19

Tool use

Humans use tools routinely and have done so for millennia. Such tools include axes, pens, plows, and pots. What makes these objects tools is that they are not attached to the body but can be held to bring about changes in the condition of other objects [772]. By extension, the idea of *tool use* in human–computer interaction (HCI) is that a computer system is a tool for *controlling something else*. According to this view, a user using a system to accomplish a task is not markedly different from a person using a hammer to drive nails or an algebraic rule to do calculations in one's head.

The computer, when viewed as a tool, is manipulated by users. Their goal is to do something that goes beyond the interface to effect some desired change in the world. By using this tool, users extend their capabilities.

In essence, tool use is about manipulating technology to achieve some aim that goes beyond the tool itself. Some researchers have taken the view of interaction as tool use at face value. For instance, Beaudouin-Lafon [53] departed from the idea that the manipulation of physical objects with our hands can be used as the basis for designing new user interfaces. He separated domain objects that are manipulated from interaction instruments, which are computer artifacts that manipulate domain objects. For example, a scrollbar is an interaction instrument, or tool, that operates on documents. Further analysis reveals it has low integration because a 1D action is controlled by a 2D mouse, and it has low compatibility in some designs because the content moves in a different direction from the movement of the scrollbar.

More broadly, the view of interaction as a tool suggests three essential components: problems, tools, and people (Figure 19.1). Several models of tool use capture these components [e.g., 135, 448, 465, 582].

This simple view can be used to derive several principles that are central to HCI. First, understanding computers as tools emphasizes the *utility* of a tool, by which we mean how well it supports what people want to do. Second, tools may be more or less easy to manipulate when used. This is called *usability*. These three aspects are interrelated. All these models emphasize that tools influence the tasks we do and how we do them. At the same time, the tools influence us. They change how we think, what we consider easy, how we express ourselves, and how we work together. This is orthogonal to utility: A tool may be useful but difficult to operate, while another tool may offer trivial value but be easy to operate. This was captured in the 1980s by Golden [280, p. 4]: "it is perfectly possible to have a program which is structured, modular, readable, flexible, self-documenting, maintainable, which performs its specified function, and which is a source of constant frustration and irritation to its users." Tools may also be more or less *accessible* to their users. And finally, tools may be more or less acceptable to their users.

These theoretical concepts have informed some key insights in HCI research, which we review in this chapter and in the chapter on practice (Chapter 22):

1. Usability is one of the best predictors of users' willingness to adopt software. For example, the User Burden Scale is a questionnaire for measuring the felt burden in software use [806]. It consists of six subscales: difficulty of use, physical burden, time and social burden, mental

Introduction to Human-Computer Interaction. Kasper Hornbæk, Per Ola Kristensson, and Antti Oulasvirta, Oxford University Press. © Kasper Hornbæk, Per Ola Kristensson, and Antti Oulasvirta (2025).
DOI: 10.1093/oso/9780192864543.003.0019

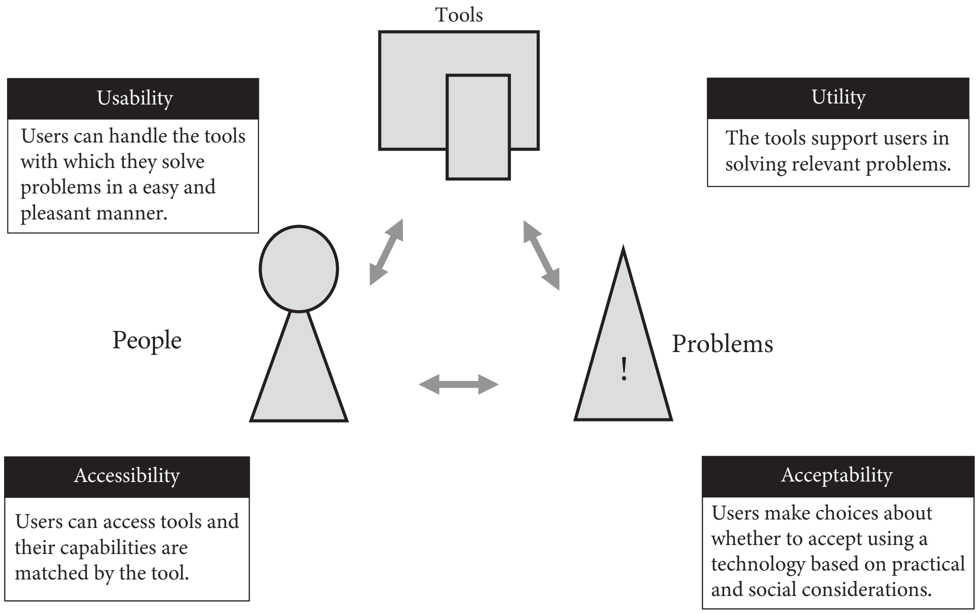

Figure 19.1 Three key components and four central considerations in tool use.

and emotional burden, privacy burden, and financial burden. Suh et al. [806] showed that software that is abandoned is associated with worse measures in all metrics except time and social burden. In other words, ease of use is an important consideration when people decide to abandon software.

