An Exploration of Collaborative Prototyping

Marcel Bogers
University of Southern Denmark
Mads Clausen Institute
bogers@mci.sdu.dk

Willem Horst
University of Southern Denmark
Mads Clausen Institute
willemhorst@gmail.com

Abstract

In new product development (NPD), prototyping is recognized as an important activity of iterative problem solving. In this inductive study, we investigate how collaborative prototyping across functional, hierarchical and organizational boundaries can improve the overall prototyping process. Our combined action research and case study approach provides new insights into how collaborative prototyping can provide a platform for prototype-driven problem solving in early NPD. Our study shows that the actual prototyping practice involves non-discrete steps in the trial-and-error problem solving cycle, that the actual prototype was used as a tool for communication or development, and that this platform for collaborative prototyping detects emerging usability problems through active engagement and experimentation. Our continued analysis moreover shows how collaborative prototyping provides an opportunity to share knowledge and expertise across functional, hierarchical and organizational boundaries, while we also identify some constraints in involving the appropriate stakeholders at the right time. Finally, we find that despite the continuous nature of the practice of prototyping, there are certain windows of opportunities during which the collaborative prototyping approach actually leads to changes in the product design.

Jelcodes:O31,O32
An Exploration of Collaborative Prototyping

ABSTRACT

In new product development (NPD), prototyping is recognized as an important activity of iterative problem solving. In this inductive study, we investigate how collaborative prototyping across functional, hierarchical and organizational boundaries can improve the overall prototyping process. Our combined action research and case study approach provides new insights into how collaborative prototyping can provide a platform for prototype-driven problem solving in early NPD. Our study shows that the actual prototyping practice involves non-discrete steps in the trial-and-error problem solving cycle, that the actual prototype was used as a tool for communication or development, and that this platform for collaborative prototyping detects emerging usability problems through active engagement and experimentation. Our continued analysis moreover shows how collaborative prototyping provides an opportunity to share knowledge and expertise across functional, hierarchical and organizational boundaries, while we also identify some constraints in involving the appropriate stakeholders at the right time. Finally, we find that despite the continuous nature of the practice of prototyping, there are certain windows of opportunities during which the collaborative prototyping approach actually leads to changes in the product design.
INTRODUCTION

The importance of design research and design thinking is increasingly being recognized within the context of new product development (NPD) due to the emphasis on the process of problem solving to address user needs and create new opportunities (Brown, 2008; Ulrich, 2011; Verganti, 2009). In particular, prototyping is considered as an important design practice, which thereby becomes a central element in corporate innovation processes (Leonard and Rayport, 1997; Mascitelli, 2000; Schrage, 2000) as exemplified by several important advances in this area (e.g., Buchenau and Fulton Suri, 2000; Hartmann, 2009; Terwiesch and Loch, 2004; Thomke, 1998). In this paper, we especially build on the recognition that prototyping can improve the NPD by performing it early in the process and by involving a number of relevant stakeholders within and outside the design team and organization (Buchenau and Fulton Suri, 2000; Mascitelli, 2000; Terwiesch and Loch, 2004).

Within NPD, the product specification is typically made before prototyping starts, which has been questioned in both software and product development (cf. Boehm et al., 1984; Rudd et al., 1996; Thomke and Bell, 2001). In fact, in the early phase of the NPD process, prototyping can play an important role in informing the development of the product specification (Mascitelli, 2000; Thomke, 1998). For example, at IDEO, the development team starts with prototyping to develop the specification and at later stages develops prototypes based on this specification. David Kelley, founder and chairman of IDEO, argues that “organizations intending to be innovative need to move from specification-driven prototypes to prototype-driven specifications” (quoted in Schrage, 1996: 195). Hans-Christoph Haenlein, (former) Director of Prototyping at IDEO, describes three prototyping phases to inspire, evolve and validate the specification. In the first two phases, “project specifications are derived from the prototypes. Towards the end of a project, very complete
prototypes are built to validate the design specification as a whole” (quoted in Hartmann, 2009: 22).

Combining the prototyping and specifying approaches in this way can also help to deal with the tension within NPD between the number of features and the usability of the product (Keijzers et al., 2008). This is especially true for electronic products with a user interface, since features can be added at relatively low cost, making it tempting to do so in a competitive market where the number of features is an important competitive factor (Rust et al., 2006). Specifying the feature-set of a product happens in the early stages of the NPD process, typically before starting the design of the user interface (cf. Thomke, 1998). However, how features are implemented and available in the product relates to their usability and usefulness. Therefore, starting the design of the user interface before finalizing the feature-set helps in defining an appropriate feature-set. Emphasizing the experiential aspect in active engagement with prototypes moreover enables coming up with new solutions in unknown design spaces (Buchenau and Fulton Suri, 2000). By making interactive prototypes that simulate the functionality at this early stage, it is possible to iteratively evaluate the prototype’s usability as features are added or removed, or implemented in different ways.

In this paper, we explore a collaborative prototyping process within a particular organization, in which a prototype was used to help define the feature-set by involving not only common stakeholders as interaction designers and usability experts but also other stakeholders as marketers, directors and users. We present an NPD project in which the prototyping process was initiated before the features were fully defined, and it continued until the final software specification over a one-year period. During this process the purpose of the prototype changed as both the feature-set and the user interface became more defined. Depending on the purpose of the prototype, different stakeholders were involved in a collaborative prototyping process.
In the next section we review the relevant literature on (collaborative) prototyping in NPD. Then, we describe our research methodology and introduce the company case. Next, we present the empirical results of our study based on “reflection-in-action” within an action research approach and “reflection-on-action” within three cases of collaborative prototyping processes with different stakeholders. We then analyze and discuss our findings, which suggest that collaborative prototyping in early NPD creates a prototype-driven approach to problem solving that collapses problem-solving cycles into a continuous iteration centered around the prototype, that it improves the functionality-usability balance and focuses on actual design changes, that active engagement in collaborative prototyping enables collaboration across various boundaries, and that there are certain windows of opportunities for actual changes in the product design. We conclude by summarizing our results and discuss implications for management and theory in relation to collaborative prototyping in NPD and beyond.

**LITERATURE REVIEW**

**Collaborative prototyping in NPD**

In the management literature, the prototyping process is conceptualized as an iterative trial-and-error learning process, following four steps: (1) design, (2) build, (3) run and (4) analyze (Thomke, 1998; von Hippel, 2005). Moreover, as a design practice, prototyping is a central element in corporate innovation processes (Leonard and Rayport, 1997; Mascitelli, 2000; Schrage, 2000), thus making design (thinking) an important part of problem solving within NPD (Brown, 2008; Ulrich, 2011; Verganti, 2009). An important development within the literature related to prototyping (and NPD and innovation at large) is a consideration of how designers can better engage other stakeholders in the prototyping process.
In the literature, various reasons are discussed for involving others in the prototyping process. Terwiesch and Loch (2004) describe the collaborative prototyping process as a search process for the ideal product specification for custom-made products involving the producer and user of the product. In participatory design, the idea of cooperative prototyping, which involves the end-users of the interface that is being designed rather than demonstrating it to them, is used to learn more about the actual use-context and shape the technology (Bødker and Grønbæk, 1991). Toolkits for innovation give users the possibility to design, simulate or prototype a custom product or service (von Hippel and Katz, 2002). An important characteristic of such toolkits is that they “enable users to carry out complete cycles of trial-and-error learning” (von Hippel and Katz, 2002: 825), which is an important aspect of the prototyping process. Although this shifts the locus of innovation to users, the design of the toolkit is a collaborative process, where the manufacturer can optimize its design based on feedback from users (Bogers et al., 2010).

Many such approaches relate to involving end-users in prototyping and as such relate to cross-organizational NPD processes. In addition, intra-organizational boundaries across the different functions involved, such as research and development (R&D), production and management, also play an important role in NPD (Atuahene-Gima and Wei, 2011; Song et al., 1997; Song et al., 1998). In such context of internal NPD processes, Henderson (1995) considers the politics of prototypes, as they embody particular points of view, and can be used as arguments in an organization. Schrage (1996: 200) describes the risk of exposing important stakeholders to the prototype at a late stage in the project, as in such cases “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.” This is especially true for top managers who are involved late in the design cycle and then “are being asked to approve—rather than to review or assist—new-product creation.” (Schrage, 1996: 200)
Towards prototype-driven problem solving

Boehm et al. (1984) make a distinction between taking a specifying and a prototyping approach in software development, concluding that a prototyping approach produces comparable prototypes but with significantly less code and effort than the specifying approach, although the latter approach produced code that was more coherent and easier to integrate.

