowers the technical expertise required and drastically reduces the cost of prototyping. These prototypes are certainly useful in some contexts, but are less useful when dynamic aspects are an important part of the use experience (cf. Sefelin et al. 2003 on the limitations of paper prototyping).

Another approach has been the development of toolkits to support designers in making interactive prototypes that are more complex. The goal of such toolkits is to enable designers to make prototypes in the early stages of a project to explore (physical) interfaces including the design and evaluation of the dynamic aspects. These toolkits contain various components that can be used in a variety of projects, and some environment to easily configure and program the prototypes without writing extensive code. Examples of such tools and toolkits are Phidgets (Greenberg & Fitchett 2001; Greenberg & Boyle 2002), the iStuff toolkit (Borchers et al. 2002; Ballagas et al. 2003), the Calder toolkit (Lee et al. 2004) and d.tools (Hartmann et al. 2006).

Although these approaches make prototyping easier for interaction designers, it remains challenging to involve other disciplines in the prototyping activity in an industrial context. Involving members of a cross-functional team as well as management in prototyping activities is important for them to be able to contribute to its development. As Schrage (1996) notes when this does not happen “the prototype becomes a medium for persuasion, rather than a vehicle to evoke discussion. It is used to prove a point, rather than to create a platform for a design dialog.” (p. 200) This is especially true for top managers that are involved late in the design cycle and then “are being asked to approve—rather than to review or assist—new-product creation” (ibid.) In the remainder of this paper I describe how the use of a flexible prototype in combination with supportive tools was an effective way to open up the prototyping process to these internal stakeholders in an interdisciplinary product development project.

## RESEARCH CONTEXT

This paper is based on my involvement in the development of the Danfoss living eco® radiator thermostat (eco®) at Danfoss Heating Solutions. The eco® is a programmable radiator thermostat containing electronics and a user interface, which can be mounted onto any radiator and has a similar form factor as a

conventional radiator thermostat. Based on the schedule set by the user and the temperature measured by the temperature sensor, a small motor controls the radiator valve to regulate the temperature. The product offers users a convenient way of saving energy, by for example automatically lowering the temperature at night and/or working hours.

The Danfoss Heating Solutions department responsible for its development normally develops mechanical products, such as conventional radiator thermostats, and the eco® is the first of its kind for this department. Since the department did not have all the necessary expertise in-house, the internal development team had to collaborate with different internal and external partners. Examples of external partners in this project are usability consultants and interaction design consultants, and examples of internal partners are other departments in the wider Danfoss Heating Solutions organization with expertise in software or electronics.

## RESEARCH AND DESIGN ACTIVITIES

In this project I was involved as an Interaction Design Consultant to develop interactive prototypes of the user interface. As a part of my research project in the role of prototypes in interdisciplinary product development, I designed and deployed three supportive tools to enable members of a cross-functional team to use the prototype as a development tool. During a 4 month period I worked closely together with the Design Line Specialist (DLS), who was responsible for the Man-Machine Interaction (MMI) as it was referred to in the company. As an external consultant, I did a lot of my work remotely, and communicated with the DLS via email. In addition to this I worked face-to-face with him during seven days spread over the 4 months and was part of 3 prototyping workshops at Danfoss Heating Solutions. After my involvement in the project the interactive prototype was used extensively for various activities over a one-year period.

This paper is based on various data sources that were collected in two stages. In the first stage during the intensive 4 month period, I gathered 9 hours of workshop videos, 60 emails and 70 prototype iterations. One year after my involvement in the project stopped, I conducted 5 semi-structured interviews reflecting on the use of the prototype as a development tool with the Design Line Specialist, the R&D Project Manager, the R&D Senior Director, the Global Webmaster and a Product Marketer.

## ABOUT THE PROTOTYPE

Very early in the project it was planned to do several usability tests of the interface, and it was clear that the user interface would have to be revised multiple times. Therefore the interface and the exact features were not frozen until these tests were done, although some decisions were made on aspects that related to the product hardware. These hardware decisions provided the framework for choosing an appropriate prototyping approach and medium.

