

# Programming in the Kitchen

**Olav W. Bertelsen, Toke Eskildsen & Werner Sperschneider**

Department of Computer Science, University of Aarhus, Denmark

olavb@daimi.au.dk

**Abstract:** In this paper we discuss issues for the integration of computing power in mundane artefacts. We do so with reference to a recent project with a manufacturer of ovens for larger kitchens and catering services. We introduce three levels of obstacles when integrating computers into mundane tools and point out how organisational issues are a source of interface problems. We conclude that the design of computer-based interfaces to mundane tools should encompass broader studies of the context of use and the history of the considered artefact.

**Keywords:** Field studies, prototyping, information appliance, ovens in large kitchens, embedded interfaces.

## 1 Introduction

When mundane artefacts embed extensive computer power, the boundaries of human-computer interaction are widened. In this paper we look into human-computer interaction issues occurring when extensive computing power is embedded into a commercial oven. We focus on users' programming the oven, which is the single most problematic issue we observed in a series of field studies and prototyping. We observed a basic tension between ovens as physically understandable devices and the flexibility enabled by introducing computer-based features. To understand this tension we introduce three classes of obstacles to user programming: personal and cultural resistance, interaction issues and organisational issues.

The cooperation project was initiated by a local manufacturer of ovens (70 employees), motivated by the fact that its existing line of interfaces suffered from being overcrowded with functions added over the years, and not having a modern appearance. Previous efforts to redesign the interface had been terminated before they were finished; therefore the manufacturer decided to team up with an external partner to complete the redesign. The overall design challenge was to make an interface that supports programmability and other advanced features, and at the same time looks like an oven.

Project time lines were tight, and several basic decisions regarding the new interface had already been made. Most importantly, the manufacturer had

decided to use a colour LCD display and a row of 6 soft-keys at each side of the display as the platform for the interface. This limited design space made it more likely for the manufacturer to get the new oven ready on time. As researchers we saw these limitations as an interesting challenge. Activities and method

Among project activities, two types proved to be particularly fruitful. Firstly, we conducted field studies of the use of the present production model line as well as of the use of competitor products. Secondly, we developed prototypes reflecting the changing design specification, and tested these prototypes with users both in the lab and on site.

Disregarding the fact that a new design had already been developed, our first activity was a general study of ovens in use. The aim of this was to find out what kind of device an oven is, to transcend the narrow idea of the oven as an assemblage of functionality. We conducted field studies in three types of settings defined by the manufacturers' product line. Ovens for *Bake-off*, aimed for in-store bakeries in supermarkets, kiosk and gasoline stations where pre-prepared pastry is finished, have dedicated buttons for activating programs for each pastry type. Combi-steamer ovens are aimed for two very different environments, for *restaurants/catering services*, and for *institutional kitchens* like hospitals. Five full days of observations were conducted at 8 sites, involving 11 users, over 3 weeks in early 2002. We observed daily routines with the oven and conducted interviews with users at small shops, can-

teens, restaurant kitchens and large institutional kitchens. Approximately 10 hours of video were recorded during field studies and subsequently logged, transcribed and analyzed; in addition a few pages handwritten notes were taken for a field diary. Interviews were open-ended focusing on day-to-day work with the oven. As part of the development project, studying ovens in use was important as a way to establish a pool of reference points in the further construction process.

The other main activity was the continuous construction of a prototype reflecting the evolving state of the manufacturers specification. Normally, in a user-centred design project, we would have based the first prototype on insights from field studies and other interaction with users. However, in this project we started by a straight implementation of a specification developed by the manufacturers' electronic engineer in collaboration with a consulting graphical designer. We did so because the manufacturer had decided on the specific platform, and also because we wanted to try to work within this sort of constraints.

In parallel, various design drafts were produced aiming to highlight and solve the major issues found in the field studies. The first versions of the prototype were operated with a mouse on an ordinary desktop computer. In the next round we built a mock-up of an oven front with real keys, and a laptop computer hidden behind the front only showing the part of the screen corresponding to the area of the LCD screen planned for in the new oven. This mock-up was brought into the field and tested with ordinary users in their own context.