2. Utility centers what users want from technology. For example, Koelle et al. [426] studied the adoption of data glasses (e.g., Google Glass, Meta Pro) over multiple years. They asked experts familiar with data glasses what would need to be improved to make data glasses more acceptable. Usefulness, functionality, and usability were the most important factors—more important than security, privacy, pricing, experience, and compatibility.
3. Users actively repurpose tools to make them more personally usable and relevant. Design should support such repurposing. For example, Renom et al. [696] conducted a study on text editing using a novel user interface. They found that exploration and technical reasoning facilitate creative tool use. Users who explore available commands in a tool are better at repurposing its functionality. More surprisingly, engaging in technical reasoning (reasoning about functionality and objects) supports repurposing more than procedural knowledge inherited from other software.

Next, we discuss these concepts and their associated measures and design implications.

19.1 Utility

The *utility* of an interactive system concerns its match with the tasks of users. If the match is good, the tool has high utility; if the tasks that users want to do are not supported by the tool, the tool has low utility. The tasks that users want to do using a tool may be existing tasks or some new tasks they have not yet realized. Users may realize their wants only when using a system, which makes the prediction of utility inherently hard. Moreover, even routine users of software may not be able to articulate the utility they gain from it.

Utility is about the relationship between functionality and users' needs and wants. It can be assessed in different ways. One way is through surveys of the perceived utility [180]. Such surveys may ask users whether a tool would "enable them to do their tasks more quickly" or "improve their job performance." Other researchers have developed criteria for assessing utility [386]. For them, utility can be ensured by checking that systems are "available on a wide range of devices and media along with integration of other resources" and "robust on basic functionality important functionality."

Grudin [300] noted that utility has typically been a key concern of research on information systems. For decades, systems have been developed in-house or contracted, and at the time of Grudin's analysis, their use was typically mandated. Hence, the question of utility in information system research has been prominent. However, many interactive systems depend on users experiencing them as easy to use and with few barriers. As such, there is a contrast between *getting the right design* and *getting the design right* [290]. The former is about utility; the latter is about usability. One shorthand way of expressing this is that utility is "whether the functionality of a system in principle can do what is needed" [591, p. 25]. In practice, whether people can do anything concerns—among other things—usability.

19.2 **Usability**

Some tools are easy to operate. It is clear what they can do, it is easy to learn to use them, and they may be safely used. For computer-based tools, these considerations are described by the term *usability*, one of the key empirically measurable constructs that the field of HCI has developed.

Usability concerns how easily computer-based tools may be operated by users trying to accomplish a task. Usability differs from utility. Usability concerns whether users can use the product in a way that makes it possible to realize its utility; utility is about whether the goal is important to the user. Ideally, the user can use the tool without unnecessary effort so that the use is direct, transparent, and unnoticeable.

Research on usability has been ongoing since the 1970s, departing from research in human factors and early notions of user-friendliness and ease of use. Since then, various ways of thinking about usability have emerged [332, 757]. The current way of defining usability is captured in the ISO 9241-11 definition, based on work by Bevan [68] and many others, which defines usability as the "extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."

19.2.1 **Characteristics of usability**

Research on usability has uncovered several key principles and insights. One insight is that *usability is relational*; it arises as an interplay between people, tasks (problems), and interactive systems (tools); see Figure 19.1. Both the ISO definition and the introduction to this chapter make this clear: Tools are usable as a consequence of their fit to people and tasks. For this reason, usability is sometimes called *quality-of-use* to emphasize that it is a quality of interactive systems, like maintainability and reliability. However, usability emerges only when people actually use tools to achieve their goals. Because of this, it makes no sense to talk about the usability of an interactive system without considering the users and their tasks. It also does not make sense to talk about usability as a property of interactive systems. A system may be usable for some tasks and less usable for others; it may be usable for some users but not for others. In any case, usability is not a property of the interactive system.

Another key insight is that *usability is measurable*, that is, it is possible to quantify usability based on users' behaviors or opinions. We may evaluate usability as part of iterating the design of an interactive system. We may also use this insight to articulate what requirements a design should fulfill. For instance, Whiteside et al. [886] showed how to make explicit quantitative goals for usability. They provided an example of the usability of software installation. This was quantified through the time it takes to install software. This could take one hour or, in the best case, just 10 minutes. These numbers may be stated upfront and tested later on. We may also use usability measures to experimentally quantify differences between user interfaces. We may measure the time used and errors made by a user while solving a problem with different tools and then compare the results (more on experimental comparisons in Chapter 43).

