Kasper Hornbæk, Per Ola Kristensson, and Antti Oulasvirta. Upcoming 2024. Introduction to Human-computer Interaction. Oxford University Press.

Part II.

Understanding People

2. Introduction to Understanding People

Can we design for people if we do not understand them? History has shown that we cannot—user interfaces that fail often lack awareness of users' behaviors, feelings, needs, and wants. For example, mobile text entry methods have failed if they have not supported users in becoming more efficient in using them [427] and collaborative systems have failed if they have been unable to incorporate an understanding of people's needs, motivations, and work practices [297].

In the introductory part of this book (Part I), it was noted that human-centredness is one of the pillars of HCI as a discipline. It requires us to take a real interest in people. It requires us to base decisions in design and study on understanding of people. But *how* is one human-centered, what does it mean to work in a human-centered way?

To claim that a choice is taken with people in mind—that it is *human-centered*—means that the choice is justified by reference to knowledge of how people feel, think, and behave. For example, we may choose colors in a user interface to match how the human visual system works, design mechanisms to an application that motivates changes in unwanted behaviors, such as smoking, or choose a product concept to design based on ideas about what people desire. Further, decisions about how to evaluate a collaborative system can be based on knowledge about how people coordinate work when aiming to achieve a shared goal.

In these and many other instances, we know things about people—from research in HCI or through other sciences—and put that knowledge to use in HCI activities. We call this knowledge and its use to improve interactive systems *understanding people.*

Understanding people is hard. Most of the factors involved in a person's observable behavior are hidden. We cannot precisely know what a person thinks, what they feel, or what drives a person's behavior. However, using scientific methods and theories, it is nonetheless possible to obtain some knowledge of these factors. Such knowledge gathering is valuable in HCI, because it allows us to reason about user behavior.

To illustrate such reasoning, consider the two photos in Figure 2.1 and compare them:

Perception The two situations differ in terms of what users must be able to perceive. During driving, the user *shares* their visual attention between the computer and the road, the latter continuously changing. In many countries, using a mobile device while driving is illegal because of the effect of the device on the user's ability to perceive events on the road.

Motor control Many games require not only fast reflexes but also the ability to intercept a fast-moving target that can be small and move erratically. Using a mobile device for non-gaming purposes mostly involves static targets selected with a very different input device, such as a touchscreen.

Thinking The gamer must keep track of several events, such as the current status of an enemy player, to choose the next action. On the contrary, the selection of options from a mobile display is based on the recognition of icons and labels that are most likely to lead to the target state.

Needs What kinds of desires and risks are associated with the two situations? Do people play games for different reasons than those that drive the use of social media? Or are there a few basic psychological needs that can be met through both activities?

Experience Immersion in a virtual world is a desired quality in gaming, but not in mobile interaction, which places greater emphasis on instrumental experiences, such as being able to complete tasks.

Collaboration In the game example, the ad hoc formation of a team of players and their mutual awareness of what is going on are essential to success. Understanding how to do that is an important part of developing interactive systems. Driving a car, in contrast, requires different collaboration (for instance with a passenger), if any.

Communication Competitive gaming is an extreme situation in which two parties have conflicting goals. Both users must infer what the other party means or desires via the limited cues on the interface. Every so often, this requires intense communication and impromptu collaboration. Likewise, driving, in this case, co-occurs with the use of a mobile phone, and thus shapes what is being communicated due to the user having to multitask.

We write about people in other parts of the book. What is the difference between this part and the others? The defining focus of this part is the pursuit of *general* knowledge about people, for instance, as theories, models, concepts, and taxonomies. We seek to understand how people feel, think, and behave in ways that hold across different times

Figure 2.1.: This section introduces methods and theories that help understand people as users of computers. Compare the two situations in the photos: desktop gaming and driving. How do you believe they differ in terms of the users' goals, experience, thinking, and cognitive demands?

Topic	**Part**	**Focus**
Understanding People	This part	Generalizable understanding of how people think, feel, and behave
User Research	Part III	Methods for obtaining insights on specific users, activities, and use contexts
Interaction	Part IV	Generalizable understanding of how users interact with computer systems
Evaluation	Part VIII	Methods for assessing interaction with specific interactive systems

Table 2.1.: Different types of knowledge produced about people in interaction with computing systems

and settings in computer use. This is a very efficient type of understanding, because it has wide applicability. For example, the mechanisms involved in how the human visual system works are universally relevant for graphical user interfaces.

However, at the same time, such knowledge is also *abstract*: it does not prescribe a *specific* user, nor a *specific* setting or configuration. Therefore, to be valuable for design and engineering, the understandings described in this part need to be complemented by other forms of insights about users. Table 2.1 shows an overview of the other parts of this book and the type of knowledge they cover. The part on user research (Part III) covers methods for gathering information on specific users, activities, and contexts of use. The part on interaction (Part IV) provides HCI theories and models that explain how users interact with a variety of computer systems. Finally, the part on evaluation (Part VIII) explains how to assess HCI systems using a variety of different approaches, such as experiments and deployment studies.

Understanding people is central to HCI and there are many examples of HCI research that contribute new insight that can inform design. Next, we give some examples.

- Borst et al. [88] proposed and evaluated a process model to predict when interruptions are disruptive to users. The experiments provide evidence for two design guidelines to minimize disruptions as a result of interruptions.

- Bachynskyi et al. [35] studied user performance and ergonomics on different touch surfaces, such as public displays, tabletops, tablets, and smartphones, thereby informing the design of user interfaces using these surfaces for interaction.

- Lottridge et al. [481] investigated how chronic multitasking with relevant and irrelevant distractors affects the quality of writing. They find that multitaskers write better essays when provided with relevant distractors and worse essays when provided with irrelevant distractors. This tells us that while multitaskers can be negatively affected by irrelevant distractors, they are also able to integrate different sources of information when writing essays.

Next, we review the different forms of understanding we employ in HCI, the different areas of understanding, and how we use them in evaluation and design.

2.1. Types of understanding

Several types of understanding are relevant in HCI. A concept may help us to see an underlying driving factor, such as a need or motivation, behind users' behavior. A computational model enables us to predict what a user can recall. A model may help separate the actions that a user performs when searching for information. What is common to the types of understanding covered in this part, and to those just mentioned, is that they are general, robust across time and place, and linkable to empirical phenomena. The types of understanding that we will discuss in this part encompass five main types.

Theories consist of constructs and relations among those constructs. Theories help to understand, explain, or predict phenomena related to interactive systems. They are more general and encompassing than models. For example, in Chapter 6 we describe self-determination theory. This theory outlines the motivations of people. It describes the factors involved in intrinsic motivation and the general human tendency to engage in activities that are seen as enjoyable and interesting.

Concepts name particular phenomena, often with additional characteristics, such as how to identify the phenomena, information on when they usually occur, or knowledge about their underpinning mechanisms. For example, in Chapter 9 we learn about turn-taking as a concept for understanding human conversations offline and online. This concept has implications for how we support communication in interactive systems.

Taxonomies propose a system of elements or mechanisms of how people think, feel, or act. For example, human memory consists of several systems (see Chapter 5). A simple taxonomy separates declarative and procedural memory, each broken down into further, distinct types of knowledge. This taxonomy may be used to analyze the types of knowledge that a particular interface requires.

Models are formally expressed simplifications of reality. Models link concepts, often in a visual form, but also mathematically or in computer code. They may be verbal or quantitative, provide numerical estimates, or allow the model to be simulated computationally. For example, Chapter 4 discusses models that predict how design affects the time it takes people to select a target, such as an icon, on a display.

Guidelines Theoretical knowledge can be summarized into practical rules of thumb or heuristics, which are frequently called guidelines. For example, the chapter on cognition (Chapter 5) provides knowledge from which one can derive guidelines for evaluating interactive systems (Chapter 41).

The above types represent different forms of understanding people. However, such an understanding cannot be static. The effective types of understanding in HCI have four qualities. First, researchers can and should update such understandings in light of new observations. An understanding of people that is immutable is not understanding—it is a dogma. Research progresses by updating and rejecting knowledge based on new evidence.

Second, simplification is often the crux of understanding – not just in HCI but in the social and behavioral sciences in general. Something complex, such as someone's behavior, is explained by reference to something more straightforward, such as a construct like personality, or a concept about collaboration. Thus, our discussions in this part of the book balance an extensive theoretical understanding of some aspects of people, and simple—actionable—approximations.

Third, understanding people should be actionable to be useful in HCI. Knowledge should help practitioners make better choices. An understanding of people that only calls for more empirical research may be interesting to academic researchers looking for a new line of inquiry but ultimately has little practical value in itself.

Fourth, claims about people should be logically sound and empirically justifiable. Theories are subject to constant scrutiny and public criticism. A theory based on someone's opinion is not really a theory. Such a scrutiny can take place both in a related discipline, such as psychology, or in the field of HCI. In the construction of theories to understand users, HCI has drawn from several areas of psychology, but most notably cognitive psychology, social psychology, and the psychology of motivation and needs. Frequently, applications of such theories in HCI have resulted in HCI feeding back information to the original disciplines that originated the theories.

2.2. Areas of understanding

HCI is fascinating because of its wide range of human activities, from delicate adaptations of movement in virtual reality to changes in adolescents' well-being due to digital communication technologies. It includes people browsing a web page and local communities using social media to organize activities. The areas of understanding that are of relevance to HCI therefore include *all* areas of understanding people, in many areas of scholarship. However, which of these areas are the most beneficial to learn about?

2.2.1. Seven areas of understanding people

In this book, we focus on seven areas that have historically been prominent in HCI research and that cover many types of understanding needed in practice. This part will summarize what each area teaches us about the individual and social factors that affect the use of computers. Table 2.2 shows an overview of these areas.

These areas allow us to understand the basic perception and motor aspects of interaction, making us aware of the limits of human performance and how design can shape such performance. The areas help to discuss people's cognition and how their needs shape people's behavior and their experiences with interactive systems. They also help us to understand communication and collaboration. Together, the areas mentioned in Table 2.2 are mainly rooted in sociology and social psychology. Knowing these areas is

Area	Focus	Relevance for design
Perception	How people see, feel, hear, taste, and smell	Informs the design of user interfaces
Motor control	How people plan and execute movements	Informs the design of input devices, interaction techniques
Cognition	How people remember, pay attention, and think	Informs the design of complex interactive tasks that require memory and reasoning
Needs	How needs work to motivate people	Tells us what is important for users in computer use and beyond
Experience	How people experience and form experiences	Tells us how users experience events involving computers
Communication	How people communicate with each other	Informs the design of services and applications for human-human communication
Collaboration	How people achieve joint goals	Informs the design of collaborative software

Table 2.2.: Seven areas of understanding people that provide useful knowledge on human use of computing.

essential knowledge for the design of applications and services that are used by groups and organizations.

All of these areas are important and interdependent. Independently of the interactive system and the people involved, any task will involve motor control. Further, all people form experiences along the lines outlined later when they interact with systems. We, therefore, reject claims that some of these areas are of priority for understanding people and that some of these areas have faded in importance for HCI. However, specific projects or interactive systems may require particular attention to certain areas. For example, a study of sharing bereavement on social media may require particular attention to the areas of communication (Chapter 9) and experience (Chapter 7). If we are interested in creating a new way of working together in virtual reality, we need to know both perception (Chapter 3) and collaboration (Chapter 8). However, this does not mean that the other areas are irrelevant.

2.2.2. Special application areas

Special application areas are specific areas of activity, such as work or games, that have characteristics that make them unique. These areas are unique in terms of what people do, how they feel, or what they think, to such a large extent that it is invaluable to study them. Research in such areas often involve both general theories and principles, as well as those that are developed especially for the area itself.

Examples of special areas that HCI engages with are continuously changing. Recent examples include personal health, leisure, sustainability, technology in the developing world, creativity, games, and learning. Much can be learned about such areas through user research, following the principles outlined in Part III. However, the scientific literature outside HCI is likely to contain a lot of general information as well. For example, there is a lot of knowledge in the literature about why people play games. This includes theories of motivation in games, particular behaviors in games, and measures of states in games, such as immersion or flow (Chapter 7). In studies of health and well-being, we deploy theories of behavior-change and motivation (Chapter 6). In studies of security and privacy, we consider users' basic needs for personal safety (Chapter 6). In studies of computer-supported cooperative work, we deploy an understanding of how people collaborate (Chapter 8 and communicate (Chapter 9). A proper understanding of such special application areas, regardless of how particular they are, should always take into account general knowledge of HCI.

2.2.3. User groups

Another area of general understanding concerns particular *user groups*. User groups are differentiated by their interests, capabilities, and the systems they use. They may differ in important ways across the seven areas of understanding people. Therefore, the literature within HCI, and in other fields, may describe the characteristics of such user groups. One example of such a user group is older adults. Much is understood about such adults in terms of sensory, motor, and cognitive capabilities, and disabilities. Such understanding can be critical for projects working with older adults. Numerous other particular user groups have also attracted a substantial amount of research, such as design and evaluation with children, users with disabilities, people in low-income countries, and families.

2.2.4. Individual differences

It is almost a cliché that everyone is different. Yet, in addition to looking at how groups of users differ, we need to understand how users' characteristics vary. No matter how homogeneous the sample is, large individual differences are the norm and not an exception [237]. Individual differences can also be pronounced in HCI tasks. In tasks such as text editing and programming, differences between individuals can be of the order of 20:1 [192]. The scale of such differences can be of great practical significance.

However, to go beyond merely stating that differences exist, we need to understand the mechanisms that produce them. Why is age, gender, or education important factors? In this part we will learn about the mechanisms that underpin differences among users.

2.3. Applying our understanding of people

HCI values *actionable* knowledge. However, what are the different ways we can apply our knowledge about people? The general answer to this question is that understanding people *helps solve problems in HCI*. Many research problems in HCI involve explaining and predicting how people use computers, considering what systems to construct, or investigating how an interactive system affects a community or organization. The theories presented in this part help tackle such problems.

Work in HCI has explained how we might use theory. For instance, Bederson and Shneiderman [57] and Rogers [687] separate descriptive, explanatory, predictive, prescriptive, and generative theories in HCI. Let us use those answers to identify some ways of using understandings of people and present some examples of such uses.

First, we may use an understanding of people *to direct what to pay attention to* in a project. Understanding people in general might help direct attention in encounters with particular users. For example, research shows that motivation plays a key role in what people do and what value they derive from their actions (see Chapter 6). Thus, in encounters with users of a particular project, we might want to pay attention to motivation. We might also use such knowledge about people when we reflect on and analyze such encounters.

Second, we may use the understanding of people *to explain empirical findings*. For example, maybe users mention the need to stay in touch during a particular activity. This may be explained by the concepts of coordination work (see Chapter 8). As another example, Steinberger et al. [773] explored how to increase engagement in driving by gamification. They constructed prototypes and could explain their empirical results of driving with a general model of opportunities and costs.

Third, we may use the understanding of people *to make design decisions*. These may be high-level decisions concerning the overall concept of an interactive system. For instance, we may depart from the typical biases of memory (see Chapter 5) to design a photo-based memory tool [339]. As another example, Consolvo et al. [167] used theories of behavior change to build a system that encourages people to be more physically active. Theories may also inform particular design decisions. Oulasvirta et al. [612] used models of how perception is integrated across different channels (see Chapter 3) to discuss the design of buttons.

Fourth, we may use an understanding of people to help *explore a design space*. That way, general understanding, rather than direct contact with users as explored in the part on user research (Part III), serves to generate design ideas. For example, Ballendat et al. [38], used proxemics—a theory of the distances people perceive and how people use physical distance in encounters with other people—to generate design ideas. They create a suite of devices and interaction techniques based on an awareness of nearby people and other devices. In short, the theory of proxemics served an important generative role.

Fifth, we may use an understanding of people to *predict people's behavior*. Predictions may refer to expected events or processes, and they may also involve numerical estimates. For example, researchers have modeled visual search patterns on web pages. Such models allow us to predict the most salient part of a website with some probability, or predict the average time to find a particular link on a web page.

2.4. Is a general understanding of people possible?

We have argued that there are general understandings of people that are useful across topics in HCI. We have argued that such understandings are useful for solving practical problems in HCI, although a consequence of their generality is that they must be augmented with user research.

However, this view is not accepted by all. Some researchers object that a general understanding of people is impossible. The argument is that general understanding is not typical of any particular situation. Therefore, it does not provide much, or any, benefit. Further, Lincoln and Guba [470] argued that generalizations imply a belief in determinism—that if the antecedent of a generalization is present, then the consequent must follow—and some form of reductionism—that phenomena relate only to one or a few generalizations. While we agree that it is difficult to arrive at generalized knowledge, we argue that such understandings exist in many areas and show examples of how they may be used. For example, we know things about the visual system in humans and about collaboration that needs to—and must—play a role in designing and evaluating interactive systems.

Another related belief is that we are all equipped with empathy—the ability to intuitively understand other people and their experiences. Why do we need theories if we have such an ability? Unlike empathy, theories can be communicated to others and subjected to scrutiny. Further, our empathy and intuition often fail us. The inferences we make about other people are often incorrect, especially when they involve people from different backgrounds (see Chapter 5). Scientific concepts are more comprehensive, precise, and—if properly applied—appropriate for describing what happens when a user interacts with a system.

Finally, why bother to understand people in the first place? Why not just create a quick prototype of an interactive system and improve the design later on? Indeed, Petroski [635] described the evolution of everyday artifacts as occurring through trial and error. The pencil, for example, has gone through hundreds of development cycles throughout its history, some related to the manufacturing of pencils, and some to its use. In a discussion about the role of theory in HCI, Landauer [444] argues that we need to "get real" about the possible impact of cognitive psychology—a type of understanding of people—in HCI. The argument is that the real world of users is not captured well or in its entirety by cognitive psychology. Instead, empirical work and formative evaluation are suggested to be sufficient to drive the development of interactive computing systems. That is, rather than using an understanding of people, we rely on empirically testing user interfaces with people. We share this empirical orientation. However, relying exclusively on empirical data is costly and prone to frequent failure.

In summary, we find theories about people essential. Although such theories need to be complemented by empirical research, we think that the HCI field can, and should, draw on general understandings of people. The rest of this part presents such understandings.

Summary

- A deep and scientific engagement with interactive systems and the phenomena that surround them begins with people. We call that approach human-centered.

- This part of the book shows how being human-centered requires us to draw on theories and models of how people feel, think, and behave.

- These understandings of people are general, holding across many individuals, many types of user interfaces, and many use contexts.

3. Perception

Consider the situation in Figure 3.1, where you are walking down a street and look around and have a sense of being lost. You grab your mobile device to search for directions from a navigation application. You unlock the device, launch the application, and press a button to locate yourself on the map. When you read the map, you may have stopped walking. Alternatively, perhaps you kept walking but slowed your pace to avoid bumping into other people. *Perception* is a critical capability that made all this possible. You used visual perception to locate the buttons and guide your finger to press them. If you had approached another person while walking, you may have noticed that through your peripheral vision. Beyond vision, you used your tactile sense to guide touches on the device, audition (hearing) to follow sounds in the background, and vestibular sense to maintain balance.

Perceptual tasks that we may consider simple, like looking for a button on a display, often have plenty going on. A display consists of light-emitting pixels. As you move your eyes, patterns of activations of rod and cone cells occur on the retina, feeding into a perception of regions with colors, shapes, sizes, and orientations. Over many glances (fixations) on the display, we construct a more coherent *percept* of it. This percept has structure: it is not a chaotic galore of colors, but some objects are in relation to each other. For example, certain elements appear being in front of each other or belonging together. These percepts ensure that when we search for something, it happens in an orderly and mostly efficient way. We focus attention not on random elements, but on elements that have some probability of being the ones we look for. This is affected both by the content of the percept itself, and expectations that you have unconsciously learned over previous encounters. After a few glances, you locate the button, read the label, confirm that it is what it is supposed to be, and click it.

This mundane example demonstrates why understanding human perception is fundamental to HCI: Perception is the main means of acquiring information about the state of a computer. A user interface ”communicates” to users through perception. Therefore, the way people sense and perceive is essential knowledge to study and design interaction. The example also demonstrates another important property of perception. Design does not fully determine the way we look at the interface; our previous experience and strategies also play a role. To understand how to design perceptually efficient interactive systems, we need to understand how prior experience, attention, and the designed world work together.

More generally, *perception* is the ability to collect and organize information about the environment through physiological sensory systems. Thus, perception refers not only to the subjective sensory experience of the interface. It also refers to the processes that help us *organize* a representation of the display. Its functioning can be understood via three prime

Figure 3.1.: Perception serves a multitude of roles in human-computer interaction. We regulate our actions in interaction via perception. User interfaces communicate their state via perception. In this example, perception has a decisive role in helping us find elements on the display, guide fingers, maintain awareness of the background, and control gait and walking.

processes (see Figure 3.2). One process is *sensory information* that our sensory systems produce. A sensory modality is a biologically specialized system dedicated to a type of transduction. In sensory transduction, a form of physical stimulation is transformed to neural events. Light, for example, when it hits the back of the retina, triggers a cascade of physiological events through the rod and cone cells, which ultimately contribute to our perceptual experience. Sensory modalities commonly used in HCI include vision, hearing (audition), and touch (tactition). However, there is also experimental research on smelling (olfaction) and tasting (gustation). Moreover, we can rely on a sense of our body, or proprioception. It refers to the knowledge of the position of our limbs in space.