Through prototype testing in the field more issues were identified. An iterative process took form in which the initial field studies, the original design specification, the insights from implementing the specification and the field tests interacted in the refinement of the evolving interface.

Prototypes were evaluated using a thinking-aloud approach in context. Users spent approximately 30 minutes using the prototype to solve tasks specified by us. A minimum help approach was applied. Each test session was recorded on video and analyzed informally the day after. Subsequently, a more formal logging was produced in order to systematically identify problems, and their relation to each of the three types of setting; possible improvements of the interface; and interesting research issues. A total of 13 tests were conducted on sites. Pre-test observations lead immediately to a re-design of the first prototype. The test of prototype II was

terminated after 4 rounds. Finally prototype III was tested with 7 users.

## 2 What is an oven?

To understand how ovens are used we had to get a broader and more detailed understanding of the oven. An oven is a hot place where raw materials are transformed into food. However, in a cultural historical perspective, the oven should be understood in terms of its function in civilisation, within the history of cooking, catering, etc. Before gas came to the kitchen, food was made on open fire. Seen in the long historical perspective the separation of the baking/heating/frying functionality into a dedicated unit was a major step, in the same sense the introduction of automatic control, e.g. programmability is currently changing the oven.

Ovens heated by gas and later electric power made cooking easier. Meat roasting and bread baking are prime examples of this. With later expansions of the oven capabilities, e.g. the introduction of steaming, the oven gradually shifted from being an appliance for specific tasks to be the general instrument for cooking the food. The oven didn't require the same degree of monitoring as pots and pans, which gave the user more time for other tasks. Effort is delegated to the oven; when meatballs are finished in the oven it is not as likely that they get burned.

The addition of timers and programming make room for further delegation of responsibility. The kitchen leader can specify a recipe and be sure that the oven will treat the food as he intends, even if he is not present. The appliance takes responsibility.

The evolution of the oven is significant compared to other appliances in large kitchens. Electronic weights and thermometers, steam-based boilers and automatic dishwashers were all improvements to existing technology, but they didn't change the basic structure of a kitchen. The oven has gradually become a place where nearly all hot food passes, at least to keep it warm. In one institution kitchen, the stove was only used for making sauce in a pot. Delegation of skill and responsibility into the oven also made it possible to introduce food preparation into areas lacking trained personnel, most prominently the bake-off sector.

Cleaning an oven was until recently a messy affair. The strong and constant heat burns fat and food parts to the inside of the oven. Traditionally, the cleaning involved strong detergents, facemasks and rubber gloves. This part of kitchen life has been eased with automatic cleaning systems.

Danish law requires still more detailed reports from large kitchens, imposing extensive measurement and documentation of temperature when heating and cooling the food on the personnel. Advances in oven technology makes it possible to automate large parts of this data collection, hence giving a further incentive to use the oven rather than the stove.

In short the oven has developed from open fire, to be a special appliance for a few special tasks like baking bread, and further into the centralised point of passage in the institutional kitchen. Problematic aspects of this development include that it is harder for learners to develop a sense of when the food is well done, and that it might be more difficult to place responsibility.

### 3 Studies of present oven use

In the field studies users referred to a broad range of things they did with the oven as “programming”: making recipes step by step in manual mode, by selecting a program by scrolling or pressing buttons, or by choosing additional features; using the timer function; and even using the cleaning program.

In the following sections we describe five observations of use of the present ovens, observations involving programming. Person names are changed.

#### 3.1 Augmenting with paper notes

Often users make hand-written guides or instructions with a number of how-tos. Such guides are placed near the oven. Handwritten annotations of guides or instructions have a tangible quality, which is impossible to transfer to an LCD display. A paper sheet guide is always at hand, also when the oven is running. If there is doubt about a parameter, it could be looked up. If a colleague asks “Are you really sure about the duration?” the proper setting could be looked up. This quality is not given on an LCD.

Basic instructions give security, e.g. in case the person normally responsible is not at work. Especially, pre-fabricated foods e.g. bakery products, vary in quality, or size, calling for adjustments of

temperature or duration. Such variations and changes are handled in a straightforward manner as annotations on printed sheet of paper.