Earlier work has also shown that *usability is multidimensional*. This means that in most settings, a valid characterization of usability will need to employ several dimensions and measures. This is typically found in studies where multiple aspects of usability are measured and modest correlations are found. For instance, Nielsen and Levy [592] compared users' performance and their preferences across 57 studies and found what they called a strong positive correlation. Nevertheless, they concluded that "there are still many cases in which users prefer systems that are measurably worse for them, so one should exercise caution" [p. 75]. Later studies explored this finding in different ways, generally found some correlation, and cautioned against treating usability as a one-dimensional construct [261, 356, 733]. Thus, one should typically pick multiple indicators of usability and assume that different aspects of usability (e.g., subjective and objective measures, performance, and outcomes) should be measured separately.

19.2.2 **Models of usability**

The characteristics described in Section 19.1.1 have led to numerous *models of usability*, dimensions of usability, and indicators or measures of those dimensions [e.g., 591, 757]. These models can help us identify some dimensions of usability that are independent. In other words, they capture different aspects of usability. These models also help us pick measures for evaluating usability and bring structure and rigor to our everyday understanding of ease of use. The quest to spell out the dimensions of usability has been going on for decades, so summarizing and integrating this work is difficult. Nevertheless, two models of usability are particularly useful.

One is the ISO 9241-11 model of usability. It defines usability as comprising effectiveness, efficiency, and satisfaction. Table 19.1 shows the definition of these dimensions as well as examples of how they may be measured. Another model for thinking about usability is that of Nielsen [591], which identifies five dimensions of usability (also in Table 19.1).

19.2.3 **How to select goals for measuring usability?**

Given these and other models of usability, an important question remains: How does one select or prioritize between dimensions and measures for a particular case? For instance, is the time it takes to complete a task always a good indicator of usability and one that we should seek to minimize? Is effectiveness always more important than satisfaction, or is it the other way around? In practice,

Table 19.1 Two models of usability, based on ISO 9241 and Nielsen [591], including their dimensions, definitions, and examples of how they may be measured.

Model	Dimension	Definition	Example measures
ISO	Effectiveness	The accuracy and completeness with which users achieve goals	Binary task completion, error rates, quality of outcome
	Efficiency	The resources spent by users to achieve goals	Task completion time, input rate, mental effort, learning
	Satisfaction	The users' comfort with and positive attitudes toward the use of the system	Preference, perception of ease of use, willingness to recommend to others
Nielsen	Easy to learn	The user should be able to rapidly start getting work done with the system	Self-reported learning effort, number of hours required to achieve a minimum level of proficiency
	Efficient to use	The highest level of productivity that proficient users can achieve	Task completion time and efficacy in representative tasks
	Easy to remember	Users should be able to return to a system without needing to relearn how it works	Times a manual or help is needed, change in user performance after a period of non-use
	Few errors	Users should be able to operate a system with as few errors as possible	Number of erroneous or hazardous actions
	Subjectively pleasing	Users should find the system pleasant to use	Questionnaire-based metrics (see chapters 7 and 13)

the selection of usability measures requires careful judgment—it is a matter of determining the importance and validity of parameters.

The validity of a measure concerns whether it measures what it is supposed to measure, in particular whether it is an accurate and complete indicator of the usability of an interactive system. Validity will be discussed in more detail in Chapter 43.

Newman and Taylor [587] presented a concrete approach to selecting usability measures, which they called selecting *critical parameters*, that is, "performance parameters that measure how well the system serves its purpose." The critical parameters of a system are those parameters that allow designers and evaluators to establish whether an interactive system serves its purpose and compare designs.

The categories of "satisfaction" in the ISO model and "subjectively pleasing" in Nielsen's model both refer to the experience of using a tool to accomplish a task. Rather than looking at the total experience, such as expectations, memories, and general affect, satisfaction has historically been understood as more narrowly concerned with the experience of using a tool to accomplish a task.

Numerous validated questionnaires for measuring satisfaction are available. They are typically administered to users after they have used an interactive system; the answers are summed or otherwise combined to form an indicator of satisfaction. Some questionnaires aim to measure the full satisfaction with a system. Paper Example 19.2.1 shows one of the most popular questionnaires for doing this: the System Usability Scale (SUS). It gives a satisfaction rating between one and 100 for an interactive system. Questionnaires with different levels of breadth exist for satisfaction:

- The questionnaire for user interface satisfaction [150] originally consisted of 90 questions but often only five questions are used. The items are rated on nine-point semantic differentials going from terrible to wonderful, from frustrating to satisfying, from inadequate power to adequate power, from dull to stimulating, and from rigid to flexible.
- Some questionnaires are very short to facilitate quick answers. The UMUX-Lite consists of just two items, “This system’s capabilities meet my requirements” and “This system is easy to use,” each answered on a seven-point scale.
- The subjective mental effort questionnaire (SMEQ) rating scale consists of just one question answered with a graphical scale.

