INTERACTION DESIGN

ROGERS • SHARP • PREECE

beyond human-computer interaction

3rd Edition

The compromises made when producing a prototype must not be ignored – whatever those compromises were. However, when a project team is under pressure, it can become tempting to pull together a set of existing prototypes as the final product. After all, many hours of development will have been spent developing them, and evaluation with the client has gone well, so isn't it a waste to throw it all away? Basing the final product on prototypes in this way will simply store up testing and maintenance problems for later on: in short, this is likely to compromise the quality of the product.

Evolving the prototype into the final product through a defined process of engineering can lead to a robust final product, but this must be clearly planned from the beginning.

On the other hand, if your device is an innovation, then being first to market with a 'good enough' product may be more important for securing your market position than having a very high-quality product that reaches the market two months after your competitors'. ■

11.3 Conceptual Design: Moving from Requirements to First Design

Conceptual design is concerned with transforming requirements into a conceptual model. Designing the conceptual model is fundamental to interaction design, yet the idea of a conceptual model can be difficult to grasp. One of the reasons for this is that conceptual models take many different forms and it is not possible to provide a definitive detailed characterization of one. Instead, conceptual design is best understood by exploring and experiencing different approaches to it, and the purpose of this section is to provide you with some concrete suggestions about how to go about doing this.

In Chapter 2 we said that a conceptual model is an outline of what people can do with a product and what concepts are needed to understand how to interact with it. The former will emerge from the current functional requirements; possibly it will be a subset of them, possibly all of them, and possibly an extended version of them. The concepts needed to understand how to interact with the product depend on a variety of issues related to who the user will be, what kind of interaction will be used, what kind of interface will be used, terminology, metaphors, application domain, and so on. The first step in getting a concrete view of the conceptual model is to steep yourself in the data you have gathered about your users and their goals and try to empathize with them. From this, a picture of what you want the users' experience to be when using the new product will emerge and become more concrete. This process is helped by considering the issues in this section, and by using scenarios (introduced in Chapter 10) and prototypes (introduced in Section 11.2) to capture and experiment with ideas. Mood boards (traditionally used in fashion and interior design particularly) may be used to capture the desired feel of a new product (see Figure 11.6). All of this is informed by results from the requirements activity and must be tempered with technological feasibility.

There are different ways to achieve empathy with users. For example, Beyer and Holtzblatt (1998), in their method *Contextual Design*, recommend holding review meetings within the

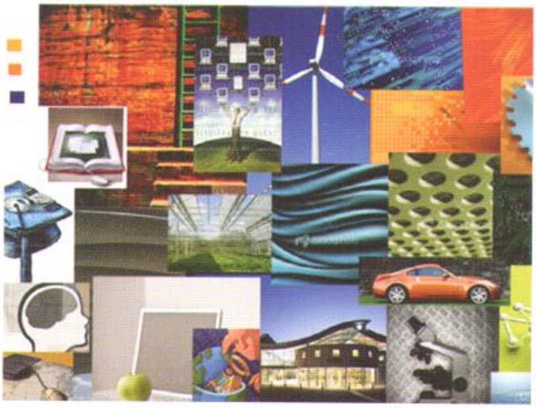

Figure 11.6 An example mood board

team to get different peoples' perspectives on the data and what they observed. This helps to deepen understanding and to expose the whole team to different aspects. Ideas will emerge as this extended understanding of the requirements is established, and these can be tested against other data and scenarios, discussed with other design team members, and prototyped for testing with users. Other ways to understand the users' experience are described in Box 11.2.

Key guiding principles of conceptual design are:

- Keep an open mind but never forget the users and their context.
- Discuss ideas with other stakeholders as much as possible.
- Use low-fidelity prototyping to get rapid feedback.
- Iterate, iterate, and iterate.

BOX 11.2

How to really understand the users' experience

Some design teams go to great lengths to ensure that they come to empathize with the users' experience. This box introduces three examples of this approach.