Moving from specifying to prototyping may enable a better understanding of both technical and customer-need related problems, for example due to the improved fidelity of the problem-solving process (Foray, 2004; Thomke and Bell, 2001; von Hippel, 2005). In the context of rapid prototyping, for example of a user interface, high-fidelity prototypes are particularly beneficial to provide a holistic perspective based on interactive exploration and testing, although possible drawbacks may include cost and time considerations (Dey et al., 2001; Rudd et al., 1996).

The transition from specification-driven prototypes to prototype-driven specifications raises the question who should be involved in the prototyping process (cf. Schrage, 1996). Usually, the people developing the specification (e.g., marketers or top management) are not the same people who make the prototypes (e.g., design engineers). In a prototype-driven process it will thus be important to involve these various stakeholders, who may be spread across various functions and hierarchical levels (Adler, 1995; Buur and Matthews, 2008; Song et al., 1998).

Experience prototyping

Collaborative prototyping can be seen as a practice that relies on the involvement of various stakeholders. Engaging in such activities involves an exploration of the relevant design space as in line with the more general discussion of how to best use prototypes within
(participatory) design (Hartmann, 2009; Lim et al., 2008; Schuler and Namioka, 1993). For example, human embodied engagement has been advocated within interaction design as a way to shape experience, understanding and interactions in design practices (Klemmer et al., 2006). Moreover, the concept of learning-by-doing highlights the important role of the use experience in identifying and solving problems within innovation activities (Macher and Mowery, 2003; von Hippel and Tyre, 1995).

Buchenau and Fulton Suri (2000: 424) propose “experience prototyping” as “a form of prototyping that enables design team members, users and clients to gain first-hand appreciation of existing or future conditions through active engagement with prototypes.” The aim of an experience prototype would be to design an integrated experience to better understand, explore or communicate what it might be like to engage with the product that is being designed (Buchenau and Fulton Suri, 2000). The high fidelity of actively engaging in a prototype will improve the overall performance of the problem-solving process (Foray, 2004; Lim et al., 2008).

METHOD AND RESEARCH CONTEXT

Study setting and data sources

In this paper, we adopt multiple methods to report and analyze a series of experiences in the development of the interface of the living eco® radiator thermostat at the Danish company Danfoss Heating Solutions. The research presented here was developed in two stages.

In the first stage, an action research approach was taken in which the second author was involved as an external Interaction Design Consultant (IDC¹) at Danfoss Heating Solutions over a period of a year (in 2009 and 2010). During this time, the collaborative

¹ A complete list of acronyms of involved stakeholders can be found in the Appendix.
A prototyping method was developed and deployed in several workshops, to develop the interactive prototype with multi-stakeholder involvement. Data collected during this period includes video material from those workshops, email conversations with the Design Line Specialist (DLS), project documentation, and the 70 iterations of the interactive prototype. See Table 1 for an overview of the different data sources that are used in this paper.

The prototype was run and developed on a PC. It was designed to be interactive and flexible. For this reason, it was broken up into three different components. First, there is the programming code, which defines the general behavior, interaction, and ties together all assets. Second, 20 textfiles describe the structure of the interface (e.g., which icons are placed on each screen, or what happens next when an icons is selected), and another 5 textfiles describing various default values and settings. Finally, there are 80 graphics that simulate each of the segments of the final segment display. Because each iteration was stored in a different folder, we were able to trace back the development of the prototype, and pinpoint when design decisions were taken by looking at the date and time these files were last modified. Using a custom-made analysis tool allowed stepping through the project day by day, which enabled us to see, for example, what kind of design decisions were made in the various workshops, when specifications were updated, and what the email response was to the different prototype versions.

In the second stage, we follow an exploratory case study approach using the case of Danfoss Heating Solutions. More specifically, the case study compared three types of prototyping within Danfoss Heating Solutions. We analyzed information about the prototyping activities from different sources, such as documentation about specifications, scenarios and strategies, email correspondence related to the prototyping activities, and video
material from various workshops. In addition, five interviews were conducted with the DLS, the R&D Senior Director (RDSD), the R&D Project Manager (RDPM), the Global Webmaster (GB) and a Product Marketer (PM) after the prototype had been used within the organization for over a year. The interviews focused on the role the prototype had played in the work of the different disciplines involved. Each interview was recorded and transcribed and lasted between 30 minutes and one hour. (See Table 1 for an overview of data sources).

**Data analysis**

In order to combine the two stages within our research design, namely action research and case study, we organize our analysis around “reflection-in-action” and “reflection-on-action,” respectively (Schön, 1983).

Essentially reflecting the main principles of action research, reflection-in-action entails that the researcher as practitioner “reflects on the phenomenon before him, and on the prior understandings which have been implicit in his behaviour. He carries out an experiment which serves to generate both a new understanding of the phenomena and a change in the situation.” (Schön, 1983: 68) Action research works within the realm of practical knowing while emphasizing cycles of action and reflection, with activities comprising planning, action and fact-finding (Coghlan, 2011; Lewin, 1946). The reflection-in-action that we present in this paper builds on an action research project that extended over a one-year period. Here we also build on design research, in which this approach is commonly taken when an important aspect of the research is to find out how new design methods work in practice (Binder and Redström, 2006). Instead of observing what happens as an objective researcher, the researcher takes a dual role of researcher-designer, who takes active part in the design activity (i.e., the design of the user interface in the NPD project). This type of research is future-oriented and action-driven—through doing interventions and analyzing the outcome it
is possible to develop “field-tested and grounded technological rules to be used as design exemplars of managerial problem solving.” (van Aken, 2004: 221) The researcher-designer works on a practice problem and designs a solution concept. This is the basis for the design of the intervention, which is a context-specific solution. Through evaluating and reflecting on the outcome the researcher can revise the solution concept and re-design a specific solution (Andriessen, 2007). In the findings section, we will provide a an overview of the development of the collaborative prototyping approach—cf. “thick description” (Geertz, 1973)—and will describe how it was used in an NPD project by organizing workshops that facilitate involving various stakeholders. We will present evidence based on the observations and reflections during relevant events as well as from the video recordings (which also allows us to present evidence of conversations and interactions). We thus present illustrative examples of relevant activities, events and discussions, while we also refer to later reflections (cf. reflection-on-action below) and related literature in our to put our reflection-in-action into a broader perspective. Especially the IDC (i.e., the researcher-designer) and the DLS will be central in this reflective narrative given their roles in this project and research study.

The second part of our study entails reflection-on-action, in which the researchers, after the first stage of the research was concluded, “think back on a project they have undertaken […] and they explore the understandings they have brought to their handling of the case.” (Schön, 1983: 61) As such, this exploratory study follows the principles of the qualitative case study approach (Eisenhardt, 1989; Eisenhardt and Graebner, 2007; Yin, 2003). In order to ensure construct validity, we used a general structure of questions and framework to investigate and analyze the different prototyping activities, while we also constructed a timeline of activities to construct a chain of evidence (i.e., how certain events caused particular effects). Triangulation was established by using multiple sources of information and by relying on multiple informants (e.g., interviews). Even though the case
study is conducted in a single organization, we essentially investigate three separate cases, the unit of analysis being the prototyping activities, which enable us to predict similar results across the cases (literal replication) and predict contrasting results for predictable reasons (theoretical replication), which thereby increase external validity of the study (Yin, 2003). In the analysis of the prototyping cases, we attempted to identify categories of findings from within the cases, while comparing the finding across cases as an analytic technique. Given the inductive nature of this research, we applied the principles of grounded theory to construct categories of findings by developing categories of information (open coding), interconnecting the categories (selective coding), and building a story that connects the categories (axial coding), upon which the final findings are based (Strauss and Corbin, 1990). As such, the construction of categories can be seen as an iterative process that establishes common meaning across multiple observations (Locke, 2001).