The second process concerns *expectations*. Over years of interacting with the world, we have accumulated prior experience. Such experiences enable the brain to *project* its expectations back to the world. To achieve this, perception utilizes *internal models* to constantly make guesses on how the world *might* be. Internal models allow us to deal with the fact that sensory stimulation is relatively poor. They fill in the blanks.

The third process is *attention*. Attention refers to the ability to focus processing on a select portion of the full perceptual scene. The decision on what to attend to is a strategic one: at any given moment, you could be attending almost anything in your vicinity, but you only attend to one thing at a time. These strategies affect what we perceive and are as important to understand as the other two processes. If you were left with sensory information only, you would have no way to form a coherent organization of a user interface. We need top-down processing to impose organization to sensory data. And we need attention to guide the formation of that organization.

But why is understanding perception important for HCI; is it not something that only psychologists and biologists should care about? As stated in the introduction to this Part, understanding people may be put to many uses, such as explaining empirical findings and designing user interfaces. In the case of perception, we do this by drawing on concepts and models developed in biology, neuroscience, and the cognitive sciences. This may help us do the following in HCI:

- Design display technology. A display is a device that presents computer-controlled patterns of physical stimulation to express information in a computer-controlled way (see Chapter 25). The paper by Denes et al. [183] represents one example of such work. It aims to achieve a high number of frames on a virtual reality headset. They do so by exploiting features of the human visual system to reduce the resolution of every other frame. In that way, they can reduce the data transmitted by about 40% at a limited cost to the perceived image quality.

- Explain why people use computers as they do. For instance, Figure 3.1 suggests that the visual search on a UI is influenced by both expectations and the visual features of the page. This is confirmed in eye-tracking studies showing a distinct pattern in search, based on users' expectations from previous pages on where the most relevant information is located. We can use such patterns to place information in the most salient places [386].

- Evaluate designs. The understanding in this chapter directly underlie guidelines for

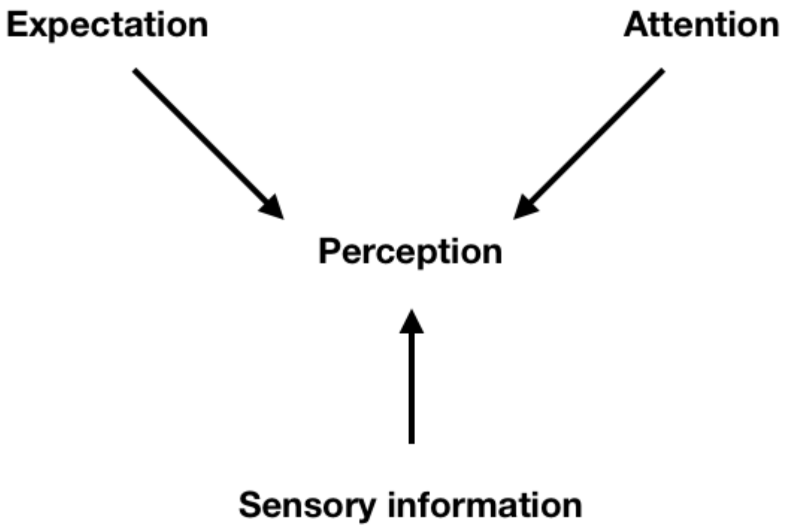


Figure 3.2.: Our perceptual experience of the world may appear like it is veridically (truthfully) reflecting the sensory data we receive. However, it is a representation that our mind actively constructs. On the one hand, perception is affected by our expectations that are drawn from prior experiences. On the other hand, perception is shaped by how to deploy attention to sample information.

effective designs and may be used to evaluate UIs. For instance, the Gestalt laws for visual perception explain how visual stimuli are organized into wholes. This allows us to check if the organization of a page that we intend is indeed the one that users are likely to perceive (see section 3.3).

- Inform the design of visualizations, interaction techniques, and user interfaces. For example, laws of contrast perception and the gestalt laws have been used to algorithmically optimize data visualizations such as scatter plots, enabling people to see structure in complex datasets that would otherwise be difficult to do (see [526]). In interaction techniques (see Chapter 26), we can find optimal ways of providing feedback based on human perception; occasionally, we might even exploit its limitations to make interaction techniques work.

In contrast, it is hard to design a good user interface without understanding the foundations of human perception. Next, we discuss the basic phenomena and principles of perception, along with selected implications for HCI. For more information on the anatomical and physiological aspects of perception, see, for instance, Ware [857].

3.1. Sensory modalities

Sensation is a physiological process that produces information about the environment for perception. Sensation feeds perception. From a biological perspective, sensation is about *transduction*: *a sensory system* transforms energy in one form – say light or physical contact – to electrochemical events in the brain that produce the experience of sensation. This is a complex physiological process. It is important to understand, on the one hand, how sensation is different from perception and, on the other, how the two are interrelated.

Human sensory modalities are served by three basic types of transduction [758]:

Mechanosensitivity Example of mechanosensitivity is kinaesthesia (sense of own movement), touch, hearing, and equilibrium. Here, the energy of physical contact is transduced, for example, through hair cells in the ear.

Chemosensitivity Examples of chemosensitivity include gustation and olfaction. In these examples, chemical properties are transduced, such as, for example, by taste buds on the tongue.

Photosensitivity Examples include the retina. Here, stimulation by photons is transduced.

Another way to look at sensory modalities is based on *what* they sense. *Exteroceptive* modalities sense stimuli outside our body. They include vision, hearing, feeling, smelling, and tasting. *Interoceptive* modalities sense stimuli inside our bodies. They include proprioception and vestibular sense.

These sensory modalities differ greatly with respect to the properties important for interaction. In HCI, we consider the following differences as important:

1. *Information rate*: How much information can be sensed per unit of time? This is tricky to measure, except for visual and auditory perception.
2. *Parallelism*: How much parallel processing of information can occur? For example, vision is highly parallel, and while audition also can do parallel processing, it is less so.
3. *Sensitivity* refers to the minimum intensity of physical stimulation that a receptor needs to exceed its sensation threshold. Vision has high speed and sensitivity. Such properties can be measured and modeled using psychophysics (see below).
4. *Receptive field* is the size of the region that produces an integrated feature. It is a measure of the association between neurons and receptors.
5. *Adaptation*: Tuning of outputs to attenuate non-informative signals. (See 'habituation' below.)

These differences are important to understand when considering which modalities to use in a user interface. Table 3.1 summarizes three main modalities and key properties for HCI. Vision is fast and has high bandwidth thanks to parallel processing, and it can be used to communicate information through visual (e.g., color, shape, size), spatial (e.g., layouts), and lexical features (e.g., words). Audition (hearing) is fast but serial in presentation and can be used to present information via sound (e.g., auditory icons)

Sensory modality	Key characteristics	Design considerations
Vision	Fast, high bandwidth for parallel processing, field of view is about 180 degrees	Visual, spatial, and lexical aspects of graphical displays, like contrast, acuity, use of color, visual primitives, symbols, and text
Hearing	Very fast (about 40ms faster reaction times than vision), serial presentation, 360-degrees detection area	Properties of sound and voice, such as pitch, timbre, melody, and phrasing
Tactition	Fast but limited to areas of physical contact	Properties of haptic stimulation like amplitude and frequency of vibration

Table 3.1.: Most common sensory modalities studied in HCI, with properties important in design.

and voice. Tactition is fast, but requires physical contact. It can be used to convey low-dimensional events and information via amplitude and frequency of vibration and haptic texture characteristics.

Although the three types mentioned in Table 3.1 are the most frequently considered in HCI, the human body also has other sensing principles. They include, for example, thermal sensing and pain, which have been explored in research as alternatives. For instance, Wilson et al. [877] developed thermal icons. Like visual icons, thermal icons have a specific thermal profile associated with a meaning.

3.1.1. Constructing percepts from sensory information

Perception must construct an actionable percept based on sensory datum it receives. However, if the input to perception are fleeting activations of receptors (think: photons stimulating cells on retina), how is it able to construct coherent, organized percepts that help guide our action? How can we perceive a button as a button and not just the numerous sensations that are caused by the involved LED lights? Two processes are important: integration and adaptation.

First, a sensory system consists of receptors – like mechanoreceptors on a finger tip. It also consists of a neural code and brain regions dedicated to *integrating* information over the distributed sensing inputs. Some sensory modalities, like vision, are *topographically projected* in the brain: that is, the topological structure of the receptors is retained in the corresponding receptive fields. In other words, peripheral receptors, for example on the fingertip, are projected to central neurons in such a way that their *neighborhood relationships* are preserved. This pertains to brain regions dedicated to visual, auditory, and somatosensory modalities. They integrate information over a larger sheet of peripheral receptors. *Convolution* is a special type of topographical projection in visual perception, where receptive fields of increasing size form an interconnected hierarchy. In other words, there is one layer of neurons dedicated to precise local information, and another to higher level of integration, and so on. The benefit is that perception has access to *both* larger homogenous regions in the visual field as well as detail in the foveated region. This makes it possible for larger objects to pop out or appear organized. Whereas the visual system is specialized in spatio-temporal information, the auditory system integrates information across frequency and time. We exploit such integrative capabilities in user interface design, for example in graphical interface design when considering how to group graphical elements (see Chapter 28).

Another key property is adaptivity. Environments change and so must percepts. Have you been startled, for example, when an ambient background noise that you did not pay attention to suddenly stops? A sensory system is adapted to maximize information gain within their sensing limits. In *habituation*, a sensing threshold adapts to continued stimulation. For example, have you ever noted that if you rest your fingers on keyboard that you must move them slightly to ”sense them again”? The reason is that there is no novel information to perceive. The opposite of habituation is *strategic adaptation*. Perception shows sensitivity to the statistical structure of the environment. For example, we develop attentional patterns of looking at mobile applications [460]. Early research on web pages found an F-shaped pattern for traces of gaze. However, when similar studies were done for mobile applications, a different pattern emerged: Although we tend to look at the left top corner the most, faces and text are also likely to attract attention. This is a learned adaptive pattern.

3.1.2. Multimodal perception

In most HCI tasks, sensory modalities rarely operate on their own. Everyday interaction *requires* integration of information not only within a modality but across modalities. Even when you are pointing with a mouse, four modalities participate: proprioception (feeling the angles of the joints moving), tactition (feeling the palm of the hand move against the surface), audition (hearing the movement), and vision (seeing the hand move and the mouse cursor move).

The McGurk effect is a great example of multimodal integration [802]. It is a perceptual phenomenon in which vision *alters* speech. If you look at a face that pronounces the syllable ’ga’, you tend to perceive a heard sound ’ba’ as ’da’. Besides specific effects like this, discrepancy between modalities in HCI can distort integration. You may have noticed that in a videoconferencing situation where there is latency between audio and video channels, it is difficult to follow what the other partners say (Figure 3.3).

The McGurk effect is an example of *cross-modal perception*. There are a number of similar cross-modal effects. For example, *the ventriloquist effect* is something that we commonly experience when watching videos with humans speaking [15]. In this effect, we perceive the speech as originating from the characters we see on the display, even if there is a large distance between the audio speakers and the person speaking on the display. One takeaway from studies of this effect is that vision tends to dominate other modalities.

Another example is *pseudo-haptics*: the creation of a tactile sensation when there is

Figure 3.3.: Multimodal integration is needed when attending to a video conference. We not only listen to other speakers but look at their facial gestures and lips when they speak. However, if audio signal is delayed, for example due to network latency, it is difficult to follow what the other partner says.

none [665]. In a pseudohaptic illusion called the hand-displacement illusion, the visual presentation of the user's hand is dynamically displaced, creating a sense of a force field when no such exist. For example, a mouse cursor can be slowed down when moving up a mountain on a map, which gets users to report a sensation of a resisting force field.

It is good to note that the term *multimodal interaction* in HCI has a different meaning. It refers to the study of novel combinations of technical modalities in input and interaction techniques (see Chapter 26). There are many (technical) modalities that can be combined for a given task, for example speech, gesture, touch, and facial expressions. However, as said, from a biological perspective, even ”regular” workstation interaction is multimodal. Even when 'just' using a mouse, we engage our tactile, auditory, visual, and proprioceptive sensory systems.

3.2. Elementary functions of perception in HCI

As an applied field, research on human-computer interaction pursues the *functional* understanding of perception. HCI has been less interested in the neural and physiological processes that underpin perception and more interested in the properties of perception that affect outcomes and performance in interactive tasks. From a functional perspective, perception is about obtaining such *information* that helps complete some task. Thus, the design of a user interface, from a functional perspective, starts with the question: What is *required* of perception for the user to successfully achieve the goal?

Perceptual task is a central concept in functional understanding of perception: Some sensory information is available, and the user must decide or act in a way that reaches some goal or subgoal. Consider the task of finding a link on a web page, keeping a finger on a particular key when playing a game and preparing to respond to an event, or detecting a beep when a new message has arrived. These are all tasks where perception is required, however, in very different ways. In this section, we look at five elementary perceptual tasks:

Discrimination The task of telling whether a difference occurs in sensory stimulation. For example: is this object brighter than this other one?

Detection The task of telling whether an event of interest occurs (or not) in the environment. For example, was there a link or not on a page you scrolled quickly through?

Recognition The task of categorizing a stimulus as something. For example, you are playing a game and see something pass your screen quickly: was it an enemy or a friend?

Estimation The task of estimating a property of an object of event in the environment. For example, how far is an object you see in the VR environment, could you perhaps reach it?

Search The task of localizing an object of interest. For example, where on the keyboard is the key for character '<'?

To understand how these tasks are 'solved' by perception, we need to understand both the 'hard' and 'soft' sides of perception; that is, the hard-wired capabilities of perception that have evolved with us as species, as well as how – via learning during the lifetime of an individual – they are adaptively controlled and shaped. Our perceptual system has specialized capabilities to support us in tasks that are important for our success. These changes occur via two processes: phylogenetically, via evolution, and ontogenically, via learning. Some physiological enablers of perception – like the structure of the retina – change slowly over a lifetime. However, many defining physiological characteristics have remained stable for millennia, as they have evolved for tasks important to our species, such as finding food, spotting predators, and understanding the facial expressions of our companions. From an evolutionary viewpoint, the time that homo sapiens has spent with computers is too negligible to have had an impact on phylogeny.

Learning and adaptation to perception occur as a consequence of experience. Consider, for example, learning where to look when looking for a link. We know that the first places where we look at a screen slowly tune to reflect the statistical distribution. On a web page, we first look in the upper left corner of the page, whereas with a mobile application, we tend to be drawn to faces and large logos [460]. But adaptation can be remarkably fast. For example, when looking for a link, we try to avoid looking at the same place that we already visited. This 'inhibition of return' is important for making visual search efficient.

The rigorous definition of perceptual tasks has been instrumental in advances in understanding perception in HCI. Tasks that can be controlled in experimental settings have enabled the exposing of underpinning phenomena and mathematical modeling of perceptual capabilities. In such models, we see the hard limits posed by the capabilities of our sensory organs, as well as the softer, more malleable limits posed by learning.

3.2.1. Discrimination

In a discrimination task, the user must decide whether a level of stimulation is available or if that level has *changed* from some reference. *Discrimination threshold* defines the minimum level of stimulation required for sensing. For example, healthy young adults can hear sounds between 20 and about 20,000 Hz. With aging or hearing loss, discrimination thresholds change. In addition, thresholds for sensing visual contrasts change. Such individual differences are important to note. Too often we design interfaces with our peers in mind and forget that many (and in some cases most) users are not the same.

Psychophysics refers to the study of psychological (psycho) responses to physical (physics) events. Its core aim is to measure an observer's ability to distinguish a difference in physical stimulation. In computer graphics, psychophysical models are used to optimize rendering, image processing, haptics, audio, and video. For example, common video compression algorithms exploit discrimination thresholds: knowing that a human perceived would not be able to observe, for example, a minute change in color, can be exploited in compressing an image. In HCI, psychophysics laws have gained recognition among design guidelines [465] and in the design of haptic and other displays.

A psychophysics model is a systematic way to describe people's ability to discriminate stimuli. It relates mathematical physical changes to the perceiver's ability to detect those changes. To obtain data for such models, in a psychophysics experiment, a user is presented with two stimuli, for example, two auditory beeps, and must tell if they are different or not. The experimenter knows if a difference exists and how large it is, and uses this information to model the participant's detection capability. By systematically controlling the properties of the baseline stimulus and its difference to the other stimuli, this capability can be described for a range of stimulation. In another common task type, the user would first be presented with a reference stimulus, and would have to *adjust* another source to match it.

A well-known psychophysical model, *Weber's law*, states that equal stimulus ratios produce equal subjective ratios [774]. Weber, a German physician in the 19th century, provided different physical stimulations like weights or intensities of light, and asked people to tell if a difference exists between them or not. He discovered that there is a systematic relationship between the size of the stimulus and the accuracy in telling a difference. Weber's law states that *just noticeable difference* (JND) is proportional to the size of the standard stimulus:

$$JND = kS, \tag{3.1}$$

where S is the size of the standard and k is a constant – a so-called Weber fraction. In practice, k describes the proportion of increase (in the standard stimulus) required before the observer can make a reliable discrimination. JND thresholds have been charted for perceptual events and qualities relevant for user interfaces: visual length and area, visual distance, visual velocity, visual flash rate, and duration [774].

An application in HCI has been *time perception*, with applications to the response time of the system and the progress bars [735]. For example, how the pace of change in a progress bar affects the perception of system speed or the felt duration of loading.

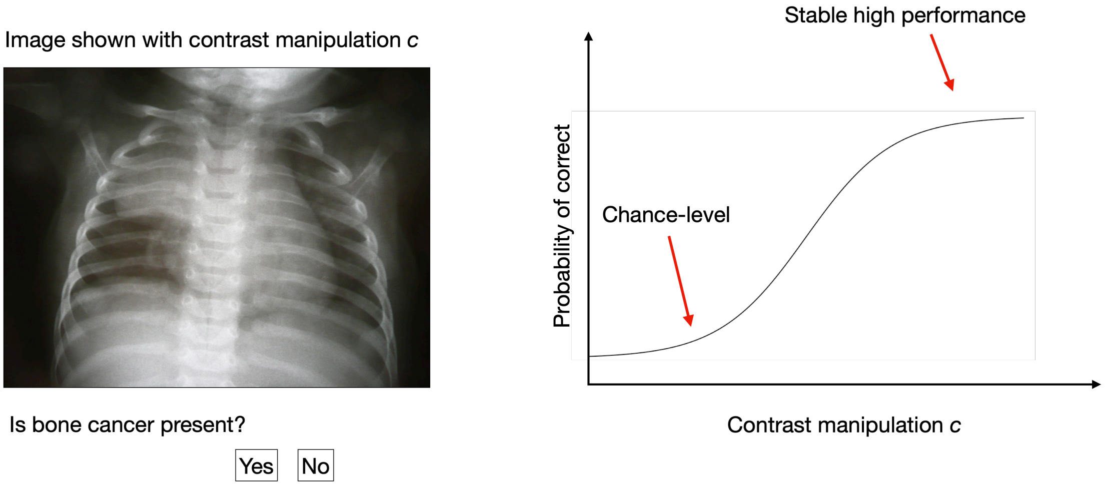


Figure 3.4.: An illustrative psychophysics experiment and a model. Here, an HCI researcher has come up with a contrast manipulation that is expensive to compute, but could help radiologists detect bone cancer from images. In the study, radiologists are presented with images and must press yes / no to indicate whether an event (bone cancer) is present or not. Some proportion (e.g., 10 %) of the stimuli have the event, but the rest do not. Over multiple trials where contrast is manipulated, a psychophysical function can be built that tells the odds of detection as a function of contrast. A psychophysical function relates a physical measure to a measure of perceptual ability. Here it shows that stable detection can only be reached when the contrast algorithm is applied at a level of $0.8c$ or higher.

3.2.2. Detection

In a detection task, a user must decide whether an event has occurred or not. Consider for example playing a game and having to decide if a shadowy dark contour in the distance is another player, or a radiologist examining a computer tomography image and having to decide whether cancer is present or not (Figure 3.4). The study of detection tasks in HCI originates from the study of radar operators in World War II. Radar operators had to attend to grainy black-and-green displays and make decisions on if a blob is an enemy plane. However, a blob could also be a bird or caused by interference in the radar reading. Models of detection can help inform the design of displays that help people detect events better.