Buchenau and Suri (2000) describe an approach they call experience prototyping, which is intended to give designers some of the insight into a user's experience that can only come from first-hand knowledge. They describe a team designing a chest-implanted automatic defibrillator. A defibrillator is used with victims of cardiac arrest when their heart muscle goes into a chaotic arrhythmia and fails to pump blood, a state called fibrillation. A defibrillator delivers an electric shock to the heart, often through paddle electrodes applied externally through the chest wall; an implanted defibrillator does this through leads

(Continued)

that connect directly to the heart muscle. In either case, it's a big electric shock intended to restore the heart muscle to its regular rhythm that can be powerful enough to knock people off their feet.

This kind of event is completely outside most people's experience, and so it is difficult for designers to gain the insight they need to understand the user's experience. You can't fit a prototype pacemaker to each member of the design team and simulate fibrillation in them! However, you can simulate some critical aspects of the experience, one of which is the random occurrence of a defibrillating shock. To achieve this, each team member was given a pager to take home over the weekend (elements of the pack are shown in Figure 11.7). The pager message simulated the occurrence of a defibrillating shock. Messages were sent at random, and team members were asked to record where they were, who they were with, what they were doing, and what they thought and felt knowing that this represented a shock. Experiences were shared the following week, and example insights ranged from anxiety around everyday happenings such as holding a child and operating power tools, to being in social situations and at a loss how to communicate to onlookers what was happening. This first-hand experience brought new insights to the design effort.

Another instance is the Third Age suit, an empathy suit designed so that car designers can experience what it is like for people with some loss of mobility or declining sensory perception to drive their cars. The suit restricts movement in the neck, arms, legs, and ankles. Originally developed by Ford Motor Company and Loughborough University (see Figure 11.8) it has been used to raise awareness within groups of car designers, architects, and other product designers.

Finally, Grammenos *et al* (2009) have been working on universally accessible computer games (see Box 11.3) but one of the games they developed is inaccessible. Called Game Over! this game breaks a range of accessibility design guidelines which makes it very frustrating for anyone to use. The game exposes designers to a range of situations commonly experienced by disabled players. ■

Figure 11.7 The patient kit for experience prototyping

Figure 11.8 The Third Age empathy suit helps designers experience the loss of mobility and sensory perception

Before explaining how scenarios and prototyping can help, we explore in more detail some useful perspectives to help develop a conceptual model.

11.3.1 Developing an Initial Conceptual Model

Some elements of a conceptual model will derive from the requirements for the product. For example, the requirements activity will have provided information about the concepts involved in a task and their relationships, e.g. through task descriptions and analysis. Immersion in the data and attempting to empathize with the users as described above will, together with the requirements, provide information about the product's user experience goals, and give you a good understanding of what the product should be like. In this section we discuss approaches which help in pulling together an initial conceptual model. In particular, we consider:

- Which interface metaphors would be suitable to help users understand the product?
- Which interaction type(s) would best support the users' activities?
- Do different interface types suggest alternative design insights or options?

We are not suggesting that one way of approaching a conceptual design is right for one situation and wrong for another; all of these approaches provide different ways of thinking about the product and help in generating potential conceptual models.

Interface metaphors. As mentioned in Chapter 2, interface metaphors combine familiar knowledge with new knowledge in a way that will help the user understand the product. Choosing suitable metaphors and combining new and familiar concepts requires a careful balance between utility and fun, and is based on a sound understanding of the users and their context. For example, consider an educational system to teach 6-year-olds mathematics. You could use the metaphor of a classroom with a teacher standing at the blackboard. But if you consider the users of the system and what is likely to engage them, you will be more likely to choose a metaphor that reminds the children of something they enjoy, such as a ball game, the circus, a playroom, and so on.

Erickson (1990) suggests a three-step process for choosing a good interface metaphor. The first step is to understand what the system will do, i.e. identifying the functional requirements. Developing partial conceptual models and trying them out may be part of the process. The second step is to understand which bits of the product are likely to cause users problems, i.e. which tasks or subtasks cause problems, are complicated, or are critical. A metaphor is only a partial mapping between the software and the real thing upon which the metaphor is based. Understanding areas in which users are likely to have difficulties means that the metaphor can be chosen to support those aspects. The third step is to generate metaphors. Looking for metaphors in the users' description of the tasks is a good starting point. Also, any metaphors used in the application domain with which the users may be familiar may be suitable.

