Chapter 2
User-Centered Systems Design: A Brief History

Abstract The intention of this book is to help you think about design from a user-centered perspective. Our aim is to help you understand what questions to ask when designing a technology or a system or when you are evaluating a design that already exists. We focus on physiological, cognitive, and social aspects of the human user, aspects that will affect how someone will use what you design. This chapter introduces some historical background to the field of User Centered System Design, and introduces current themes.

2.1 Introduction

It has long been recognized that we need to consider human capabilities and characteristics when designing technologies and systems. As Nickerson summarized in 1969, when the potential for computer-based technologies was first being fully recognized: "the need for the future is not so much computer oriented people as for people oriented computers" (Nickerson 1969, p. 178 in the IEEE version).

Since then a number of fields have grown up, expitly concerned with how to design effective technologies and systems that are intended for human use. User-Centered Systems Design (UCSD or HCSD or when the word "human" is used instead of "user"), User Experience (UX), User-Centered Design (UCD), Interaction Design (IxD) and Human–Computer Interaction (HCI) are areas of research that have taken up that call and are concerned with improving how people interact with computers. Each of these has a slightly different focus and breadth, and each encompasses many different approaches. What they all have in common is that they grow their methods and deliverables in response to changes in the technological landscape.

In this chapter we offer an overview of the intellectual roots of these areas of research and development. Early work focused on the learning and use of command-line interfaces and on programming languages. Following the development of now familiar) WIMP interfaces (Windows, Icons, Menus, Pointer) and

F. E. Ritter et al., *Foundations for Designing User-Centered Systems*,
DOI: 10.1007/978-1-4471-5134-0_2, © Springer-Verlag London 2014

Graphical User Interfaces (GUIs), the focus shifted to understanding how to design visual layouts and the optimization of input devices. Developments in the technology of the devices (e.g., mobile computing, embedded computation, and sensor technologies), and in input methods (e.g., sound, vision, and gesture), have led to a proliferation of design and evaluation methods and a focus on the effects of context on user's experiences of devices, applications, and services. Further, as the uses of technology have been ever more democratized—computers are no longer only available to a few expert users—the effects of individual differences or divergences in use by different user populations have also been increasingly of interest (e.g., differences between children and adults, cultural differences in uptake and use, gender differences in use). Research has also focused on much broader concerns, such as the effects of technology design and uptake in terms of social impact and cultural/global sustainability.

2.2 Influential and Related Research Fields

The intellectual roots of User-Centered Design (UCD, also sometimes called User-Centered System Design, UCSD) lie in several areas of basic and applied research. These include:

- Cognitive and social psychology
- Linguistics
- Mathematics
- Computer science
- Engineering
- Human factors and ergonomics
- Socio-technical systems design
- Scientific management
- Work, industrial, and occupational psychology
- Human relations
- Organizational behavior.

User-centered systems designers also draw on basic research in anthropology, sociology, and information science, and in recent years there has been considerable overlap with ideas flowing between UCD researchers and practitioners and those in research areas such as user experience (UX), human–computer interaction, computer supported cooperative work, computer-mediated communication, and ubiquitous/pervasive computing.

Figure 2.1 presents a simple summary of roots of UCD and how they are related. It is deliberately simplistic but should provide you with some insights into how UCD came about.

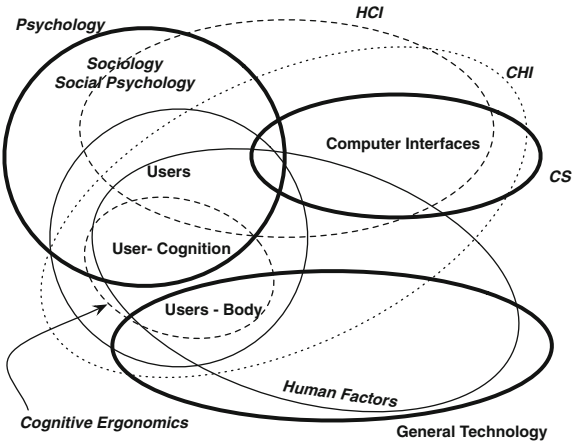

Fig. 2.1 A pictorial summary of some of the fields related to the user. The major fields are shown with *solid lines*

In this section we offer a brief introduction to the fields we consider to be most influential to our approach in this text. These include various branches of human factors and ergonomics, user-centered design, and human–computer interaction.

2.2.1 Ergonomics and Human Factors

Derived from the Greek word *ergon* for work and *nomos* for natural laws, ergonomics draws on a number of research areas including anatomy, engineering, physiology, and psychology. The purpose of ergonomics and human factors research and practice is to maximize the safety and healthiness of work environments and work practices, and to ensure the usability of tools, devices, and artifacts in general. More specifically, ergonomics and HF are concerned with providing a good fit between people and their work or leisure environments. There are a number of sub-fields in ergonomics that have arisen as a result of the increasing penetration of technology into everyday lives. We give a short overview of each of these below. First, however, it is worth considering the notion of "fit."

Many of us are familiar with ergonomic assessments in the workplace; these assessments are conducted to minimize the risk of hazards to health and to prevent ailments such as upper limb disorders. In the UK, however, human factors have embraced the broader context of work practices, going beyond physical environment considerations and biomechanics to include selection and training. Thus, fitting the person to the environment is the responsibility of selection and training, whilst ergonomists fit the environment to the person. Although in this book we are not concerned with selection and training, it is worth noting that there is a complementary relationship between these activities–user groups may be selected or

required by the working environment and/or training and selection are employed to modify the user population to provide the most advantageous fit between the user and the technology.

We can borrow from Rodger's (cited in Holloway 1991) encapsulation in the 1950s which was summarized as "fitting the man to the job and the job to the man"[1] (FMJ/FJM). This is broken down into:

Fitting the man to the job through

- Occupational guidance
- Personnel selection
- Training and development

and fitting the job to the man through

- Methods design
- Equipment design
- Negotiation of working conditions and (physical and social) rewards.

Although Rodger's definition is limited, as it does not take into account the organization in which the person works, we can see this useful encapsulation for occupational psychology as extended by consideration of issues dealt with by human factors.