#### 3.2 Tailoring programs

Programming a recipe into the oven is, from a technical point of view a simple thing. However, from the point of view of kitchen practice it might be considered extensive tailoring. John, a restaurant cook, deleted all “programs” stored and delivered by the manufacturer. John doesn’t fancy pre-programmed recipes. He made his own “programs”. All in all he has made five programs, each of them with about 3-4 steps. None of his assistants are allowed to make changes. To John consistency, reliability is the main concern. From his point of view consistency is the strength of the oven as a computing appliance.

#### 3.3 Reluctance to change programs

Some users of the bake-off ovens regard the program for warming up the oven less usable. The “warming up program” heated the oven to 225°C, assuming that the temperature will drop by 40-60°C when the door is opened. However, in practice users experience a much less dramatic drop. Furthermore, the heating time is not as long as assumed by the manufacturers program. Users have developed ways to work around the “warming up program”. They interrupt heating by opening the door when 180°C has been reached, or by simply shutting down the oven and restarting it immediately. Changing the “warming up program” to deal with this problem was not considered by the shop assistants we talked to. To cope with pre-programmed recipes they would rather adjust duration or temperature manually every time. They would develop workarounds rather than fix the problem by dealing with programming.

#### 3.4 Programming for flexibility in use

Users perceive programming to be a means for solving site-specific problems. Christine, a bake-off shop assistant says: “It’s difficult to make a program. There should be better default programs. We should be able to select our own menus/recipes. And there



Figure 1: The present production model.



Figure 2: The physical prototype interface.

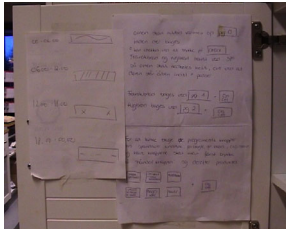


Figure 3: users notes augmenting the oven

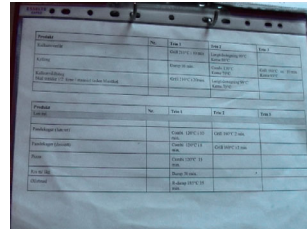


Figure 4: Binder holding the programs

should be two-step-programs, with an alarm after the first step; so we could bake white bread and rolls at the same time". If it was easy to do, she imagines she would tailor the recipes to better fit with her daily needs.

### 3.5 Re-entering every time

Beth, working in a hospital kitchen, says she "programs" an oven when using more than one step for a recipe. All recipes applied in this kitchen are written down on paper and stored in plastic covers in a binder. The binder holds 55 recipes for all kinds of meals to prepare in the hospital kitchen. Some of these recipes specify a 2 or 3 step procedure. The recipes are typed on a computer and printed out.

Beth takes this printed recipe from the binder, approaches the oven, and starts programming according to the records made by either herself or a colleague. Beth does so whenever she needs to set more than the basic features time and temperature. If she keys in a second or third menu/recipe step, she says, she "programs" the oven. So she actually uses the programming-function manually, because such programmes are not stored in the memory. She is well aware that she could. But she doesn't do so.

From Beth's point of view, programming starts when keying in a program, that is, when writing a recipe in more than one step. This is what, the computing appliance in an oven is good for: you can tell it to do something while you don't supervise the machine and are free to do something else. Everything else, e.g. storing recipes, is "advanced" and thus not part of her programming routine. Beth puts faith in what is written down, in what she can see. So do other oven users, the shop assistants, the supermarket assistants, and the cooks of several institutional kitchens we visited.

## 4 Oven Programming

Programming is the single aspect of the oven that complicates the design of the interface. Programming enables users to apply predefined sequences of heating, cooling, timing etc. From the very beginning of our field studies we were puzzled by the fact that everybody seemed to agree that the features for programming the oven could be a great help, but that almost none of the users actually used these features. They seemed to perceive programming to be very difficult or even painful. The field studies show that users are reluctant to "program" their ovens, even at hospital kitchens, where the same recipes are prepared repeatedly. They all use manual run.

"It's too complicated. It's the same as with the computer. I haven't had the time for dive into this

stuff [the manual]. I know there are courses offered. But when should I participate? Probably I'll do it one day." Another voice: "Well my job is to prepare food. It should be on the table at noon. Later the mess needs to be cleaned up. Then I am ready to go home. So, you ask me why I don't use the computer function?"