When suitable metaphors have been generated, they need to be evaluated. Again, Erickson (1990) suggests five questions to ask.

1. How much structure does the metaphor provide? A good metaphor will provide structure, and preferably familiar structure.
2. How much of the metaphor is relevant to the problem? One of the difficulties of using metaphors is that users may think they understand more than they do and start applying inappropriate elements of the metaphor to the product, leading to confusion or false expectations.
3. Is the interface metaphor easy to represent? A good metaphor will be associated with particular visual and audio elements, as well as words.
4. Will your audience understand the metaphor?
5. How extensible is the metaphor? Does it have extra aspects that may be useful later on?

In the shared travel organizer introduced in Chapter 10, one metaphor we could use is a printed travel brochure. This is familiar to everyone, and we could combine that familiarity with facilities suitable for an electronic brochure such as videos of locations and searching. Having thought of this metaphor, we need to apply the five questions listed above.

1. Does it supply structure? Yes, it supplies structure based on the familiar paper-based brochure. This is a book and therefore has pages, a cover, some kind of binding to hold the pages together, an index, and table of contents. Travel brochures are often structured around destinations but are also sometimes structured around activities, particularly when the company specializes in adventure trips. However, a traditional brochure focuses on the details of the vacation and accommodation and has little structure to support visa or vaccination information (both of which change regularly and are therefore not suitable to include in a printed document).
2. How much of the metaphor is relevant? Having details of the accommodation, facilities available, map of the area, and supporting illustrations is relevant for the travel organizer, so the content of the brochure is relevant. Also, structuring that information around types of vacation and destinations is relevant, but preferably both sets of grouping could be offered. But the physical nature of the brochure, such as page turning, is less relevant. The travel organizer can be more flexible than the brochure and should not try to emulate

its book nature. Finally, the brochure is printed maybe once a year and cannot be kept up-to-date with the latest changes whereas the travel organizer should be capable of offering the most recent information.

3. Is the metaphor easy to represent? Yes. The vacation information could be a set of brochure-like pages. Note that this is not the same as saying that the navigation through the pages will be limited to page-turning.
4. Will your audience understand the metaphor? Yes.
5. How extensible is the metaphor? The functionality of a paper-based brochure is fairly limited. However, it is also a book, and we could borrow facilities from e-books (which are also familiar objects to most of our audience), so yes, it can be extended.

ACTIVITY 11.2

Another possible interface metaphor for the travel organizer is the travel consultant. A travel consultant takes a set of requirements and tailors the vacation accordingly, offering maybe two or three alternatives, but making most of the decisions on the travelers' behalf. Ask the five questions above of this metaphor.

Comment

1. Does the travel consultant metaphor supply structure? Yes, it supplies structure because the key characteristic of this metaphor is that the travelers specify what they want and the consultant goes and researches it. It relies on the travelers being able to give the consultant sufficient information to be able to search sensibly rather than leaving him to make key decisions.
2. How much of the metaphor is relevant? The idea of handing over responsibility to someone else to search for suitable vacations may be appealing to some users, but might feel uncomfortable to others. On the other hand, having no help at all in sifting through potential options could become very tedious and dispiriting. So maybe this metaphor is relevant to an extent.
3. Is the metaphor easy to represent? Yes, it could be represented by a software agent, or by having a sophisticated database entry and search facility. But the question is: would users like this approach?
4. Will your audience understand the metaphor? Yes.
5. How extensible is the metaphor? The wonderful thing about people is that they are flexible, hence the metaphor of the travel consultant is also pretty flexible. For example, the consultant could be asked to bring just a few options for the users to consider, having screened out inappropriate ones; alternatively the consultant could be asked to suggest 50 or 100 options! ■

Interaction types. In Chapter 2 we introduced four different types of interaction: instructing, conversing, manipulating, and exploring. Which is best suited to your current design depends on the application domain and the kind of product being developed. For example, a computer game is most likely to suit a manipulating style, while a drawing package has aspects of instructing and conversing.