**About the company and NPD project**

This paper is based on the development of the Danfoss living eco® radiator thermostat 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 products offers customers a convenient way of saving energy, by for example automatically lowering the temperature at night and/or working hours.

The department, responsible for the development of the eco®, 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 users, 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. Each partner, both internal and external, was located at a different city, and this was the biggest managerial challenge according to the RDPM:

*The biggest challenge is to manage a project where the resources are on three different geographical places: in the UK, and in Denmark, a couple in Vejle. That was the biggest one, the space between us.*

And although it started as a regular project, it rapidly expanded as the project gained momentum, and grew to be the biggest project in the organization as the RDSD explains:

*When we started the project, we did not think it would be any more important than any other projects, but right now it is by far the biggest and most important project in the whole Heating Solutions organization, also including electronics. It is by far the biggest project. Looking at the cost and investment in the project, but also looking at the market scope and money we will spend on communication in the market and the interest from top management. I think that is one of the first projects where they are linked very close to our timeline and looking into all details.*

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 for example 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, a touchscreen PC running a virtual prototype of the interface scaled to the real dimensions was used, 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 (see Figure 1). This is a different type of prototype than usually used during development, as the DLS explains:

Unless 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.

**FINDINGS**

**Reflection-in-action: Developing a collaborative prototyping approach**

*Designing a collaborative prototype*

An important challenge in this project was to design a user interface on a small product surface, which is easy to use, yet offers a high level of functionality to the end-user, such as the ability to fully customize a week schedule. The design of the user interface consisted of three stages that are described in this paper, each with particular challenges and stakeholders the development team had to relate to. Finding the limits of how many features can be offered without making the interface too complicated and cluttered was important in the early stage when the exact feature-set was specified. Important stakeholders at this stage were PMs, the Product Portfolio Director (PPD) and the RDSD. When the feature-set was defined, the challenge was to optimize the interface to ensure the usability of the product. The
external usability consultants—the Senior User Experience Manager (SUEM) and User Experience Consultant (UEC)—who conducted several usability tests were key stakeholders in this stage. Furthermore, fine-tuning the interface details after a usable interface was developed was important in the last stage of the interface design to focus on the use experience beyond ease-of-use. In this stage, end-users were the main stakeholders.

In terms of developing an active involvement of the relevant stakeholders in this project, it turned out that flexibility and communicability were two important issues in developing the specific prototyping approach. That is, the IDC designed the prototype with flexibility in mind, so changes could be made rapidly, allowing for many design iterations—cf. Verganti (1999) on planned flexibility and Thomke (1997) on flexibility in NPD. In fact, it was so flexible that the IDC could change it on the fly during collaborative prototyping sessions, going through the four steps of the informal prototyping process; participants could try (run), evaluate (analyze), give input and make suggestions (design), which the IDC implemented (build) in an iterative process until the participants were satisfied. Secondly, the IDC designed supportive tools to be used together with the prototype, by both the IDC and others. 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. Moreover, an export tool was built into the virtual prototype to export a picture of the current screen with a single key-press, to support effective communication as most of the IDC’s work was done remotely (cf. Horst, 2011).

Thus, it was the active engagement of various stakeholders with the prototype—in line with the principles of “collaborative prototyping” (Terwiesch and Loch, 2004) and “experience prototyping” (Buchenau and Fulton Suri, 2000)—that enabled an efficient overall prototyping process.
Balancing functionality and usability in a collaborative prototyping workshop

Because of the importance of the user interface within the collaborative prototyping of this product, the tension between the number of features and the usability soon became apparent (Keijzers et al., 2008; Rust et al., 2006). Building on the main features that were developed in several workshops before the IDC joined the project, an initial version of the prototype was made based on 13 click-thru scenarios describing those features. Participants in the first collaborative prototyping workshop were the IDC, the DLS, an innovation intern (INT); two Design Engineers (DE1 & DE2), the Communication Technology Manager (CTM) and Marketing Engineer (ME) from the development team, the R&D Director (RDD), the RDSD, the PPD, and two external usability consultants (SUEM and UEC). Using the prototype triggered a discussion about whether there were enough features to distinguish the product from competitors, or too many features that made the interface overly complicated.

By seeing the user interface on the real scale, the trade-off became very concrete, especially for more advanced features, such as full customization of the schedule:

DE1 Now we have the possibility to have several different periods, but we aren’t able to have different temperatures in those periods?
DLS No, that is something we decided not to do.
DE1 Can our competitors do that?
PPD Yes. Yes, that is something we have to include.
DLS So we have to be able to, on the size of a 10 kroner [coin with diameter of ca. 25 mm], to choose a flexible temperature change for every single period?
RDSD So if I understand what DE1 is saying, you want to be able to put it down to 17 degrees on Monday evening and to 18 on Tuesday evening, and...
DE1 To have it on 23 degrees during one day, and 22 on the next, and so on.
DLS What adds value? Does it add value to have a completely clogged user interface? We just have a set-point [normal temperature] and then a set-back temperature [energy saving temperature] that you can change.
UEC A competitive parameter is also user-friendliness, not just the amount of features.
ME I think it is much more important to have the freedom to set the period from this time to this time where we should lower the temperature.
DLS Of course it is possible, and we have also played with it, but there is a significant difference in the user interface, if we include the temperature parameter in programming the schedule.
DE1 So when they are setting the schedule they think: “Why can’t I do this?” Our competitors can.
DLS Yes, but does it add value?
SUER In any case there is the problem with this trade-off on the user-friendliness. It means that if you have to compromise the user-friendliness, you could say, that you can’t even reach that functionality. You could say it has a price.
DLS It doesn’t need to be worse, just because it is different [from our competitors]. We can argue for why we are doing it the way we do.
This led to some reflection on the amount of functions the product should have:

CTM  We have really many functions, and I seriously doubt to what extent these advanced functions will be used in really many installations, because it’s simply too difficult to figure out. We can of course write that we can do these fine things on the box, which our competitors can also do, so we have a product that matches that.

ME  We still have to turn the decision when it is out in a shop. Then people have to say: “ok, this one can do this, and Danfoss can’t do nearly as much, but it costs the same. Then I will choose the competitor.”

CTM  I think that the user interface as it is now, with all the functions that we desire, yes then it gets complicated. But if we could reduce the function list, that would help a tremendous amount.

The important role of the prototype in finding a balance between functionality and usability is confirmed by a later reflection of the DLS:

This [prototype] has also been used to set the boundaries for the basic specification during its development. [...] It has also been used like some sort of evidence for when we have to stop to put features into this product. Because, again, we can see at an early stage: “Well, we will never ever get this particular feature up and running with this screen size and this number of buttons.” So there it has been really strong, [...] There are many funny and wacky ideas for such an interface. “Why can’t we do this, and why can’t we also do that too?” And so on. But again, it is enormously powerful to be able to prove why we do it like this, relatively easy. [...] With the pressure and the [feature] requests that have come from marketing I feel we have struck, or found, the balance fairly well. [The balance] between what can and what cannot be done. It could easily have gone to the wrong side without this tool. There have been really many feature requests, both from the market and from marketing.

Not only the function list, but also how these functions should be available on the product was discussed. One particular example is the “mounting” function. To put the eco® on the radiator or to take it off, the motor has to rewind the regulating pin to be able to screw it on/off. Everybody agreed this function should be in the product, but who should access this function was a point of discussion:

DE2  On the other side, we always hear from marketing that the end-user shouldn’t be able to take it off so easily. For that they have to use a tool. That was the idea: they shouldn’t actually do that.

CTM  Why should they use a tool if they want to take it off to program it?

SUEM  I think many will feel insecure about taking it off, I mean. If I would be at home…

UEC  I wouldn’t do it either. I’m not a plumber.

CTM  But we have to somehow, no matter what, be able to put it in that state.

UEC  But the point is, it doesn’t have to be that intuitive if it is anyway a service [and installation is done by a trained service mechanic].

DLS  It shouldn’t be intuitive.