The concept of 'fit' is a useful one. For physical devices and designed environments, the term *fit* is used literally. For example, on amusement park rides there are height restrictions—children have to wait till they are a certain height and weight before they are allowed on rides. However, the concept of fit is also used when people need to conform the way they act and think to accommodate how tasks are laid out in interfaces. Sometimes this is appropriate, but sometimes alternative designs which modify themselves to accommodate human traits would be more effective. For example, Figs. 2.2 and 2.3 show two example web sites that invite the user to fit themselves to the interfaces, suggesting they modify their behavior to the interfaces. In Fig. 2.2 the web site can recognize that users often put their email address in (e.g., fer2@psu.edu), but rather than remove the domain for the user (@psu.edu), it instructs the user to do so. Figure 2.3 shows a low-cost airline web site where the user is trying to find a cheap flight from Edinburgh to Naples in the run up to Christmas. The results on the *3 day view* and *3 week* view tabs simply show there is nothing available. Even the *Year view* tab only shows the cheapest prices against the months when flights take place. The user then has to infer from the results on the *Year view* tab that the last flights take place in October. The problem arises because the user is thinking in terms of flight dates—

[1] We note that the language at the time used the word *man* to include both genders, a practice that, appropriately, is no longer acceptable.

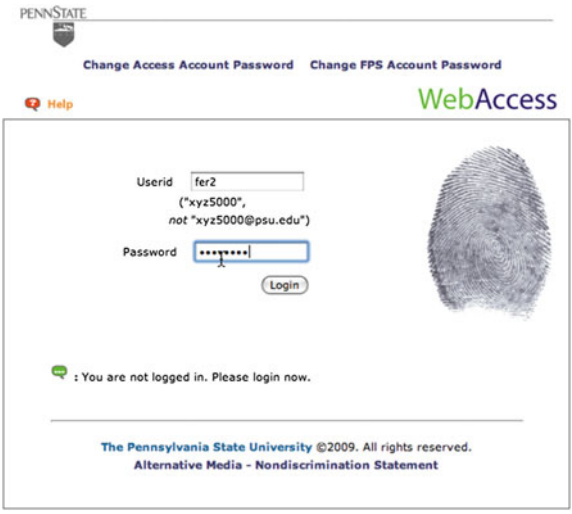

Fig. 2.2 An example interface that attempts to "fit the user to the machine". In the top entry field the user is expected to remove the domain rather than have the system do that (many sites, including GMail, will let users login with either user-name or user-name@domain, this one does not)

they do not care about dates when there are no flights—but the web site only works in terms of calendar dates.

2.2.1.1 Classical Ergonomics

Classical ergonomics has also been called interface ergonomics. The interface referred to is the person/machine interface of controls and displays, and the principle contribution of the designer is the improved design of dials and meters, control knobs, and panel layout. Notably people are usually referred to as *users* or *operators* in this literature. The expert's concerns can extend beyond the design of chairs, benches, and machinery to specify at least partly the optimum physical work environment, including temperature, humidity, and location of work surfaces.

This classical approach started with the design of military equipment, but now considers the design of items and workspaces in civilian contexts. This approach often takes a consultancy mode, with advice usually being delivered in the form of principles, guidelines, and standards. This can cause problems for two reasons: (1) classical ergonomists are only called in at the end of development and asked to advise on the final product, rather than being involved throughout the development process—this means that ill-thought out design decisions with poor rationale may already be "baked into" the design, and no easy fix (or no fix at all) is possible; and (2) guidelines and prescriptions for design activity are usually generic, and lack context specific details. We return to this issue in Sect. 2.3.

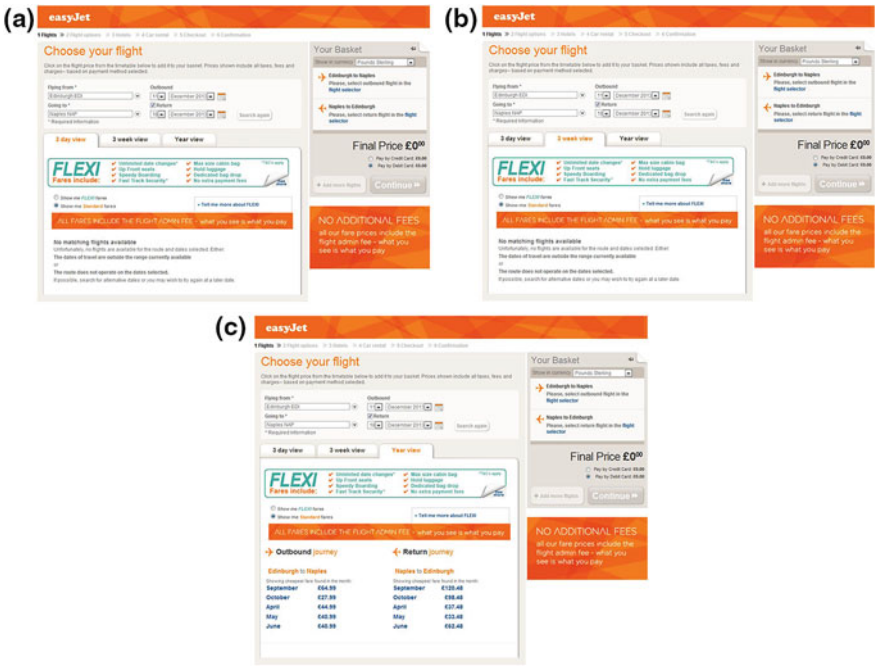

Fig. 2.3 In this interface the airline presents flight information using calendar dates. On the *left* is the 3 day view, in the *center* is a 3 week view, and on the *right* is a year view. Whereas the user most likely wants to view flight dates (with a pointer to the last, or next available flight, e.g., something like "there are no flights available for the period you have selected, the nearest available dates are October xx 2013 and April yy 2014"). This interface offers hints and encourages the user to search repeatedly rather than do the search for the user

2.2.1.2 Error Ergonomics

Error ergonomics is the study and explanation of human error in systems. The zero defects approach assumes that human error is the result of inadequate motivation, c.f. the examples of accidents and error attribution including the official report on Kegworth noted in the Appendix. Reason (1997) describes this as the person model or person approach. This approach tends to result in campaigns for safety procedure training and for safety oriented materials. These drives attempt to raise awareness and incentives for the workers. Even during the World War I, where the work force was *highly* motivated, error ergonomists discovered that fatigue was a major cause of errors.