It was decided that the product would have 3 buttons (up, down and enter), and a circular segment display with a diameter of 25 mm. The choice for a segment display, as opposed to e.g. a matrix display was an important constraint. With a matrix display, the exact icons can be changed at a later stage of the project at low cost because it is possible to make them in code. With a segment display, all segments (icons, digits, etc.) have to be specified and 'frozen' during the electronics development. The (cost of the) chip required to drive the display depends on the number of segments it has to control. Moreover, segment displays are tailor-made, and once such a display is made it is very costly to change it. Therefore, defining the (minimum) number of segments required to make up all the possible screens, and finding the right layout with appropriate icons on the right scale was an important objective. For this reason the interactive prototype had to be very detailed with regards to graphics and be on the right scale, without the high cost of changing the segments. To do this we chose to use a touchscreen PC running a virtual prototype of the interface scaled to the real dimensions (see figure 1), which meant we could do valid tests on the legibility of the icons in usability tests and change the virtual segments if necessary at low cost. This is a different type of prototype than the company usually uses during development:

*Usually when we talk about prototypes, then we are much further in pure hardware terms before we can call it a prototype. So it is perhaps the final display we sit and play with, which then gives us a lot of limitations, because now we have this display and we cannot go back. So that is where the value really kicks in, that we have something that resembles reality early on.* (Design Line Specialist in interview)



Figure 1: The virtual prototype

Using a virtual prototype, i.e. a piece of software which can run on a computer, has the advantage that it is easy to share, which was particularly relevant since I was working remotely. The Design Line Specialist also shared it internally and mentioned this as a clear benefit to both get input from various people, and allow them to use the prototype in their work:

*You could say that everyone who tries it internally comes with comments. Especially in the early phase, when things can be improved. So in that way you of*

*course also get an enormous amount of input, when it has been so easy to share this piece of software. The whole thing became one long test. [...] It has been so nice and easy to have been able to share this piece of software with internal people, so they could sit with it on their computer, either to play with it, or to use it concretely in their work.* (Design Line Specialist in interview)

## SUPPORTIVE TOOLS FOR NON-DESIGNERS

Designing the user interface of this particular product interface involved paying a lot of attention to the details. Because the interface surface was very small – everything had to fit on a display with a diameter of 25 mm – every detail could affect the product's usability. These details could be the exact size of an icon, how fast icons would blink, what heating schedule should run by default, or how long the backlight would stay on. These details are difficult to specify without trying them out and seeing how they work in a dynamic prototype, and evaluating them with others. Supporting other developers to 'play around' with these values, involving internal stakeholders as well as users, would enable them to use the prototype as a development tool. This could be done in evaluation sessions or in collaborative prototyping workshops.

To do this, I designed three supportive tools to be used together with the prototype. Two of these tools were designed to make changes to the prototype without coding: the first to edit basic parameters, such as blink-frequencies and timeouts, and the second to edit the virtual segments. Finally, I built in an export tool into the virtual prototype to export a picture of the current screen with a single key-press, to support effective communication as most of my work was done remotely. The prototype itself is structured around external files, such as graphic resources and sounds, and textfiles describing the layout of each screen and values for settings and parameters. The tools take advantage of this flexible structure and make changes to some of these external files (see figure 2). This enabled making changes to these aspects of the prototype without changing code or compiling a new version of the prototype. Some changes did of course require coding, and this could only be done in the Adobe Director environment, which I used to develop the virtual prototype. In the following subsections I briefly describe each tool and how it was used in the development process.

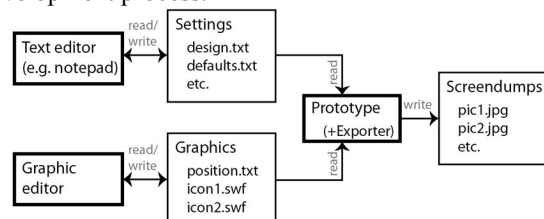


Figure 2: The relation between the prototype and the supportive tools

### PARAMETER EDITOR

To enable other developers to make changes to the parameters and settings of the prototype I stored all the

settings in separate textfiles. In DesignSettings.txt I stored all settings that had something to do with the overall look and feel of the interface. This could be the click-sound to use, the blink-frequency or various timeout values. In gDefaults.txt all the global defaults were stored. This could be the default temperature, the default date and time, or maximum and minimum values for the temperature. I also provided a readme textfile where each parameter was described and what type of values it could have (see figure 3).

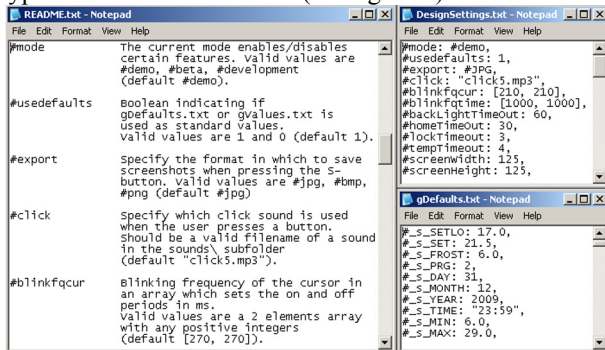


Figure 3: Text files with parameters and readme

These textfiles could be easily edited with standard software, such as Notepad or TextEdit, and enabled others to try out different values. Before I structured the parameters like this, I was the only one who could change these values, and would for example get emails with a request to change something:

*Could we try out a frequency of 1 to 2, so that it is gone twice as long as it is shown? (Intern – email)*

After the other developers could change these settings they first used it to find values they preferred themselves:

*I think it starts making sense to me. I do to however get a bit confused and have a tendency to get stressed with all the blinking, but I can just try and change the speed and see if that helps. (Intern – email)*

Later in the process this was also done in collaboration with end-users:

*The good thing about this tool is that you can try things out immediately, especially with these ‘soft-coded’ things, where you can very easily change a parameter and test it right away. [...] Things we have played around with a lot are the default values in it: blinking frequencies, time-outs and such things. You call it participation workshops, where you test directly with a user and correct immediately, until you reach a satisfying result. [...] As a developer you lose the feeling for those parameters, time-outs and frequencies and things like that. So there it was very easy to go in and find the values, by being able to adjust them until the end-user says: “Now it is good.” [...] It is of course about being able to correct things onsite, directly, instead of having to write something down, and tomorrow you have forgotten what it was you had to correct and what it was he said, and these kinds of things. So to get it adapted to the test person you are sitting with, immediately, has also sped up the process*

*tremendously, and moreover you get everything. (Design Line Specialist in interview)*

#### GRAPHIC EDITOR

All graphic assets used by the prototype were stored externally in a separate folder to allow for easy updating of graphics. To further facilitate making changes to graphics without the need for either using external graphic design tools or coding, I developed a graphic editor which provided an overview of all segments of the display and allowed making basic changes to existing graphics, such as repositioning, scaling and deleting, as well as importing new graphics or replacing graphics (see figure 4). New graphics would have to be developed with a graphic design tool, such as Adobe Illustrator – the editor had no drawing capabilities. Using the tool made it easy to make changes to icons that would carry through the whole prototype without any code, and made making 'cosmetic changes' to address cosmetic usability problems easy.



Figure 4: Graphic editor

The graphic editor was very useful in collaborative prototyping session, since it enabled me to quickly make changes to the graphics, e.g. change the size or position, on any computer. Before I made the editor, I would have to edit the graphics in Adobe Illustrator and export them, which took some time and could only be done on my computer, since nobody in the development team had this program installed on their computer.

#### SCREEN EXPORTER

I developed a screen exporter that was integrated in the virtual prototype; pressing the 's'-button would take a screendump and store it in a specific folder. I envisioned it would be used primarily to facilitate communication between the Design Line Specialist and me, but it turned out it was used for other reasons as well. The pictures were used for internal presentations, making the user guides, software specification and the product website. This tool was used both during and after the development of the interface by different departments: *The exporter has been used extensively for presentations, and then gone straight into the manual. The quality has been so high that there has been no need for drawing anything in. So that is also a lot of*



time saved. [...] When we come to the documentation part of the user interface and the handover to the programmers, well then it was of course also incredibly helpful and timesaving for me to be able to use this. Especially the export function; instead of writing it down with words, the specification actually consists of hundreds of small comic strips. (Design Line Specialist in interview)

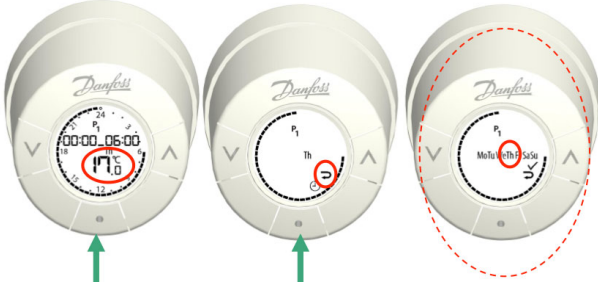


Figure 5: 'Comic strip' from the exporter (annotated in PowerPoint)

When changes had to be made to the prototype, e.g. after a usability test, the Design Line Specialist would send me 'comic strips' annotated in PowerPoint with the changes (see figure 5). In a similar way, the exporter was used to document the interface to the software developers.

After the interface development was finished, I made a stand-alone version of the screen exporter specifically for the user guide developers (see figure 6). They requested a higher resolution, and a glowing effect to indicate the icon that was highlighted. This also came with a parameter editor, which enabled the user guide developers to change the highlight-glow size, strength and colour.