The aforementioned questionnaires cover satisfaction in general. In addition, numerous specialized questionnaires have been developed to explore different dimensions of satisfaction. For instance, Suh and colleagues [805] developed a questionnaire for user burden.

Paper Example 19.2.1: System Usability Scale (SUS)

In the mid-1980s, the SUS was developed [97]. While other questionnaires for satisfaction were developed around the same time, SUS has become one of the most widely used scales for measuring the users’ perception of usability.

SUS was developed to be relatively quick to use and freely available. Users should answer the questions reported below immediately after using the interactive system being assessed. The SUS consists of 10 questions answered on a Likert scale (from “strongly disagree” to “strongly agree,” coded from 1 to 5):

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very awkward to use (changed from the original “cumbersome,” as recommended by Bangor et al. [39]).
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The answers to these questions are summed to calculate the overall SUS score. For odd-numbered items, subtract 1 from each score (1–5); for even-numbered items, subtract each score from 5. Then, sum these values to obtain the total SUS score. Brooke stressed to only use and interpret the summed score, not the answers to the individual questions. The summed score is indicative of the level of satisfaction with the interactive system and may be compared to the scores of other versions of the system as well as those of other systems.

Research has shown that SUS can discriminate between systems with poor and good usability, can be used with a range of technologies, correlates modestly with task performance, correlates well with other questionnaires, and has good reliability [39, 96].

19.3 Acceptability

People may choose to use a particular tool or something else to solve a task, or they may give up solving the task. These possibilities are related to the *acceptability* of the tool, that is, whether users choose to use the tool when given that option. HCI researchers have worked on understanding what factors shape acceptability; designers aim to create interfaces that users accept.

Acceptability has two main dimensions [591]. The first dimension, *practical acceptability*, includes costs, the reliability of the interactive system, and its compatibility with other systems. The perceptions of utility and usability may also influence the judgment of practical acceptability. This follows the model of inference about experiences outlined in Chapter 7.

The second dimension, *social acceptability*, concerns whether interactions map well to the social norms and roles in the settings where they occur. Koelle et al. [427] surveyed approaches to social acceptability and found indicators such as "disturbing," "inappropriate," "creepy," and "impolite." These indicators suggested that users, or indirect users such as onlookers, did not find an interaction socially appropriate. For example, social acceptability was an important consideration for early smart glasses, that is, eyewear with computational capabilities, particularly models fitted with cameras [426].

Acceptability includes the choice to *not* use a system. The studies of non-use in HCI suggest some processes and reasons why a useful, accessible, and usable interactive system may not be used by the intended user group. Such considerations include bias and accessibility.

19.3.1 Technology acceptance model

The most common way of working with acceptance is the technology acceptance model (TAM). Underlying TAM is the theory of reasoned action. Davis [180] proposed that whether an individual ends up using a system, that is, their *usage behavior*, depends on their *intention to use* the system. However, what affects the intention to use? TAM answers this question.

TAM posits that the intention to adopt a particular technology is driven by two kinds of perceptions: (1) how easy it is to use a system and (2) how useful it will be to use it [180]. Furthermore, the perceived ease of use affects the perceived usefulness: If technology is hard to use, it is less useful.

Since the 1980s, many studies have empirically validated TAM and extended it to include indicators of social norms, the availability of support, and other factors. From the HCI viewpoint, TAM has a number of strengths and limitations [355]. It says very little about interaction, which in the end is the basis of the two critical types of perception. It also has very few nontrivial

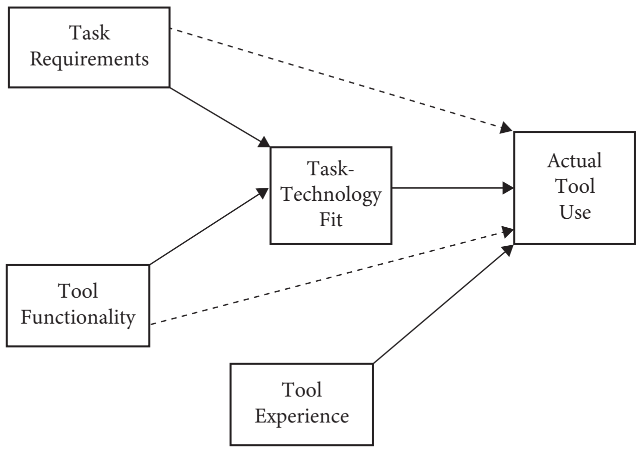

Figure 19.2 Task–technology fit refers to how well technology supports the task demands and individual capabilities of the user [197].

implications for design. While it may indicate which aspect of tool use is more important—utility or usability—it does not indicate *what* makes a system useful or usable.