Similarly, the error data store approach, which forms a part of methods like THERP (Technique for Human Error Rate Prediction, Swain & Guttman, 1983) assumes that human error is inevitable (this is discussed further in Chap. 10). This approach produces data banks of error probabilities for a variety of tasks executed under various conditions. It is therefore necessary to predict the incidence and consequences of human errors in any given situation. The results inform the design of systems in a way that minimizes the occurrence and effects of errors.

2.2.1.3 Systems Ergonomics

This approach was developed in the USA in the 1950s, and takes a more holistic approach to understanding users and systems as they work in concert. That is, the user and the system are seen as a single interacting system that is placed within a work context. Within this approach, system design involves parallel development of hardware and personnel issues, with training and selection issues considered. The ergonomist acts as an integral member of the design team, working throughout the life cycle to inform the system design. Therefore, in addition to the physical, behavioral and cognitive considerations of the finished product itself, the human factors expert (or "ergonomist") is involved in: (1) determining the required task functions (by activity and task analysis in conjunction with the consideration of the task requirements) and allocating the functions between the user and the system (2) the design of personnel subsystems, and (3) the design of job descriptions and job support materials (e.g., manuals and training schemes).

The approach differs from user-centered design as the designers and human factors experts still view the user as just one part of the system, whereas user-centered design focuses more on the user's needs and perspective than those of the system, tasks, and activities per se. In computer system development, for example, a systems approach would consider the task from a logical, syntactic perspective and then the computer system implementation issues with a view to allocating function between the user and the computer system. A user-centered approach would consider the processing capabilities of the human user and analyze tasks from the perspective of the user.

2.2.1.4 Cognitive Ergonomics/Cognitive Systems Engineering

Since the mid-1960s and the development of integrated circuits and third generation computer systems, research has been carried out in user-centered aspects of data processing, management information systems, information systems, and information technology. The 1970s saw a rapid increase in the use of computer-based technologies, resulting in the body of knowledge about user-centered design methods in areas such as Office Information Systems, industrial process control systems, and transportation systems. The role of people changed from one of directly controlling machinery and equipment to one in which they were interacting with computer based technology. In industrial systems this was characterized by a change in the operator's role from one of hands-on control to one of monitoring and supervisory control. This change from *doing* to *thinking* meant that it became more important to understand the way that people perceived problems, made decisions, and took actions. This led to the development of the field of cognitive ergonomics, which is nowadays more frequently described as cognitive systems engineering (CSE), or just cognitive engineering.

Originally developed in the 1970s and early 1980s (Hollnagel and Woods 1983), cognitive systems engineering has continued to evolve since that period

(Hollnagel and Woods 2005; Woods and Hollnagel 2006). Cognitive ergonomics is concerned with *human–machine systems*. Here, *machine* is taken to represent any artifact that is designed for a specific purpose. There are technological aspects to these systems, which are investigated from the perspective of how they affect use. These systems are always embedded in a socio-technical context because people are involved in the design, construction, testing, and use of these systems. Although CSE practitioners regard all systems as *socio-technical systems*, they usually draw a distinction between the *technological* system, in which the technology plays the central role in determining what happens, and the *organizational* system, in which people mainly determine what happens.

CSE is concerned with applicable, approximate models of how people perceive, process, attend to, and use information to achieve their goals. Aimed at designers, instructors, and users, CSE draws on many areas of psychology that are often taught separately, such as planning, language, problem solving, learning, memory, and perception. However, CSE addresses how such processes work together. It is different to the direct application of cognitive psychology in that it does not look at cognitive processes in isolation, but at their integration and how they are involved in particular activities or situations. CSE also differs from cognitive psychology in focusing on theories which can predict behavior in what have been called real world settings, rather than laboratory settings, although results from laboratory settings are considered informative. Real world settings may require a more detailed treatment of, for example, individual differences, uncertainty, ad hoc problem solving, and so on, than many other branches of psychology. CSE also places greater emphasis on the co-agency of action between the user and the machine, but, again, this is a difference in emphasis and these fields overlap to a great extent.

CSE is thus largely concerned with applications in complex dynamic domains, such as aviation, industrial process control, healthcare, and so on. It normally starts by attempting to understand the issue at hand, using observation to try to understand the patterns of work. It then uses this understanding to guide the search to identify what would be useful to support the types of work that have been observed. These insights are then used as a basis for (innovative) design, in participation with others, to support the work, the processes of change, and optimizing the process.

2.2.2 Socio-Technical Systems Design

The term *socio-technical systems* was originally coined by Emery and Trist (1960) to describe systems that involve a complex interaction between humans, machines, and the environmental aspects of the work system—something that is true of most systems in the workplace. The corollary of this definition is that all of these factors—people, machines, and context—need to be taken into account when developing

socio-technical systems using so-called socio-technical system design (STSD) methods such as ETHICS (Effective Technical and Human Implementation of Computer-based Systems; Mumford 1983, 1995). In reality, these methods are more like guiding philosophies than design methods that are usually associated with systems engineering (Mumford 2006). In other words, the STSD methods tend to provide a process and a set of guiding principles (e.g., Cherns 1987; Clegg 2000) rather than a set of detailed steps that have to be followed.

From its inception in the period immediately after World War II, by what is now called The Tavistock Institute, until the present day, there have been several attempts at applying the ideas of STSD, although these have not always been successful (e.g., see Mumford 2006 for a critical review of the history of STSD methods). Early work in STSD focused mostly on manufacturing and production industries such as coal, textiles, and petrochemicals. The general aim was to investigate the organization of work and to see whether it could be made more humanistic, incorporating aspects such as the quality of working life. In other words, the idea was a move away from the mechanistic view of work that is usually associated with Taylor's principles of scientific management, which largely relied on the specialization of work and the division of labor.

The heyday of STSD was probably the 1970s. This was a time when there were labor shortages, and companies were keen to use all means available to keep their existing staff. This was also the period where more and more computer systems were being introduced into the workplace. Apart from the usual cultural and social reasons, companies could also see good business reasons for adopting socio-technical ideas. As just one of many such examples, Digital Equipment Corporation (DEC) had a family of expert systems that were developed using STSD (e.g., see Mumford and MacDonald 1989) to support the configuration and location of DEC VAX computers that saved the company tens of millions of dollars a year (Barker and O'Connor 1989).