CTM  No, because if it’s intuitive then the end-user will also do it. “Oh, what is it doing now? Ok, the radiator is getting really hot…” [with the regulating pin pulled back the heating valve will be completely open] (laughing) But it should be there in some way.
In this early stage of the development of the product and related prototyping approach, there was a clear recognition and discussion of the tension between functionality in terms of features and usability, given the strategic importance of such issues (Keijzers et al., 2008; Rust et al., 2006). The prototype essentially offered a platform to explain and relate various perspectives on the interface that enabled cross-fertilization of knowledge (Atuahene-Gima and Wei, 2011; Carlile, 2002; Nicolini et al., forthcoming).

Translating usability problems into design changes within collaborative prototyping

In the next stage, the first usability test was done by the external consultancy and the DLS and IDC worked on the interface to address the usability problems described in the usability report. Decoding the usability report and translating usability problems into design changes was challenging, since the development team had not participated in the usability test. Then, the developers organized a prototyping workshop in which the SUEM participated. Having conducted the usability test, she knew a lot more about the usability problems and how they arose than was (and could be) described in the usability report.

An example of an instance where she drew on her deep understanding of a problem was when she asked if it was possible to make a certain icon smaller, since it was “overshadowing” nearby icons. This was easily done with the graphic editor, but it was difficult to evaluate if it was small enough. She then directed the developers to go to the particular screen where the problem had been most prominent in the test. This enabled all participants to see that the icon needed to be even smaller and a few seconds later the next problem could be tackled.

Another example illustrates how she applied her expertise in usability testing during the session. After setting relatively complex things like a schedule, the user has to go through an extra step to confirm, cancel or go back. For setting the date and time, it had been decided
this was not necessary. When she tried the function, she asked what would happen if she made a mistake, which was the start of the following discussion.

SUEM If you’re going through the time, and you make some mistake, then that’s just too bad? Or can you go back?
IDC No, then you have to go to the advanced menu.
SUEM Then you will have to get your little user manual.
DLS We could of course have the ok icon…
IDC … and the back icon there.
SUEM You could easily imagine since this is the very first thing they do when they get this one that something can go wrong. That they do what I just did, that they press next too quickly. And then you’re sitting there like: “Oh… already now the first user makes a mistake…” (laughing)

At this point we realized that for the second usability test, we were going to start with setting the date and time instead of the normal home screen, to better simulate the user experience after inserting the batteries into the product for the first time. But, rather than seeing it as something that could go wrong, we saw it more as a warm-up exercise—how hard could it be to set the date and time? However, having conducted many usability tests, she knew that the first task is important since it can either give the user confidence or insecurity for the rest of the test. Without this intervention and the fast update of the screens, this problem would not have been detected until the next usability report. Thus, the active engagement of various stakeholders, in this case to better understand user needs within the context of usability, continued to enable the possibility to bring together different perspectives and shortcut some parts of the prototyping process (Buchenau and Fulton Suri, 2000; Nicolini et al., forthcoming; Terwiesch and Loch, 2004).

Finally, when the end of the workshop was approaching, she also raised the point that, although she felt a lot of the usability problems were going to be solved, the changes might have undesired side effects resulting in other usability problems:

DLS I'm looking forward to seeing the next round.
SUEM Me too, definitely. It is hard to anticipate how much will still be needed. We can sit here and change this and that, but... The things we have changed now, I'm thinking... We are of course working towards eliminating problems, but it is hard to exclude that what we are changing now will introduce new problems. That would be unfortunate, but that is how it will be, and in the end it will help to prioritize some things, so they become more clear. Those are the trade-offs, right.
IDC Then we also know what the focus should be in the user manual, what they can do without and what they need the manual for and show it clearly and step-by-step how to.
The role of the prototype in this setting is referred to as a “sketching tool” by the DLS when discussing this particular workshop in a later reflection:

*It is always exciting to get things tested. It is a difficult process, but also incredibly healthy. It has really given so much, it has. And one could say that, with the way in which we have used the tool as a sketching tool in the beginning, it has also been an easy and fast process in the adaptation towards the tests, and we could also easily correct things between the different tests. We have done multiple sessions of these end-user tests, and so the three of us could sit together and update it quickly in between the tests. So from the one test to the next relatively little time passed.*

It is therefore clear that, even though the prototype helps to create a climate of problem solving and collaboration (cf. Brown, 2008; Ulrich, 2011), there are also constraints in terms of creating a full understanding of the user experience. One of the lessons here is that the collaborative prototyping approach, which at this particular part of the development process only used a usability report and expert to represent the users’ perspective, can mostly shortcut the front-end of the prototyping process, namely design and build, but the run and analyze steps remain the domain of actual use (cf. Thomke, 1998).

*Detecting emerging usability problems through active engagement and experimentation*

After the second usability test, a number of problems were identified that were undesired side effects of some of the changes—as inherent to any design process. In order to explore whether collaborative prototyping during the run step (usability testing) could help detecting these side-effects immediately, end-users were involved in prototyping workshops. The IDC ran four such workshops with potential end-users, and the DLS also ran 10 workshops, using the supportive tools to make modifications to the prototype.

During the first workshop by the IDC, there was an instance of an emerging usability problem as a side effect of a change suggested by the end-user (EU1). The change related to which icon should be the default one to be selected when coming back from the schedule function after making a mistake. The IDC made an update and EU1 explains what he is doing:
EU1: [...] If I make a mistake, for example on Thursday. That I set it too early, and on a weird time period, so it isn't quite right, and I want to do it differently, that I then come back to Thursday. That is nicer. (comes back to “Thursday” and hits enter) Ah… this is confusing...

IDC: Why?
EU1: Because now, look... So Thursday, Thursday, see, this one isn't right. So if I click here, I want to return to Thursday. Yes. But then I have to do this again (press down), and then I can set it again. So just before I clicked too quickly, and then I jumped back too many steps.

IDC: The reason you can't stay on that day is because otherwise you wouldn't be able to come back. So you need an extra step.
EU1: Yes, but this is already nicer... Oh no, wait... See, if I for example make a mistake here, then I have to go back to this one. Yes. But actually it's not so much easier [than before].

IDC: So would it be easier... So, would it be more logical if you return here... (swaps two lines of code) That then the schedule is the default?
EU1: Yes.
IDC: I'll quickly go through this. Can you try it now? If it is more logical?
EU1: Thursday, change it, because I want to go to bed early and I make a mistake. Then if I want to come back, yes, that I'm directly on the clock, so I can adjust it again. Yes, I like this better.
IDC: Okay.
EU1: I think this makes a big difference.

The user and prototyper together make sense of why the change confused him, whether the change is an actual improvement, and how the negative side effect can be addressed. In the course of a few minutes, these problems are effectively solved. What is noteworthy is that in the final design of the interface, there are several (minor) elements that can be traced back to these workshops—i.e., users made contributions to product details.

Another aspect of doing prototyping in the run phase is that the revised prototype embodies the input from users on the usability. The DLS compared receiving an updated prototype with receiving a usability report:

In that case [of getting a usability report] I can also misunderstand the words, and I have to get in and interpret these words in such a report myself and develop my own impression of this user interface. But when I am provided with this piece of software, for example from [the IDC]; [the IDC has] done such a test, and then this modified piece of software comes back. Well then I can directly relate to it and see the result of this test. In that way you save a lot of, firstly time, but also misunderstandings, which can happen with a report.

The DLS, who had no background in interactive prototyping, used the supportive tools, and especially the tool to edit dynamic settings, in collaboration with different internal users. In a later interview:

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.” [...] So it is of course about being able to correct things on-site, 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.

It thus became clear that the involvement of (in this case) a possible end-user was essential in extending the collaborative prototyping approach to all phases of the problem-solving process (Thomke, 1998). It was the active engagement of the user that lead to a better understanding of the relevant experience and related design parameters (Buchenuau and Fulton Suri, 2000; Klemmer et al., 2006), which was enabled by the direct interaction with the prototype and the instant changes by the designer (cf. Mascitelli, 2000; Terwiesch and Loch, 2004). The face-to-face interaction provided an opportunity for mutual improvisation to unstick and share relevant knowledge as well as to increase NPD performance (Bogers and Larsen, 2012; Vera and Crossan, 2005).