19.3.2 **Task–technology fit**

The theory of *task–technology fit* (TTF) can illuminate what users consider useful and how this affects their decision to adopt a particular technology. TTF refers to the ability of technology to support a task [197]. The capabilities of the technology should match the demands of the task and the skills of the individual; in this case, the fit is perfect. TTF theory posits that a rational user will choose the tool with the highest fit due to its efficacy and efficiency. Conversely, a system that does not offer a good fit will not be used.

TTF has been modeled with several statistical models, including the one shown in Figure 19.2. The goal of the model is to predict a user's willingness to use a tool, which is affected by the user's experience with the tool and by the fit. The fit is affected by two key factors, task requirements and tool functionality, which must match to have a high fit. A TTF model may include a component for individual abilities, the *tool experience* component, to account for the fact that prior experience with computers can positively affect utilization.

TTF has been used to assess users' willingness to use various technologies such as email or spreadsheets. However, such models have been criticized for assuming that users are rational. Like TAM, these models tend to dismiss "irrational" factors that affect acceptance, such as one's attitude.

19.4 **Accessibility**

While an interactive system may offer utility and usability, it may only offer these to a limited group of people. In HCI, we typically want products to be usable by as many users and in as many diverse situations as possible; in other words, we want high *accessibility*.

One prominent definition of accessibility is given by ISO 9241-171, which defines it as "the usability of a product, service, environment or facility by people with the widest range of capabilities." Accessibility is also referred to as universal design, universal usability, or inclusive design. These terms reflect similar ambitions, insights, and methods.

In this way,accessibility concerns the match between a user's abilities and the system's required abilities. As such, it differs from usability (which is about the relationship between users, tools, and tasks) and utility (which is about whether a tool may be used to complete a task). Thus, if one wants an interactive system to be usable by all major demographic groups, one should also be concerned with accessibility rather than only usability.

There is a range of communities with accessibility issues. A comprehensive literature survey of accessibility research in HCI [500] identified the following groups: (1) blind or low-vision, (2) deaf or hard of hearing, (3) autism, (4) intellectual or development disability, (5) motor or physical impairment, (6) cognitive impairment, (7) older adult, (8) general disability or accessibility, and (9) other. However, each user manifests these factors at different degrees and in different combinations.

This is a key challenge in accessibility research: There are many disabilities, and solutions that work for one community or a particular disability may not work for others. Users may also have multiple disabilities, resulting in more or stricter design requirements. Aging is sometimes grouped with disabilities, as in the above list, even though it is not a disability. A discourse analysis of over 600 HCI research papers [856] found that aging is typically framed as a "problem" that can be managed by technology, a stance that has been framed as a form of ageism in critical gerontology.

There are many reasons for designing accessible interactive systems. Perhaps the foremost argument is that it is the right, ethical thing to do. A second reason is the large group of users affected by one or more disabilities. According to the World Health Organization, over one billion people have some form of disability.[1] A third reason is that it is a legal requirement in many jurisdictions to ensure interactive computer systems and services are accessible to a wide range of users with different disabilities. Finally, a fourth reason is that considering accessibility throughout can result in an improved design for all. The so-called curb-cut argument is an example of this. Cutting the curb and providing a ramp for approaching the pavement does not only benefit people with motor disabilities; this design benefits anyone with a situational impairment, such as a person with luggage. This thinking is sometimes referred to as "universal design" or "design for all."

19.4.1 Designing accessible interfaces

A large survey on disability research in HCI [500] identified three main strands of research: (1) increasing access to digital tools by offering users new forms of technology; (2) understanding users' needs, preferences, and abilities; and (3) increasing access to the physical world.

In practice, a significant proportion of HCI research is focused on arriving at technological solutions and designs for specific problems. Several approaches to accessibility design and research have emerged. One is ability-based design [906], which involves adopting a design stance in which a design is specifically tailored to the end-users' needs and abilities. Ability-based design thus requires a deep understanding of the target groups. Various practices, such as co-design, are often used to involve representative users of the target audience in the design process.

[1] https://www.who.int/en/news-room/fact-sheets/detail/disability-and-health.

Another approach is inclusive design, which is a design method for ensuring products and services are usable by a very wide spectrum of users. Inclusive design is process-oriented and adopts, modifies, and extends several product design processes originating in engineering design. It advocates the use of specific methods and toolkits to achieve inclusivity, such as the inclusive design toolkit. An important aspect of inclusive design is obtaining accurate statistics on relevant characteristics of the user population. These parameters (e.g., grasping strength) are then fed into various toolkits used by designers to ensure a wide range of users with varying abilities can be accommodated. To effectively assess the target audience, it is vital to sample representative individuals and user groups. This is challenging because it requires knowledge of which user groups must be sampled and in what proportion as well as the ability to reach user groups that may be difficult to reach.