There was a downturn in the use of STSD in the 1980s and 1990s as lean production techniques and business process re-engineering approaches dominated system development. STSD is, however, still widely advocated in the field of health informatics for the development of health care applications (e.g., Whetton 2005). Many medical systems are still never used because they introduce ways of working that conflict with other aspects of the user's job, or they require changes to procedures that affect other people's responsibilities. By focusing on the underlying work structure, STSD approaches facilitate the development of medical systems that are acceptable to the users (Berg 1999, 2001; Berg and Toussaint 2003).

Socio-technical ideas pervade a lot of thinking around information systems, although they may not always be explicitly referred to as such (Avgerou et al. 2004). The ideas appear in areas such as participatory design methods, computer supported cooperative work (CSCW), and ethnographic approaches to design. Recently, Baxter and Sommerville (2011) have outlined the need for socio-technical systems *engineering*, which integrates the ideas that have been developed in these different areas.

2.2.3 Cognitive Modeling and Programmable User Models

A cognitive model is an approximation of how people reason. The goal of a cognitive model is to explain scientifically very basic cognitive processes, explain how these processes interact, account for errors and breakdowns in these processes, and derive predictions about how those reasoning processes will proceed under different conditions.

Cognitive modeling is a method developed from early work in the late 1950s when psychologists realized that computational processes may be a good analog of human reasoning processes: like humans, computers take input in the form of symbols, require memory for information storage, and manipulate those symbols with algorithms to produce output. It was therefore proposed not only that human reasoning could be an inspiration for thinking about computational processes but also that computers may be a good way for us to simulate human reasoning and therefore derive deeper understandings of how humans think (Newell et al. 1960; Newell and Simon 1972).

Cognitive models in the late 1960s, the 1970s, and the 1980s focused on how people solved problems *symbolically*: humans take input in and form symbols, require memory for information storage, and use algorithms to manipulate those symbols to produce output. The models were usually limited to one task (or one type of task) and usually simulated reasoning in terms of what was going on in the user's mind. They addressed *human information processing* but did not address how information is taken in from the external world, how actions are performed in the world, and the ways in which real world settings impact the pace at which those processes take place. Each model was essentially a micro-theory of how some part of behavior occurred, and it was independent of other micro-theories. Over time, the need to integrate the micro-theories increased, which led to the idea of *unified theories of cognition* (UTCs; Newell 1990).

These theories are implemented as cognitive architectures available as computer simulations that constrain how models (based on task knowledge) can perform tasks in psychologically plausible ways. So, for example, often when humans perform two tasks simultaneously, the performance on one is affected by the performance on the other. Cognitive models are essentially programs written in a specific language to run on particular cognitive architectures. The models can perform complex tasks including perception, learning, reasoning, problem solving, remembering, decision making, proprioception (how people manage their bodies in space), and ambulation (how people move around physical spaces).

There has long been an overlap between cognitive modeling and human–computer interaction. Drawing on these developments in psychological theory and in simulation modeling, design researchers started investigating the possibility of building models of how people reason and problem solve when using complex interfaces, so that predictions about the pros and cons of different interface and information representation choices could be tested prior to investing in any interface or interaction development (e.g., Pew and Mavor 2007). Models force the

designer to consider psychological factors systematically and explicitly as they make usability predictions. Examples of some influential approaches in the world of human–computer interaction are the Model Human Processor (MHP), GOMS (which stands for Goals, Operators, Methods, and Selection rules), the Keystroke Level Model (KLM) (see Card et al. 1983), and Programmable User Models (PUMs: Young et al. 1989). We will discuss this further in Chap. 11 on task analysis.

In recent years we have become more and more familiar with concepts such as Artificial Intelligence (AI) and Machine Learning (ML). These fields share roots with these cognitive modeling efforts. It has also been more recently acknowledged that cognitive models can combine symbolic and *sub-symbolic* processes–such as neural net modeling, for example. These hybrid models allow us to consider the characteristics and constraints of the brain's architecture of neurons and how the neural underpinnings of cognition impact cognitive processes (Busemeyer and Dieterich 2010) on both a symbolic and sub-symbolic and also an emergence level.

In the first part of the book we introduce several ideas and theories about the way that people behave. These ideas and theories are all encapsulated in cognitive architectures like ACT- R (Anderson 1993, 2007; Anderson et al. 2004) and Soar (Laird 2012; Newell 1990). There are still active research communities for both of these architectures. We introduced ACT-R in Chap. 1 and return to use it to help summarize design relevant user characteristics in Chap. 14.

2.2.4 User-Centered and Human-Centered Design

Through the 1980s, user-centered design (UCD, Norman and Draper 1986) came to the fore. User-centered design involves focusing on the user's needs, carrying out an activity/task analysis as well as a general requirements analysis, carrying out early testing and evaluation, and designing iteratively. As in the systems approach, this has a broader focus than the other approaches, but here there is a greater emphasis on the user and less of a focus on formal methods for requirements gathering and specification, and a move from linear, rigid design processes to a more flexible iterative design methodology.

A related movement, Human-Centered Design (HCD), expanded the focus from the user in interaction with the system to considering how human capabilities and characteristics are affected by the system beyond direct interaction with the interface or system itself. Humans should be seen as the most important element of information systems and should be *designed in*. The people context of information systems must be studied and understood. In more recent work, dimensions such as gender, race, class, and power are also being explicitly considered with respect to people's interactions with interactive technologies.

This sensibility surfaces in three ways. First, consideration is given to the fact that the introduction of a new system engenders changes in the organization of

peoples' behaviors and activities—that is in how people do things. These behavioral changes also affect others. So, user needs and demands, situational effects, and technological requirements are considered in tandem. The boundaries between which issues are defined as *technical* and which are *organizational* or *social* are considered to be malleable, not fixed, and need to be negotiated. This kind of approach is also prevalent in socio-technical systems design, described above.

Second, human-centered design addresses the fact that more and more systems are being built where users do not interact directly with the technology as "users." Examples may be telecare assistive technologies—bed sensors which are programmed to track automatically when a person gets out of bed and to raise an alarm if they are not back in bed within a programmed time limit.

Finally, human-centered design tends to look to the longer-term effects, as well as the immediate, task-related issues that occur at human-system "touchpoint" moments. New applications of technology should be seen as the development of permanent support systems and not one-off products that are complete once implemented and deployed. In other words, the way in which technological change alters the organization of activities, and what are likely ongoing interventions, need to be considered.