Reflection-on-action: Exploring collaborative prototyping

Building on the above-presented action research as reflection-in-action, we now build on those findings and further explore the notion of collaborative prototyping as it has been developed in this particular organization. We first further analyze the three types of prototyping as already described above and then continue our general analysis in order to further conceptualize the main elements and issues within our collaborative prototyping approach.

Three cases of collaborative prototyping

We described three cases of our use of the collaborative prototyping approach within Danfoss Heating Solutions. In each case, the prototype was brought in for a different purpose, based on the practice problem at hand, and was staged in a different way to involve the
relevant stakeholders, which may be within or outside the design team and organization. Table 2 gives an overview of the cases, describing the role of the prototype, the practice problem, research problem, solution concept, specific solution and outcome of each of the interventions (cf. Andriessen, 2007). As such, it gives an overview of the types of problems for which a collaborative prototyping approach can be used, and who to collaborate with.

<table>
<thead>
<tr>
<th>Insert Table 2 about here</th>
</tr>
</thead>
</table>

In the first case, the prototype played an instrumental role in negotiating the features by making the trade-off between functionality and usability very concrete. This enabled the participants to prioritize the features that were most important to have and leave out the features that added a lot of complexity but little value for the consumer. Exposing the prototype to a broad range of stakeholders from the different departments involved in the project at this early stage was important not only to communicate (as in Mascitelli, 2000) but also for them to get ownership over the prototype as they saw their suggestions implemented (Amabile et al., 1996; Siegel and Kaemmerer, 1978). The ability to make suggestions that could be implemented directly in the prototype also played an important role for the part of the specification that was not yet detailed (Boehm et al., 1984) in line with the principles of experience prototyping (Buchenau and Fulton Suri, 2000). Essentially, it created shortcuts in the design and prototype cycles, which speeded up the process because misunderstandings were discovered immediately.

In the second case, the prototype was important as a way to gain a deeper understanding of the usability problems described in the usability report. Involving the usability consultant who conducted the usability test was very important because she knew a lot more about the problems than were described in the usability report. By going through the different screens and making changes, the developers were able to address the problems
(Buchenau and Fulton Suri, 2000). Through the workshop, she could bring in her tacit knowledge and expertise (Mascitelli, 2000). However, design changes have side effects and it was difficult for the internal developers to assess if a change would introduce new usability problems, thus calling for a need to implement the collaborative prototyping approach across a larger set of stakeholders and within all phases of the general problem-solving process (Atuahene-Gima and Wei, 2011; Thomke, 1998).

In the third case, the prototype was evaluated with end-users, and usability problems were addressed as they were discovered. By implementing design changes and evaluating them, emerging usability problems were quickly identified and solved. The resulting modified prototype proved to be an effective way for the other developers to relate to, since they can immediately see the changes. This is more difficult with usability reports, but it should be stressed that collaborative prototyping with users cannot replace regular usability tests, as it is important to confirm how the changes improve usability for a broader range of users than the particular user who suggested it. Moreover, the end-user learns more about the design rationale behind the interface through discussion with the prototyper, which has an influence on their ability to understand it and the perceived usability of the system. Therefore, formal usability tests are complimentary to collaborative prototyping sessions with users, and the collaborative prototyping approach not only gives the designer empathy for the user context (Brown, 2008; Leonard and Rayport, 1997) but also created “reverse empathy” of the user for the design context. However, it should also be noted that not all user involvement is equally productive. More generally, it became clear that there may be varying capabilities and thus roles for different users in collaborative prototyping (cf. Terwiesch and Loch, 2004). For example, one end-user (EU2), who is familiar with traditional prototyping, noted that it requires another attitude and more skills of the user to be involved in such a
collaborative prototyping approach—which we could thus label as turning users from problem-finders to problem-solvers (cf. von Hippel, 1994, 2005; von Hippel and Katz, 2002).

**Conceptualizing the prototyping process**

Building on the above cases, we now further analyze how our collaborative prototyping approach compares to the general notion of prototyping. In the management literature, the prototyping process is conceptualized as an iterative trial-and-error learning process, following four steps: (1) design, (2) build, (3) run and (4) analyze (Thomke, 1998; von Hippel, 2005). The results show that, based on the interactive and experiential characteristics of the collaborative prototyping approach (Buchenau and Fulton Suri, 2000; Mascitelli, 2000; Terwiesch and Loch, 2004), the general steps within the prototyping process happened on two levels within NPD project. First, there is a formal or “managerial” prototyping process, in which the four steps are separate activities that can be planned on the project timeline. Second, there is an informal or “designer” prototyping process, which is more fluid and the four steps are more difficult to distinguish.

In the context of developing a user interface for a product, the managerial prototyping process could consist of the following steps: (1) designing the user interface, (2) making a prototype of the interface, (3) doing a usability test and (4) analyzing the results from the test, and repeat from (1) if necessary (see Figure 2). What we refer to as the designer prototyping process is what happens at the operational level during prototyping when the involved stakeholders may not be aware of which step of the prototyping cycle they are active in.

For example, imagine a prototyper gets a specification of the user interface as the outcome of the design step and has to implement it (normally in the build step). However,
often the specification does not contain all the details necessary for straightforward implementation. An early specification may for instance not specify in milliseconds how fast an icon should blink. The prototyper then has to go through another, much faster and informal, prototyping process. The four steps could be to (1) make an educated guess about an appropriate blink-frequency, (2) implement it in the prototype, (3) run the prototype and (4) evaluate if the frequency indeed is appropriate, and repeat from (1) if necessary. In this particular example, these steps can take a few seconds each, and the prototyper is likely not aware of following a four-step process, but engages in “reflection-in-action” (Schön, 1983) or what Klemmer et al. (2006) call “thinking through prototyping.”

Table 3 provides an overview of the different kinds of learning that happen in the managerial and designer prototyping processes and indicates who is the learner, who is a stakeholder with an interest in what is learned, and how long each trial-and-error learning cycle typically takes in the context of the design of a product interface. Moreover, Table 4 gives an example of how the prototyping steps become fuzzy when going down to the more informal designer level, which deals with the actual prototyping practice.

---

In essence, the collaborative prototyping approach developed in this project activates the informal prototyping process, which then brings together a wide variety of stakeholders who interact and collaborate in a way that shortcuts the general (managerial) prototyping cycles (Atuahene-Gima and Wei, 2011; Mascitelli, 2000; Terwiesch and Loch, 2004). Thus, also in line with the principles of experience prototyping (Buchenau and Fulton Suri, 2000), our collaborative prototyping approach creates a direct and active involvement of various stakeholders in which the respective (discrete) prototyping activities are collapsed into an ongoing and almost continuous prototyping process. Figure 3 shows a representation of how
collaborative prototyping serves as a hub between the design, build, run and analyze activities. The traditional or managerial prototyping cycles are hereby made less discrete as there is an (almost) continuous iteration between these activities through the collaborative prototyping approach.

Collaborative prototyping as a platform for collaboration

The actual practice of collaborative prototyping on the informal level of the problem-solving process, as described above, involves an approach in which the actual prototype acts as a platform for collaboration. As a prototype-driven approach to problem solving within NPD, the results show a number of themes that become relevant in this context. As shown in Table 5—which provides evidence from the workshops and interview—we find that our collaborative prototyping approach especially enables communication and collaboration across different kinds of boundaries. An important element in this is that the prototype serves as a communication tool in which direct interaction is possible, which facilitates the informal practice of prototyping.

In line with Carlile’s (2002) pragmatic view of knowledge and boundaries, the prototype, as a communication tool, acts as a boundary object that can be used to represent, understand and transform knowledge across boundaries. In our case, these boundaries can be between functional areas and between the organization and external stakeholders, which are all important to facilitate NPD and innovation performance (Atuahene-Gima and Wei, 2011; Bogers and Lhuillery, 2011; Leonard-Barton, 1995). In some cases, the prototype also enabled communication and collaboration across hierarchical levels. As a platform for
collaboration, collaborative prototyping thus develops and integrates knowledge that can serve as a source of innovation within NPD, although the actual design will determine that facilitating of hampering nature of the platform (Brown, 2008; Carlile, 2002; Ulrich, 2011). As such, it not only directly triggers cross-disciplinary collaboration and facilitates work across different boundaries but moreover provides the basic infrastructural support of collaboration, thereby addressing the different roles of objects in cross-disciplinary collaboration (Nicolini et al., forthcoming).