Signal detection theory offers a theory and a model to understand detection performance. It assumes that detection is affected by two components: (1) sensitivity and (2) response bias. In a noisy environment, detection performance is limited by the observer's ability to discriminate the signal from noise (sensitivity). In the example of Figure 3.4), the HCI researcher wanted to study whether a contrast enhancement algorithm improves the ability of radiologists to detect bone cancer in images. The quality of the x-ray imaging equipment would affect the ability to detect cancer. Similarly, a novice radiologist, or one with poor eyesight, would be less sensitive to differences. However, people may not respond simply on the basis of what they sense. They may also strategically adjust their responses to bias the outcomes. Such adjustment may reflect the expected values of failing vs. succeeding in the task. For example, radar operation in WW2 was not without risks. The consequences of a poor choice could be devastating: If an enemy went undetected, people could lose lives; If noise was falsely detected as an enemy, time, and money would be lost. In a situation like this, the radar operator may be biased to respond 'yes' (enemy) even when somewhat unsure, showing bias. Let us look at these two components more closely.

Sensitivity of an observer depends on the perceived properties of the signal compared to noise. Sensitivity can be empirically measured in a task where a participant is shown signals and non-signals (noise) and asked to report yes/no accordingly. We tabulate the responses to four classes: true positive, true negative, false positive, and false negative. The sensitivity can be calculated from this table. It is often described using a statistic called d', which describes the difficulty of detecting a signal, in particular how difficult it is to discriminate it from noise. It can be computed from observed hit rate (H) and false alarm rate (FA):

$$d' = z(FA) - z(H) \tag{3.2}$$

Here, $z(FA)$ and $z(H)$ give the right-tail probabilities of the rates on the normal distribution. We cast the rates to the normal distribution to ensure we can compare sensitivities across different types of task.

Intuitively d' represents an index of how far apart two distributions are:

$$d' = \frac{\mu_{SignalPresent} - \mu_{SignalAbsent}}{\sigma} \tag{3.3}$$

Figure 3.5 shows to examples, with low and high d'. d' provides a way to compare

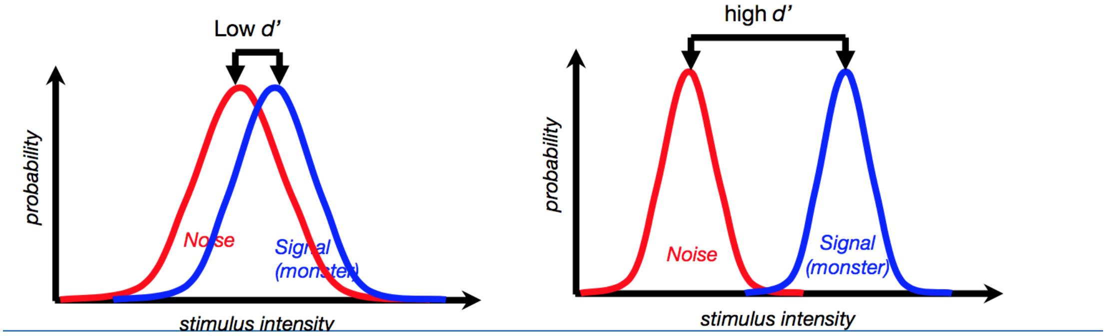


Figure 3.5.: You hear a sound during your sleep: is it a monster or just noise? In a hard detection task, the distributions of the two sounds are similar, they are therefore hard to discriminate and many errors occur. In an easy task, the two sounds are so different that they can be easily discriminated. d' is a statistic that describes how far apart the two events are.

detection performance across an experimental task. The closer d' is to zero, the less discrimination there is possible. When $d' = 0$, performance is pure chance. The higher the d', the more sensitive the observer. When $d' = 4.65$, performance is already very high; for example, the hit rate is 0.99, and the false alarm rate is 0.01.

Response bias denotes a participant's bias (tendency) toward a particular response. For example, if an enemy in a competitive game shoots me because I failed to shoot first, that can be a high cost. However, if I erroneously shoot an important co-player in my team, that can be an even higher cost. In signal detection theory, the response criterion is denoted with β. A rational player would adjust β to favor a more reserved policy, to avoid shooting co-players. High criterion means fewer hits, but fewer false alarms. A low criterion means more hits, but also more false alarms. Response bias can also occur for other reasons. For example, if the 'yes' button is easier to reach, a response bias can emerge even if it is not in favor of the player.

3.2.3. Recognition

In a recognition task, a stimulus must be classified as one out of a number of classes. For example, icons, words, faces, alarm sounds, etc., need to be recognized as a *particular* member of a set. "This is the icon of Excel." All other things being equal, the larger the set, the harder the recognition. If you have 12 icons versus 100 icons on a display matters. Discrimination becomes harder as more candidates compete.

People are surprisingly good at recognizing faces and objects they have seen before. In controlled studies, they can accurately recognize hundreds of objects presented to them previously [600]. Performance in a recognition task can be measured by precision and recall. *Precision* is the number of true positives divided by the sum of true and false positives. *Recall* is the number of true positives divided by the sum of true positives and false negatives[1].

Human ability in visual recognition is leveraged in the direct manipulation paradigm of graphical user interfaces (see Chapter 28). It is important for training better machine learning algorithms. Humans are often asked to classify data that are used to train algorithms. However, people do not always agree on what the ’true’ classes are. Categories that show disagreement among human observers are problematic for machine learning, because they add noise to the training data. HCI researchers can help build user interfaces that improve recognition accuracy and help annotators solve conflicts effectively.

3.2.4. Estimation

Estimation is the task of assessing a property of an object from stimulus information. Users may be estimating, for example, the size of a 3D object in a VR environment or the (true) color of a product on an e-commerce site.

According to the *cue integration theory*, our prior experiences affect such estimations. Over time, users become sensitive to how likely different properties are [227]. The theory makes the assumption that people consider all information available to them optimally. For example, when trying to respond to a serve in a virtual tennis game, we need to estimate the ball’s speed. To this end, we have two types of cues in our disposal: (1) the sound we hear when the ball was hit by the opponent and (2) the motion of the ball as we see it move on display. The cues add information that is not contained in the other one.

The theory suggests that we integrate such cue-specific percepts into one estimation of the ball’s velocity. The idea of optimality is that we make such estimate under which the observed data (the two cues) are most probable. In new tasks, where users have little exposure to the cues, they would struggle to make a good estimate. However, with more time, their performance is better and better described by the theory. Despite the optimistic assumption of optimality, the theory has turned out to be accurate in fast-paced perceptual tasks in HCI, for example in tasks where a moving target need to be selected [456].

3.2.5. Search

In search tasks, the location-in-space of some object must be determined. In auditory search, the task is to localize the source of a given sound. We talk about visual search as a case below.

3.3. Visual perception and attention

Visual perception refers to perception through sensing of stimulation by light. It is a sensory modality of prime importance in human-computer interaction. Vision offers a rich and efficient way of conveying information to a user. First, visual perception is fast. Unlike olfaction, visual sensory experience is fast, formed within milliseconds from the

---

[1] Not to be confused with memory recall, see Chapter 5.

onset of the stimulating event. However, constructing a more comprehensive percept, which requires several eye movements, can take seconds when the display is complex.

Second, it offers unparalleled parallel processing. It benefits from massively parallel processing over the entire field of perception. Unlike the sense of touch, no physical contact is required with the perceived objects. The human visual system is proposed to involve three neuroanatomically distinct pathways [644]:

'What': The ventral pathway, which encodes the identity of visual objects, such as those of tools, faces, or animals. In the case of graphical user interfaces, it recognizes the types of elements and objects on the displays.

'Where' and 'How': The dorsal pathway encodes the locations of visual objects and actions related to these objects. In the case of graphical user interfaces, it registers which elements can be clicked and tracks moving objects.

"Who': More recently, a third pathway has been discovered. The superior temporal sulcus is proposed which specializes in dynamic social processing. It processes the actions of moving objects and bodies, such as their expressions, gaze, intentions, and moods. In the case of graphical user interfaces, they are deployed in situations such as virtual reality, games, and videoconferencing.

Third, the visual system offers hierarchical and highly interconnected structure. *Visual primitives* – like shape, size, orientation, color, and motion of objects – are extracted in massively parallel processing in lower-level areas. The low-level areas feed into higher-level areas that recognize objects, and encode their identity. These, in turn, project back to the lower-level areas. This structure realizes the top-down and bottom-up processing discussed in the previous section.

In the following, we discuss known limits of the visual system, and then cover basics of eye movements, perceptual organization, and visual attention.

3.3.1. Limits to the human visual system

Take a look at Figure 3.6. The photo shows an extreme case of environments encountered when using computers. Even if the face of the smartwatch may look fine when printed on the pages of this book, this real-world experience would be challenging. Before reading on, think about the following: What is it exactly that makes the case challenging for the user?

Ware [857] summarized key limits to the visual system under the concept of *Windows of Visibility*. A 'window' is a metaphor to describe physiological limits posed to visual perception. We here describe four windows important for HCI: (1) visible spectrum of light, (2) field of view, (3) contrast, (4) foveated vision. First, we can only perceive a limited range in the spectrum of light, from about 380 to 780 nm. The rod and cone cells we have permit only three chroma of color (trichromaticity). This sets limits to the range of light that displays need to cover.

Figure 3.6.: Ware's Windows of Visibility can be used to understand how viewing conditions affect perceptual tasks in the wild. For example, consider wanting to glance at the smartwatch while outdoors. Using the Windows of Visibility, assess the situation: what are the key limits to the user's visual performance? Photo used under Creative Commons Zero.

Second, our field of view is limited to about 190 degrees horizontally and 125 degrees vertically. This sets limits to how large displays can be and how where they must be located in relation to our eyes.

Third, our perception of detail is limited. *Contrast* refers to difference in luminance and color that make something in the field of view to stand out from the rest. *Contrast sensitivity* refers to our ability to distinguish levels of contrast. The contrast sensitivity function (CSF) is a standard way of expressing this as a function of spatial cycles per degree. The function shows thresholds for contrast perception in a typical viewer. At first, contrast sensitivity is low because the information density is low. The contrast sensitivity is highest in the middle when cycles-per-degree matches the properties of the human visual system. As cycles-per-degree increase, sensitivity decreases again, as the system is not able to discriminate the cycles.

Fourth, the retinal image is limited and very non-uniform. It loses accuracy at the periphery. Vision is physiologically divided into two regions:

Foveal vision refers to a narrow but precise, high-fidelity perception around the point where you fixation. This area spans only 2 degrees of angle (roughly corresponding to the size of your thumb held out at arms length).

Peripheral vision refers to the rest of the field of view, which has eccentrically decreasing fidelity. The farther away from the foveal area, the lower the fidelity.

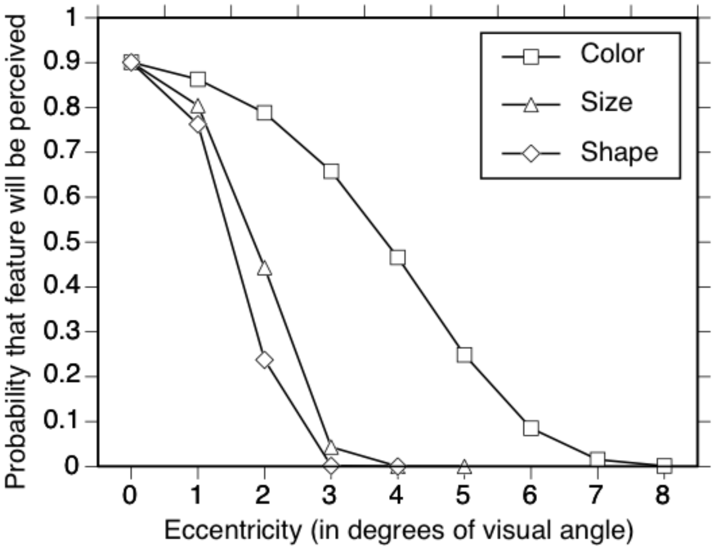


Figure 3.7.: Peripheral drop-off functions: The further away a feature on display is from the foveal region (area of high acuity), the less probable it is that is available for perception [406]. Note that in different perceptual tasks, different drop-off curves can be observed.

In the peripheral region, vision retains some access to visual primitives, but not so much to objects. The visual primitives of human visual perception are color, shape, size, and depth. *Drop-off functions* describe how information is lost from foveal to the peripheral region. An example is given in Figure 3.7.

In the case of the smartwatch (Figure 3.6), which Windows are relevant? At least three of them. First, the watch may reside outside of the field of view, depending on the posture of the user. For viewing, it must be brought closer to where the eyes are operating. Second, even then it is likely to be foveally viewed, because the eyes need to attend the shoelaces. To attract the user's attention, the smartwatch would need to use highly salient visuals (e.g., salient colors, motion). Third, given that watch face is small and far away, all text is beyond human contrast sensitivity. Typically users bring the watch closer to the eyes for reading.

Visual impairments

Visual impairments that limit a person's ability to view computer displays are surprisingly prevalent. Visual impairments include a wide range of physiological and neurological problems that cause color blindness, low-level vision, blindness, and deficiencies controlling eye movements.

Color blindness, or color deficiency, manifests itself in difficulties in perceiving differences between colors, the brightness of colors, or different shades of colors. People with color blindness are often unable to distinguish certain colors, most commonly green–red or blue–yellow. Humans perceive color using cone cells in the retina that either detect red, green or blue light. The inputs of these cone cells are then integrated in the brain to give the impression of a particular color. When cone cells do not function as expected, or are absent, this gives rise to color blindness. Color blindness covers a spectrum from mild color blindness that may be affected by, for example, dim lighting, to severe color blindness resulting in perceiving the world in shades of gray only. Color blindness is usually stable and affects both eyes. There is no treatment for people born with color blindness. There are also various visual aids, such as apps, that allow a user to take a photo and point anywhere on the photo and the app will explain the color at that point to the user. More generally, the effects of color blindness can be mitigated in GUI design by careful chose of how information is encoded using color. In Chapter 25 we talk about HCI research automating color decisions to help users suffering from deficiencies with color perception.

3.3.2. Eye movements

The human oculomotor system controls eye movements. It offers three main modes of controlling where we look:

Fixations Fixations encode information about the visual scene consisting of multiple micro-fixations, each a few tens of milliseconds. What eye trackers do is that they cluster micro-fixations into fixations of 200–400 milliseconds. The way we pick information during a fixation is task-dependent: when reading text vs. searching for an icon, we pick different visual information (e.g., color, text, shapes, orientations, motion).

Saccades Saccades move the gaze point in ballistic leaps that are not perceived during the scene. They are ballistic in the sense that the target of the saccade is not changed after the onset of the movement. They are also 'blind': No information is sampled during a saccade.

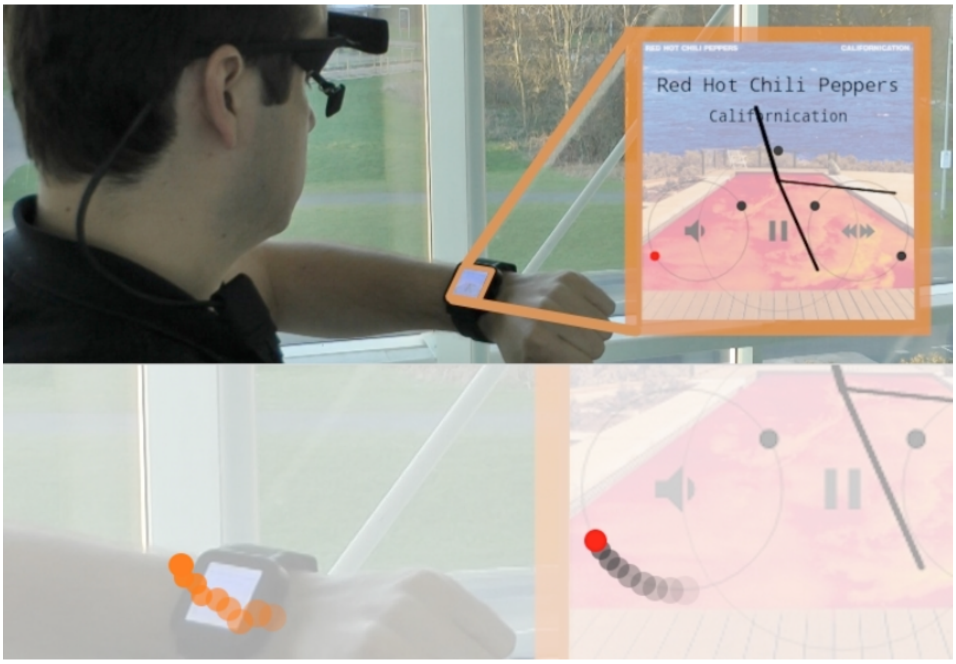
Smooth pursuit Smooth following of moving targets, such as when following an animated character moving on the display. No saccading occurs. An example of an innovative application in HCI is shown in the side box.

The speed and accuracy of eye movements can be modeled mathematically (see the side box below).

Fixations are the main means for the perceptual system to sample the visual environment. Every microfixation can gain progressively more information. As described above, this amount decreases *eccentrically*; that is, the further away an object is from the foveal region, the less information we gain about it in a fixation.

This property explains how eyes move when reading text. When reading, the visual system gains information about the current fixated word and the following word [155]. The control of eyes is affected by lexical processing. Lexical processing is affected by three properties: the frequency, length, and contextual predictability of the word. For example, if a phrase is hard to understand, we may need to fixate on words fixated already. When a phrase is easy to grasp or familiar, we may not need to look at every word but we can skip them.

Paper Example 3.3.1 : Exploiting smooth pursuit for input



Orbits is a selection technique that exploits smooth pursuit as an input modality [228]. Gaze-based selection normally uses dwell time to detect the object the user wants to select. For example, if one looks for one second at an icon, the icon is selected. The idea of orbits is matching. The user needs to move the eyes to match the motion or trajectory indicated by a to-be-selected widget. For example, user can raise the volume of a player on the smartwatch by following the trajectory of a moving circle.

Paper Example 3.3.2 : Speed and accuracy of discrete eye movements when viewing displays

Mathematical models of eye movements are important tools in HCI. They can be used to predict the time costs of using complex displays. Such costs depend on a complex way on the layout (distances and sizes of targets) and the expertise of the user. Salvucci [704] proposed a model called EMMA (Model of Eye Movements and Visual Attention) to more predict the performance of eye movements in HCI. It consists of equations for predicting time spent encoding fixated objects as well as for the movement time and accuracy of saccades.

First, when fixating an object, the time to encode it, T_e, is a function of amount of experience with the object. Objects we have encountered more often take less time and vice versa. This is captured as follows:

$$T_e = K \cdot [-\log(f)] \cdot e^{k \cdot \epsilon}, \tag{3.4}$$

where K and k are constants, and f is the frequency of the object. ϵ is the eccentricity – measured as the distance of the target from the current eye fixation (in degrees). When eccentricity is high, to account for the cost of eccentricity, the visual system may initiate a saccade to get closer to the target. According to EMMA, an object (e.g., word) can be skipped if sufficient information has been encoded during a fixation.

The duration of a saccade, T_s, on the other hand, is dependent on the distance to the object D, measured in degrees:

$$T_s = t_{prep} + t_{exec} + D \cdot t_{sacc}, \tag{3.5}$$

where t_{prep}, t_{exec}, and t_{sacc} are constants related to the human visual system related to movement preparation and movement execution.

An important property of eye movements is that they are noisy and therefore inaccurate. The actual landing point differs from the intended landing opint. This error can be modeled as signal-dependent noise. Inaccuracy a normal distribution with a standard deviation of σ_V times the distance D. That is, the greater the distance, the greater the noise [704].

When applying the model, the constants need to be calibrated to the particular viewing conditions. Previous applications provide indications for them. Note that the model does not predict the duration of fixations but encoding time. Encoding can happen over multiple fixations.

Another feature of attention is that we do not always need to move the gaze to encode information. When the encoding time is smaller than the movement preparation time, the target can be encoded without moving the eyes. These eye movements are called *covert* movements, rather than *overt* movements. Covert movements explain many of the variations we see in the duration of fixation.

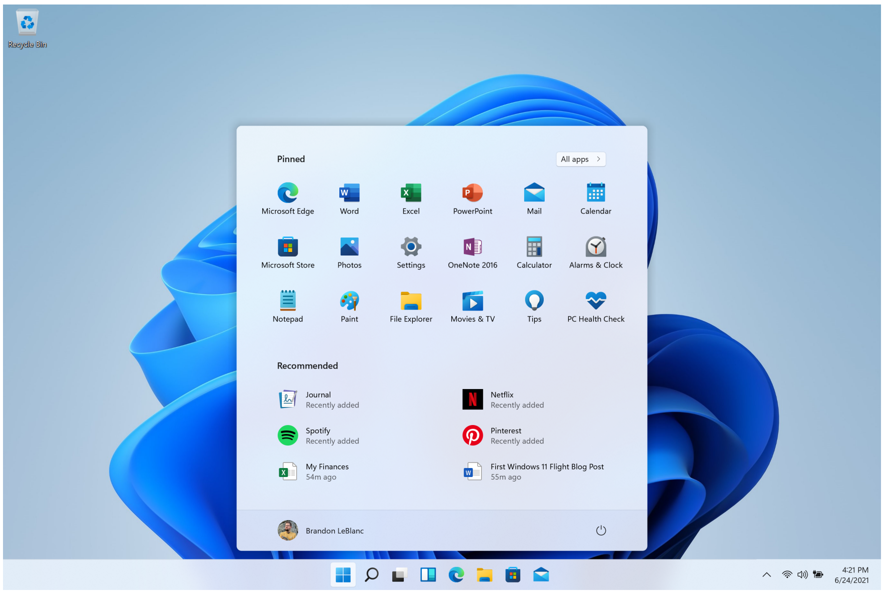


Figure 3.8.: Perceptual organization refers to (1) the division of elements into figure vs. ground and (2) their grouping into coherent regions. In this example, the application window (figure) is clearly in front of the desktop wallpaper (ground). There are also many visual groups on the display. For example, the region of recommended apps forms a distinct region from the rest of what is presented in the window. Similarly the icons on the horizontal task bar form a visual group.