User-centered (and human-centered) design methods tend to emphasize user participation in the design process for ideation and evaluation of design options. In this book, we have adopted the user-centered perspective, but we do not focus on the interaction with the interface; our intention is to broaden the scope of analysis to the user + technology system in the task context. Hence we have adopted the term "user-centered system design".

2.2.5 User Experience

User experience has been described as "a person's perceptions and responses that result from the use or anticipated use of a product, system, or service" (ISO 9241-210). According to this definition, *user experience* goes beyond interface design to address a person's emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviors, and accomplishments that occur before, during, and after use. Three factors that influence user experience are considered—the system, the user and their characteristics, and the context of use of the technology or system. User experience is often used interchangeably with *usability* but there is clearly a different focus that is signaled: usability and *usability engineering* focus on task related aspects (getting the job done); user experience and *experience design* focus on and foreground the users' feelings, emotions, values, and their immediate and delayed responses.

2.2.6 Human–Computer Interaction

Human–computer interaction (HCI) is the study of interaction between people (user) and computers. Although often confused with interface design, the remit of HCI is considerably broader. Further, while HCI draws insights from the foundations of interfaces design (design sciences and graphics), the roots of HCI lie in the social sciences.

The Association for Computing Machinery (ACM), the major professional association for computer science, has a subgroup, a special interest group (SIG) on Computer–Human Interaction (full name SIGCHI). SIGCHI was fundamental in creating, nurturing, and defining HCI as a field. There are a number of excellent texts that summarize the history and current activities in HCI that are shown below. SIGCHI (Hewett et al. 1996) defined HCI as:

> … a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

It is worth noting that HCI as a field is constantly changing in response to technological innovations and consequent emerging user needs and demands, and this response is also updated in the ACM's recommended curriculum where HCI is a core area of computer science (http://www.acm.org//education/curricula/ComputerScience2008.pdf).

In 2006, Suzanne Bødker (2006) outlined three "waves" in the development of HCI as a field. The first wave drew insights from cognitive theory and human factors predominantly (see also Bannon 1991). We believe this perspective is still relevant while the perspectives of the second and third waves broaden HCI's remit, increasing its influence. The second wave that developed through the late 1980s into the early 2000s focused on groups working with collections of applications, drawing on theories of "situated action," "distributed cognition," and "activity theory." Scholars wrestled with how to capture the effects of context on activity. Bødker suggests that at this point "rigid guidelines, formal methods, and systematic testing" were no longer the central focus as HCI researchers and practitioners moved to "proactive methods such as a variety of participatory design workshops, prototyping, and contextual inquiries…". Finally, the third wave of HCI acknowledges that computers are increasingly being used in private and public spheres, moving out of workplace contexts and into everyday life for "non-work, non-purpose, and non-rational" uses. This third wave necessarily addresses the "expansion of the cognitive" to include emotional and esthetic aspects of experience, but also the pragmatic and cultural-historical.

A recent report summarizes current and future teaching in HCI and documents some of the changes that have occurred in the field since its beginnings in the early 1980s (Churchill et al. 2013). It is also worth noting that the role and involvement of the HCI

expert varies in design. The nature and level of involvement depends on the ethos of the design setting (the relative importance of usability issues and the degree of focus on supporting the user). We will deal with more HCI research in upcoming chapters.

2.3 Standards, Principles, and Guidelines

All of the disciplines mentioned above have a goal of answering specific research questions using experimental and observational methods. For example a research project may ask:

- Is this chair comfortable over an 8 h working day?
- Can the user get their task done with this application?
- Is the font used in this interface readable?
- Have we made the most important information in this interface stand out?
- Is this interface esthetically appealing to the user demographic I am interested in?
- Will the user get the information they need in a timely fashion if there is an emergency?

It is not always possible to carry out this research to answer questions of this sort oneself, so researchers turn to lessons learned from previous studies that are codified as *standards, principles, and guidelines* that can be applied to the problem situations they encounter.

Formal standards are generated by experts. They are intended to capture the agreed-upon wisdom and best practices of the field. Once created, they offer a common vocabulary for designers/developers and, ideally, result in systems that are more consistent for users, more easily inter-operable, and easier to integrate. In the design world, standards tend to be concerned with human adaptability and human variability. They are prescriptions for safe, acceptable designs, detailing the limits outside which the user may suffer from stress, and accidents may be caused. Standards are and can become part of the law. For example, British Standard BS 5330 deals with the relationship between sound levels in the workplace and the incidence of hearing loss.

Principles are prescriptive and specify general theoretical ideas that can underpin design decisions. They do not specify the limits of human capabilities like standards do and tend to be more general than guidelines. (Note that although we make this distinction between guidelines and principles, the ergonomics literature generally does not.) Ideally, such principles are encapsulations of theoretical insights that have been derived from extensive data gathering and testing. For example, Norman (1988, 2013) outlines a set of principles that designers should consider, e.g., making things visible and providing feedback. Norman's principles are based on his theory of action and interaction (noted further in Chap. 12). In particular, he emphasizes that the state of the system should be visible, that feedback on user's actions should be

Table 2.1 Principles for design to avoid exasperating users (Hedge 2003)

• Clearly define the system goals and identify potential undesirable system states
• Provide the user with appropriate procedural information at all times
• Do not provide the user with false, misleading, or incomplete information at any time
• Know thy user
• Build redundancy into the system
• Ensure that critical system conditions are recoverable
• Provide multiple possibilities for workarounds
• Ensure that critical systems personnel are fully trained
• Provide system users with all of the necessary tools
• Identify and eliminate system "Gotchas!"

provided, and that the system should be consistent across its subsystems. Similarly, Hedge (2003) offers the principles shown in Table 2.1.

Guidelines are prescriptive and offer some general guidance for making design decisions. They tend to be more specific than principles, but still relate existing theory and knowledge to either new design or established design problems (e.g., Brown 1988; Mosier and Smith 1986; or the Apple design guidelines, available online). Below we offer a number of principles and guidelines that we have found useful in our own work. In designing and evaluating systems we ask questions about the design's *functionality, usability, learnability, efficiency, reliability, maintainability*, and *utility* or *usefulness*. These are all discussed below.

(1) Functionality, what something does, is often the first thing to be considered while consideration of usability issues is sometimes tacked on at the end of development. This can lead to poorly designed artifacts that are hard to use but that offer new functionality. Sometimes this is enough. Sometimes it is not. Often, with more thoughtful design, one can have both (Pew and Mavor 2007).