Given the general difficulties to transfer and develop innovative knowledge across boundaries, our approach to collaborative prototyping provides an explicit way to overcome interpretive barriers in NPD (Dougherty, 1992), for example by direct feedback and face-to-face interaction in relation to proposed solutions (Mascitelli, 2000). In particular, it was the active engagement of the various stakeholders that lead to a better understanding of the relevant experience and related design parameters, as in line with experience prototyping (Buchenau and Fulton Suri, 2000), while the high fidelity enables the use of the prototype as an interactive tool for development, marketing and sales (Rudd et al., 1996). In a sense, the collaborative prototyping platform created a local community-of-practice (Brown and Duguid, 1991) centered around the prototyping activity and the interface development in particular, which in turn created a collective transformation of understanding through a process of shared meaning and sensemaking (Bechky, 2003; Weick et al., 2005).

Windows of opportunities within collaborative prototyping

The practice of collaborative prototyping—on the designer level—which thus serves as a platform for prototype-driven problem solving, can be described as a process in which the traditional phases within problem solving (see Figure 2) are collapsed into an almost continuous iteration that blurs the boundaries between those phases (see Figure 3). In this
process, the collaborative prototyping approach brings together a wide variety of stakeholders who interact and collaborate to solve relevant problems in the early stage of NPD (cf. Atuahene-Gima and Wei, 2011; Mascitelli, 2000; Terwiesch and Loch, 2004). However, the highly interactive and collapsed/continuous nature of this prototyping process does not mean that all stakeholders are solving problems and adjusting the prototypes all the time. In fact, by relying on several sources—including prototype iterations, event descriptions, emails and other documentation—we constructed a history of interactions and iterations related to the actual prototype, which shows that the overall process, on the managerial level, is highly discontinuous.

Figure 4 shows a timeline with an overview of all iterations that were made in the course of a six-month period within the project. These iterations comprise changes in programming code, structure of the interface, and graphics that are part of segment display. As the figure shows, the iterations tend to be performed in narrow time windows, separated by longer periods of limited or no iterations—much in line with Tyre and Orlikowski’s (1994) finding of “windows of opportunities” within the process of technological adaptation.

Most of these “windows” were triggered by certain events or interactions. The top of Figure 4 (above the chart) shows the main events and interactions—such as workshops, tests, and intensive email exchange—that are often linked to a chain of iterations as triggers or invitations for change, typically preceding the iteration, while in some cases the iterations are done leading up to a certain event.

What is then interesting to observe is that if we investigate the iterations within a short period of time, within one of the windows of iterations, the changes become much less discontinuous—as in line with our proposed process of collaborative prototyping (see Figure 3). Accordingly, Figure 5 shows the cumulative iterations within a specific two-hour period (on 08/25/09) as an indication of the more continuous or step-wise process as enabled by the
collaborative prototyping approach in which the relevant scale is minutes or even seconds rather than days or weeks (cf. Dey et al., 2001; Rudd et al., 1996). Thus, while the designer level of prototyping essentially entails a continuous problem-solving cycle, the managerial level of the prototyping project is in fact more discontinuous.

| Insert Figures 4 & 5 about here |

CONCLUSION

In this paper, we presented a particular approach to collaborative prototyping, which involves the active engagement of various stakeholders across functional, hierarchical and organizational boundaries. Our combined action research and case study of the development of a radiator thermostat at Danfoss Heating Solutions provides a rich description and analysis of how collaboration across various boundaries was needed in order to successfully develop the product. The action research approach was embedded in the challenge to design a user interface on a small product surface, which is easy to use but also offers a high level of functionality to the end-user. The design of the user interface consisted of three stages, which serve as three cases of collaborative prototyping practices, each with particular challenges and involved stakeholders, followed by a further analysis of the overall case.

In terms of developing an active involvement of the relevant stakeholders in this project, it turned out that flexibility and communicability were two important issues in developing the specific prototyping approach. Therefore, a prototype was developed that allowed for rapid changes and many design iterations (cf. Dey et al., 2001; Thomke, 1997; Verganti, 1999). By combining perspectives within design and management research, we conceptualize that breaking up the formal prototyping process into discrete steps that are done by different experts (e.g., a prototype that is designed by interaction designers, built by design engineers, and tested by usability experts) is problematic, because it is a learning
process in which not all the learning can be easily coded and subsequently decoded from the outcome of each step (e.g., a specification, prototype, or usability test). For example, separating making from evaluation in the prototyping process is problematic because the design rationale of the prototyper may be hard to decode by the evaluators and the feedback the prototyper receive through evaluation may be hard to implement, since the evaluators are not aware of the concrete constraints the prototyper has to deal with (cf. Leonard-Barton, 1988; von Hippel and Tyre, 1995).

Our findings suggest that involving these experts in the other steps of the problem-solving process and introducing an element of prototyping, which allows going through full trial-and-error learning cycles, is a way to share this tacit knowledge and integrate it into the developing prototype (Mascitelli, 2000; Thomke, 1998). By focusing on the actual practice of prototyping, we moreover found that the ordinary prototyping process mostly refers to a managerial level whereas our approach activates a designer level in which the entire problem-solving cycle is collapsed into an ongoing, almost continuous process. This approach of collaborative prototyping was used in the NPD project by organizing workshops that facilitate involving various stakeholders across functional and organizational boundaries. In particular, it was the active engagement of various stakeholders with the prototype that enabled an efficient overall prototyping process, in line with the principles of “collaborative prototyping” (Terwiesch and Loch, 2004) and “experience prototyping” (Buchenau and Fulton Suri, 2000). As such, our collaborative prototyping approach enabled a shift from specification-driven prototypes to prototype-driven specification.

The prototype essentially offered a platform to explain and relate various perspectives on the interface that enabled cross-fertilization of knowledge (Atuahene-Gima and Wei, 2011) and thus effectively acted as a boundary object for cross-disciplinary collaboration (Carlile, 2002; Nicolini et al., forthcoming). As such, the collaborative prototyping approach
also enabled the translation of usability problem into actual design changes. The active engagement of various stakeholders, in this case to better understand user needs within the context of usability, continued to enable the possibility to bring together different perspectives and shortcut some parts of the prototyping process (Buchenau and Fulton Suri, 2000; Klemmer et al., 2006; Nicolini et al., forthcoming; Terwiesch and Loch, 2004).

More generally, the prototype helps to create a climate of problem solving and collaboration (cf. Brown, 2008; Ulrich, 2011), although we also identify some constraints in terms of creating a full understanding of the user experience, while the knowledge and skills of the user to act as a problem-solver should also be considered (cf. von Hippel, 1994, 2005). One of the lessons is that the collaborative prototyping approach can mostly shortcut the front-end of the prototyping process (design and build), but that the later run and analyze steps remain the domain of actual use (Terwiesch and Loch, 2004; Thomke, 1998). Therefore, end-user involvement is an important element in extending the collaborative prototyping approach to all phases of the problem-solving process (cf. Bogers et al., 2010).

The case thus shows that the active engagement of the user leads to a better understanding of the relevant experience and related design parameters (Buchenau and Fulton Suri, 2000; Klemmer et al., 2006), which was enabled by the direct interaction with the prototype and the instant changes by the designer (Mascitelli, 2000; Terwiesch and Loch, 2004).