The concept "programming" seems to mean different things seen from different perspectives. Consequently, the question is: what is programming, and how do we address programmable features in an appliance, which is not to be experienced as a computer on the first hand? The notion of programming is loaded with connotations. From a technical point of view, programming is a tool for performing tasks and processing information. For a cook programming is a "computer" thing, and the cook is a "chef de cuisine", not a programmer. Programming the computer is different seen from different perspectives. In the users' perspective programming is a scary computer feature; however, at the same time acknowledged as convenient automation of steps in a recipe. In contrast the manufacture's perspective is that programming is a technical feature that unproblematically makes it easier to use ovens and also a feature that makes them sell better. Finally, the researchers' perspective is that oven programming is about storing and reusing sequences of machine operations, programming is not always something users can do, there might be all sorts of obstacles for the users when programming.

End-user programming has been a recurring theme in HCI since the early eighties (Nardi 1995); and programming languages like SIMULA and Smalltalk reflected concerns for tailoring by end-users. Mørch (1997) makes a distinction between three levels of end user tailoring from "end user customization", over "end user integration" to "end user extension". Oven programming can be seen as a form of integration, compared to macro programming in desktop systems. Our observations consistently show that users have problems with programming. We hypothesized that it was a problem that the users thought the oven looked too much like a computer, inducing an uneasy sort of computer fear. But we also knew that some of the users were using computers regularly, and we knew that common household appliances like the VCR could be very hard to program because the interface was too complicated. Thus, we assumed, in accordance with Mackay (1991), that the problem could be solved with a better interface. However, as we analysed more closely the field data we discovered a third class of obstacles to easy adoption of oven pro-

programming features. Those obstacles came from the fact that ovens are used in an organisational and cultural context, and that introducing programmability fundamentally displaces (or disturbs) those contexts.

In sections 7, 8, and 9 we discuss three classes of obstacles for programming. Firstly, *personal and cultural resistance* is seemingly irrational issues like profession specific idiosyncrasies, stylistic preferences, and computer fear. Secondly, *interaction issues* relate to poorly designed interface and awkward interaction - the kind of issues normally dealt with in HCI. Thirdly, *organisational issues* cover relations in the work setting that have to be considered and supported in the interface, e.g. division of labour and responsibility.

## 5 Prototyping the new interface

By testing the prototype we learned that the new interface indeed solved problems observed with the old interface.

The prototype discussed below was the result of three major iterations of design, implementation, and testing. While it didn't represent the blueprints for a final product, it did convey most of the major decisions regarding interaction with the oven.

The prototype was designed to be very true to a real-life oven. Physically the prototype was mounted on the back of a metal plate with a hole for the LCD panel, resembling the front panel of an oven for large kitchens. The software permitted nearly all intended operations on an accelerated timescale, complete with audio alarms and programming capabilities. The interface made use of a LCD panel surrounded by 12 soft-keys, a "recipe button" and a start/stop-button.

The manufacturers' experience from years of manufacturing ovens implied that new features

would be added over time. Their latest ovens suffered badly from the fitting of extra functions to existing buttons, by introducing counter-intuitive modes of operation, like holding down a button for a brief period in order to make it activate a secondary function. The use of soft-keys provided flexibility for future changes in design and allowed for later upgrades of sold units.

The interface consistently used 5 rows for manipulating the elements on the LCD and a single row for navigation. The navigational row also contained the status for the current operation. The manipulation of elements made use of the standard western connotations for left being less and right being more. By pushing the soft-key to the left of the temperature, the temperature was lowered and vice versa. The tests showed that the participants easily grasped the concept.

Roughly, three modes of operation were available to the users: Manual run, recipe selection and programming. For restaurants and institutional kitchens, the manual run was the starting point. For bake-off it was recipe selection. The switch to recipe selection was initiated by pressing the recipe button (lower left). Pressing the button with the recipe selection already displayed switched to manual run.

All three modes contained too much information to be shown on one screen. The solution was to use navigational soft-keys (row 6). Tests showed that users became confused by sudden changes; therefore smooth scrolling was introduced. Instead of having to grasp the connection between multiple pages, it was possible to perceive them as just one long page.