Figure 6: Stand-alone version of the screen exporter

## CONCLUSIONS AND FUTURE RESEARCH

In this paper I have presented three supportive tools for collaborative prototyping that were used in a product development process. The tools enabled me as a designer to effectively work together with different members of an interdisciplinary development team. The role of the prototype also changed, because it was no longer static, but could be tinkered with by non-designers. The Design Line Specialist compared this

prototype to another virtual prototype developed by another company:

*You could say that the [virtual] prototype [developed by another company] is not flexible; we cannot change anything. I cannot change anything in it myself. I can use it as a communication tool, or some status update: "Well, now it is like this." But I have no possibility to modify it, or develop with it, you could say.* (Design Line Specialist in interview)

Developing and maintaining the different tools took 12 hours (on a total of 172) of development time (which includes 8 hours to optimize the screen exporter for the people making the user guide). The supportive tools added value during the interface development, and extended the lifespan of the prototype far beyond development – due in large part to the screen exporter. My findings suggest that investing in the design of supportive tools in the context of cross-functional product development is well worth it, which was underlined in the interviews:

*I am sure that we have saved both money and time in this project using this tool. It has been involved in so many different parts of the project. So it is not only to settle the Man-Machine Interface, but also as documentation in different ways.* (R&D Senior Director in interview)

The flexible prototype and its supportive tools were also used in collaborative prototyping events (described in more detail in Horst and Bogers, forthcoming) and enabled *live prototyping*, i.e. making changes to the prototype on the fly based on input from various participants. As such, these tools supported these participants to engage in the prototyping activity, which gave them a better understanding of the design constraints and implications of design decisions. This supported the collaborative and interdisciplinary development as the prototype and its tools acted as a boundary object for the different stakeholders involved. The tools presented in this paper can be improved and expanded in several ways. A special tool to edit the parameters, integrating the instructions of the readme and the actual values could be an example of making the tools more user-friendly, where the users are the interdisciplinary development team in this context. Specifically designing these supportive tools based on the needs and skills of the different developers involved is an area to explore further.

My original intention with these tools was to open up the prototyping process to the developers I was collaborating with. The fact that the prototype and its tools were so easy to share made it possible to open up the prototyping process to a much broader range of people, who used and appropriated it in ways I had not imagined, without my involvement. Designing supportive tools that open up even more of the prototype to enable different stakeholder to contribute to its development in a collaborative process is an interesting area for future research.

## ACKNOWLEDGMENTS

I would like to thank the people involved in the project described in this paper at Danfoss Heating Solutions for their support during the project and their cooperation afterwards. Special thanks go to Anders Østergaard Clausen for his support and engagement throughout the process.

## REFERENCES

Ballagas, R. et al. 2003. 'iStuff: a physical user interface toolkit for ubiquitous computing environments', In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, pp.537–544

Borchers, J. et al. 2002. 'Stanford interactive workspaces: a framework for physical and graphical user interface prototyping', *IEEE Wireless Communications*, 9(6), pp.64–69

Ehn, P. & Kyng, M. 1991. 'Cardboard Computers: Mocking-it-up or Hands-on the Future', In Greenbaum, J. & Kyng, M. (Eds.) *Design at Work: Cooperative Design of Computer Systems*, Hillsdale, NJ: Laurence Erlbaum Associates, pp.169–195

Greenberg, S. & Boyle, M. 2002. 'Customizable physical interfaces for interacting with conventional applications', In *Proceedings of the 15th annual ACM symposium on User interface software and technology*. New York, New York, USA: ACM, pp.31–40

Greenberg, S. & Fitchett, C. 2001. 'Phidgets: easy development of physical interfaces through physical widgets', In *Proceedings of the 14th annual ACM symposium on User interface software and technology*. ACM, pp.209–218

Hartmann, B. et al. 2006. 'Reflective physical prototyping through integrated design, test, and analysis', In *Proceedings of the 19th annual ACM symposium on User interface software and technology - UIST '06*. New York, New York, USA: ACM Press, pp.299–308

Horst, W. & Bogers, M. forthcoming. 'Enabling the prototyper: A case study of collaborative prototyping', In *Proceedings of the 18<sup>th</sup> International Product Development Management conference*, June 5-7, 2011, Delft, the Netherlands

Klemmer, S.R., Hartmann, B. & Takayama, L. 2006. 'How bodies matter: five themes for interaction design', In *Proceedings of the 6th conference on Designing Interactive systems*. pp.140–149

Lee, J.C. et al. 2004. 'The Calder toolkit: wired and wireless components for rapidly prototyping interactive devices', In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*. ACM, pp.167–175

Rettig, M. 1994. 'Prototyping for Tiny Fingers', *Communications of the ACM*, 37(4), pp.21–27

Schrage, M. 1996. 'Cultures of prototyping.' In T. Winograd, ed. *Bringing design to software*. New York: ACM Press, pp.191–205

Sefelin, R., Tscheligi, M., & Giller, V. 2003. 'Paper Prototyping - What is it good for? A Comparison of Paper- and Computer-based Low-fidelity Prototyping', In *Proceedings of CHI 2003: New Horizons*, April 5-10, Ft. Lauderdale, Florida, USA, pp.778–779