The continued analysis moreover shows that despite the continuous nature of the (designer) practice of prototyping, there are certain windows of opportunities (on a more managerial level) during which the collaborative prototyping approach actually leads to changes in the product design (cf. Tyre and Orlikowski, 1994). Thus, while the designer level of prototyping essentially entails a continuous problem-solving cycle, the managerial level of the prototype and NPD project is in fact more discontinuous, triggered by certain events of interactions that act as invitations for change.
IMPLICATIONS

Our results have several implications for prototyping theory and practice. Most generally, our conceptualization of the prototyping process at different levels, comprising a managerial and designer level, brings together literature from both design and management to work towards an integrative model of the prototyping process. In particular, the prototype-driven problem-solving process as an informal practice implies an important role of the active engagement of various stakeholders, thereby integrating the principles of “collaborative prototyping” (Terwiesch and Loch, 2004) and “experience prototyping” (Buchenau and Fulton Suri, 2000) as well as the more general principles of prototyping (Schrage, 1996; Thomke, 1998; Ulrich and Eppinger, 2008) and (participatory) design (Hartmann, 2009; Leonard and Rayport, 1997; Schuler and Namioka, 1993; Ulrich, 2011).

Making a distinction between the designer and managerial prototyping process is important because there is a difference in how internal stakeholders can be brought into the process. In the managerial prototyping process, it is relatively easy to involve for example a large team in a brainstorming session in a build step, or do an internal evaluation of the product with top management in a run step. This is more difficult in the designer prototyping process, because it happens fast and requires technical expertise to be involved in. However, in the designer process, many small decisions are made on the exact implementation, and the prototyper learns what the implications are of design decisions and becomes familiar with the design space and constraints. This knowledge is “sticky” in that it is not easily transferred by exposing others to the prototype that comes out of this process (von Hippel, 1994). In line with Mascitelli (2000), face-to-face interaction between relevant stakeholders in NPD can enable creative improvisation and real-time knowledge sharing. Involving others in the informal prototyping then would be a way for them to learn about the design space, constraints and implications. This could be done, either by involving them in this process
during the build step, or by introducing a degree of prototyping in the design or run steps since the informal process consists of the same four steps. Prototyping could be a part of for example team brainstorming sessions or evaluation sessions.

The early phase of the NPD project that we studied shows that there is always a balance between the specification and the prototype in the sense that there were a number of features that were set at the outset but otherwise many design parameters were left open (cf. Boehm et al., 1984). Starting prototyping before and during the development of the specification is especially relevant for products with interfaces. Prototype-driven problem solving allows the evaluation of the usability of an interface and makes the trade-off between adding features and adding complexity very concrete. This can help specifying a feature-set, which does not only look good on the box, but also adds most value for the customer in use.

Another aspect of product interfaces is that they often include dynamic aspects, such as how fast icons blink or how long a backlight stays on after the last interaction. Although it is possible to estimate appropriate times for a blink-frequency or a backlight timeout, finding a good value is often a trial-and-error process. When the prototype is used to help define the feature-set it will be important to not only involve stakeholders such as interaction designers and usability experts but also for example marketers and the product portfolio director.

Our study moreover has important implications for managers and academics alike who are interested in collaborative prototyping as a way to effectively involve not only external stakeholders such as users (Bogers et al., 2010; Poetz and Schreier, 2012; Terwiesch and Loch, 2004) but also internal stakeholders across hierarchical and functional boundaries (Adler, 1995; Buur and Matthews, 2008; Song et al., 1998). Furthermore, we show the importance of both separating and linking the formal managerial level of prototyping and the informal practice of prototyping where the actual prototype acts as an important boundary object for communication (Carlile, 2002; Nicolini et al., forthcoming). Collaborating with
various experts in different steps of the (formal) prototyping process in prototyping workshops enables the different participants to unstick their knowledge, which is directly implemented in the evolving prototype. However, we have also identified some constraints in prototype-driven problem solving, while our finding of windows of opportunities at the level of the prototype project emphasize the triggers or invitation for change as well as that the periods of actual change are characterized by more continuous adaptation—thus reinforcing the need to consider different levels of the problem-solving process.

This approach is most relevant for managers in NPD projects where prototyping starts at an early stage, and who are managing a cross-functional team (Song et al., 1997; Ulrich and Eppinger, 2008). We provide problem-classes that this approach successfully addressed in this particular case, although we must note that the results from this case study are not yet “field-tested and grounded technological rules” (van Aken, 2004). In the terms of van Aken (2004), we are now at the stage of α-testing where the solution concept has been tried in multiple prototyping cases within a single company context. In subsequent research, the technological rules should be applied to practice problems in different cases and by third parties to do β-testing until theoretical saturation is reached (Eisenhardt, 1989) to establish field-tested and grounded technological rules.

More broadly, out inductively derived framework could be further developed and applied within the context of a number theoretical frameworks and perspectives. On the one hand, there are a number of perspectives that we draw on or develop in our analysis that could be further elaborated in future research. For example, we extended Terwiesch and Loch’s (2004) definition of collaborative prototyping by not only focusing on the user-producer interface but instead we developed a more general multi-stakeholder view on collaborative prototyping, which can include any stakeholder from within and outside the organization. As such, it would be useful to link our collaborative prototyping approach to
perspectives that deal with the involvement of a variety of stakeholders in NPD within more open innovation models (Adner and Kapoor, 2010; Chesbrough, 2006; Dahlander and Gann, 2010). We embedded a multi-stakeholder approach within the concept of experience prototyping, which enables involved stakeholders to gain first-hand appreciation of existing or future conditions through active engagement with prototypes (Buchanan and Fulton Suri, 2000; Klemmer et al., 2006). This approach to prototyping, as a way to design an integrated experience to better understand, explore or communicate what it might be like to engage with the product that is being designed, could be further explored in future research. Moreover, a particular finding in this context is that the collaborative prototype integrates different roles of boundary objects (Carlile, 2002; Nicolini et al., forthcoming), which offers a basis for further exploration of the mutual benefit of collaborative prototypes and boundary objects.

On the other hand, there may be value in applying some of our finding to a number of established or growing research streams. For example, different areas of design may be affected by a better understand of collaborative prototyping, including product design and organization design (Tushman and Nadler, 1978; Ulrich, 2011), also with possible reference to the concept of modularity (Baldwin and Clark, 2000; Sanchez and Mahoney, 1996). Moreover, it may be useful to integrate our conceptualization of collaborative prototyping into the complementarity between intra- and inter-organizational sources and processes of innovation (Bogers and Lhuillery, 2011; Hillebrand and Biemans, 2004; Neyer et al., 2009). Finally, our perspective on collaborative prototyping could in some way inform how organizations build innovation capabilities and routines, in line with the recent call more research into the micro-foundations of such processes (Felin and Foss, 2005; Lewin et al., 2011).
REFERENCES


APPENDIX

Acronyms of Involved Stakeholders and Participants:

- CTM: Communication Technology Manager
- DE: Design Engineers (DE1 & DE2)
- DLS: Design Line Specialist
- EU: External User (EU1 & EU2)*
- GW: Global Webmaster
- IDC: Interaction Design Consultant†
- INT: Innovation Intern
- ME: Marketing Engineer
- PPD: Product Portfolio Director
- RDD: R&D Director
- RDPM: R&D Project Manager
- RDSD: R&D Senior Director
- SUEM: Senior User Experience Manager*
- UEC: User Experience Consultant*

* External stakeholder
† Researcher-designer
TABLES

Table 1: Overview of Data Sources

<table>
<thead>
<tr>
<th>Data source</th>
<th>Amount of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prototype iterations (total)</td>
<td>(70)</td>
</tr>
<tr>
<td>Shared with larger team</td>
<td>25</td>
</tr>
<tr>
<td>Developed in workshops</td>
<td>13</td>
</tr>
<tr>
<td>Emails (total)</td>
<td>(60)</td>
</tr>
<tr>
<td>Design Line Specialist</td>
<td>23</td>
</tr>
<tr>
<td>Interaction Design Consultant</td>
<td>23</td>
</tr>
<tr>
<td>Intern</td>
<td>9</td>
</tr>
<tr>
<td>R&amp;D Project Manager</td>
<td>3</td>
</tr>
<tr>
<td>Other</td>
<td>2</td>
</tr>
<tr>
<td>Documentation (total)</td>
<td>(22)</td>
</tr>
<tr>
<td>Click-thru scenarios</td>
<td>13</td>
</tr>
<tr>
<td>Design change proposals</td>
<td>4</td>
</tr>
<tr>
<td>Usability reports</td>
<td>3</td>
</tr>
<tr>
<td>Organization structure</td>
<td>2</td>
</tr>
<tr>
<td>Video material (total)</td>
<td>(24 hours)</td>
</tr>
<tr>
<td>Workshops</td>
<td>9 hours</td>
</tr>
<tr>
<td>Usability tests</td>
<td>15 hours</td>
</tr>
<tr>
<td>Interviews (total / time)</td>
<td>(5 / 3 hours)</td>
</tr>
<tr>
<td>Design Line Specialist</td>
<td>1 hour</td>
</tr>
<tr>
<td>R&amp;D Project Manager</td>
<td>30 min.</td>
</tr>
<tr>
<td>R&amp;D Senior Director</td>
<td>30 min.</td>
</tr>
<tr>
<td>Global Webmaster</td>
<td>30 min.</td>
</tr>
<tr>
<td>Product Marketer</td>
<td>30 min.</td>
</tr>
</tbody>
</table>