Most conceptual models will include a combination of interaction types, and it is necessary to associate different parts of the interaction with different types. For example, in the travel organizer one of the user tasks is to find out the visa regulations for a particular destination; this will require an instructing approach to interaction. No dialog is necessary for the system to show the required information; the user simply has to enter a predefined set of information, e.g. origin of passport and destination. On the other hand, the user task of trying to identify a vacation for a group of people may be conducted more like a conversation. We can imagine that the user begins by selecting some characteristics of the destination and some time constraints and preferences, then the organizer will respond with several options, and the user will provide more information or preferences and so on. (You may like to refer back to the scenario of this task in Chapter 10 and consider how well it matches this type of interaction.) Alternatively, for users who don't have any clear requirements yet, they might prefer to be able to explore the information before asking for specific options.

Interface types. Considering different interfaces at this stage may seem premature, but it has both a design and a practical purpose. When thinking about the conceptual model for a product, it is important not to be unduly influenced by a predetermined interface type. Different interface types prompt and support different perspectives on the product under development and suggest different possible behaviors. Therefore considering the effect of different interfaces on the product at this stage is one way to prompt alternatives.

Before the product can be prototyped, some candidate alternative interfaces will need to have been chosen. These decisions will depend on the product constraints, arising from the requirements you have established. For example, input and output devices will be influenced particularly by user and environmental requirements. Therefore, considering interfaces here also takes one step towards producing practical prototypes.

To illustrate this, we consider a subset of the interfaces introduced in Chapter 6, and the different perspectives they bring to the travel organizer:

- Shareable interface. The travel organizer has to be shareable as it is intended to be used by a group of people, and it should be exciting and fun. The design issues for shareable interfaces which were introduced in Chapter 6 will need to be considered for this system. For example how best (whether) to use the individuals' own devices such as smartphones in conjunction with a shared interface.
- Tangible interface. Tangible interfaces are a form of sensor-based interaction, where blocks or other physical objects are moved around. Thinking about a travel organizer in this way conjures up an interesting image of people collaborating, maybe with the physical objects representing themselves traveling, but there are practical problems of having this kind of interface, as the objects may be lost or damaged.
- Augmented and mixed reality. The travel organizer is not the kind of product that is usually designed for an augmented or mixed reality interface. The question is what would the physical object be in this case, that the virtual element could enhance? One possibility might be to enhance the physical brochure to provide more dynamic and easily changed information.

ACTIVITY 11.3

Consider the movie rental club introduced in Chapter 10.

1. Identify tasks associated with this product that would best be supported by each of the interaction types instructing, conversing, manipulating, and exploring.
2. Consider the movie rental club and pick out two interface types from Chapter 6 that might provide a different perspective on the design.

Comment

1. Here are some suggestions. You may have identified others:
   - Instructing: the user wants to see details of a particular movie, such as script writer and shoot locations.
   - Conversing: the user wants to identify a movie on a particular topic but doesn't know exactly what is required.
   - Manipulating: the movies might be represented as icons (maybe still images from the movie) that could be interrogated for information or manipulated to represent the movie being reserved or borrowed.
   - Exploring: the user is looking for interesting movies, with no particular topic or actor in mind.
2. Movie rental services tend to be web-based, so it is worth exploring other styles to see what insights they may bring. We had the following thoughts, but you may have had others.

   The movie rental club could be used anywhere – at home, on a boat, in the park – wherever a member might want to book a movie for immediate viewing or for viewing later. If the movie has to be viewed through the same interface as the rental is booked then this would limit the suitability of small mobile devices or wearables, but would make shareable interfaces appropriate. Ideally, the movie would be rentable through a wide range of interfaces, and viewable through a more limited set. A multimodal interface would offer the opportunity of experiencing movie clips and trailers (and indeed the movie itself) in different ways – especially with 3D movies, and future promise of successful smellovision. ■

11.3.2 Expanding the Initial Conceptual Model

Considering the issues in the previous section helps the designer to produce a set of initial conceptual model ideas. These ideas must be thought through in more detail and expanded before being prototyped or tested with users. For example, concrete suggestions of the concepts to be communicated between the user and the product and how they are to be structured, related, and presented are needed. This means deciding which functions the product will perform (and which the user will perform), how those functions are related, and what information is required to support them. These decisions will be made initially only tentatively and may change after prototyping and evaluation.