3.3.3. Perceptual organization

When you look at a graphical display, it nominally consists of just pixels on a two-dimensional array. How is perception able to make sense of it? Consider the example in Figure 3.8. It appears organized into regions defined by the desktop, the menu bar, and the window. Some regions, like the window, are standing in front of other regions, like the wallpaper. Some elements belong together and form regions. For example, the list of recommended apps consists of icons that we tend to perceive as a group that is separate from the other application icons.

Figure/ground perception refers to the organization of visual experience in a visuo-spatial hierarchy. That is, some objects belong to figure, or an object in front, and others to objects in the background, or the ground. Graphical interfaces exploit figure/ground perception to show a *display hierarchy* (or screen hierarchy). It defines to what element or whole another element belongs to: this belongs to that, and that is different from this. We use cues such as shadows, occlusion (an object covers another), and size to parse the display. When binocular stereo displays are used, also stereoscopic depth cues can also help in figure/ground perception.

The perception organization decomposes a display into *regions*. Look at the graphical user interface in Figure 3.8: what you most likely experience is a layout consisting of a few main regions, which in turn consist of elements with text or images. But how do we achieve this? If you think about it, there are actually many ways to parse a layout into regions. Two elements A and B are associated by similarity in color, distance, and alignment. However, also B and C are associated by the same qualities but in different degree. How is it determined that two elements belong to a group with each other and not to some other group?

Visual grouping refer to tendencies of elements presented on a display to form visual groups. *Gestalt laws* originate from a German branch of psychology from the early twentieth century. The goal of Gestalt psychology was to understand what constitutes ”a whole”. Wertheimer studied visual factors that cause the perception of elements as belonging together on a complex display. He identified seven laws now called Gestalt laws. They have endured well [845]. In the following, we present four commonly used principles of visual grouping with an example shown in Figure 3.9:

Proximity The closer some elements are together and the farther apart they are from others, the stronger they are grouped together.

Common area Elements that are located in the same closed region are grouped together.

Similarity Elements that are similar in e.g. color, size or orientation are grouped together.

Continuation Elements that are connected by continuation of flow are grouped together.

Brumby and Zhuang [104] found that when menu groups were marked using visual grouping cues, items were found quicker. However, this strategy only worked when there were fewer, larger groups. Having multiple small groups dilutes the facilitatory effect of visual grouping.

Although Gestalt laws were called *laws*, they are less like scientific laws in, say, physics and more aptly thought of as heuristics to understand certain tendencies in how perception works. The laws are popular among designers because they offer a good estimate of how users perceive structure in graphical interfaces and layouts. They help us to understand how to present the hierarchy of a user interface in a way that users find easier to match. They are also part of design education and are actively exploited by interaction designers. However, they ignore the fact that perception is an active and adaptive process: Depending on the task and our prior beliefs, users can organize the display in different ways. We return to discuss active perception at the end of the chapter.

3.3.4. Visual Attention

Attention means the focussing of perceptual processing on a region or object in the perceptual field. *Visual attention* is traditionally considered to consist of three processes:

Selective attention The ability to shift attention to a desired object or location

Vigilance The ability to sustain attention on something for a longer period

Divided attention The ability to share attention between one or more objects, locations, or tasks.

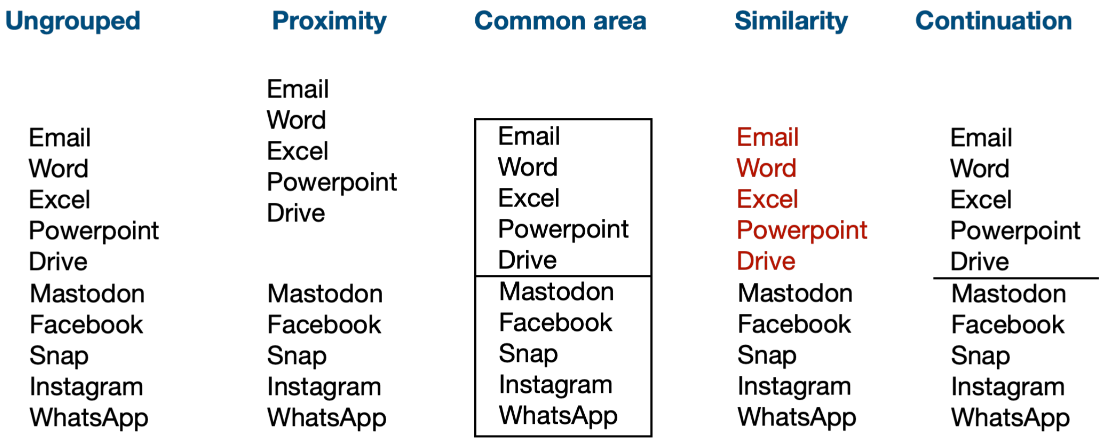


Figure 3.9.: Principles of visual grouping. On the left, an ungrouped menu design. The other images show how the principles can be utilized to visually communicate which menu items belong together.

Selective attention refers to the deliberate refocussing of attention to another object or region in the perceptual field. *Overt attention* refers to movement of eyes to fixate on something. However, we can also sample information without moving the eyes. *Covert attention* is processing of extra-foveal information without moving the eyes. When reading displays, both cover and overt attention occur. Eye trackers are measurement instruments for overt attention. The limit of how long we can hold attention on a task or object is called *vigilance*.

Divided attention refers to the division of perceptual processing to multiple objects or regions. For example, *situation awareness* is the ability to keep track of events and objects in a dynamically moving scene. Consider driving a car through a busy intersection, or the task of an air traffic controller in keeping track of planes in the airspace of an airport. These require us to continuously shift attention between multiple objects and update mental representations on their properties. For example, when walking down a busy street, visual attention to a mobile device fragments. Average durations of glances can be just 3-4 seconds [609]. However, if you walk down a less crowded street, you can maintain your gaze on the device for longer periods.

3.3.5. Visual search

Visual search is a common task with graphical interfaces involving selective attention. The goal is to find an element on the display, for example an icon or a label of interest. This task poses a control problem the visual system: it must decide where to place the next fixation. How does it do this? Three things interact to affect how eyes are guided when searching: (1) visual features of the display; (2) learning; (3) strategic decisions on *how* to search.

First, the display consists of a spatial distribution of visual features. Compare for example an application menu and a mobile app in terms of how colors, sizes, and shapes are distributed. Depending on where you are currently looking at (current fixation), these features are differentially available to the visual system. If you look at the top right corner of a mobile app screen, different parts of the visual feature distribution are available than if you were looking at the center.

In *feature (or disjunctive) search*, visual search is based on so-called visual primitives like color, shape, size, curvature, orientation, or motion. When the target is unique in any of these features, search is fast (typically < 100 ms). For example, if you are looking for a green icon from a display that does not have other green elements, you can spot it almost instantaneously assuming that it is large enough. Feature search is fast, because such features can be spotted peripherally by the perceptual system. On the other hand, in *conjunctive search*, the target cannot be singled out with a single feature: We need to look for a combination of features. For example, there are multiple green elements and the only way to know which one is the icon you are looking for is to look at the icon and its label. In such cases, search cannot be 'solved' with peripheral processing: It requires fixating on candidate items to pick up the relevant information. This search is much slower. In the design of graphical user interfaces, elements that should be quickly found can be designed to be found via disjunctive search (Chapter 28).

The *Feature Integration Theory* of Treisman and Gelade [809] explains why this difference emerges (Figure 3.10). Low-level visual features of a stimulus environment are directly represented in what she calls *Feature Maps*. Attentional spotlight, or information collected via fixations, on the other hand, is needed to find *combinations* of these features. This combined map integrates lower-level features; it maps objects (with their features) to their locations. In other words, selective attention is needed to bind features to recognizable objects that can be further processed.

What this model does not account for is the fact that there is less information in the peripheral view than in the foveal view. Kieras and Hornof [406] proposed a model called *Active Vision*. It emphasizes that people choose where to look based on the description of the target and the visual features available via peripheral vision. However, peripheral information is compromised: the farther away eccentrically an object is from the current fixation point, the more it is compromised. The visual system has uncertainty about the regions of the display it is *not* attending. Yet, it must 'gamble' and look at there anyway. However, according to the theory, it does that only after exploiting candidate items close to the foveal region. The model is dubbed 'active' because users are not passively drawn to visual features, but their choice of next fixation points depends on what they are looking for and what they presently see.

However, what these models do not account for is the effect of *learning*. If you have seen an item for a few times, it is easier to locate it. In menu systems, for example, a user who has experienced the design a few hundred times, can fixate it virtually immediately when looking for it [127]. The *Guided Search* model of Wolfe [891] attempts to explain this (see Figure 3.11). It proposes that decisions where to fixate are guided by *priority maps*. The priority map combines bottom-up and top-down information to guide where to look at next. Depending on how strong the long-term memory is, it differentially contributes to the map and therefore to guidance. Users who have seen a UI a few times, have positional memory of where objects on it are located, especially those that are often

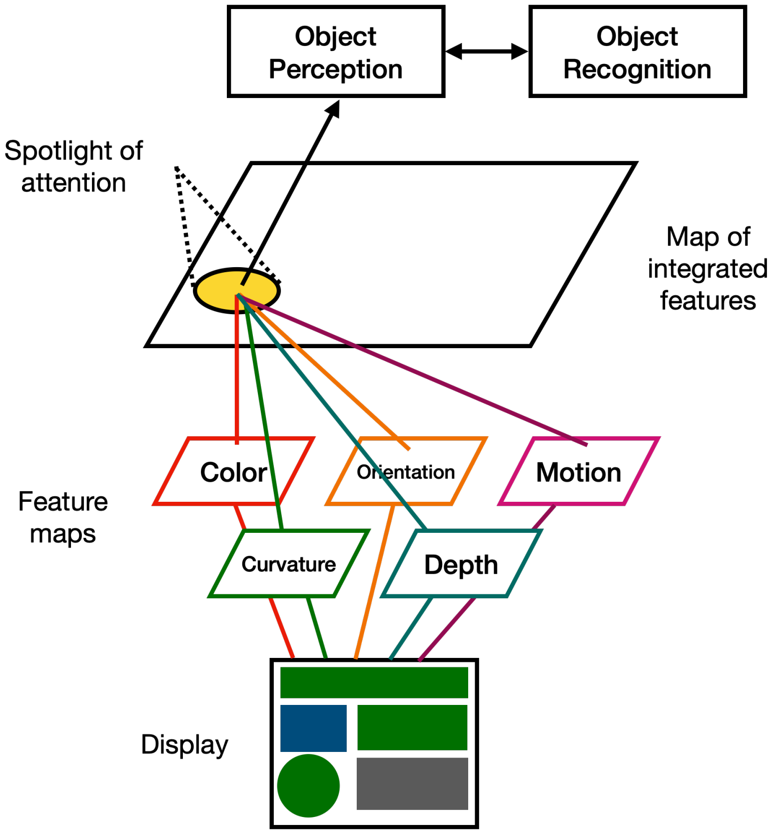


Figure 3.10.: Feature integration theory explains why conjunctive search is slower than feature search. Feature search can be carried out using parallel processing of the peripheral vision, whereas conjunctive search requires users to fixate on the region. User interface designers can exploit these differences.

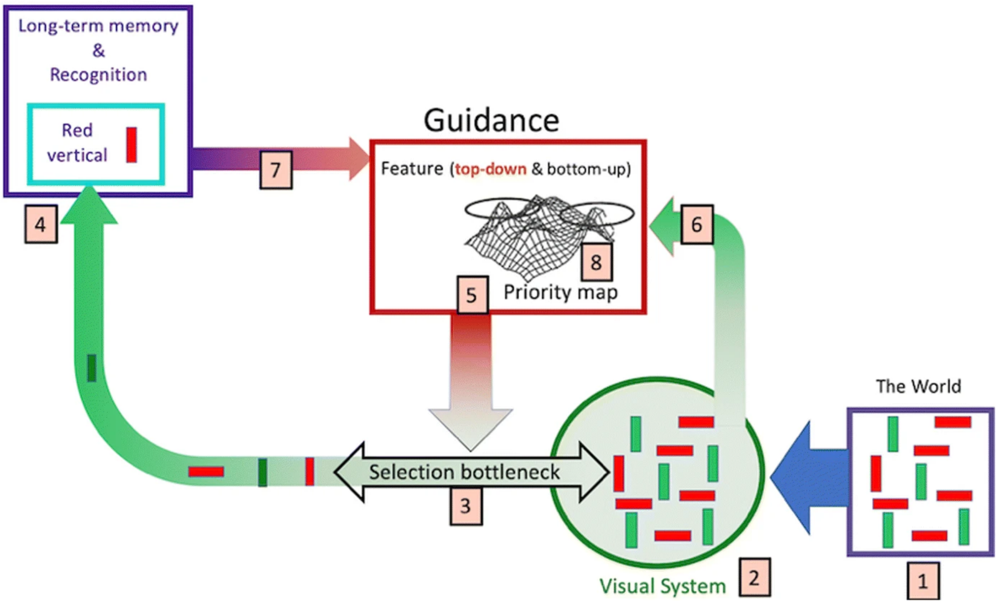
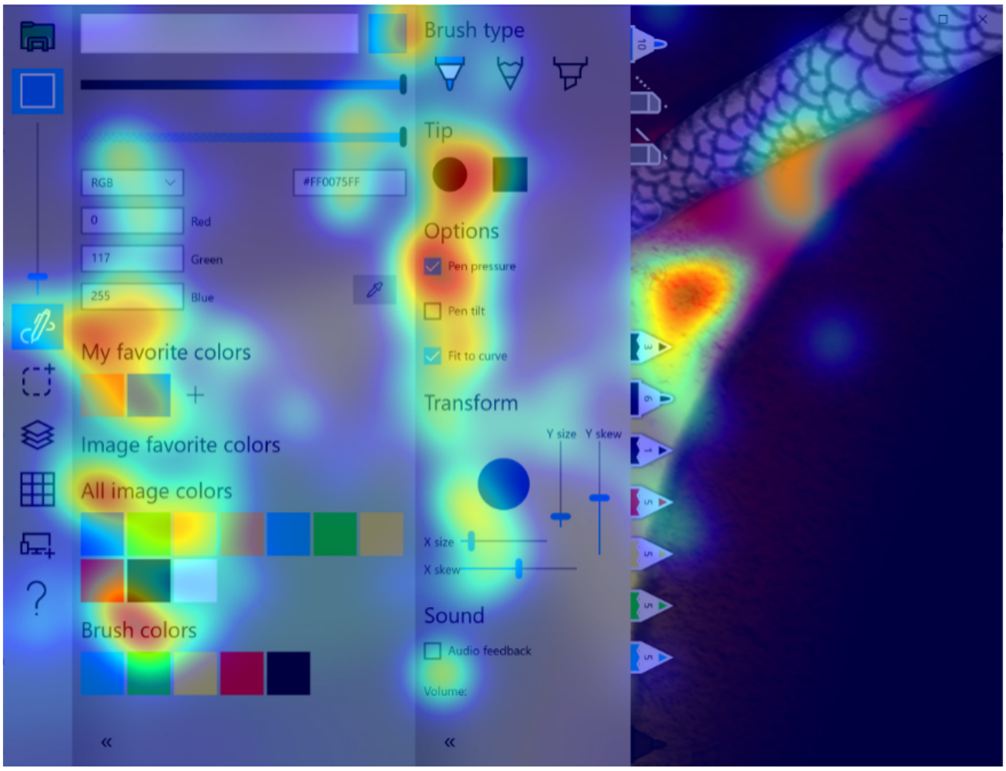


Figure 3.11.: A model of visual search by Wolfe [891]. It suggests that both long-term memory of locations (top-down) and perceived features (bottom-up) contribute to the selection of where to place a fixation. This takes place via what is called *the priority map* where these information are combined. This mode (Guided Search 2.0) has been updated to considers also scene composition, lexical and semantic features, as well as task rewards.



RGB
#FF0075FF
0 Red
117 Green
255 Blue
☑ Pen pressure
☐ Pen tilt
☑ Fit to curve
☐ Audio feedback

Figure 3.12.: This heatmap shows example eye tracking data on what people look at when seeing a UI [378]. Visual saliency denotes the probability with which visual features attract attention when we see a UI for the first time. Saliency depends on the distribution of visual features on the UI, expectations (prior experience to related designs), as well as attentional strategies.

searched. With more repetitions, the locations of elements will be remembered and do not need to be attended actively. However, if the display changes, the value of long-term memory is lost, and the user needs to rely again on bottom-up guidance. Wolfe's model has been updated a few times. The version shown in the figure is 2.0. Since then, the later models also cover the effect of task rewards (see Chapter 21), as well as scene syntax and semantics.

Users often explore displays to form an actionable priority map. For example, when seeing a new ticket vending machine, should one try to directly find the entry point to the display (exploit) or first familiarize with the overall layout of the system (explore)? We often explore briefly and search then. The controller must in that case decide whether *enough* is known about the scene to find the target of interest, or if more information needs to be collected.

3.3.6. Visual saliency

Visual salience is an application of visual attention modelling that has become popular recently. Saliency refers to the probability with which a graphical element can attract visual attention during the first seconds of viewing a display (see Figure 3.12). Salience depends on the visual properties of the target and those of the rest of the display. It also depends on tendencies that are learned over repeated exposure.

Attention, generally, is drawn to **visually unique elements**. Through color and bold font, the words stand out relative to the rest of this paragraph. More generally, regions and elements that are unique in terms of visual primitives, such as color, shape, size, orientation, or motion – stand out.

Saliency emerges in parallel processing of retinal input at lower levels in the visual cortex, on the one hand. On the other hand, visual saliency is also affected by top-down factors such as memory and expectations of informative locations. We learn to look. For instance, in an image full of green tones and green-filled shapes, if a color such as red appears, observers tend to look at the red shape. Top-down factors include task goals and expectations based on the learned statistical distribution of features. For example, in many natural scenes that show a horizon, most of the information lies close to the horizontal medial line, which also attracts attention. When the visual task or content changes, both bottom-up and top-down factors may change. Therefore, an empirical effect related to saliency reported for one context does not carry over trivially to another.

Research has found that there is something special about user interfaces. They differ from natural scenes (think: looking at a forest) in several respects. In natural scenes, there is a tendency to look at the center and the horizontal line. This is understandable, as the horizontal line often has most information about a natural scene. However, for user interfaces, this is often not the case [460]. Attention tends to be drawn to the top-left quadrant of the page. In addition, color is a poorer predictor of saliency than in natural scenes. Instead, attention is drawn to large texts and images with faces. However, despite visual saliency having emerged as one of the better-known aspects of the human visual system, designers often use rules of thumb and gut feeling instead of theory or rely on experimental evaluations.

Paper Example 3.3.3 : What makes a display cluttered?

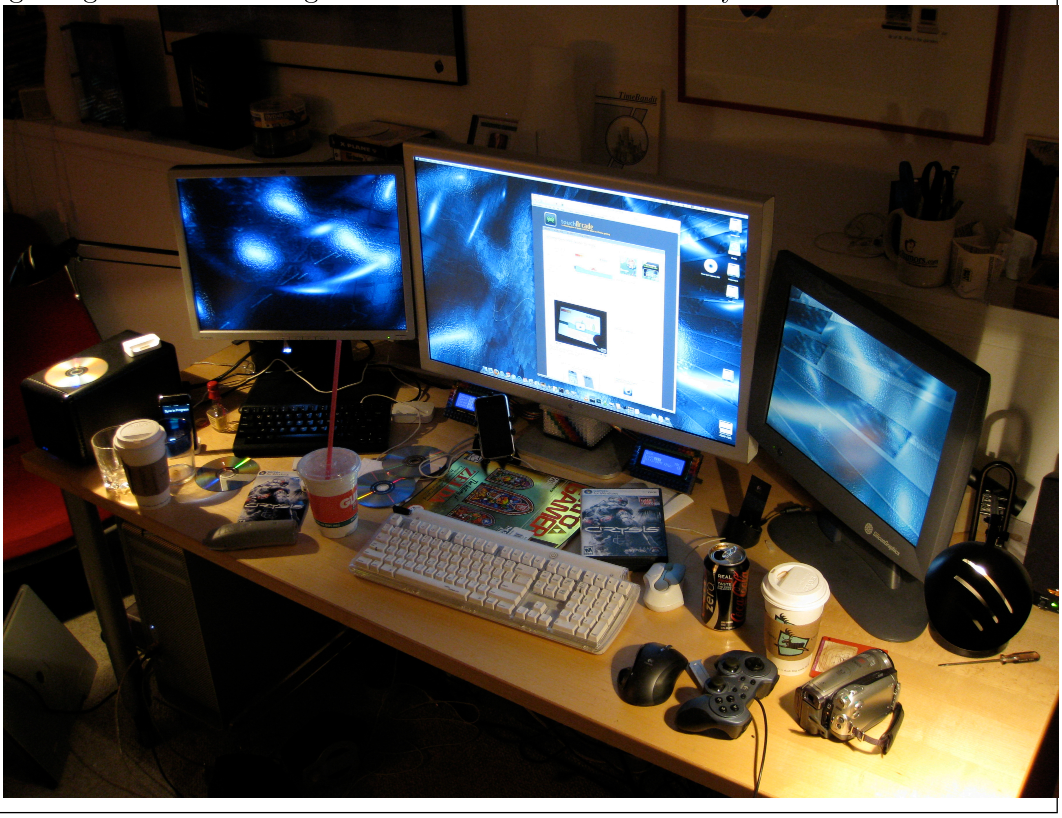
Take a look at the example of a desk in this side box (picture used by Creative Commons license). Anyone can see that it is cluttered. But what does clutter mean?