Inclusive design is desirable because, when consistently applied, it ensures many widely used products and services can be used by as many people as possible. At the same time, there is often a limit to how broad a range of users a product or service can be designed to accommodate. For example, augmentative and alternative communication (AAC) is concerned with supporting non-speaking individuals with motor disabilities. AAC users rely on speech-generating devices (SGDs) to communicate with other people. For example, amyotrophic lateral sclerosis and cerebral palsy patients may be AAC users.

19.4.2 **Designing speech-generating devices**

SGDs vary depending on the needs and wants of the user. An illiterate AAC user typically relies on a symbol- or picture-based interface. A literate AAC user enters letters, words, and sentences that are then communicated using speech synthesis. As AAC users rely on their SGDs, many aspects of this speech output are important, such as the tone and accent of the speech synthesis voice. For literate AAC users, many text entry mechanisms are available, such as physical keyboards and touchscreen keyboards. These ordinary keyboards are usually coupled with word, phrase, and sentence prediction tools intended to increase the text entry rate.

Other SGDs use interfaces based on a single switch. These interfaces may be used when a user is unable to use a physical keyboard or a touchscreen keyboard. An example of such a switch is an eyebrow switch, which activates when the user activates the muscles in the vicinity of the eyebrow. To use such a switch for typing, the SGD interface must be designed with this in mind from the beginning. The most common solution is a scanning keyboard. A scanning keyboard presents all letters in the alphabet along with some ancillary punctuation symbols, such as period and question mark, on a grid. The system begins by highlighting each row in turn. If the user activates the switch, the row is selected. The system then highlights each letter in the row. The letter that is highlighted when the user activates the switch is selected.

Activating the switch is cumbersome and error-prone. In addition, the user must wait for the desired item to be highlighted, meaning the entry rate is low. For this reason, scanning systems are often augmented with some form of word prediction system. It is also possible to optimize the scanning pattern to minimize the average waiting time for a designed selection. Regardless, false activation is a significant problem with single-switch systems.

Another SGD variant relies on eye tracking, which may be suitable for users who have no effective means of interaction beyond eye control. The typical communication solution uses a technique known as eye-typing. To communicate using eye-typing, the user moves their gaze across an onscreen keyboard. However, the system does not know whether the user wants to type a key or is merely looking at a key. This is known as the Midas touch problem. To disambiguate these two possible user intentions, eye-typing relies on a dwell timeout. If the user fixates on a key for a sufficiently long time, the system assumes the user means to type that key. Typically, this dwell timeout is shown visually to the user to both indicate the timeout and to draw the user's visual attention during the timeout.

Eye-typing is an effective means of communication; however, it is not efficient. Three fundamental problems prevent high entry rates. First, the eyes are sensory organs and not control organs. It is difficult for users to artificially maintain fixation on specific keys. Second, the dwell timeout provides a low ceiling on performance. Third, people think in terms of words, phrases, and sentences when they communicate. Eye-typing forces users to think in terms of individual letters. This has a cognitive cost and is not a fluid means of communication. Many of these considerations also apply to single-switch systems.

Eye-typing has been improved over time. Commercial eye-typing products rely on word and phrase suggestions to improve the entry rate. Another solution is to use dwell-free eye-typing, which removes dwell timeouts. To achieve this, the system uses a statistical decoder to translate observations of the user's gaze into hypotheses of what the user wishes to write. Dwell-free eye-typing has been released as part of a commercial eye-typing product but has so far not seen widespread adoption.

These examples illustrate some of the challenges that accessibility poses to design and highlight the need for thorough user research.

Paper Example 19.4.1: Generating user interfaces based on user's abilities

Supple++ [266] is a computational method developed in HCI that can improve graphical user interfaces to better fit a user's unique motor and vision abilities. In Supple++, the user is first asked to perform a series of motor tasks. This information is used to calibrate an internal computational model of the user's motor ability. Once the calibration is complete, Supple++ optimizes the user interface automatically by changing the size and location of user interface elements and the organization of the user interface, subject to constraints specified by the designer.

This constraint-based optimization allows for changing the user interface according to individual abilities without losing important qualities of a well-functioning user interface, such as consistency and clear organization. For example, a user who can make rapid but imprecise movements is provided with a user interface with large user interface elements that are farther apart. By contrast, a user who can only make slow but precise movements is provided with an interface in which user interface elements are smaller and closer together.