What functions will the product perform? Understanding the tasks the product will support is a fundamental aspect of developing the conceptual model, but it is also important to consider which elements of the task will be the responsibility of the user and which will be carried out by the product. For example, the travel organizer may suggest specific vacation options for a given set of people, but is that as much as it should do? Should it automatically reserve the booking, or wait until it is told that this travel arrangement is suitable? Developing scenarios, essential use cases, and use cases will help clarify the answers to these questions. Deciding what the system will do and what must be left for the user is sometimes called *task allocation*. The trade-off between what the product does and what to keep in the control of the user has cognitive implications (see Chapter 3), and is linked to social aspects of collaboration (see Chapter 4). In addition, if the cognitive load is too high for the user, then the device may be too stressful to use. On the other hand, if the product has too much control and is too inflexible, then it may not be used at all.

Another aspect concerns which functions to hard-wire into the product and which to leave under software control, and thereby indirectly in the control of the human user? Answering this depends on the product's architecture, although a clear architectural design at this stage of development is unlikely.

How are the functions related to each other? Functions may be related temporally, e.g. one must be performed before another, or two can be performed in parallel. They may also be related through any number of possible categorizations, e.g. all functions relating to telephone memory storage in a cell phone, or all options for viewing photographs in a social networking site. The relationships between tasks may constrain use or may indicate suitable task structures within the product. For example, if one task depends on another, you may want to restrict the order in which tasks can be completed.

If task analysis has been performed, the breakdown will support these kinds of decision. For example, in the travel organizer example, the task analysis performed in Section 10.7 shows the subtasks involved and the order in which the subtasks can be performed. Thus, the system could allow potential travel companies to be found before or after investigating the destination's facilities. It is, however, important to identify the potential travel companies before looking for availability.

What information needs to be available? What data is required to perform a task? How is this data to be transformed by the system? Data is one of the categories of requirements we aim to identify and capture through the requirements activity. During conceptual design, we need to consider these requirements and ensure that our model provides the information necessary to perform the task. Detailed issues of structure and display, such as whether to use an analog display or a digital display, will more likely be dealt with during the physical design activity, but implications arising from the type of data to be displayed may impact conceptual design issues. Information visualization was discussed in Section 6.2.

For example, identifying potential vacations for a set of people using the travel organizer requires the system to be told what kind of vacation is required, available budget, preferred destinations (if any), preferred dates and duration (if any), how many people it is for, and any special requirements (such as physical disability) that this group has. In order to perform the function, the system needs this information and must have access to detailed vacation and destination descriptions, booking availability, facilities, restrictions, and so on.

Initial conceptual models may be captured in *wireframes* – a set of documents that show structure, content, and controls. Wireframes may be at varying levels of abstraction, and may show a part of the product or a complete overview. See Case Study 11.2 for more information and some examples. Physical design involves considering more concrete, detailed issues of designing the interface, such as screen or keypad design, which icons to use, how to structure menus, etc.

11.4 Physical Design: Getting Concrete

There is no rigid border between conceptual design and physical design. Producing a prototype inevitably means making some detailed decisions, albeit tentatively. Interaction design is inherently iterative, and so some detailed issues will come up during conceptual design; similarly, during physical design it will be necessary to revisit decisions made during conceptual design. Exactly where the border lies is not important. What is relevant is that the conceptual design should be allowed to develop freely without being tied to physical constraints too early, as this might inhibit creativity.

Design is about making choices and decisions, and the designer must strive to balance environmental, user, data, and usability and user experience requirements with functional requirements. These are sometimes in conflict. For example, the functionality of a wearable interactive product will be restricted by the activities the user wishes to perform while using it; a product may need to be learnable but also challenging.

User characteristics also have a significant impact on physical design, and two aspects that have drawn particular attention for physical design are accessibility and national culture. Accessibility was discussed in Box 1.2. Researchers, designers, and evaluators have investigated a range of techniques, toolkits, and interaction devices to support individuals with different accessibility needs. For example, tactile graphics are sometimes used to visualize information for blind and partially sighted users (see Chapter 6 for more on haptic feedback). Jay *et al* (2008) experimented with combining haptic and audio cues for visually impaired participants retrieving information from graph structures. They found that haptic cues combined with audio information significantly improved participants' performance in recognizing nodes and structures, over the use of audio alone. Visual appearance may also be modified to support users with other disabilities. For example the chess game described in Box 11.3 supports a range of disabilities including visual cues for users with mild memory problems.