(2) Usability is a complex concept that can be defined in several ways. For example, Ravden and Johnson (1989) specify the following as all relevant to an assessment of whether a system or technology is usable or not:

Visual clarity
Consistency
Informative feedback
Explicitness
Appropriate functionality
Flexibility and control
Error prevention and control
User guidance and support.

Eason (1984) offers the following definition of usability: the "major indicator of usability is whether a system or facility is used." However, this is patently not the case as many devices that are used are hard to use. More usefully, Eason notes that usability is not determined by just one or two constituents, but is influenced by a

number of factors. These factors do not simply and directly affect usability, but interact with one another in sometimes complex ways. He focuses on three elements in particular that need to be taken account of explicitly: system function-task match, task characteristics, and user characteristics. Eason argues that these are independent variables that lead to changes in user reaction and scope of use that could be restricted, partial, distant, or constant.

In 1991 the ETSI (European Telecommunications Standards Institute) proposed two kinds of usability dimensions, those linked to performance and those related to attitude, where performance is measured objectively and attitude represents subjective dimensions (see http://www.etsi.org).

Although Shackel (1991) maintains the distinction between performance and attitudinal dimensions, he defines four distinguishable and quantifiable dimensions which can assume varying degrees of importance in different systems: effectiveness, learnability, flexibility, and attitude. These dimensions are not mutually exclusive in the sense that measures of effectiveness, for example, can at the same time also give some indication of system learnability. However, they provide a good starting point.

Finally, Booth (1989) says that usability is usefulness, effectiveness, ease of use, learnability, attitude, and likeability. A useful system is one that helps users achieve their goals. This more pragmatic approach is also taken by the International Standards Organisation (ISO) in their 9241 series of standards: "the usability of a product is the degree to which specific users can achieve specific goals within a particular environment; effectively, efficiently, comfortably, and in an acceptable manner."

(3) Learnability is how easy the system is to learn. This is affected by a number of factors: for example, how complex it is, how well the system behaviors are signaled in the form of feedback, how consistently the system behaves, how mode changes which may lead to different kinds of behavior are signaled to the user, and so on. Learnability can also be affected by how well the system is documented, either formally (though instructions) or informally through the availability of other users who may be more expert and can help the novice learner.

Learnability is also affected by how similar the new system is to other systems that the users know, because there may be transfer of knowledge from previous system use. How similar it is to previous systems not known to the user can also be important because, if there are other users, they may be able to help novices with new systems if the new systems are similar to previous systems, and existing consultants and teachers may be available if the systems are similar.

(4) Efficiency of a system can be measured through the use of resources such as processor time, memory, network access, system facilities, disk space, and so on. Programmers tend to focus mostly on efficiency, because it ensures that systems work fast and do not frustrate users by keeping them waiting. Note that this is a *computer* not a *human* centric view of efficiency. It is a relative concept in that one system can be evaluated as more efficient than another in terms of some parameter such as processor use, but there is no absolute scale on which to specify an optimum efficiency with regard to people's experience of a system when carrying

out a task. Optimum efficiency from a human-centric perspective requires consideration of the task, the task-context, and the characteristics of the users. One needs to consider the users' knowledge level and disposition, including their motivation. It is also important not to confuse efficiency with speed of execution; speed may be important, or it may also be ultimately inefficient.

In the early days of computers, when programs were small and computer time was relatively expensive, efficiency of computer time was considered to be of paramount importance, and it probably was. With today's faster machines, designers need to consider the effects of choices upon all resources and the consequences of different kinds of efficiency. For example, when considering Internet sites, slow download times are an efficiency issue caused by site/application design and connectivity bandwidth. Users can get frustrated if they are in a hurry to complete a transaction. However, the opposite also occurs—when a transaction is *too* efficient, users can get disoriented and dissatisfied (e.g., one-click payments without asking the user to review orders before placement). Thus efficiency must be calculated in terms of technical efficiency that matches user efficiency expectations for the task at hand.

(5) Reliability is concerned with the dynamic properties of the eventual system and involves the designer making predictions about behavioral issues. We need to know whether the system is going to be complete (in the sense that it will be able to handle all combinations of events and system states), consistent (in that its behavior will be as expected and will be repeatable, regardless of the overall system loading at any time, and across components of the system), and robust (when faced with component failure or some similar conflict, for example, if the printer used for logging data in a chemical process-control plant fails for some reason, the whole system should not crash, but should instead follow a policy of *graceful degradation*).

As systems get larger the problems of ensuring reliability escalate. For safety critical systems where this factor is most important, various techniques have been developed to help overcome limitations in design and implementation techniques. For example, in a system used in a fly-by-wire aircraft in which the control surfaces are managed by computer links rather than by direct hydraulic controls, the implementation will be by means of multiple computers, with a strong likelihood that each will have been programmed by a separate development team and tested independently. Any operational request to the control system will then be processed in parallel by all the computers and only if they concur with the requested operation will it be carried out.

(6) Maintainability is how easy a system is to maintain and upgrade. As systems get larger and more costly, the need for a life-long time in service increases in parallel. To help achieve this, designs must allow for future modification. Designers need to provide future maintainers with mental models of the system and the design rationale so that future maintainers can gain a clear understanding of the system and how it is put together (Haynes et al. 2009). Development of modular designs helps, but larger systems present further problems. While small systems can be modeled with a structural model (i.e., laying out

the component parts of the system), as systems get larger it is important to develop functional models that simulate what the component parts do themselves and in interaction with each other.

(7) Utility/Usefulness is an important concept always to consider when designing systems. Is it ultimately useful for users and how long is its likely usefulness? Is this something that will become an everyday system or an infrequently used system? When it is used, how useful do users find it or are there other workaround that users would rather engage with? Usefulness can be measured both in terms of how often and in what way something is used, but can also be measured with subjective scales like 'how much do you like this?' People may find something useful because it makes them feel good about themselves rather than because it is an efficient, reliable system with a highly usable interface from our perspective as designers.