Consider visual clutter ”the evil twin” of saliency. Rosenholtz et al. [692] notes how visual saliency can be used to explain and measure how cluttered something is perceived. *Clutter* means that *everything* attempts to be salient.

The underpinning hypothesis is that the human visual system has evolved to detect unusual objects in scenes. A salient item is unusual given the visual context. Clutter, by contrast, is the state in which excess items compete for attention. In the state of clutter, all visual features are ”congested”. For example, color clutter means that many colors from the color spectrum are used in the display. Many elements are unique in this and therefore nothing is salient.

Rosenholtz proposed a simple test to understand if a display is cluttered. If you wanted to place a note on the desk (e.g.,. a post-it note) that the user should notice, how would the design be? Which color, size, or shape would you choose? If the desk is cluttered, you cannot pick a visual feature that would make the note stand out on the display.

Clutter is detrimental to visual performance. It is empirically associated with a decrease in object recognition performance and visual search performance, as well as forgetting due to exceeding the limits of short-term memory.



3.4. Perception is an active process

One of the main takeaways from this chapter is that users are not at the mercy of their senses. They are not passive recipients of stimulation determined solely by the display design. Instead, they actively construct percepts.

Users use existing knowledge to actively decide where to attend; guessing where things are or what they are called. In addition to learning, users also adapt strategically how they perceive. Experienced users, for example, rely more on long-term memory than novices. Also the reward/cost structure of the task affects perception. When task rewards change, for example to penalize mistakes and errors, however, users may change their strategy.

Users also actively explore their environments using their bodies. They reorganize the relationship their body has with their task environment. For example: Were you for some reason abruptly re-positioned one meter further away from this text, you would want to step closer to continue reading. In short, people use their bodies to help perceive. This view is called *active perception*. Active perception refers to the characteristic of perception as actively sampling, exploring, and learning in order to better support the organism. *Ecological perception* refers to the idea that these characteristics are sensitive to and adapt to the structure of the environment. For example, we learn to look at web pages in a particular way, expecting items to appear in a certain location that reflects the probability of that item being there. *Embodied perception* is a more radical position that states that perception and bodies cannot be decoupled: All perception is neurobiologically coupled to motor systems. No matter what the view, it is clear that understanding embodiment is as important for understanding perception than is the understanding of its underpinning sensory modalities.

A prominent example of active perception is *perceived affordance*, a term coined by Norman in the book *The Design of Everyday Things*. Affordance is a property of an element that invites interaction. For example, a door handle may afford an opening mechanism that either invites the user to push a plate on the door to open it or pull a handle on the door to open. Users would, in general, know which method is suitable for the door without trying by the affordance of the door handle. Similarly, graphical objects can also afford interaction, for example, by signalling whether they should be clicked, dragged, etc. Norman argues that, however, that the useful property for a graphical user interface is not affordance but perceived affordance: “In similar vein, because I can click anytime I want, it is wrong to argue whether a graphical object on the screen “affords clicking.” It does. The real question is about the perceived affordance: Does the user perceive that clicking on that location is a meaningful, useful action to perform?”[2].

Buttons, for example, can communicate affordance by shapes, colors, and shadows. However, perceived affordance is cultural. Our ability to perceive affordance in graphical objects relies on our existing knowledge and experience. For example, the blue hyperlinks that are now common in browsing would not afford to click until people learned that it was possible to click on links in the text.

---

[2] `http://www.jnd.org/dn.mss/affordances_and_design.html`

Summary

- Perceptual experience is not determined by what is shown on the display alone, but in conjunction with our expectations and strategies of deploying attention.
- Perception in HCI can be understood through elementary tasks it needs to serve: discimination, detection, recognition, estimation, and search.
- Another main role of perception is to 'bring order into chaos', in other words organize the overwhelming sensory experience into something we can act on.
- Visual saliency refers to the probability with which something on display attracts our visual attention. When the user interface is cluttered, all visual primitives are 'congested' and nothing can be made to stand out from the page.

Exercises

1. Analysis of a visual design. Choose a graphical user interface you think is difficult to use. Could this be because of the organization of the display? Take a screenshot and analyze it in terms of how it uses the principles of (1) visual saliency (what attracts attention) and (2) Gestalt principles (grouping of information).
2. Alternative modalities. Section 3.1 lists three types of sensory modalities prevalent in computer use. Consider designing a display that shows the speed of a car to a driver. What are the pros and cons of each sensory modality when used as a display in this case?
3. Perceptual tasks. This chapter has discussed five perceptual tasks in HCI: discrimination, detection, recognition, estimation, and search. Which perceptual task the defining task for the following cases: (1) Playing a first-person shooter, you see a shadowy character move in a distance: is it a friend or a foe? (2) You arrive at the airport and look at a display of departing flights: from which gate is your flight leaving? (3) You're in subway and feel a slight vibration in your pocket: is it your mobile device or tremble from the subway? (4) You're wearing a VR headset and see a button close to you: Is it far enough so that you can press it without moving or should you first walk closer to it? (5) You see an icon that has a circle with a cross (X) in the middle: is this a 'Close window' button?
4. Using smell in user interfaces. Compare olfaction (smell) and vision as sensory modalities. Think about everyday human–computer interaction: Even if technology would allow olfactory displays, why are they unlikely to replace vision?
5. Understanding clutter. Pick a user interface (e.g., a ticket vending machine) you think appears cluttered. Why is it? Refer to Treisman's conjunction search to identify three regions of the UI that might be hard to find. Circle and number the regions in the photo above and explain why they are hard.

6. Visual grouping in graphical user interfaces. Take a graphical user interface and look at it: Which elements belong together, i.e. are grouped? Which visual grouping principles are used to achieve that?

7. Unhelpful use of color. A simple mobile device is operated by holding the device in the non-dominant hand and a plastic pen in the dominant hand. The device enables text entry through a graphical keyboard presented on a resistive screen. The system does not perform word prediction or auto-correct. A designer proposes to improve text entry performance by changing the colors on the keys on the graphical keyboard. Based on principles taught in this chapter, explain why this approach is fundamentally flawed.

4. Motor Control

When users type on a keyboard, tap on a touchscreen, steer a car, or generally perform an action using their body, they rely on *motor control*. Human motor control refers to the regulation of all movement in a human, including integrating relevant internal and external sensory information to determine the necessary signals to trigger muscles to activate. Motor control is critical for fundamental human tasks such as balancing the body and pointing and grasping an object.

Motor control in HCI focuses on humans interacting with computing systems, for instance, typing on a keyboard, moving a mouse pointer to an icon, and reacting to a prompt. Precise understanding of motor control allows the design of user interfaces that better fit the capabilities of users. Such an understanding is often captured in the form of mathematical or computational models. For example, the average time it will take an average user to touch an app icon on a touchscreen phone can be predicted with a model if we know (1) the distance to the app icon and (2) the size of the app icon. Predictive models like these have seen many applications in HCI:

1. Graphical layouts can be personalized to better match a user's motor abilities [262]. Depending on the amount of tremor, elements on an interface can be made bigger and the layout reorganized so that they can be better reached.

2. Target selection techniques utilize motor control models to dynamically make objects on display more easily selectable. For example, they can change their selection areas to assess fast selection (see Chapter 26).

3. Keyboard layouts have been optimized using models of pointing. For example, letters and characters can be assigned to keys so that the average selection time for a given language is minimized. DSK (Dvorak Simplified Keyboard) is a famous early example of thinking how to break the hegemony of the Qwerty layout. While optimized keyboard layouts demonstrate faster and less error-prone typing, this happens at the expense of having to learn a new layout (e.g., [71, 235]).

4. Input methods and devices can be compared for performance using metrics rooted in models of motor control (e.g., throughput). For example, we can compare an in-air input method and a method using the fingertip; the two have widely different movement sizes and types. Yet, with the concept of throughput, we can compare them.

Motor control models can also help to understand how difficult certain tasks are for users. Figure 4.1 shows two slightly different examples. Navigating a dropdown menu is surprisingly complex, but Ahlström [13] showed how to do that with two laws—Fitts'

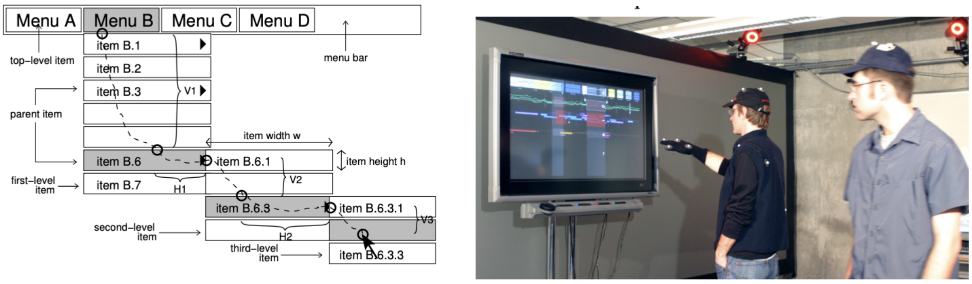


Figure 4.1.: Two examples of motor control in HCI. The figure to the left show how to select an item in a hierarchical dropdown menu. Fitts' law and the steering law can be used to predict users' performance in this task [13]. To the right is shown a system where the user can point to a large display at a distance [840]; photo from Vogel's master thesis, page 51.

law and the Steering Law–discussed in this chapter. Before we get to a more in-depth understanding of such examples, let us first look at elements of motor control tasks in HCI.

4.1. Elements of a motor control task in HCI

Movement control concerns how our nervous system produces purposeful and coordinated movement. In HCI, movement most often occurs with our fingers, but some systems have used movement of the head, feet [30], or entire body. Even if the movement is done by multiple body parts, in HCI we are typically only interested in the part that we use for something, the *end-effector*.

End-effectors controlled by different body parts have different *degrees-of-freedom*. For instance, the knee has one degree of freedom (it is basically a hinge), whereas the hip has three degrees of freedom (it can rotate around three axes). These degrees of freedom are often used to translate an end-effector in 3D space and rotate them along three axes (called roll, pitch, and yaw). This gives a total of six degrees of freedom.

Moving is complicated. Often, joints need to work together to produce movement, the environment's effect on your body varies, and the levels of fatigue in your muscles vary. All of this causes variability in movement. Learning to move your body and learning to control computers through movement is about regulating that variability.

Movement of end effectors may be coordinated in two principled ways. One is called *open loop*, referring to the fact that there is no feedback loop during the movement. When you quickly press a blinking button or grab your coffee mug, you are doing open-loop control. In *closed loop control*, feedback is received and incorporated during motion. This feedback allows you to correct the movements, making them more precise. (We discuss these control principles in Chapter 17.)

In HCI, we care mostly about *aimed movements*. That is, a user attempts to move an end-effector (say, their finger) to a certain location (say, the button on a touchscreen).

However, a large class of non-aimed movements find applications, too, in HCI. Gestures are one example (think swipe); normal walking is another.

More formally, aimed movements are movements in which success is defined by external constraints. To elicit the right command, we need to move the body in the right way and at the right time. For example, to send character 'a', a sufficiently small body part must land exactly on the cap of the button with sufficient (but not excess) force. Except for brain-computer interfaces, most UIs are operated by aimed movements. Two fundamental types of *movement constraints* can be distinguished.

Spatially constrained aimed movements These movements are restricted at the end or during the movement to a specified region or point. *Discrete aimed movements* are movements *to* spatially bounded targets.

An example is moving a mouse cursor on top of a button to select it. This movement type is prevalent in HCI, it is, in fact, one of the prime paradigms used to communicate intentions and commands to a computer. Consider for example buttons, widgets, links, icons and so on.

Continuous aimed movements, in contrast, require keeping the control point within a bounding box during the whole duration of the movement. For example, keeping a cursor within a tunnel when navigating a hierarchical menu requires continuous "steering" (Figure 4.1). We discuss steering and gesturing in the two next sections.

Temporally constrained aimed movements In these movements, a target defined in time must be hit. The target can be hit during a specific interval, or the goal is to be as close to the target as possible. An example of the latter is playing notes on a piano, and an example of the former jumping over obstacles in a video game. Often these two types of constraints occur together.

In an *interception* task, spatial and temporal demands co-occur. For example, we need to catch a moving object by (1) placing a selector on its future path and (2) pressing the button when the object is within the selector's effective region. Consider for example hitting a tennis ball served by the opponent or sniping an enemy player in a first-person shooter game.

A fundamental tendency holds for virtually all aimed movements. The *speed-accuracy tradeoff* refers to a limit of the motor system: it cannot be both fast and accurate at the same time. If you think about moving your hand to a target, it is clear that you can either choose to move your hand quickly towards the target or to be precise in hitting it. Alternatively, you can choose some mix of these two objectives. However, you cannot be good at both at the same time.

4.2. Target Acquisition

Target acquisition is a discrete, spatially constrained aimed movement. *Pointing* is perhaps the most typical target acquisition task in HCI. Here, the goal is to move the control point on top of an area denoting the target. If the control point ends up missing the area, the movement is considered having missed the target. Typically, there are two performance objectives: be as fast as possible and do not miss the target; that is, keep the error rate below a threshold.

Although pointing is the most common target acquisition task in HCI, three other tasks are studied in the literature:

- A point target: The target is defined as a point in space. The user's accuracy is measured as the Euclidean distance of the end point from the point. This is natural for targets with diffuse shapes, such as a moving game character on a display that changes it shape.
- A line/surface target: The target is a line in space, and the user's goal is to move the control point to cross the line. Performance is measured in terms of movement time. When the target is a line segment (with finite length), accuracy can be measured as hit/miss. For example, we can define items in a menu as lines as opposed to regions, thus making them faster to select.
- A postural/angular target: Joints must be rotated to a particular angle. For example, in some dancing games, users need to replicate a specific posture shown on display.

An immediate question for HCI is when are these tasks difficult to do and how may we design them to be simpler. Fitts' law provide answers to this question.

4.2.1. Fitts' law

The time it takes a user to point to a *target*, for example a 2D button, depends on how large it is and how far away it is. Average movement time in this task can be mathematically predicted using Fitts' law. Fitts' law is a mathematical model that predicts that the average movement time required to hit a target along a one-dimensional path is proportional to the difficulty of hitting a target. This difficulty involves the distance to the target and the width of the target.

The law is not a law in the sense of a natural law, but rather statistical regularity discovered through experimentation. In the original research paper, Fitts [239], an American psychologist interested in human performance, used what is now known as the reciprocal task paradigm. Participants were asked to select targets of varying distances and widths by alternatively pointing to the left and to the right. Figure 4.2 illustrates this reciprocal task paradigm as it was used in the original experiment.

For such a task, Fitts' law predicts that the average movement time MT can be predicted by a linear relationship:

$$MT = a + bID, \tag{4.1}$$

where a and b are regression coefficients and ID is known as the *Index of Difficulty*. The Index of Difficulty is an encoding of how far away the target is and how large it is, and it is defined as:

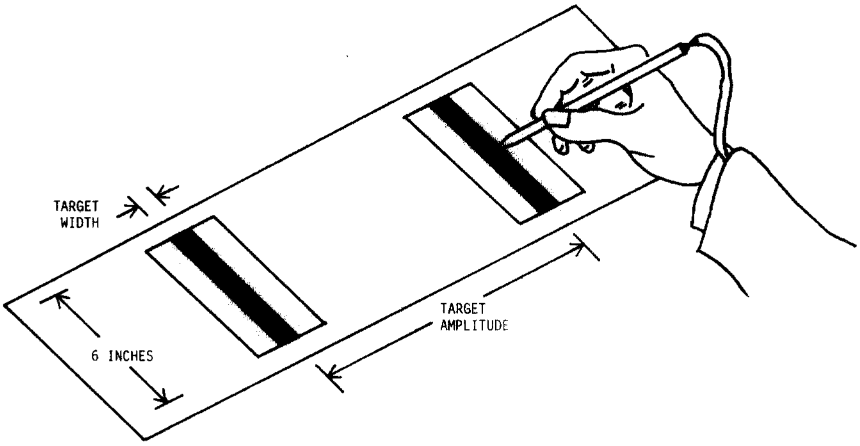


Figure 4.2.: The experimental setup of Fitts manipulated movement D and target width W in a reciprocal tapping task. The two metal plates were to be hit with a stylus in an alternating sequence as fast as possible. [Figure adopted from [491].]

$$ID = \log_2 \left(\frac{D}{W} + 1 \right), \tag{4.2}$$

where D is the *distance* to the target and W is the *width* to the target. Index of Difficulty (ID) is easy to interpret: The closer the target is, the lower D is and hence the lower the ID. Similarly, the larger the target is, the larger W is and the lower the ID. In other words, targets that are either close, large, or both, exhibit a low ID, and targets that are far away, are small, or both, have a high ID.

Another intuition behind $\frac{D}{W} + 1$ is the following: it denotes how many possible targets of width W would fit within distance D. In other words, ID denotes how many targets could have been selectable (even if only one was selected in the end). We add one (1) to the fraction to account for for the fact that pointing aims at the center of a target, which means that we have to add $0.5W$ to both both ends of the movement. By convention, ID for pointing tasks is nearly always defined in terms of logarithm base 2, which means ID has a unit of *bits*. The more bits that are required to express the $\frac{D}{W}$ relationship, the higher the ID. This intuition is discussed in more detail in Chapter 17, when we discuss the relationship of the law to information theory.

Mathematically, Equation 4.1 is identical to the equation of a straight line $y = a + bx$, where a is the intercept and b is the slope of the line. This is because Fitts' law describes a linear relationship. The variables a and b are regression coefficients as the specific intercept and slope of the line governing a Fitts' law relationship is determined by experimentation: by varying the distances and widths of targets, and measuring what the average movement times are, it is possible to infer the intercept–the a parameter—and the slope—the b parameter—of the line using a mathematical procedure known as linear regression. This is what Fitts did in his original 1954 paper when he discovered the regularity.

4.2.2. Using Fitts' law to assess input performance

The model has another interesting property: it can be used to compare users' performance across conditions that had different target properties. Consider for example wanting to compare a joystick and a touchpad for target selection; how would you do that? Throughput is a property that tells that. Several definitions of throughput exist, as none is ideal. One definition of throughput is [491]:

$$TP = \frac{ID_{avg}}{MT_{avg}}. \tag{4.3}$$

The downside with Equation 4.3 is that it relies on an arbitrary average ID.

An alternative definition of throughput is given by Zhai [902]:

$$TP = \frac{1}{b}, \tag{4.4}$$

which relates throughput solely to the slope b of the line in Fitts' law. The downside of Equation 4.4 is that it ignores any effect of the intercept a. Using either definition of throughput it is possible through experimentation to determine the throughput of various pointing methods. A higher throughput is better.

As stated above, the parameters a and b may vary depending on specific tasks, user groups and use contexts and there is a rich body of research investigating how these parameters may change. Also, several extensions to the model have been considered in the HCI literature, such as modeling 2D target selection [492] and targets that move on the screen.

4.2.3. A Worked Example

Let us show in more detail how to apply Fitts' law. The experimental task in Fitts' work, shown in Figure 4.2, is called *reciprocal tapping task*. This is a study protocol one can arrange relatively easily to obtain a Fitts' law model. In this task, users are asked to tap two targets of width W placed at distance D from each other.

In the original study, Fitts systematically manipulated D and W – distance and width of target, respectively. His data is given in Table 4.1 [239]. It tabulates average MT as a function of D and W. The insight that Fitts made is that while there is no obvious relationship between MT and neither D nor W, they can be combined into a single term that does. This is the basis of Fitts' law.

Why are tasks with higher ID motorically harder? You can approach this question by thinking about what happens when distance increases and when the width is decreased. Both increase ID and therefore also MT. In other words, other things being equal, targets that are further away or smaller are harder and therefore slower to the point. Thus, movement time is related to the inverse of spatial error. Using ID also makes computations with the model simpler: instead of the non-linear relationship, we can now deal with a simpler, linear relationship.