It has been argued that designing products with accessibility in mind is good for everyone, and so it is good design for all products to include accessibility features. An example of this is the iPhone, which includes as standard a gesture-based screen reader that can be used with voice input, audio predictive text for text messages, wireless Braille displays and keyboards, all translatable into 21 different languages. It also includes the ability to zoom in and out of the screen, enhance the contrast between screen elements, and control the device by voice.

BOX 11.3

Designing universally accessible games

Computer games form a large part of the entertainment industry across the world, and are being increasingly used in training and education settings. Games often rely on complex interaction, requiring sophisticated motor, sensor, and mental skills to control interaction devices, respond to stimuli and, formulate game plans. This renders them inaccessible to many people with disabilities. Grammenos *et al* (2009) have developed some universally accessible games which are designed to optimally fit individual characteristics and to be played concurrently by gamers with different abilities. They promote a design approach that is highly participative, user-centered, and iterative involving accessibility experts as well as experienced gamers.

One of their UA games, UA-Chess, is designed to support users who are sighted, users with low vision, users who are blind, users with hand-motor impairment, and users with mild memory or cognitive impairment. The interaction for each player is tailored to their own characteristics, so each may have a different experience while playing the same game – whether the two players are collocated or not. Common accessibility features are incorporated in the game, such as re-sizeable images, voice input, speech output, and various input and output devices. The designers faced several challenges. One of these focused on players with mild memory or cognitive impairments. To support these users, the game includes visual cues such as the last move made, available valid moves, the current selected piece and move, and whether the king is in check (see Figure 11.9). Another challenge was related to the requirement for players to enter their specific requirements and preferences, so that the game interaction could be tailored to their own needs. The problem was that the relevant dialogue box needed to be accessible to the user before the system knew what features to enable in order to make the system accessible to the user! The designers overcame this challenge by building into this interaction as many accessibility features as possible – many of which would be redundant for any one user.

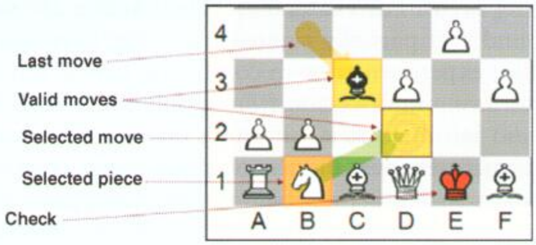

Figure 11.9 Available visual cues for players of UA-Chess with mild memory or cognitive impairments

Throughout the book, you will find interaction design examples relating to different national cultures, e.g. Chinese ATM users, Mexican and Tanzanian children, American government, European families, and so on. As companies and communities go global, designing products for a wide range of cultures has become more significant. Aspects of cross-cultural design include use of appropriate language(s), colors, icons, and typographic layouts. Example design guidelines include ensuring that the product supports different formats for dates, times, numbers, measurements, and currencies, and that generic icons are designed where possible (Esselink, 2000). The reaction of users to icon displays (Choong and Salvendy, 1998) and aesthetic design (Irani *et al*, 2010) has been found to vary across cultures, and so designing for an international audience can be tricky, and organizations face a dilemma when designing for an international brand (see the Dilemma box in Section 10.2).

One of the most well-known sets of guidelines for cultural web design was proposed by Marcus and Gould (2000), building on the cultural dimensions proposed by Hofstede (1994). However, as we noted in the Dilemma box, Hofstede's work and its application in interaction design has been challenged, and designing for a cross-cultural audience is now recognized as more than a translation exercise. As Carlson (1992, p. 175) has put it, successful products "are not just bundles of technical solutions; they are also bundles of social solutions. Inventors succeed in a particular culture because they understand the values, institutional arrangements, and economic notions of that culture." The need for a more in-depth understanding of different cultures when designing interaction has been recognized and alternatives to user-centered design (e.g community-centered design) have been proposed for some settings (Marsden *et al*, 2008). For more information on this topic, see Case Study 11.1 and Gary Marsden's interview at the end of Chapter 12.