Figures 19.3 and 19.4 show examples of automatically generated interfaces based on individual abilities. Figure 19.3 shows an interface for a user with muscular dystrophy who relies on the mouse to make selections. Figure 19.4 shows an interface for an eye-tracker user. Note that these interfaces did not just change icon sizes and locations; they also changed the user interface organization to best accommodate their respective users.

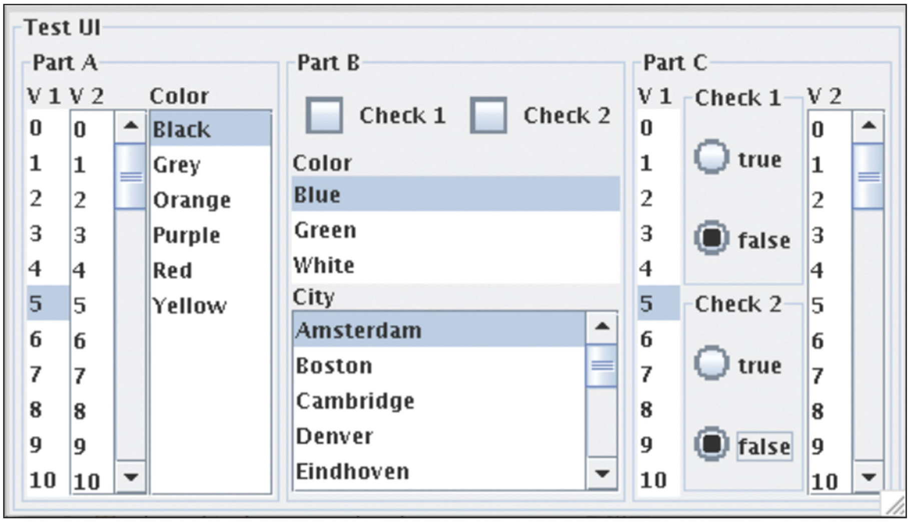

Figure 19.3 An interface generated by Supple++ for a mouse user with muscular dystrophy [266].

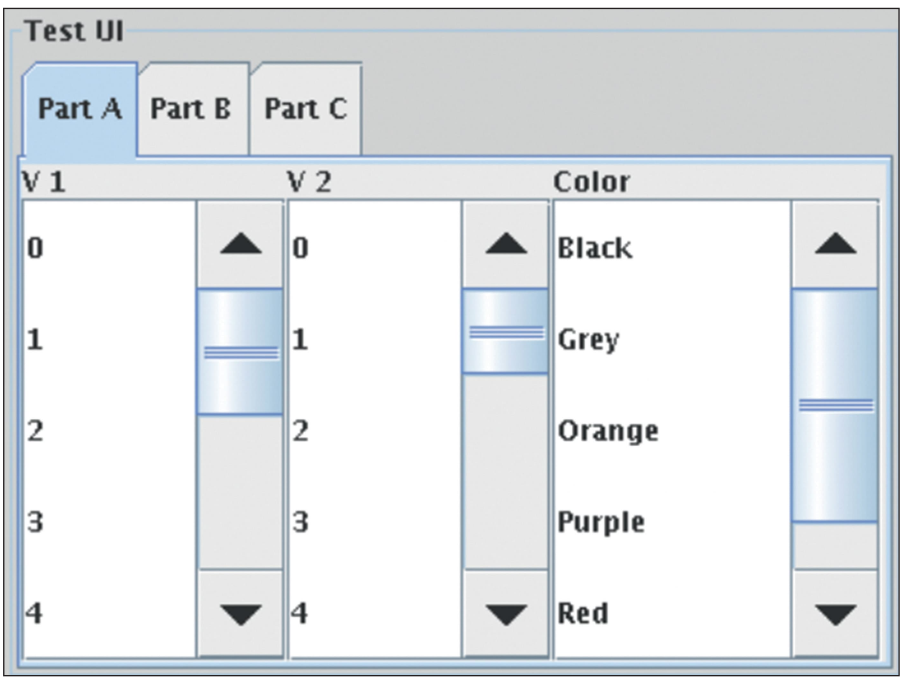

Figure 19.4 An interface generated by Supple++ for an eye-tracker user [266].

19.5 Tools change how we think and perceive

Using a tool for extended periods can fundamentally change the way a user thinks and perceives both the tool and the world. First, using a tool for an extended period can result in the *cognitive integration* of the tool. Cognitive integration means that we internalize the operation of the tool.

We not only act but also start thinking as defined by the unique constraints and mechanisms of the tool.

What is such integration based on? Tversky and Jamalian [833] proposed that embodied action is at the core of this. We move our bodies and toss, push, and pull objects. These movements can be thought about, imagined, and referred to in language. This, in turn, can change the substrate of thinking. For example, the expression "on the one hand, on the other hand" is based on our experience of hands.