*Manual run* used the first horizontal line for selection of mode and the next four lines for the parameters relevant to the mode. In this prototype modes could be normal hot air, steam, combined hot air and steam, reheating of food and raising.

*Recipe selection* lets the user select from a list of recipes. It also allowed for the selection of special

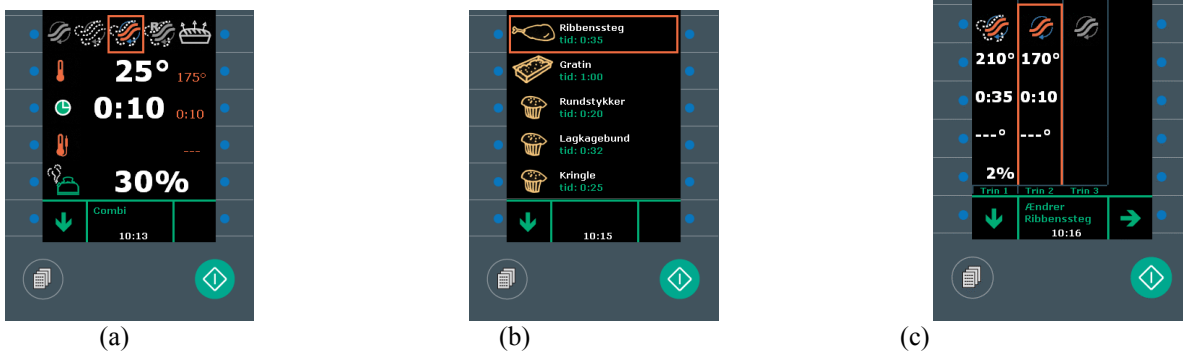


Figure 5: Prototype displays. (a) Manual run, (b) Recipe selection and (c) Programming.

functions like heating the oven, a timer and an automatic cleaning-system. The mixture of special functions and recipes was a cause for great discussion, but we found that it was preferable to the introduction of a fourth mode.

*Programming* presented the individual steps in a recipe as columns. The position of the parameters in each column was the same as for manual run. A red box was used as a marker to specify the current step for manipulation.

Direct manipulation was an ideal for the interface: If an element was visible, it could be manipulated by pressing the corresponding soft-key. This proved to be a challenge for the programming section. The principle could have been upheld by using a full panel for each step in the recipe, but this would mean the loss of breath of view. Showing all primary parameters for a full recipe at a time allowed the user to check for errors. This recipe control is especially valuable in kitchens with multiple oven users and changing recipes. It reduces the need to trust the oven to be untouched by other users.

## 6 Personal and cultural resistance

After doing the first round of field studies, our impression was that the most prominent obstacles to the adoption of advanced features of current ovens were related to an unwillingness to deal with a tool resembling a computer. It seemed irrational but it was not a big surprise.

Older users without computer experience, seemed intimidated by the computer-like features. Some users, particularly the restaurant cooks, told us directly that it was not a part of their professional culture to deal with computers; they were only interested in preparing food. In other words we saw both personal resistance resembling computer fear, and cultural resistance related to professional pride. However, we did also meet users for whom the resemblance with a PC was helpful when getting to understand the interface of the oven. Thus, one user commented during prototype testing: "oh, I see, the arrow for navigating is just like on the computer".

Upon closer analysis these (irrational) issues turned out to be hard to deal with, but also less important to address directly. General uneasiness with computers cannot be solved through interface design, but uneasiness caused by difficulties in understanding basic principles of the computer-based artefact can be addressed by a comprehensively laid out interface revealing the basic model. Resistance originating in lacking fit between the oven interface and

the professional culture tend to be easier to deal with if understood as an interaction issue. Restaurant cooks repeatedly praised a physical and direct interaction form; they preferred to turn knobs rather than pressing buttons when e.g. adjusting temperature the pressing a button is less direct than turning a knob because the button does not provide direct haptic feedback. This means that the user is dependent on visual feedback, which may indeed conflict with cooks' very physical navigation in the chaotic kitchen environment. In other words it is not merely a cultural idiosyncrasy but can as well be understood as an interaction issue about not breaking the flow or rhythm of work. It is far more idiosyncratic when cooks express strong objections to the whole idea of delegation; in particular because it conflicts with the actual history of oven use in large kitchens.