DILEMMA

Using Hofstede's dimensions in interaction design

One of the most influential pieces of work on characterizing national culture differences was carried out by a management theorist called Geert Hofstede around 1970. He was given access to responses from a survey of IBM employees in over 50 countries worldwide and from this he identified four dimensions of national culture: power distance (PD), individualism (IND), masculinity–femininity (MAS), and uncertainty avoidance (UA). As a result of work done in Hong Kong at a later date by a Canadian, Michael Bond, a fifth dimension was added which deals with time-orientation.

Although influential, Hofstede's work does have limitations. For example, he admits that the people involved in designing the original questionnaire were all from Western cultures. In addition, his studies have been discussed and challenged over the intervening years: e.g. Oyserman *et al* (2002) challenge his finding that European Americans are more individualistic than people from other ethnic groups. The application of his ideas in interaction design has also been challenged – e.g. work by Oshlyansky (2007) found that Hofstede's model does not help explain cultural differences in affordance; nor does it seem to apply to technology acceptance. So, although popular, Hofstede's dimensions may not be the best approach to accommodating national culture differences in interaction design. ■

Case Study 11.1

Deaf telephony

This case study by Edwin Blake, William Tucker, Meryl Glaser, and Adinda Freudenthal discusses their experiences of community-based design in South Africa. The process of community-based co-design is one that explores various solution configurations in a multi-dimensional design space whose axes are the different dimensions of requirements and the various dimensions of designer skills and technological capabilities. The bits of this space that one can 'see' are determined by one's knowledge of the user needs and one's own skills. Co-design is a way of exploring that space in a way that alleviates the myopia of one's own viewpoint and bias. As this space is traversed a trajectory is traced according to one's skills and learning and according to the users' expressed requirements and their learning.

The project team set out to assist South African Deaf people to communicate with each other, with hearing people, and with public services. The team has been working for many years with a Deaf community that has been disadvantaged due to both poverty and hearing impairment. The story of this wide-ranging design has been one of continual fertile (and on occasion frustrating) co-design with this community. The team's long-term involvement has meant they have transformed aspects of the community and that they have themselves been changed in what they view as important and in how they approach design.

Deaf users in this community started out knowing essentially nothing about computers. Their first language is South African Sign Language (SASL) and this use of SASL is a proud sign of their identity as a people. Many are also illiterate or semi-literate. There are a large number of Deaf people using SASL; in fact there are more than some of the smaller official languages. Since the advent of democracy in 1994 there has been an increasing empowerment of Deaf people and it is accepted as a distinct language in its own right.

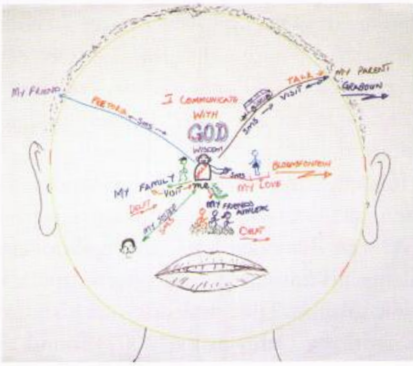

Figure 11.10 One participant's view of communication

Figure 11.11 Participants discussing design in sign language

In this case study a brief historical overview of the project and the various prototypes that formed nodes in a design trajectory are presented. The methodology of Action Research and its cyclical approach to homing in on an effective implementation is reviewed. An important aspect of the method is how it facilitates learning by both the researchers and the user community so that together they can form an effective design team. Lastly, such a long-term intimate involvement with a community raises important ethical issues which are fundamentally concerns of reciprocity. ■

There are many aspects to the physical design of interactive products: visual appearance such as color and graphics, icon design, button design, interface layout, choice of interaction devices, and so on. Chapter 6 introduced you to several interface types and their associated design issues which gave you a flavor of decisions needed at physical design stage. A wide range of guidelines, principles, and rules has been developed for different interface types to help designers ensure that their products meet usability and user experience goals, but even within one interface type, there are significant physical design issues to address. Case study 11.2 illustrates the impact that different cell phone devices may have on the same application.