Fitts' model for the obtained data is shown in Figure 4.3. Averaging the MT data obtained in each ID condition, and fitting the empirical parameters a and b, Fitts found

MT	D	W	ID	Predicted MT
180	2	2	1	107
212	2	1	2	202
203	4	2	2	202
281	2	0.5	3	297
260	4	1	3	297
279	8	2	3	297
392	2	0.25	4	392
372	4	0.5	4	392
357	8	1	4	392
388	16	2	4	392
484	4	0.25	5	486
469	8	0.5	5	486
481	16	1	5	486
580	8	0.25	6	581
595	16	0.5	6	581
731	16	0.25	7	676

Table 4.1.: Data for Fitts' original stylus pointing task. The predicted MT is 12.8+94.7ID, where $ID = log_2(2D/W)$.

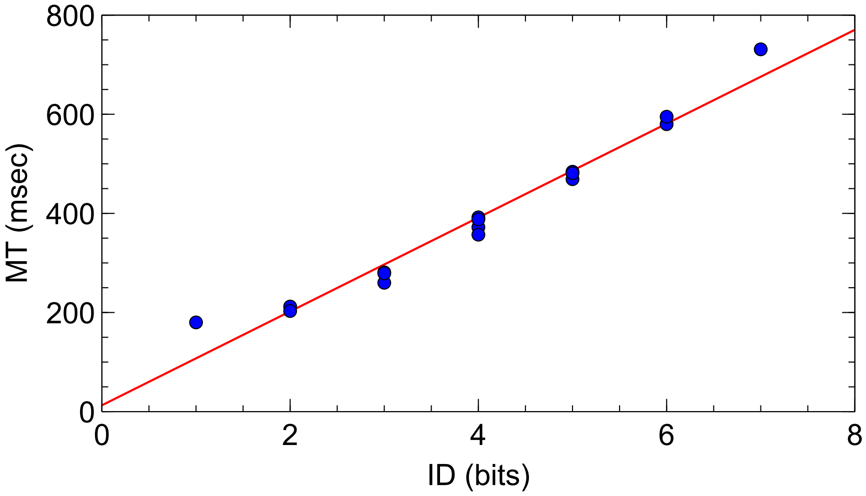


Figure 4.3.: A Fitts' law model of stylus pointing. The plot shows observed movement time MT versus index of difficulty ID. The linear trend depicts the model $MT = 12.8 + 94.7 \times ID$. The model fit is $R^2 = .967$. Data presented in Table 4.1.

the relationship in Equation 4.1. After computing ID, parameters a and b are estimated[1], yielding in this case:

$$MT = 12.8 + 94.7 \times ID, R^2 = .967 \tag{4.5}$$

The fit of the model, indicated by R^2, is high. R^2 is a statistical measure that tells how much of variance in MT can be explained by the regression model.

---

[1] Parameters a and b can be fit to data with a statistical method called OLS (Ordinary least squares), which can be found in most statistical packages and even Excel.

Paper Example 4.2.1 : Effective width

So far, we have assumed that W is defined by the interface or a researcher. Another way to look at it is that it is something that the user can affect. A precise user may be more accurate than the actual target design insist, and vice versa, an imprecise may be unable to reliably hit the designed target because the effective width is larger. To account for this behavior, MacKenzie [491] provided a variant of the model:

$$MT = a + b\log_2\left(\frac{D}{W_e} + 1\right), \tag{4.6}$$

where W_e is *effective width*. It refers to the empirical spread of end-points around the target center. While W refers to the actual width of the target, W_e is the target that users *can* hit most of the time, or its *effective* width. You can consider effective width to be motor variability that we can measure when a user is repeatedly carrying out the motor task.

Effective width can be computed if we can collect data on endpoints of movements; that is, where the users' aimed movements end. When end-points are normally distributed, the standard deviation can be used to determine W_e. The typical cut-off is $\sigma = 4.6$, which amounts to 4 % of end-points. In other words, effective width W_e would here determine the width of the target that would be hit 94 % of time. The idea is shown in Figure 4.4. Because the cut-off defines an acceptable proportion of errors, it should be decided case-by-case.

This formulation makes the relationship of the model with speed–accuracy trade-off clearer. If the user tries to be faster, the actual target width may not change, but the effective width will increase. When ID increases – in other words, the task becomes harder – either noise (effective width) will increase or the user will have to be slower.

4.2.4. Applications of Fitts' law

Fitts' law is widely used in user interface design and evaluation. If the a and b parameters are known, then it is possible to calculate how average movement times will change depending on the sizes and distances of targets. Fitts' law and its variants are used for designing better layouts, interaction techniques, and input devices. By exposing how user performance is affected by design-relevant factors, and by offering a unified account of both speed and accuracy, Fitts' law has become the core of our understanding of aimed movements. Fitts' law has also provided a basis for empirical comparisons of input devices, and it has driven innovation in interaction techniques. Good examples are the key-target-resizing technique used in virtual keyboards and layout optimization algorithms that have challenged Qwerty as the dominant keyboard layouts [71].

Fitts' law permits rigorous empirical comparison of input methods. Consider the problem of comparing two input devices. Figure 4.5 shows a plot comparing data from MacKenzie [491] The plot shows that the stylus is better throughout the ID range, it is thus preferable. However, if there was a *crossover* point, it would be exposed by the

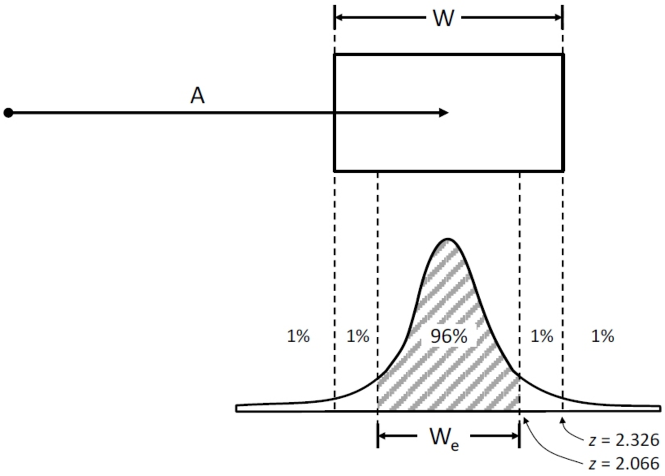


Figure 4.4.: Effective width denotes the target size that the user would be able to hit X% of time, where X is usually set to 96 %. Effective width is computed by assuming that end points are normally distributed and setting a cut-off based on standard deviation. Image source: I.S. Mackenzie 1992, Fitts' law

model, even if there was no observation at that ID point.

The concept of ID is powerful here. It collapses two parameters that describe a motor task into a single variable that is linearly correlated with MT. The alternative, called the naive-but-tempting approach by Zhai [902], would be to measure speed and accuracy on a pointing task. However, comparison would be limited to the selected observation point. Conducting a Fitts' law study invites us to systematically vary D and W, and the model will provide a point of reference – the a and b parameters, for comparing the input methods across the full scope of the motor task.

But Fitts' law is not limited to empirical comparison. Fitts' law can be used *analytically* – that is, prior to collecting empirical data – in two ways:

- Predict mean MT for a pointing task; however, empirical parameters a and b must be known. They can be obtained, for example, from literature.
- Compare pointing tasks: When a and b are equal, they can be ignored, because MT will be determined by ID only.

ID – the index of difficulty – thus offers a handy entry point for analyzing a motor task. As we learned, increasing ID is associated with increasing MT. All other things being equal,

- An increase in D will increase MT
- A decrease in W will decrease MT
- If D or W changes, MT can be kept constant by changing the other.

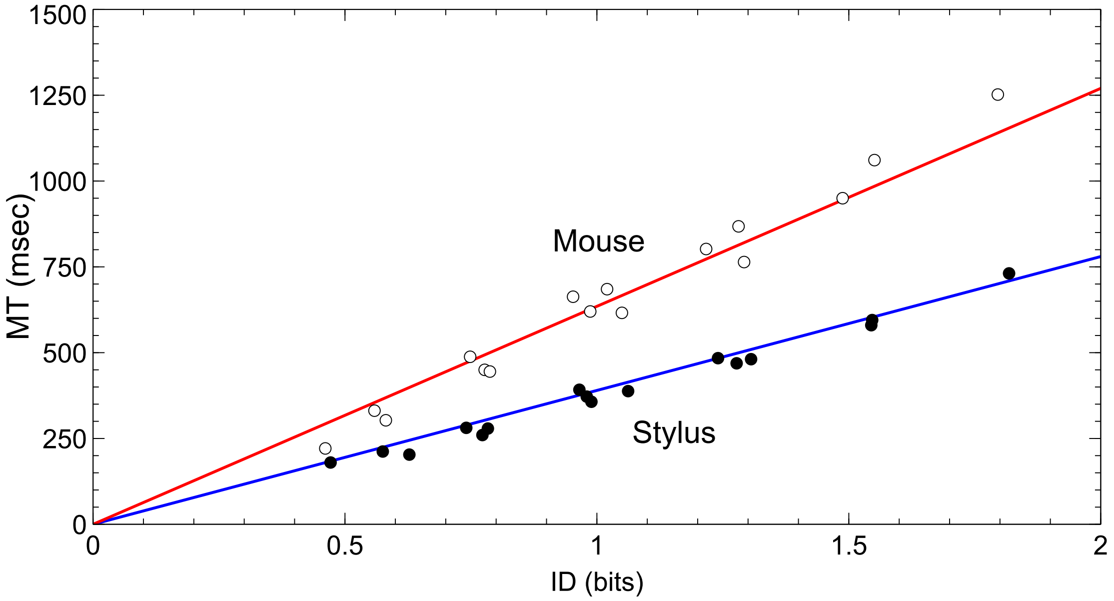


Figure 4.5.: Fitts' law models allow the comparison of user interfaces. Here, the stylus shows superior performance over the mouse. Data from [491].

Paper Example 4.2.2 : Accounting for corrective movements

Fitts' law is perhaps the most widely tested model in HCI research. It has been evaluated in many user groups, input devices, and contexts, including underwater (see [765])! It has weathered numerous challenges to its theoretical and mathematical assumptions. However, it tells very little about what happens during movement. During pointing, we have several feedback signals available: vision, audition (sound of movement, clicks, etc.), and proprioception. We use such signals to guide movement. Several models have been proposed that account for corrective movements.

One such variant is the *iterative corrections model* [174]. The idea is that any pointing movement consists of several *ballistic* movements, in-between of which there are corrections. A ballistic movement itself cannot be modified after triggering it; however, the next one can be planned considering sensory feedback. These redirections are 'corrections', thus the term 'iterative corrections'. However, detailed recordings of how laboratory participants move showed only one or at most two corrections. Moreover, considerable variation has been found in the duration of the initial sub-movement, thus the idea of equal durations is violated.

An extension of the idea was proposed by Meyer et al. [525]. *The stochastic optimized sub-movement model* defines MT as a function of not only D and W, but also the number of sub-movements n:

$$MT = a + b\left(\frac{D}{W}\right)^{1/n}, \tag{4.7}$$

where n is an upper limit on sub-movements. The authors found empirically that $n = 2.6$ minimizes RMSE[a]. Several extensions have also been proposed to compute end-point variability, similar to the concept of effective width.

These two models assume *intermittent feedback control.* Intermittent control means that control actions cannot be carried out at any time but only after 'locked' periods. In this case, the ballistic part of a motor action cannot be altered, but there is a window of time afterward to make corrections. The two models assume that such corrections are based on the error at the start of that action. Meyer's model also assumes that the neuromotor system is noisy, and that this noise increases with the velocity of the sub-movements. This causes the primary sub-movement to either undershoot or overshoot the target. One known shortcoming of the model is that the number of sub-movements is fixed. For a given D and W, the sequence of sub-movements would always be the same, and it is not possible to explain why the target is missed at times.

---

[a] The RMSE is the Root Mean Square Error of the distance between the model's prediction and the data. It serves as an indication of a model's fit to the data.

4.2.5. Limits of Fitts' law

Fitts' law is a simplification of what happens in pointing. Because it models average MT in ID conditions, it effectively hides *variability* other than the resulting performance. Variability is inherent in all motor control, affected by factors such as movement strategy (e.g., eye-hand coordination), feedback, and the involved muscle groups. For example, if you ask a user to carry out a pointing experiment with 5-10 cm targets, the resulting model may not generalize to a condition where the targets are 10-20 cm. The model is also brittle. Even small changes in the task, user, or conditions can require the collection of a new dataset and the adjustment of the model parameters.

4.3. Simple Reactions

Simple reaction is another fundamental motor action in HCI: Something appears on the display or in the environment, and the user must respond to it as quickly as possible. *Motor response* refers to the elicitation of a motor movement appropriate to a presented event or stimulus. This happens for instance when:

- reacting to a flash on display
- blocking an enemy's move with a counter-move in a computer game
- answering an incoming call on a mobile device
- braking a car
- getting rid of an annoying dialogue box asking if you want to install a new version of software

Only one response alternative is available in simple reaction—this is why it is called simple. The response can be stated by saying something aloud, pressing a button with a finger, or even by thinking of a response in brain–computer interaction. A simple reaction carries a single bit of information: that a response has taken place. Typing a phrase, for instance, is much more complex, since one must consecutively select (choose) the right key from a set of at least 26 alternative characters. We call those *choice reactions.*

Simple reaction tasks are among the fastest human responses in HCI. Performance in this task is measured in milliseconds as the time duration between the onset of the event and the user's response. Typical responses, depending on the input and output modalities, are in the range of a few hundred milliseconds (200-250 ms). However, within this rather narrow range, relatively large differences can be observed depending on the condition.

Simple reactions have been studied in psychology for more than a century, and the involved cognitive processes are somewhat known. Some of the earliest studies considered the vigilance of soldiers in World War II: After sleep deprivation, or in a stressful event, how would reaction time be altered? In HCI, it can be used to understand fast reactions, such as when playing games or musical instruments on a computer. The response consists of a decision process ("I must react") and a motor response (the pressing of the button). In the following, we present one way of understanding this process.

Figure 4.6.: **Simple reaction** is a motor task where the user must respond to a prompt as quickly as possible by pressing a button. **Choice reaction** generalizes this to the case where more than one response option is available. Image by Evan Amos, used with permission.

4.3.1. Drift diffusion model

Ratcliff and Van Dongen [671] presented a *evidence accumulation model* that predicts the distribution of reaction times as a function of what happens after the stimulus has appeared. While it has been mostly used in safety-critical reaction tasks, such as driving, it is illuminating also for HCI applications, where it is important to understand what affects reaction time.

The idea in evidence accumulation models is that perceptual evidence for and against responding accumulates until some *threshold* is met, and the motor response is launched.

- Stimulus onset: The event that one should respond to appears, for example a big figure suddenly on the screen in a first-person shooting game;
- Perceptual encoding: The event is encoded as a candidate for one that should be responded to; Example: encoding the visual shape and figure of the thing that appears;
- Evidence accumulation: Every fixation samples more evidence pro/against the decision to respond; Example: ”Is this a friend or a foe in the game?”
- Decision: When enough evidence has accumulated to meet a decision threshold, the corresponding motor action is launched; Example; ”Yes, this is an enemy!”
- Motor action: Launching the overt, movement response to trigger the proper response.

Hence, between the stimulus and the observable response many things happen. Evidence accumulation models can account for some effects of user interface design as well as

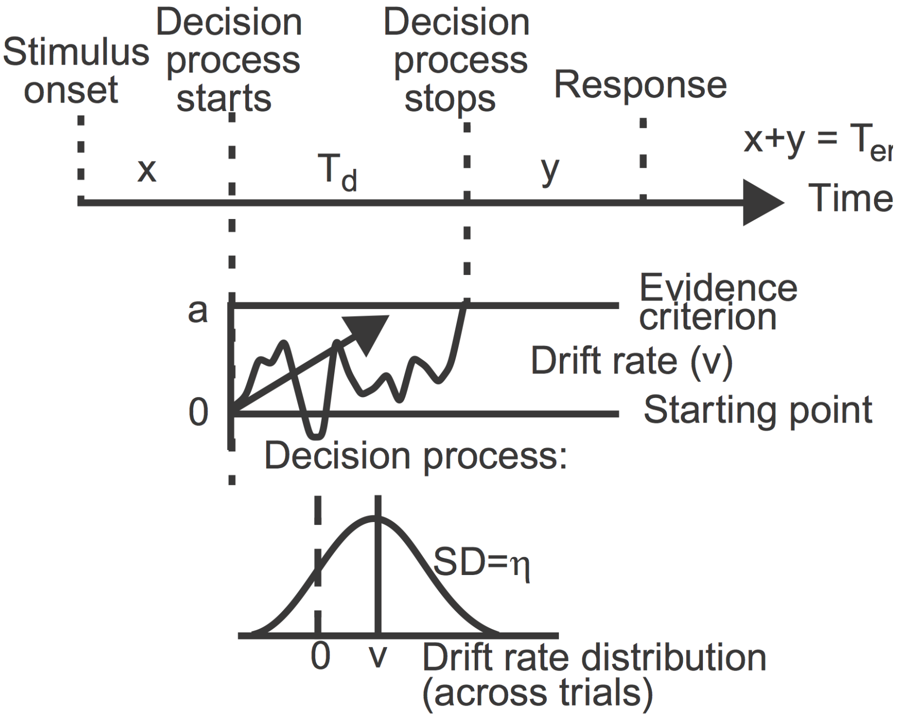


Figure 4.7.: The model of Ratcliff and Van Dongen [671] explains simple reaction as consisting of a decision task and two non-decision tasks. After perceiving the prompt, a decision task starts. It is assumed to be a stochastic diffusion process where evidence accumulates toward threshold a, at which point the response is emitted.

various task-related, individual, and contextual factors. They predict naturally occurring variations we can observe in performance when the same reaction task is repeated.

The model assumes that simple reaction RT has two sources of variation: decision time T_d and nondecision time T_{er}. Nondecision time is further broken down into two subcomponents x and y. The first nondecision event is the perceptual encoding of the stimulus that lasts for some duration, marked with x. After perceiving the stimulus, a stochastic decision process starts. During this period, evidence is accumulated ("diffused") in the brain that the stimulus should be responded to. This evidence accumulation phase is affected by perceptual conditions like noise (e.g,. poor resolution, poor eyesight) and the complexity of the visual scene.

Finally, after sufficient evidence has diffused to surpass threshold a, the decision process stops and continues to motor response process with duration y. Summing up, (average) reaction time is given as:

$$RT = T_d + T_{er} = T_d + x + y \tag{4.8}$$

Figure 4.7 illustrates the model.

This model sheds light on how user interfaces could improve users' reaction time. The duration of the decision process T_d is user- and task dependent and varies across trials. The drift rate (or the accumulation of evidence) is assumed to be normally distributed with mean v and variance η. [2] The two nondecision components x and y are summed to T_{er} and treated together in the model. They can also change according to the user interface. For example, an auditory prompt may take longer to register than a simple visual symbol. This nondecision component is also assumed to vary across trials with SD s_t.

The standard way of plotting the predictions is *the hazard rate function*. It gives the probability that the decision process terminates in the next instant of time, given that it has survived to that time. Formally, $h(t) = f(t)/(1 - F(t))$, where $f(t)$ is the probability density function and $F(t)$ is the cumulative density function. Figure 4.8 shows three examples assuming the same decision threshold a:

- *Slow but perfect responder*: In the top-most figure, drift rate v is mediocre (0.4) with no variation ($\eta = 0$). The shape of the function is "perfect" in the sense that it is only achievable by a user who can decide the stimulus with no hesitation.
- *Fast responder*: Here drift rate v is higher, but there is more variation ($\eta = 0.3$). This yields a much faster response, peaking around 300 ms.
- *Slow responder*: Here drift rate v is mediocre (0.4) but there is some variation ($\eta = 0.3$). The hazard distribution has a long tail. This kind of variation could be produced, for example by sleep deprivation or by noisy, hard-to-interpret display.

Consider an application in controlling an avatar in a 3D world. A user controls an avatar from an egocentric perspective and has to accelerate and decelerate and steer the

---

[2] The diffusion process is given by $dX(t) = vdt + sdW(t)$, where $dX(t)$ is the change in the accumulated evidence X for a time interval dt, v is drift rate, and $sdW(t)$ are zero-mean random increments with infinitesimal variance s^2dt.