## 7 Interaction issues

Ovens restricted to heating, steaming and simple timing typically used three knobs or three pairs of buttons: A mode selector, a temperature selector and a time selector. Parameters were etched directly on knobs or displayed on LEDs adjacent to relevant buttons. A concept easily grasped.

Until recently, programmability advanced timing, customizability and more was implemented in most ovens by extending these mechanisms. Some ovens also added tiny character-oriented LCDs.

This extension causes two big problems: awkward interaction and lack of overview. The lack of overview is unavoidable, unless more advanced display technology is introduced. Most manufactures of industrial ovens introduce LCD displays in their new models, possibly as much for marketing as for usability reasons.

Awkward interaction emerging from the addition of functions was typically addressed by grouping. One such grouping was the separation of core features (heating and steaming) from advanced features (programming, timing and more) by juxtaposition. The empirical studies showed that the separation made sense in a backwards way: The core features were used daily and the advanced features were generally ignored.

The lack of overview was mostly connected to programming. The selection of a program was performed by selecting a number or by pressing a few keys in a specific order. This required the use of a physical list with the numbers for the recipes and required the user to *blindly* trust the author of the recipe. The addition of tiny character-based LCDs up to 5 lines high, made it possible to verify the

name of the selected recipe, but was a poor solution for selecting among many recipes and for presenting the individual steps in a recipe.

The manufacturers introduction of a LCD based display made it straightforward to find the relevant recipe in a list on the panel and made it possible to have 4 steps from a recipe shown at the same time. When asked, the participating users stated that 2-3 steps in a recipe were the norm.

The ability to see name and steps for a recipe changed the base of trust. With missing feedback, the user needed to trust himself and his colleagues to activate the right functions. With improved feedback, the user knows that the oven has been given the right orders, so he only needs to trust the capabilities of the oven and his own ability to prepare food. Isolated, oven capabilities and the users' food preparation skills are well established.

Unfortunately, the lack of overview was not eliminated; it just changed character. Navigation between panels became the primary concern. The evolution of the navigational system was a constant reduction of the numbers of perceived screens, either by removing them or by using animation to soften the transitions.

## 8 Organisational issues

Users' difficulties and uneasiness with programming were not only a result of badly designed interaction. As described in the previous section, taking the recipe "out of the black box" of the old interface by making the steps of the program visible, was an improvement, obviously because the user can now see what the program is about, but the studies also indicate that the organizational disturbance of the feature is minimized.

Introducing the possibility of storing recipes in the oven is a more dramatic change to the users than expected. Their uneasiness with storing recipes was indicating that there was more to it, by storing a recipe in the oven it becomes reified in a new way, and the oven changes from being a neutral place for heating to be a bank for recipes. In the hospital kitchen, this would mean that the binder with recipes on paper would lose its clear status as "centre of coordination".

Beth re-entering the same recipes everyday is a clear case of this. In the hospital kitchen nobody has taken the responsibility for moving the recipes into another medium. In contrast John had implemented an organisational structure in his kitchen for securing the quality of the programs stored in the oven. Re-use of programs is not merely about saving key-

ing time. As pointed out by Trigg and Bødker (1994), reuse does not work without trust, overview, and feedback. Re-use is always organisationally situated. Introducing an interface based on a fairly large LCD display, addresses some of the issues by providing feedback and overview, but it does not address the organisationally situatedness of reuse. To deal with that problem, it is not sufficient that the oven interface resembles a binder with more or less personal notes.

Supporting situated reuse with features built into the interface would at least include a clear separation between organisationally reified programs and more ephemeral programs conveniently easing day-to-day situations. The manufacturers' new interface partly supports such separation by having different representations of parameters in manual run and when entering programs. The dilemma, however, is that such separation generally tends to disable users in learning to program.

A different approach, based on the organisational need for separating recipes from the oven could be taken by enabling users, by adding features to their word processor, to store programs as barcodes on the sheets of paper they store in binders; and then enter programs into the oven by using a bar code reader attached to the oven through the maintenance USB port. Such an arrangement would provide increased efficiency without disturbing the organisational order in the kitchen. Even though this, from the perspective of kitchen work would be a simple solution, it seemed overly complicated from a technical point of view; therefore it was rejected during the project. However, taking our analysis of the organisational issues into account, we in retrospect acknowledge that such a barcode-based solution could strike a balance between technical elegance and maintenance of organisational order.