Case Study 11.2

Designing mobile applications for multiple form factors

Trutap is a social networking service for more than 350 different models of mobile device, which was built for a UK startup between 2007 and 2009. It aggregates online blogging, instant messaging, and social services like Facebook, allowing its users to interact with these

(*Continued*)

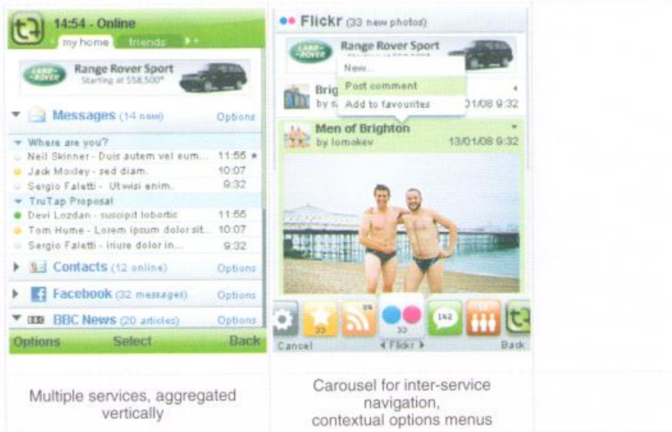

Figure 11.12 Trutap: version 2.0 design concepts

Figure 11.13 Trutap: version 2.0 screenshots, inbox

even when away from the PC (see Figures 11.12 and 11.13). The design of the Trutap application, which took place over two major releases, posed significant challenges in terms of how to integrate disparate sources of data onto small-screen devices, and produce a design which would scale between form factors, i.e. different physical cell phone designs.

The product was designed with a clear goal: teenagers and young adults were spending half of their social lives online, but had to leave that half behind when they walked away from the PC. Trutap would help them keep connected, even when they were away from the PC.

Two versions of the product were launched: Trutap 1.0 offered its own mechanisms for managing people's contacts and communicating with them, and tied into a range of existing instant messaging networks (Yahoo!, MSN, AOL, and the like). Launched in 2008, this version saw far greater take-up in India and Indonesia than with its original target audience of UK students.

This take-up, combined with the successful launch of the iPhone in July 2008 and the increasing prominence of Facebook as the dominant site for personal social networking, led to a change in emphasis for the 2.0 release of Trutap. Launched a year after 1.0, and technically an evolution rather than a reworking, 2.0 emphasized the aggregation of existing online services, tying into Facebook, weblogging software, and photo management, and extending the number of instant messaging services covered. Publicly, the product was presented as a means for aspirational middle classes in the developing world to experience many of the same benefits that the iPhone promised, but on their conventional mobile devices.

This case study, by Tom Hume, Johanna Hunt, Bryan Rieger, and Devi Lozdan from Future Platforms Ltd, explores the impact that different form factors had on the design of Trutap. ■

11.5 Using Scenarios in Design

In Chapter 10, we introduced scenarios as informal stories about user tasks and activities. Scenarios can be used to model existing work situations, but they are more commonly used for expressing proposed or imagined situations to help in conceptual design. Often, stakeholders are actively involved in producing and checking through scenarios for a product. Bødker identifies four suggested roles (Bødker, 2000, p. 63):

1. As a basis for the overall design.
2. For technical implementation.
3. As a means of cooperation within design teams.
4. As a means of cooperation across professional boundaries, i.e. as a basis of communication in a multidisciplinary team.

In any one project, scenarios may be used for any or all of these. More specifically, scenarios have been used as scripts for user evaluation of prototypes, as the basis of storyboard creation (see Section 11.6.1), and to build a shared understanding among team members. Scenarios are good at selling ideas to users, managers, and potential customers.

Bødker proposes the notion of plus and minus scenarios. These attempt to capture the most positive and the most negative consequences of a particular proposed design solution (see Figure 11.14), thereby helping designers to gain a more comprehensive view of the proposal. This idea has been extended by Mancini *et al* (2010) who use positive and negative video scenarios to explore futuristic technology.