2.4 Summary

In this chapter we have provided an overview of research areas that have contributed to our understanding of user-centered design. User-centered design draws on *multiple* sources of knowledge to support creating systems that are based on users' abilities, capabilities, and task. What all these approaches have in common is the perspective that when designing we need to consider variation and similarity in the contexts, people, and tasks that characterize different design situations and settings. A one-size-fits-all approach seldom works to achieve the most productive, safe, and enjoyable design solution. We summarize this perspective by inviting you to remember that design is about considering *particular* people doing *particular* tasks in a *particular* context—our focus in this book is people doing tasks using technologies, but this perspective can be more generally applied.

It is worth highlighting at this point that, in order to comply with ISO standard 9241-210 (which now refers to *Human*-Centered Design, rather than *User*-Centered), the following four activities are now *requirements* (previously they were recommendations):

1. Understanding and specifying the context of use (including users, tasks, environments)
2. Specifying the user requirements in sufficient detail to drive the design
3. Producing design solutions that meet these requirements
4. Conducting user-centered evaluations of these design solutions and modifying the design to take into account the results.

Our aim in this book is to provide you with the foundations that will help you to meet these requirements. In the first part of the book we focus on the capabilities of users. We categorize these capabilities into anthropometric, behavioral, cognitive, and social aspects. Although we separate issues into these categories, we

acknowledge that the boundaries between them are somewhat blurred: our bodies affect how we act, and our behaviors affect how we participate socially. Thus, all these factors interact. A key skill for effective human-centered system design is to understand which factors are central or primary in any design situation and which are peripheral or secondary.

In the latter part of the book we provide introductions to some methods that can be used to guide design and evaluation. These include task analysis, evaluation methods, and the notation of cognitive dimensions. These methods differ in terms of their preferred unit of analysis, the kinds of data collected, and the analyses that are conducted. We finish the book by providing a framework that will allow you to integrate your knowledge of the user with the methods in a systematic way.

2.5 Other Resources

There are a lot of helpful texts that can give you some background to the field of user-centered system design. Some of the texts that we have cited above are particularly helpful. For more on the history of this field read this book:

Shachtman, T. (2002). *Laboratory warriors: How Allied science and technology tipped the balance in World War II*. New York, NY: HarperCollins.

A classic text that laid many of the basics out for the field of user-centered systems design is Don Norman and Steve Draper's 1986 text:

Norman, D. A., & Draper, S. W. (Eds) (1986). *User centered system design: New Perspectives on human–computer interaction*. Hillsdale, NJ: Erlbaum.

Jack Carroll's summary of Human Computer Interaction in the *Encyclopedia of Human Computer Interaction* is a great place to start if you want an overview:

Carroll, J. M. (2009). Human computer interaction (HCI). In *Encyclopedia of Human–Computer Interaction*. M. Soegaard & R. F. Dam (Eds.). Aarhus, Denmark: The Interaction Design Foundation.

Two good textbook style overviews are:

Sharp, H., Rogers, Y., & Preece, J. (2011). *Interaction design: Beyond human–computer interaction* (3rd ed.). Chichester, UK: John Wiley and Sons Ltd.

Shneiderman, B., & Plaisant, C. (2009). *Designing the user interface: Strategies for effective human–computer interaction* (5th ed.). Reading, MA: Addison Wesley.

One of the best introductions to the practice, the how-to's, of user-centered design is by Elizabeth Goodman, Mike Kuniavsky, and Andrea Moed. They cover basic techniques and methods that will help you design better interactions. They also offer case studies and examples that you can compare to your own design situations:

Goodman, E., Kuniavsky, M., & Moed, A. (2012). *Observing the user experience: A practitioner's guide to user research.* San Francisco, CA: Morgan Kaufman

For more formal methods and models of interaction programming, read Harold Thimbleby's text *Press On*:

Thimbleby, H. (2007). *Press on—Principles of interaction programming.* Cambridge, MA: MIT Press.

If you want to know more about field based and participatory requirements gathering, a well known method is Contextual Design. This is described in this text:

Beyer, H., & Holtzblatt, K. (1997) *Contextual design: Defining customer-centered systems*. San Francisco, CA: Morgan Kaufmann.

Finally, cognitive modeling can offer enormous gains when you are thinking about how users think. An excellent introduction to this area of research and application is:

Gray, W. D. (Ed.). (2007). *Integrated models of cognitive systems*. New York: Oxford University Press.

2.6 Exercises

2.1 Consider a smartphone, either a specific one or a composite one, and consider the human factors of using it. What are the issues that each field of HCI, human factors, and cognitive ergonomics address?

Write short notes (about one side of a page in total) noting the issues on these three types of analyses.

2.2 Pick a company's web site or a university department's web site. Summarize in note form how each of the major fields noted in this chapter would analyze it and its users. Note what would be the outputs and typical recommendations. Which approach would you prefer to apply to the web site you choose? Note the relative value and the absolute value of each. That is, which gives the best results for the amount of inputs, and which gives the best value without regard to cost?

2.3 When you go home tonight, take a look at your kitchen. Look at all the displays in the kitchen and summarize what information they contain and when you would use that information. Look at the layout of the kitchen and think about whether things are placed in the most convenient place to make your movements through the kitchen when you are cooking as efficiently as possible. Make your favorite snack and draw a picture of how you move through the kitchen. Note how the kitchen can be improved based on your analysis, including both no-cost and expensive changes. This exercise is designed to make you think more deeply about the physical, cognitive, behavioral, and information issues that go into how optimized your kitchen is for you to use.

2.4 Analyze Hedge's (2003) set of design principles in Table 2.1. These principles arose out of installing a popular operating system.

(1) For each principle, note the support it has in general, when it would be true, and exceptions where it would not be true.
(2) Comment on the usefulness and usability of the principles as a set.
(3) Compare these principles with another set of HCI design principles that you find (and note and reference).

References

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Erlbaum.

Anderson, J. R. (2007). *How can the human mind exist in the physical universe?*. New York, NY: Oxford University Press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111*(4), 1036–1060.

Avgerou, C., Ciborra, C., & Land, F. (2004). *The social study of information and communication technology*. Oxford, UK: Oxford University Press.

Bannon, L. (1991). From human factors to human actors: The role of psychology and human–computer interaction studies in systems design. In J. Greenbaum & M. Kyng (Eds.), *Design at work: Cooperative design of computer systems* (Vol. 25–44). Hillsdale, NJ: Erlbaum.

Barker, V. E., & O'Connor, D. E. (1989). Expert systems for configuration at Digital: XCON and beyond. *Communications of the ACM, 32*(3), 298–318.