2 Most data is available in Danish, some in English, and some fractions in Dutch (depending on the people involved). Audio and video material was transcribed in the original language and translated into English for the analysis. Thus, all quotations used in this paper were translated into English (if not in English already).
<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role of prototype</td>
<td>Communication tool</td>
<td>Sketching tool</td>
<td>Testing tool</td>
</tr>
<tr>
<td>Practice problem</td>
<td>When specifying the features the company was afraid that they would add too much complexity, which would have a big negative impact on usability</td>
<td>The usability report, which described a number of usability problems, did not provide concrete directions on design improvements</td>
<td>After the second usability tests a number of usability problems were detected that were side effects of “improvements” we made to the prototype</td>
</tr>
<tr>
<td>Research problem</td>
<td>How can the impact of features on usability be evaluated when the feature-set is still flexible?</td>
<td>How can the usability report be translated into design improvements?</td>
<td>How can undesired side effects and emerging usability problems be detected when “improvements” are implemented?</td>
</tr>
<tr>
<td>Solution concept</td>
<td>Start prototyping before the feature-set is finalized to talk about concrete features and their impact on usability</td>
<td>Involve the usability tester who wrote the report in prototyping during the design step</td>
<td>Involve end-users in prototyping during the run (test) step</td>
</tr>
<tr>
<td>Specific solution</td>
<td>Prototyping workshop with development team, usability consultants and management represented</td>
<td>Prototyping workshop with the developer, interaction design consultant and usability consultant</td>
<td>Prototyping workshops with end-users, implementing and evaluating design suggestions during the workshops</td>
</tr>
<tr>
<td>Outcome</td>
<td>A revised prototype (and revised specification), reflecting the input from all stakeholders, which could be used in a usability test</td>
<td>A revised prototype addressing the usability problems detected to be used in the next usability test</td>
<td>A revised version of the prototype embodying the feedback from the users, which could be shared with the development team</td>
</tr>
</tbody>
</table>
Table 3. The Managerial and Designer Processes in Collaborative Prototyping

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Managerial prototyping process</th>
<th>Designer prototyping process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning</td>
<td>Usability and performance of the product interface</td>
<td>Familiarity with the design space, constraints, and implications of decisions on usability</td>
</tr>
<tr>
<td>Knowledge</td>
<td>Typically codified (e.g., usability report)</td>
<td>Typically tacit or sticky (costly to transfer)</td>
</tr>
<tr>
<td>Learner</td>
<td>Development team</td>
<td>Prototyper(s)</td>
</tr>
<tr>
<td>Stakeholders</td>
<td>Management</td>
<td>Development team and management</td>
</tr>
<tr>
<td>Full learning cycle</td>
<td>Few weeks per iteration</td>
<td>Few minutes per iteration</td>
</tr>
</tbody>
</table>

Table 4. Exemplifying the Prototyping Process and Practice

<table>
<thead>
<tr>
<th>Phase</th>
<th>Prototyping process</th>
<th>Prototyping practice</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Build phase</td>
<td>Run phase</td>
</tr>
<tr>
<td><strong>Design</strong></td>
<td>Develop the interface specification</td>
<td>Can we make this icon smaller?</td>
</tr>
<tr>
<td></td>
<td>Make a document with suggested changes and improvements</td>
<td></td>
</tr>
<tr>
<td><strong>Build</strong></td>
<td>Implement the specification or changes into a simulation of the interface</td>
<td>Make the icon smaller</td>
</tr>
<tr>
<td><strong>Run</strong></td>
<td>Conduct a usability test with end-users</td>
<td>Run the simulation, go to a specific screen</td>
</tr>
<tr>
<td></td>
<td>Run an internal test</td>
<td></td>
</tr>
<tr>
<td><strong>Analyze</strong></td>
<td>Identify usability problems and document them in a usability report</td>
<td>Assess if the icon is the right size</td>
</tr>
</tbody>
</table>
## Table 5. Collaborative Prototyping as Platform for Communication and Collaboration

<table>
<thead>
<tr>
<th>Element</th>
<th>Evidence from interviews</th>
<th>Evidence from workshops</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prototype as communication tool</strong></td>
<td>DLS</td>
<td>At a very early stage […] it has been a really strong tool, where we were able to exploit the flexibility of it, precisely by using it early on. In addition it was [used in] the whole communication part, which turned out to be very important. And we have used it a lot as an internal facilitator to get people to get a thorough understanding of what concept this is, and what kind of product it is, also at a very early stage.</td>
</tr>
<tr>
<td></td>
<td>IDC</td>
<td>No, then you have to go to the advanced menu.</td>
</tr>
<tr>
<td></td>
<td>DLS</td>
<td>We could of course have the ok icon…</td>
</tr>
<tr>
<td></td>
<td>SUEM</td>
<td>You could easily imagine since this is the very first thing they do when they get this one that something can go wrong. That they do what I just did, that they press next too quickly. And then you’re sitting there like: “Oh… already now the first user makes a mistake…” (laughing)</td>
</tr>
<tr>
<td><strong>RDPM</strong></td>
<td>It is always good to use tools and information that are as easy to understand as possible. If I had written this in words – you know, one picture tells more than a thousand words. That was very good to use that as communication material.</td>
<td></td>
</tr>
<tr>
<td><strong>Cross-functional communication</strong></td>
<td>RDSD</td>
<td>It has also been used as a communication tool towards our sales people, sales and marketing. So 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.</td>
</tr>
<tr>
<td></td>
<td>DLS</td>
<td>No, that is something we decided not to do</td>
</tr>
<tr>
<td></td>
<td>PPD</td>
<td>Yes. Yes, that is something we have to include.</td>
</tr>
<tr>
<td></td>
<td>RDSD</td>
<td>So if I understand what DE1 is saying, you want to be able to put it down to 17 degrees on Monday evening and to 18 on Tuesday evening, and…</td>
</tr>
<tr>
<td></td>
<td>DLS</td>
<td>What adds value? […]</td>
</tr>
<tr>
<td></td>
<td>DE1</td>
<td>I think it is much more important to have the freedom to set the period from this time to this time where we should lower the temperature.</td>
</tr>
<tr>
<td></td>
<td>DLS/DE1</td>
<td>In any case there is the problem with this trade-off on the user-friendliness. […]</td>
</tr>
<tr>
<td><strong>Collaboration across organizational boundaries</strong></td>
<td><strong>RDPM</strong></td>
<td><strong>EU1</strong></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>In the beginning the purpose of using the prototype was to develop a user-friendly User Interface. That was in the beginning of this prototype, but because this prototype was so good and excellent we used it in different ways. Then we used it also for communication towards customers; possible, potential future customers. And we also used it for how to do the instructions, user guides.</td>
<td></td>
<td>[...] if I for example make a mistake here, then I have to go back to this one. [...]</td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FIGURES

Figure 1. The First Virtual and the Final Physical Prototype

Figure 2. Managerial Collaborative Prototyping Process (adapted from Thomke, 1998)

Figure 3. Designer Collaborative Prototyping Process as Collapsed Problem-Solving Cycle
Figure 4. Timeline of Prototyping Iterations on Managerial Level (Six Months)

Figure 5. Cumulative Prototyping Iterations on Designer Level (Two Hours)