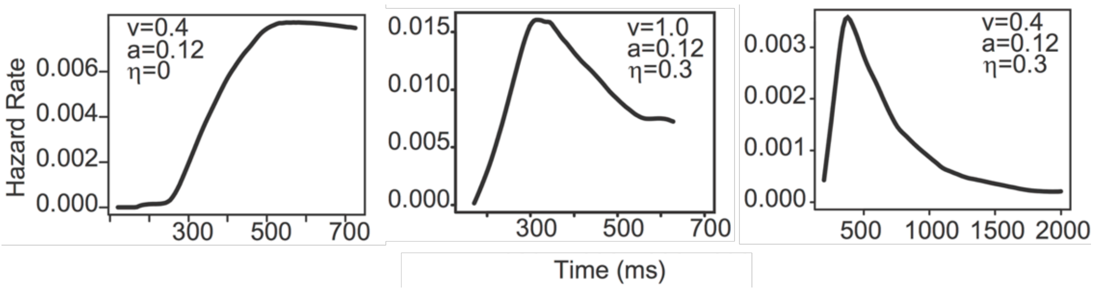


Figure 4.8.: Three hazard function examples. For comparison, notice the different scales of axes The figure is adopted [671].

avatar. If the user is asked to speak to a phone simultaneously, for example, response times to abrupt events will increase. This can be attributed to reduced drift rate. That is, when speaking on the phone, their sampling rate slows down and, therefore, a longer time is needed to respond to the event.

4.4. Choice reaction

In *a choice reaction task*, instead of one response option like in simple reaction, n options are available. When a cue (stimulus) appears, the user must execute the *corresponding* response as quickly as possible by pressing the associated key. Each cue is associated with a single response; cues can appear with different probabilities. The fingers are supposed to rest on the associated keys, to minimize the effect of pointing in the response. Consider this example: You are playing a racing game and your car is approaching a T-crossing at fast pace. Which way should you turn? This is a 2-alternative forced choice task.

Performance in choice reaction tasks is measured as *choice reaction time* (CRT): It is the time that has elapsed from presentation of the cue to the response. Errors – choosing wrong response or not responding – can be dealt with either by insisting on correct response, or by allowing errors to happen and reporting error rate alongside with CRT.

Choice reaction is a generalization of the simple reaction. When $n = 1$, we have the simple reaction. When $n = 2$, we have the 2-alternative forced choice (2AFC) task, there are two response options and one must be chosen. 2AFCs are common in HCI. Consider an incoming call, for example; It forces the user to pick between Answer and Reject call options. When n increases beyond two, we find an interesting and practically important relationship between n and CRT.

4.4.1. Hick–Hyman Law

The Hick–Hyman law is a statistical relationship between n and CRT discovered independently by Hick and Hyman [334, 362]. The law states that when the number of response options increases, choice reaction time increases. Trying to be faster than what the law suggests will lead to errors. By allowing more time for responding, fewer errors will occur.

Formally, given n equally probable response options, the average CRT follows approximately:

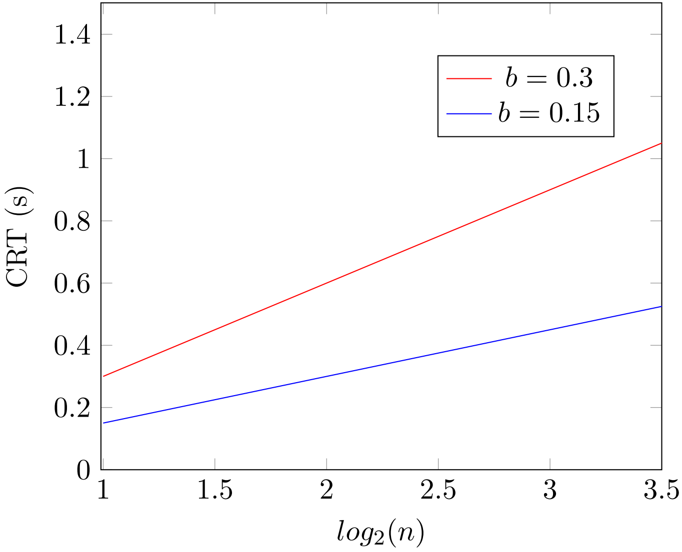
$$CRT = a + b \cdot \log_2(n) \tag{4.9}$$

where a and b are empirical constants, determined by fitting the line to data. One can be added to n in cases there is uncertainty about whether to respond or not:

$$CRT = a + b \cdot \log_2(n + 1) \tag{4.10}$$

Not-to-respond is just one more option.

Parameter b controls how strongly the increase in n affects CRT:



Why is there a *binary* logarithm in the standard formulation? According to one theory, it reflects a binary search on the part of the user. When a cue appears, the user first picks one-half of the options and rejects the other half, then picks half of the remaining options, and so on, until finally identifying the correct response. This form also links Hick–Hyman law to information theory [734], see Chapter 17.

Another interpretation of the law is that it denotes *uncertainty about the stimulus*. In the case of choices with unequal probabilities, the law can be expressed as:

$$CRT = a + bH \tag{4.11}$$

where

$$H = \sum_{i}^{n} p_i \log_2(1/p_i + 1) \tag{4.12}$$

and p_i marks the probability of response option i. Even though there is a large number of options available, if only a small subset is effectively in use (i.e., the sum of their probabilities is close to 1), CRT can be low.

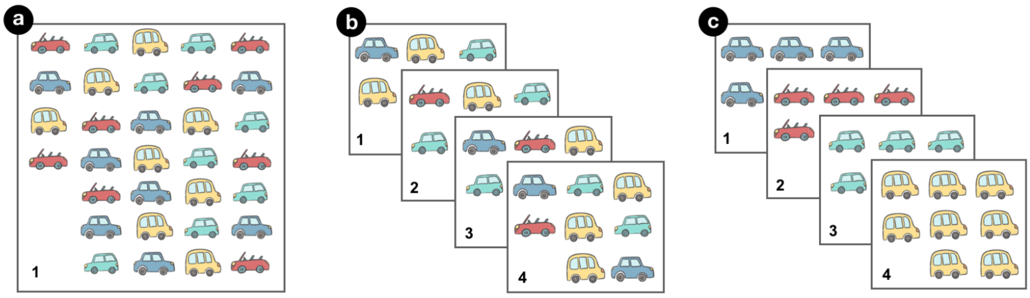


Figure 4.9.: You need to show 32 items to the user. According to Hick–Hyman law, which of the designs is the fastest to use? (Figure from Liu et al. [476].

4.4.2. Applications in HCI

Hick–Hyman law is one of the two laws – the other is Fitts' law – that Card et al. [129] introduced to HCI as motor control principles that can be used to improve the usability of user interfaces. However, it saw relatively much less applications than Fitts' law. Why? Because the implication of Hick–Hyman law appears trivial: it states that less is better. If you can design an interface that has fewer responses, users will be quicker responding to it.

However, if you need to decide how to show n elements on a display, the law predicts that there is a benefit for showing *all* elements at once. Consider the design example in Figure 4.9. Because of the logarithmic term, the best design, contrary to our intuition is Design a). Hence, the design principle should be 'more is better'. However, when other factors are considered, the situation is not so simple. There are benefits to pagination and hierarchy, which are not governed by Hick–Hyman law. Information foraging theory explains that well-organized hierarchies help users save time by *skipping* whole sections of elements (see Chapter 21). Visual search time and pointing also start taking time when the number of elements increases.

The second form of the law we discussed, which states CRT as a function of entropy H, implies that decreasing uncertainty will improve performance. How can we exploit this in practice? *Stimulus–response compatibility* has a strong effect on CRT. When stimuli and responses are 'compatible', they are ordered or otherwise structured in a consistent way. Some simple mapping exists, for example cues have the same spatial order as responses. This means that the response should be similar to the stimulus itself, such as turning a steering wheel to turn the wheels of the car. The action the user performs is similar to the response the driver receives from the car. Liu et al. [476] showed that in HCI tasks where compatibility is high, the Hick–Hyman slope almost flattens out. It is generally desirable to find consistent mappings between stimuli and responses.

Design and training also have an effect on the slope b and intercept a of the model. The Hick–Hyman law governs novice-to-intermediate range of performance. When the user receives extensive practice on responding to the task, the slope diminishes, eventually flattening out. With thousands of practice trials on the same task, response time can be effectively constant when n is smaller than ten [546].

One should be careful when applying the law to cases where $n > 10$. In its basic interpretation, the n end-effectors are supposed to rest on the n keys associated with the n responses. When $n > 10$, other factors come into play, like moving fingers, finding targets visually, and reasoning.

4.5. Gesturing

Gestural interfaces are based on continuous shapes as input. Consider, for example, handwriting as text input: in order for a letter to be recognizable by the decoder, the shape must obey certain segment lengths and curves.

4.5.1. Crossing

A *crossing* task is a task that relaxes the stopping constraint in Fitts' law, the D parameter of the target [8].

Recall that Fitts' law describes a one-dimensional pointing action. Hence, when a user is attempting to hit a target, the user needs to both move the pointer to the target *and* stop the pointer before leaving the target. In contrast, in a crossing task, the user does not need to stop within the target. However, the user does, in fact, need to ensure the target is crossed. The width parameter W here refers to the size of the target the user is crossing.

Through experimentation, it has been established that crossing tasks follow a similar statistical relationship as Fitts' law and can be described using the same equation as Fitts' law:

$$MT = a + b\log_2\left(\frac{D}{W} + 1\right). \tag{4.13}$$

Unlike pointing, crossing actions can be chained together, allowing the user to cross multiple targets in one motion. An example of a system leveraging crossing actions is CrossY [27], a drawing program specifically designed for allowing the user to input commands using crossing actions with a pen.

4.5.2. Steering

A *steering* task is a task where the user is moving a cursor through a form of tunnel constraint, proposed by Accot and Zhai [7].

In general, the time T it takes a user to steer a cursor through a tunnel is:

$$T = a + b\int_C \frac{ds}{W(s)}, \tag{4.14}$$

where a and b are empirically determined parameters, C is the tunnel constraint parameterized by s and $W(s)$ is the width of the tunnel at s. As with Fitts' law and the crossing law, the parameters a and b may vary depending on the user group, task and use context.

It is possible to define an Index of Difficulty for steering tasks, however, in this case the Index of Difficulty is directly related to the parameterized curve C:

$$ID_C = \int_C \frac{ds}{W(s)}. \tag{4.15}$$

By differentiating both sides of Equation 4.14 with respect to s we obtain:

$$\frac{ds}{dt} = \frac{W(s)}{b}, \tag{4.16}$$

where we can observe that, as expected, instantaneous movement speed $\frac{ds}{dt}$ at point s in the tunnel is proportional to the width $W(s)$ of the tunnel at the same point.

The steering law can be used to model users moving cursors within tunnel constraints, such as a user moving a mouse pointer along a hierarchical linear pull-down menu structure.

Paper Example 4.5.1 : Relationship between Fitts' law and steering law

It is worth noting that there is a mathematical relationship between the steering law and Fitts' law [7]. A steering task with a single goal constraints on each end follows the same logarithmic relationship as Fitts' law:

$$ID_1 = \log_2\left(\frac{D}{W} + 1\right) \tag{4.17}$$

.

This task can be extended by adding a single further goal constraint which yields the following relationship:

$$ID_2 = 2\log_2\left(\frac{D}{2W} + 1\right) \tag{4.18}$$

.

Note that as the number of goal constraints increase, the user has to be more careful in ensuring they pass through all the goal constraints.

Then, by generalizing the number of goal constraints the user has to pass through to N, the goal constraints form a tunnel constraint:

$$ID_N = N\log_2\left(\frac{D}{NW} + 1\right) \tag{4.19}$$

.

In the limit $N \to \infty$ we then obtain the following the following relationship:

$$ID_\infty = \frac{D}{W\ln 2}. \tag{4.20}$$

Note that in the limit Index of Difficulty is no longer related to $\log_2\left(\frac{D}{W}\right)$ but to $\frac{D}{W}$ directly. In other words, in an N goal passing task, as N approaches infinity the difficulty in achieving the task is no longer related to the logarithm of the distance W and the width W. This explains why the Index of Difficulty for a steering task in Equation 4.15 lacks a logarithmic relationship.

4.5.3. Viviani power law of curvature

The models discussed thus far in this chapter predict task completion time. A limitation of the steering law is that it does not account for complex shapes or changing shapes. Consider, for example, tracing a shape or drawing. But what happens *during* movement? We need to understand that, if we wish to understand gestures, where curvature is changing.

The *Viviani power law of curvature* (PLoC) is a kinematics model for smooth curved trajectories [839]. *Kinematics models* cover aspects of motion during pointing: position, velocity, acceleration, or jerk—however, without consideration of the time-varying phenomena that produce them. They predict the moment-by-moment motion or its properties, such as radius of curvature or tangential velocity at any point. This makes

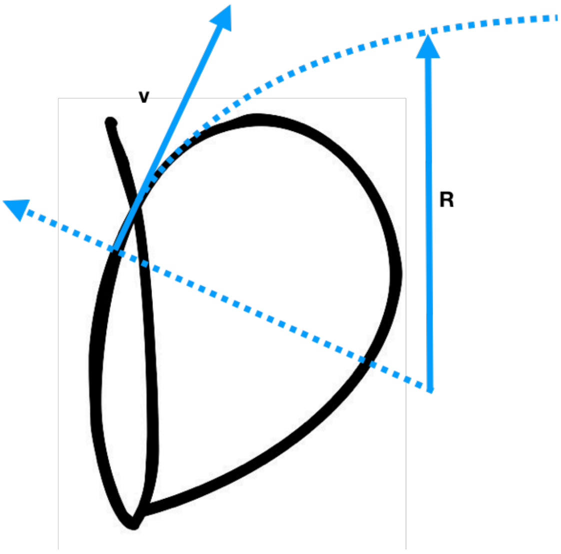


Figure 4.10.: Try drawing letter 'd' with stylus or pen. The Viviani power law of curvature predicts momentary velocity v when drawing smooth curved trajectories like this. R is radius of the curvature.

kinematics models useful for modeling gestures.

PLoC pertains to handwriting and drawing behavior, in particular when the trajectories are smooth; that is, they do not have sharp corners. PLoC relates the radius of curvature $r(s)$ at any point s along the trajectory with its corresponding tangential velocity $v(s)$:

$$v(t) = kr^{\beta} \tag{4.21}$$

where k is an empirical gain factor and β an empirical parameter. The model states that the larger curvature, the slower the motion of the end effector will be at that point.

Now, the total time for a full segment S, assuming only smooth curvature (no corners), can be computed as:

$$T = \frac{1}{k} \int_0^S r(s)^{-\beta} ds \tag{4.22}$$

For example, in a study using a stylus as the input device, $K = 0.0153$ and $\beta = 0.586$ [125]. The model has reached high fit with empirical data in drawing, with and without visual guidance.

What happens if the movement is physically larger or smaller? *Isochrony* is the empirical observation that average velocity of movements increases with distance [839]. Thus, movement distance is a weak predictor of movement time in a trajectory. Users simply move larger distances faster. The Viviani PLoC has been shown to cover isochrony. The power law -like pattern has been argued to be due to pattern generators in the neuromotor system that operate in an oscillatory fashion [717].

Summary

- Motor control is necessary for users to perform actions in a user interface.
- The time it takes users to perform actions can be predicted for some fundamental activities, such as pointing, crossing, steering, and reacting to stimuli.

Exercises

1. Applying Fitts' law. Given the following target widths and distances, calculate the average movement times for all combinations using Fitts' law. Comment on the validity and implication of the calculated average movement times: $D = [0.01, 0.05, 0.1, 1, 100]$ m; $W = [0.01, 0.050.1, 1, 100]$ m.

2. Keypad design. Think through Fitt's law about a layout of 3 x 3 icons; e.g., a numeric keypad. What happens to movement time when you make the icons bigger? How much faster will the interface be if you arrange the icons in a list (1 x 9) instead of a 3 x 3 layout? Could you do anything else to reduce the movement time?

3. Applying Hick-Hyman law. Use the Hick-Hyman law to calculate the choice reaction time for an interface with four options where the probability of each option is 0.1, 0.1, 0.1 and 0.7 respectively. Then recalculate the choice reaction time assuming the probability of each option is 0.25. Comment on the this result.

4. Understanding menu navigation. Consider a cascading linear pull-down menu. The user wishes to select a menu item in a submenu. In one variant the user has to steer the cursor through a top-level submenu item in order to trigger the submenu. When the cursor is outside the top-level submenu item then the submenu instantly disappears. This type of implementation is common in web interfaces. In the other variant the submenu triggers when the user steers the cursor at the top-level submenu item and the submenu stays in place even though the user has navigated the cursor outside the top-level submenu item. Comment qualitatively on the type of movement time model that is suitable for both types of menus and which design would be preferable and why. Optionally, further consider how the preferable submenu trigger could be implemented in an as simple way as possible.

5. Comparing input devices. Carry out a controlled pointing study with two input devices of your choice, fit the free parameters (a, b) of Fitts' law, and plot the resulting models. To collect the data, you can use an implementation of the task in the web, such as `http://simonwallner.at/ext/fitts/`. Tip: Pay attention to the range of D and W you pick.

6. Beating Fitts' law. User interfaces make it possible to do things to targets, like change their size or distance, that are impossible in physical environments. Given Fitts' law, ideate interaction techniques that would facilitate pointing (i.e,. decrease MT). Tip: Ravin Balakrishnan discusses several techniques in a classic

paper 'Beating Fitts' law': `https://www.sciencedirect.com/science/article/pii/S107158190400103X`

7. Use the drift diffusion model (code available via the book homepage) to estimate a gamer's reaction time in two conditions: in good health and after sleep deprivation. To this end, you need to think which process in the model sleep deprivation affects.

5. Cognition

Have you ever stopped to wonder about some seemingly simple interaction, asking how people can do it? For example, consider Figure 5.1: What should the user press to be able to edit a photo? This can be solved by recall from memory: ”Hmm, does Powerpoint have image editing capabilities?” But it could also be solved by means of visual attention, simply by looking around to see recognizable icons. Some reasoning may be required, for example about whether Powerpoint allows importing the image type that is being used. In the end, the user may need to decide on whether to try it with Powerpoint or just try some other means or give up. Thoughts and processing like these that go on in one's mind are jointly called *cognition*.

The term *cognition* comes from the Latin word 'cognoscere', which means to know or learn. As a scientific term, it refers to mental activities involved in thinking and understanding. It is often taken to refer to how people process information in their mind; that is, how they perceive, learn, remember, reason about, and utilize knowledge. This view is called the *information processing view* of cognition. The study of cognition is a multi-disciplinary effort spanning psychology, cognitive science, and neuroscience.

The main function of cognition in HCI is to help a user control a computer to do what they want. Attention is needed to find elements, memory to learn and recall their properties, linguistic abilities to infer text and generate command names, and reasoning to make use of knowledge obtained so far and deduce what happens inside of the computer. Unfortunately, however, these cognitive capabilities are limited. People also differ greatly in their cognitive abilities. However, the way these abilities are used is affected by the task and the design of the user interface. Cognitive capabilities are used adaptively. Users learn and try to discover new possibilities and strategies; they change behavior when a practice does not work. These properties make cognition at the same time important but challenging to understand.

In the rest of this chapter, we discuss *elementary cognitive capabilities* that relate to our ability to interact with computers:

Cognitive control Adaptively deciding goals, (what to do), allocation of cognitive resources to tasks, and change of course of action when needed. For example, cognitive control is needed when a user decides to multitask by looking at their phone while driving [705].

Memory Forming, maintaining, and accessing beliefs about objects that are not directly perceivable. For example, long-term memory helps users locate previously seen icons faster [386].

Attention Selectively processing some part of the perceptual field, for example by deploying visual attention on a screen or by sensing a tactile display sweeping fingertips

Figure 5.1.: To use a computer, users need to choose actions that get the computer to do what they want. This is challenging, however, because the computer is a non-transparent system. That is, users cannot "see" what actually happens inside the computer. The role of cognition in interaction is to help us overcome this challenge. Cognition underpins our ability to control actions, reason about the computer, learn from experience, and take decisions on what to do.

on it. We discussed visual attention in Chapter 3.

Reasoning Applying transformation rules to beliefs to form *new* beliefs. For example, reasoning is needed to infer if software can do certain things in the case where one is not able to recall doing that earlier [653].

Decision-making Interaction often requires deciding. For example, a user might decide to install application A and not application B because A appears to be more favorable with respect to features and costs.

Before we look at the key functions of cognition, we first summarize a few high-level findings on human cognition. The purpose of this is to obtain a broad understanding of basic properties of cognition. Although cognitive psychologists still study the intricacies of these findings, they are useful heuristics for practitioners and researchers to consider when thinking about interaction.

5.1. General Findings about Cognition

Many researchers have investigated cognition. We summarize some of the general findings that are relevant for HCI.

Cognition helps set goals and maintain focus on them. Cognition helps users achieve their goals. *A goal* is some desirable state of affairs. In HCI, a goal often relates to something that people would like the computer to do for them. For example, a user's goal could be to email a particular message to someone or to change privacy settings.

Goals, in turn, affect how cognition processes information. Human memory, for example, is less about storing experienced events veridically—accurately. Rather it offers access to memories that may be useful for a given task at hand. Goals set expectations about how the world is structured and what might happen next.

An important phenomenon related to visual attention is called *inattentional blindness*. If users are given a goal related to one part of a user interface, they recall features related to that part and forget the other parts. This holds true even if users did indeed look at those other parts, as confirmed by, for example, eye-tracking data. When the users are asked about those other parts later, they cannot recall the information contained in the parts that were unrelated to the goal.