Tversky and Jamalian demonstrated this claim in a clever experiment. People listened to descriptions of events in three conditions: circular gestures, linear gestures, and no gestures. Then, they were asked to draw a diagram of the event. The authors were interested in *how* they depicted the event. They found that after listening to the descriptions with linear gestures or no gestures, participants mostly drew linear diagrams, following the narrative structure of the descriptions. However, after listening to the descriptions with circular gestures, they predominantly drew cyclical diagrams. Thus, gestures affected the way participants constructed mental representations of the events.

Many tools we use, both digital and physical, are based on movement; by extension, they may change the way we think and reason. For example, the abacus is a wooden device used for teaching basic calculations. It consists of a frame with rows of wires along which beads can slide. Students who learned to do calculations with an abacus solve mathematical problems differently from others [796]. They rely more on mental imagery of the movement of beads on the abacus, which makes their mental calculations highly efficient for certain types of calculations.

The second influence from using tools, is that it may change the way *we perceive the world*. The tool itself may become "transparent" and we start perceiving "through it." In philosophy, a tool is argued to be "ready-at-hand." The tool itself is forgotten, and we construct percepts of the environment mediated by it.

Blind cane users are a good example [756]. When blind users learn to sense the environment with a cane, their perception of tactile and auditory stimuli slowly changes. Instead of sensing stimuli close to their hand, when they hold the cane, they can integrate tactile (vibration) and auditory stimuli close to the tip of the cane. They develop multimodal, integrated percepts that correspond to the tip of the cane. This was demonstrated in an experimental study where blind and sighted users were asked to react to auditory and tactile stimuli either close to their hand or farther away in the environment. Blind participants were faster in reacting to stimuli close to their hands when holding a short cane. When holding a longer cane, they were better at reacting to far sounds than to near sounds. Interestingly, when sighted users spent just 10 minutes practicing with a cane, their peripersonal space also expanded. However, when the tool was taken away, it contracted back. Learning tool-mediated multimodal integration requires time.

The cognitive integration and perceptual influence of tools are not without problems. While a tool can enhance performance in cognitively challenging tasks, its extended use may erode the cognitive capability of the user. Galletta et al. [267] warned against the effect of spell checkers on verbal ability. Having a spell checker in a word processing program may make users overly rely on the tool even if it makes several mistakes, both false positives and false negatives. The authors showed experimentally that university students who had a spell checker on during a document editing task had more errors left in the document than those who did not. Even though the spell checker only fixed the most obvious errors, having it on made users overly dependent on it.

Navigation aids are another case in point. Before mobile maps and navigation aids, we did not routinely consult maps when navigating in cities. When navigating in a city without a map, our brains actively represent and reason about our whereabouts. We use at least three kinds of neural representations: place cells, which encode locations; head-direction cells, which represent the direction of our field-of-view; and grid cells, which position the world in a coordinate-like system with the ability to compute scale and distance. In the era preceding digital navigation aids, to get from Point A to Point B, we needed to form spatial representations before the trip. These representations would get us most of the way, if not all the way. When moving around in a city, we needed to track our position on our mental map. Mobile navigation aids fundamentally changed this. Presently, we can obtain directions for navigation virtually at any time, which decreases the need for spatial cognition.

Does this mean that users who use navigational skills lose their spatial cognitive capabilities? McKinlay argued that spatial skills are "use it or lose it" [533]. However, navigation aids are only part of the picture. We still need to navigate in digital environments like games and hypertext; such activities help us retain our spatial skills. These examples show that viewing interaction as tool use reveals a surprisingly complex influence between people and interactive computing.

Summary

- The use of interactive systems can be viewed as tool use—using a system to achieve a particular objective.
- The fit of a tool describes how well its functionalities match the demands of the task and the capabilities of the user.
- Tools should be accessible to users, provide value in the world, be acceptable, and be usable.
- Tools can profoundly transform the way we act and think.

Exercises

1. Understanding tool use. This chapter has presented the view that computer use is similar to tool use. However, it is not always clear what this view encompasses. Consider whether (a) computer games, (b) visiting online virtual museums, and (c) instant messaging with friends are examples of tool use. If they are not, how should we think about these interactions? If they are, what are their outcomes and how do the principles about tools discussed in this chapter apply to the examples?
2. Concepts. Brainstorm and write down the names of two different systems, apps, web pages, or other types of computer systems. For each, identify what you deem the most important metric of success, taken from measures of usability and user experiences (Consult chapter X for a refresh). Then, consider why you find those metrics important.
3. Usability metrics. Compare the models of usability presented in this chapter: ISO 9241 and Nielsen's model. How do they overlap, and what are their non-overlapping parts?
4. System Usability Scale. Pick up the email application you use on your phone. Then, refamiliarize yourself with it. Then, fill in the SUS questionnaire and calculate the SUS score.