## 9 Discussion and conclusions

We have showed how the increasing computer power in everyday artefacts changes the basic character of the artefacts, and that programmability is a qualitative break. Based on field studies and prototyping, we have introduced a framework for analysis of three levels of obstacles in the adoption of the programming features. The seemingly irrational personal and cultural obstacles were hard to deal with directly but it also turned out that an understanding of e.g. the professional idiosyncrasies of the restaurant cooks was an important resource in dealing with interaction issues. Not surprisingly, interaction issues could account for many of the problems in

adopting programming in the kitchen. More surprisingly, it turned out that problems that seemed to be straight interaction issues had an organisational dimension. Hence we introduced organisational issues as a third level. These issues turned out to be far more important than expected. We learned that these issues, to some extent in combination with cultural resistance, were also the hard part to deal with. When the oven becomes a computing device, responsibility is changed in the kitchen. We suggest that studies of organisational and cultural issues are included in the design process, no matter how simple the interface seems to be.

Mackay (1991) and Nardi (1995) both emphasised problems with tailoring arising from the lack of consistency between the language of day-to-day use and the language used for tailoring. According to that tradition in end-user programming our initial hypothesis was that ovens would support users programming better if the distance between ordinary use and programming was minimised. However, based on our studies and experiments we conclude, in line with Trigg & Bødker (1994), that support for organisational division is similarly important. Introducing specialised mechanisms different from day-to-day interaction may be the best support for tailoring and re-use.

The basic contradiction between the flexibility, consistency and efficiency gained by introducing computer-based control, and the oven as physically understandable appliance, is to some extent a reflection of the basic difference between the planning and slowness of the personnel in institutional kitchens, and the physicality and speed of the restaurant cooks. By considering the organisational issues, we were able to realise the potentials of advanced computer control without violating the day-to-day reality of the institutional kitchen. However, we were not able to find a solution for the lack of tangibility and directness that restaurant cooks found to be the main drawback of the "soft-key and LCD display" arrangement.

Initially in the project, we emphasised that an oven had to continue to be an oven and look like an oven when advanced computer features were introduced; it should not look like a computer. However, through our cultural and historical studies of the oven we saw that the oven had undergone major changes already before the introduction of computer control – the oven has a history of change. In this

respect, the oven is not different from desktop systems. Furthermore, resemblance with computers is not in itself a problem; what matters is whether the computer-like features contribute transparently to the use of the oven in cooking.

The most important implication for future design of information appliances are that we have showed the importance of making a sufficiently in depth study of the history and use of similar artefacts. Such an analysis could be conducted applying tools from a broad range of standard methods. Furthermore, our analysis of the three levels of obstacles proved to be a useful tool in understanding the design task.

The unsolved problem with making a tangible interface based on the "soft-key and LCD display" arrangement is worthwhile pursuing further, thus a study of restaurant cooks' use of the new oven interface will be highly relevant.

## Acknowledgements

Thanks to Hans Harbek and others at Hounö A/S and to the studied users. Thanks to Susanne Bødker for comments and to Marianne Iversen for proof-reading. The project was supported by Centre for Pervasive Computing, project #222, and Centre for Human-Machine Interaction.

## References

- Nardi, B. (1995). *A Small Matter of Programming*. Cambridge, MA: The MIT Press.
- Mackay, W. E. (1991). Triggers and barriers to customizing software. In *Proceedings of CHI '91 Human Factors in Computing Systems*. NYC, NY: ACM Press. pp 153-160.
- Mørch, A. (1997). Three Levels of End-User Tailoring: Customization, Integration, and Extension. In Kyng & Mathiassen (eds.). *Computers and Design in Context*. Cambridge, MA: The MIT Press. pp 51-75.
- Trigg R. H. & Bødker, S. (1994). From implementation to Design: Tailoring and the Emergence of Systematization in CSCW. In *Proceedings of CSCW 94*, ACM Press.