Baxter, G., & Sommerville, I. (2011). Socio-technical systems: From design methods to engineering. *Interacting with Computers, 23*(1), 4–17.

Berg, M. (1999). Patient care information systems and healthcare work: A sociotechnical approach. *International Journal of Medical Informatics, 55*(2), 87–101.

Berg, M. (2001). Implementing information systems in health care organizations: Myths and challenges. *International Journal of Medical Informatics, 64*(2–3), 143–156.

Berg, M., & Toussaint, P. (2003). The mantra of modelling and the forgotten powers of paper: A sociotechnical view on the development of process-oriented ICT in health care. *International Journal of Medical Informatics, 69*(2–3), 223–234.

Bødker, S. (2006). When second wave HCI meets third wave challenges. In A. Mørch, K. Morgan, T. Bratteteig, G. Ghosh, & D. Svanaes (Ed.), *NordiCHI Nordic Conference on Human-Computer Interaction October 14–18,* (pp. 1–8). Oslo, Norway.

Booth, P. (1989). *An introduction to human-computer interaction*. Hove, UK: Erlbaum.

Brown, C. M. L. (1988). *Human-computer interface design guidelines*. Norwood, NJ: Ablex.

Busemeyer, J. R., & Dieterich, A. (2010). *Cognitive modeling*. Thousand Oaks, CA: Sage Publications.

Card, S. K., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.

Cherns, A. (1987). Principles of socio-technical design revisited. *Human Relations, 40*(3), 153–162.

Churchill, E., Bowser, A., & Preece, J. (2013). Teaching and learning human-computer interaction: Past, present, and future. *Interactions, 20*(2), 44–53.

Clegg, C. (2000). Sociotechnical principles for system design. *Applied Ergonomics, 31*, 463–477.

Eason, K. (1984). Towards the experimental study of usability. *Behaviour and Information Technology, 3*(2), 133–143.

Emery, F. E., & Trist, E. L. (1960). Socio-technical systems. In C. W. Churchman & M. Verhulst (Eds.), *Management science models and techniques* (Vol. 2, pp. 83–97). Oxford, UK: Pergamon.

Haynes, S. R., Cohen, M. A., & Ritter, F. E. (2009). Designs for explaining intelligent agents. *International Journal of Human-Computer Studies, 67*(1), 99–110.
Hedge, A. (2003). 10 principles to avoid XP-asperation. *Ergonomics in Design, 11*(3), 4–9.
Hewett, T. T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., et al. (1996). *ACM SIGCHI curricula for human-computer interaction*. New York, NY: Association for Computing Machinery. http://sigchi.org/cdg/index.html
Hollnagel, E., & Woods, D. D. (1983). Cognitive systems engineering: New wine in new bottles. *International Journal of Man-Machine Studies, 18*, 583–600.
Hollnagel, E., & Woods, D. D. (2005). *Joint cognitive systems: Foundations of cognitive systems engineering*. Boca Raton, FL: CRC Press.
Holloway, W. (1991). *Work psychology and organizational behaviour: Managing the individual at work Thousand Oaks*. CA: Sage.
Laird, J. E. (2012). *The Soar cognitive architecture*. Cambridge, MA: MIT Press.
Mosier, J. N., & Smith, S. L. (1986). Application of guidelines for designing user interface software. *Behaviour and Information Technology, 5*, 39–46.
Mumford, E. (1983). *Designing human systems for new technology—The ETHICS method*. Retrieved March 10, 2014, from http://www.enid.u-net.com/C1book1.htm
Mumford, E. (1995). *Effective systems design and requirements analysis: The ETHICS method*. Basingstoke, UK: Macmillan Press.
Mumford, E. (2006). The story of socio-technical design: reflections in its successes, failures and potential. *Information Systems Journal, 16*, 317–342.
Mumford, E., & MacDonald, W. B. (1989). *XSEL's progress: The continuing journey of an expert system*. New York, NY: Wiley.
Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.
Newell, A., Shaw, J. C., & Simon, H. A. (1960). Report on a general problem-solving program for a computer. In *International Conference on Information Processing*, (pp. 256–264). UNESCO: Paris.
Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.
Nickerson, R. (1969). Man-Computer interaction: A challenge for human factors research. *Ergonomics, 12*: 501–517. (Reprinted from *IEEE Transactions on Man-Machine Systems, 10*(4), 164–180).
Norman, D. A., & Draper, S. W. (Eds.). (1986). *User centred system design*. Hillsdale, NJ: Erlbaum.
Norman, D. A. (1988). *The psychology of everyday things*. New York, NY: Basic Books.
Norman, D. A. (2013). *The design of everyday things*. New York, NY: Basic Books.
Pew, R. W., & Mavor, A. S. (Eds.). (2007). *Human-system integration in the system development process: A new look*. Washington, DC: National Academies Press. http://books.nap.edu/catalog.php?record_id=11893. Accessed 10 March 2014.
Ravden, S., & Johnson, G. (1989). *Evaluating usability of human-computer interfaces: A practical method*. Chichester, UK: Ellis Horwood.
Reason, J. (1997). *Managing the risks of organizational accidents*. Aldershot, UK: Ashgate.
Shackel, B. (1991). Human factors for informatics usability book contents. In B. Shackel & S. J. Richardson (Eds.), *Usability—context, framework, definition, design and evaluation* (pp. 21–37). New York, NY: Cambridge University Press.
Swain, A. D., & Guttman, H. E. (1983). *A handbook of human reliability analysis with emphasis on nuclear power applications*. Washington, DC: US Nuclear Regulatory Commission.
Whetton, S. (2005). *Health informatics: A socio-technical perspective*. South Melbourne, Australia: Oxford University Press.
Woods, D. D., & Hollnagel, E. (2006). *Joint cognitive systems: Patterns in cognitive systems engineering*. Boca Raton, FL: CRC Press.
Young, R. M., Green, T. R. G., & Simon, T. (1989). Programmable user models for predictive evaluation of interface designs. In *Proceedings of CHI'89 Conference on Human Factors in Computing Systems*, (pp. 15–19). ACM Press: New York, NY.

http://www.springer.com/978-1-4471-5133-3

Foundations for Designing User-Centered Systems
What System Designers Need to Know about People
Ritter, F.E.; Baxter, G.D.; Churchill, E.F.
2014, XXX, 442 p. 108 illus., Softcover
ISBN: 978-1-4471-5133-3