Cognition is limited. Human cognition is limited. For example, working memory is limited: we can only keep a few mental representations active in our mind. The typical working memory capacity that can be simultaneously maintained active in our mind is thought to be about 2–4 items [590]. Forgetting occurs in long-term memory: we cannot remember everything we have experienced and as a result we forget details of things we have attended. Visual attention is also limited: people can extract more information from the foveal region and less from the periphery (Figure 3.1). Finally, our capacity for abstract reasoning and planning is limited. We often resort to external aids, such as calculators and notes, to help us go beyond the limits posed by our own cognition.

These limits are real and potent, although we may not be consciously aware of them. In computer use, these limits affect our ability to find items on display, make sense of information shown to us, navigate through the information space, and remember instructions on how to use something.

Cognition reasons based on internal models of reality. Cognition has the ability to reason about things that are not directly perceivable. This ability necessitates the construction of internal models of reality that can be used to formulate goals and plans. An example of such an internal model is a *metaphor*. As an example in HCI, metaphors help users understand what to expect from a user interface. The desktop metaphor uses spatial concepts that are rooted in our everyday experience of physical worlds. For example, folders are *on*, not under, the desktop and folders can be *moved into* a bin when they are no longer required.

Cognition is necessary for learning and adaptation. For some time, it was believed that human action is driven by *plans and reasoning*. That is, we plan sequences of actions and execute them. Suchman [784] challenged this view in studies of repair and maintenance personnel of Xerox photocopiers. Suchman [784] found that even in a highly regulated profession, a worker's action is not heavily scripted, but requires constant adaption and planning. While planning is carried out, such plans are often noncommittal sketches, which necessitates a need to adapt.

The present understanding is that our cognitive, motor, and perceptual processes are constantly adapting—forming beliefs, trying new tactics, and fine-tuning. This is needed to respond to the experienced structure of our environments. This need is pronounced in HCI in that the systems people use and the way people carry out work keeps changing. This phenomenon results in users having to continually formulate new plans and adapt to new ways of working with computer systems.

Hence cognition is not simply passively processing information from external environments and reacting to it. Instead cognition takes actions and intervenes in order to facilitate its own functioning. We use external aids, such as notes, calculators, and browsers to augment our abilities. Over time, such dependencies affect the way we use cognition in interaction. For example, since the uptake of the graphical user interface, which relies more heavily on visual recognition, we have less need to use our long-term memory to store computer commands that would be typed in a command prompt.

Cognition requires energy and effort. *Mental effort* relates to the use of energy we need when controlling thinking to achieve our goals [232]. Mental effort is distinguished into two components. *Task effort* refers to responses to increase computational demands. This can occur when we face novel environments, for example, when learning to use a new user interface. *State effort* refers to the energy required to protect performance from physiological fatigue, which can be caused by sleep deprivation, for example.

Effort may sound like a negative concept in that effort is something that limits performance. However, it also serves a positive function: the feeling of effort protects us from overconsuming energy in less important activities.

5.2. Cognitive control

Cognitive control refers to our ability to direct thinking and action toward some goal. When using a computer, cognition must decide among possible actions what to do in order to reach a goal state. It must activate the right representations to serve to achieve that goal [653]. It may also need to share representations among multiple tasks at the same time. In dynamic task environments, it needs to learn to control in a predictive fashion.

The basic problem of cognitive control is the following. At any given time, innumerable stimuli bombard our senses and we have several options on how to share our limited resources among them. How to ensure that the *right options* are allocated the *right resources* at the *right time*? For example, just now you could stop reading and start watching a movie. Also, if a notification pops up, do you immediately attend it or rather complete what you were doing? If you knew there was an important message coming, such as a letter of acceptance for a job application, you would probably want to attend it immediately, and this might be a rational decision. Cognitive control is critical to performing tasks in information-rich environments. The concept has been used to derive implications to the design of multi-part tasks (e.g., the use of ATMs), and regulations for multitasking settings such as in driving (e.g., suggestign minimum limits for maintaining attention on road).

5.2.1. Activating goals

Cognitive control is also needed to activate the right subgoals at the right time. Faced with a complex task with multiple goals, cognition must break it down into simpler, better manageable parts. Altmann and Trafton [17] explain this with a *goal activation model*. Contextual cues *prime* subgoals. For example, when we see a familiar intersection on the way home, it subconsciously activates the next sub-goal. *Priming* is about subconscious activation of concepts by perceived cues. However, previously activated goals can *interfere* the retrieval of the relevant goal. Cognitive control is required to suppress their activation. The model can explain *the postcompletion error*. In a postcompletion error, a user forgets to carry out an action that should be taken *after* achieving a goal. For example, one may forget to take the credit card out of an ATM after receiving cash, unless the ATM issues a reminder to retrieve it.

5.2.2. Activating task representations

When a task is performed, cognition needs to make relevant task representations available. Cognitive control operates through two mechanisms: *activation* of relevant representations in memory and *inhibition* of irrelevant representations. For example, consider the act of searching for an application icon on a desktop. Let us assume that this task was very challenging and that you have already looked at several locations. Cognitive control is needed to select, activate, locations to visit next, but also to inhibit revisiting locations you have already visited. This is called *inhibition of return.* Without this ability, you would be constantly revisiting locations, which would make the search wasteful.

This kind of regulation is important in information-rich environments. Inhibitory control is needed to avoid attending to every flash or blink on an interface and maintain focus on a task. In this way, cognitive control helps us balance between internal and external locus of control. Without internal control, we would be at the mercy of our environments. Any pop-up or notification would capture our attention, and we would not be able to complete our tasks.

5.2.3. Choosing actions

Cognition is necessary to choose what to do. A central problem in computer use is that *rewards are delayed.* In other words, we cannot immediately obtain what we are interested in. Instead, we must choose one action now, then another, and so on. Cognition is the solution to a sequential decision-making problem. But how do we know which action to choose at an immediate moment if the goal is not immediately available but instead distal? An essential ability for interaction is to evaluate actions by considering their long-term future rewards.

Two mechanisms are posited by cognitive sciences [699]. First, through experience, users learn to associate an action to *recurring contextual cues.* For example, the familiar wallpaper of your mobile device is associated certain action possibilities like pressing some icons. How are such mappings learned? Trying things allows users to learn *value estimates* of actions available in a particular state. Associative learning of such value estimates occurs involuntarily over time. It does not require effortful cognitive control but is largely automatic. For example, you may notice that when you start using a new mobile device or operating system, at first it is hard to find where things are. After a while, you can actually find the items that you frequently use more directly. However, this type of associative learning is slow to obtain and slow to change. If the user interface changes, users need to relearn the associations.

The other mechanism is *reasoning about and simulating possibilities in the mind.* For example, before selecting what to do, we may think about possible routes in a virtual environment. This requires cognitive control: you need to activate representations and compare the options, which requires effort.

5.2.4. Multitasking

We often need to carry out multiple tasks at the same time. Consider driving a car, for instance. At any given time, you would have numerous things you *could* do. You could be texting, attending to the road, talking to co-passengers, checking gasoline level, attending the AC, and so on and so forth; Cognitive control is vital for *multitasking.* The problem posed to cognitive control in multitasking is how to allocate limited resources among those tasks we want to proceed with simultaneously.

Multitasking is a resource-sharing problem. The resources include those defined by attention, motor system, and working memory.

Some of these resources can only be shared in an all-or-nothing fashion, while some can be shared in a graded manner. For example, you only have one dominant hand: in most cases it cannot be shared in a graded manner to two tasks, an exception is when the tasks can be done with different fingers. Visual attention, likewise, can be focused on a single location in space at a time. On the contrary, auditory attention can be somewhat better shared among sound sources around us, even when those sources are spatially apart.

The **multiple resource theory** (MRT) of Wickens [869] provides a rough but useful heuristic for understanding how resource conflicts arise in multitasking. The ”Wickens' cube” is presented in Figure 5.2. The cube can be used to estimate whether two tasks, when carried out simultaneously, will cause mutual conflict for resources and should therefore be avoided. For example, consider two tasks: A and B:

- Task A is playing a video game on a console
- Task B is following a conversation in social media on a mobile device.

The MRT cube in Figure 5.2 consists of segments divided into three axes. These segments denote limited-capacity cognitive resources. To apply MRT, you mark all segments that the to-be-done tasks need. If two or more tasks end up occupying the same segment, conflict emerges and will deteriorate performance.

Returning to our case:

1. Both A and B rely on visual modality, hence there is resource competition in this aspect
2. Both A and B rely on manual responding, leading to resource competition
3. While A relies on spatial processing, B relies more on verbal processing; hence, no competition
4. Both A and B can assume ambient auditory and visual stimuli (e.g,. notifications, sounds)

MRT predicts that tasks cannot be carried out simultaneously, unless task B relies on notifications. Task A can only be carried out with visual attention, and so does Task B. This means that at any given time, the cognitive controller must decide which task gets attention. Hence attention must be switched between the tasks. However, they may not need to be processed simultaneously, in case the conversation is asynchronous.

If notification sounds are turned on, visual attention does not need to be sustained on the social media app all the time. When a notification then pops up, it must be decided whether to disconnect the video game or not. Attention can be shared momentarily to read the message, while keeping hands on the video game controller.

What happens if two tasks require the same cognitive resource at the same time? This necessitate cognitive control to decide which task receives the cognitive resource at a

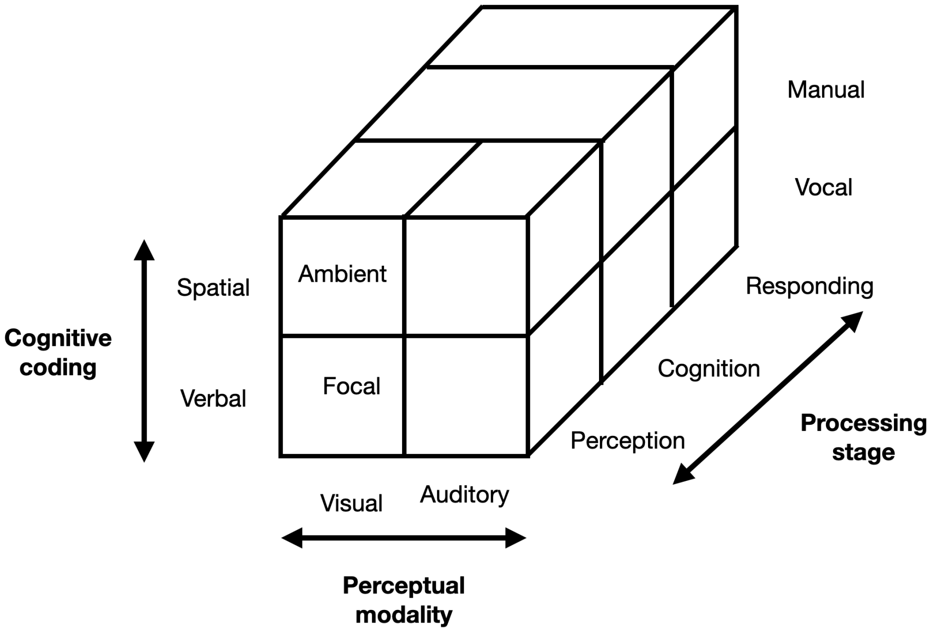


Figure 5.2.: The multiple resource theory (MRT) explains why certain task combinations are more prone to conflicts when multitasking than others [869]. The cube shows several axes that help understand the motor, perceptual, and cognitive demands of a task. To assess if two tasks lead to resource competition, you mark the slots in the cube that they occupy. If the two tasks occupy the same slots there is resource competition and we can therefore predict performance degradation. Different types of resource competition occur in different cells.

given time. The end result is interleaving: switch to task A for a few seconds, then switch to task B, etc.

This kind of interleaving is adaptive. In driving, for example, we do not switch between tasks arbitrarily. Doing so would result in a dramatic increase in accidents.

Since every task switch is somewhat costly, as you lose a few hundred milliseconds just for the switch, and you may forget part of the task if you switch too abruptly, the cognitive controller must be sensitive to both the estimated gains of switching to other tasks, as well as the benefits of carrying out a current task. For example, switching too late may result in a missed opportunity in another task. On the other hand, switching to another task for too long may also incur to much cost. This is because the longer a user is interrupted from working on a task, the longer it typically takes to resume that task later on.

Empirical studies have investigated people's ability to switch between tasks and learned that our task-switching strategies are sensitive to switching costs [377]. Cognition seeks to perform switches at natural breakpoints. In other words, users prefer to avoid switching immediately and instead postpone interruptions to natural boundaries between subtasks.

Uncertainty is a prevalent characteristic in almost all multi-tasking environments. Many tasks we need to follow in everyday life are dynamic. This means that they change on their own, even when we do not attend to them.

An implication is that at any given time when we are not attending them, we are uncertain about their state. Further, this uncertainty accumulates: the longer you leave a task unattended, the more uncertainty you have about its state.

For example, consider someone driving a car: the longer the driver attends their smartphone, the more uncertainty the driver will have about the driving environment. Guidelines for safe driving recommend maximum of 2.0 seconds shifts away from the road [595].

To decide when to switch to task, cognitive control needs access to maintain estimates of subtask states and their uncertainty. It needs to track what it does *not* know about the world. When the risk of not attending a specific task becomes too high, the cognitive controller needs to move to this task.

For example, when driving during rain, glances to secondary tasks tend to be shorter than when driving during good visibility. This is because there is more uncertainty about the driving environment when visibility is poor. When cognition does not have access to sufficiently accurate estimates of rewards or costs of a task, or of its uncertainty and risk, it may incorrectly engage with an irrelevant task.

5.2.5. Predictive control

When the task environment is dynamic, that is, changing on its own regardless of a user's action, cognitive control needs to be *proactive*. Instead of waiting to be triggered by the right stimulus, it needs to anticipate the need for a representation even before the event occurs.

To enable this, the cognitive controller requires an *internal model* of the world. The internal model should be able to predict what *might* happen if an action was taken. However, this internal model should not determine action as perception is also an important determinant in choosing an action. Thus cognitive control needs to integrate these aspects.

One hypothesis about how we do this is called the *Bayesian brain* [416]. According to this hypothesis the brain is viewed as a statistical estimator that assigns probabilities to hypotheses about the state of the world. As the brain receives new sensory information it then updates the probabilities about these hypotheses. This is referred to as updating the beliefs. This updating follows the rules of probability theory, including the well-known Bayes' theorem. A Bayesian brain uses past experiences to anticipate new experiences. The less expected a new experience is, the greater the surprise. A Bayesian brain attempts to optimize its knowledge about the world in such a way that such uncertainty is minimized.

For example, when you are interacting with a AI-driven features, like autocomplete, one may think 'what would happen if I did this, how would the feature react'. We become better and better at such predictors thanks to belief-updates. Even in case of a surprising new situation, like typing a word borrowed from another language, we utilize such beliefs to predict how the AI might react – would it know how to complete the word or not.

Consider a collaborative puzzle game. Your teammate is running very fast from one side of the display to the other. How do you know when to trigger an action throws a piece to her? A Bayes-optimal perceptual system does not represent the property of an object, for example, the velocity of an object moving on display, as a single variable but as a conditional probability density function $p(V|I)$, where I is available sensory information (e.g., change in object location between last observations). This specifies the conditional probability that the object is perceived to be moving at different speeds V, given available sensory information I.

In other words, the probability that a user interprets a certain velocity depends on what is shown on the display. However, an optimal observer must also take into account the relative uncertainty of each source when computing an integrated estimate. When one cue is more biased or unreliable than others, it should be given proportionately less weight. For example, when gamers estimate when to press a button to intercept an object, they give different weights to visual information, auditory information, and their own estimates of the object's movement (prior) [456]. By integrating all the evidence, they get a more reliable estimate than by considering either alone (posterior).

5.2.6. Cognitive workload

Cognitive control is tiresome. Do you remember the first time when you learned how to program or how to use a spreadsheet application? Do you recall how taxing it felt to carry out a simple task, such as keeping track of a mouse cursor and remembering what to do, and when? Having to continuously decide what to do is mentally demanding and doing this for an extended period of time may cause fatigue or stress. Moreover, besides having to decide what to do, users often also act under uncertainty, which may include risk, which in turn increases stress.

The experience that follows from effort and stress is called *cognitive workload*. The perceived workload can be measured with the NASA-TLX (Task Load Index) questionnaire (see Chapter 7). Over time, with experience, people develop *minimum effort task strategies*, routines that diminish perceived workload. In practice, this often means requiring less executive control. This requirement also decreases with *automaticity*; that is, when users learn a skill, it consolidates and becomes more automatic. This permits users to build on previously learned skills to refine the skill or acquire new skills.

5.3. Memory and Learning

Memory is critical in computer use. Without memory, you would struggle to complete even simple tasks, such as filling out a registration form, changing settings, or navigate a page. We need memory to keep track, reason, recall, and recognize. However, memory does not refer only to consciously accessible memories. According to dual process memory models, there are separate neural processes for conscious and unconscious remembering [884]. What one has experienced can affect performance in an implicit way. For example, you may 'just know' – without explicitly recalling – where certain common commands are, such as 'File' in an application menu.

Current neuroscience evidence suggests that several distinct memory systems are involved in interaction, each with different functions, purposes, and neural bases [769]. It is important to understand these systems, because in almost every interactive task we have more than one in play. It also challenges a common lay conception of human memory. Memory is *not* like a storage house or hard disk where you store items and retrieve them later.

5.3.1. Working memory

Working memory (WM) refers to temporary maintenance and manipulation of representations in mind needed for action [590]. WM is set apart as a memory from long-term memory due to its time- and capacity-limited nature. We can only maintain a few items at a time in our working memory. The nature of this capacity has been the subject of several decades of research.

Early research suggested that WM is limited to only 2 to 6 items at a time. This was earlier known as ”Miller’s magical number 7 ± 2”, however the view has been corrected to a more conservative 4 ± 2. According to this view, at any given time, we can only hold about 2 to 6 items in WM; trying to hold more than that will result in losing some existing ones. Contents in WM will be lost for two reasons: first, exceeding its capacity limit, and second, not rehearsing it. Active rehearsal is needed to keep the items activated. This is what makes the use of working memory seem mentally taxing. Access to items fades away quickly unless maintained actively.

The capacity limitation of working memory can be demonstrated and measured with a simple task called *the n-back task*: Ask your friend to read aloud random numbers between 1 and 9 with a constant pace. Your task is to report back the nth item *before* the last one. To be able to do this, you must now keep active n items in your WM and update the memory contents upon hearing a new number. The task is much harder than it sounds and exposes the radically limited ability we have for simultaneous maintenance of ideas. It is best learned by trying out (see the side box).

Currently, it is thought that the capacity of WM is even smaller than what Miller suggested, perhaps just 3-5 items in young adults [170]. Moreover, WM is argued not to consist of ”slots” but its limits emerge due to the inability to maintain several active representations in associative memory networks. In other words, items in WM are not fully ”lost” when we stop rehearsing them, they just lose activation and it becomes increasingly harder to reactivate them.

WM is needed in many interactive tasks. Consider, for example, the – rather annoying – task of copying text from one application to another. You need not only remember what to copy, but also keep track of where you are in the to-be-copied text. Another example is interruptions. We get notifications and change tasks all the time. Interruptions are the more disruptive, the more we rely on working memory at the time of interruption. If there is a single takeaway to design from this research, it is to avoid relying on working memory. This can be done in cases where representations can be encoded to the long-term memory.

5.3.2. Long-term memory

Luckily, users do not have only WM to rely on. *Long-term memory* (LTM) refers to memory systems that are responsible for exploiting past experiences. Figure 5.3 shows a taxonomy of long-term memory systems. Evidence for these systems comes from brain imaging and from studies of patients with abnormal brain functioning.

Declarative memory refers to long-term memories that can be consciously experienced, or *explicit memory*, while *non-declarative memory* are *implicit memories*: They affect our behavior without conscious recollection.

Example: How do you remember your password? This is a good example of how both declarative and nondeclarative memories contribute to our ability to perform interactive tasks. For one, you can recall a password by recalling the act of typing it, a memory of moving an index finger on a keypad to enter the PIN one by one. You just know this. But you might also be able to recall the password as a word. When users are asked to *generate* a password, they are asked to think about the letters and words. Thus, they may first remember the password using semantic associations. However, for a password that we have entered several times, we may have forgotten the semantic representation of the password and rely on our ability to 'just type it'. For example, you may have noticed that if the keyboard changes, it will be difficult to type a familiar password.

The declarative system is further divided into *semantic* and *episodic* sub-types. Semantic memory is responsible for propositional knowledge, such as:

- 'Folders contain documents'
- "Save' is related to 'Open"

Episodic memory, in turn, can be thought as mental time travel: it allows us to re-experience past events. 'What did I do last time I wanted to print in A3 size?' Programmers who scroll program code keep track of where they are using episodic memory.

Non-declarative memory systems are further divided into procedural, priming, conditioning, and non-associative learning. Non-associative learning refers to reflexes, such

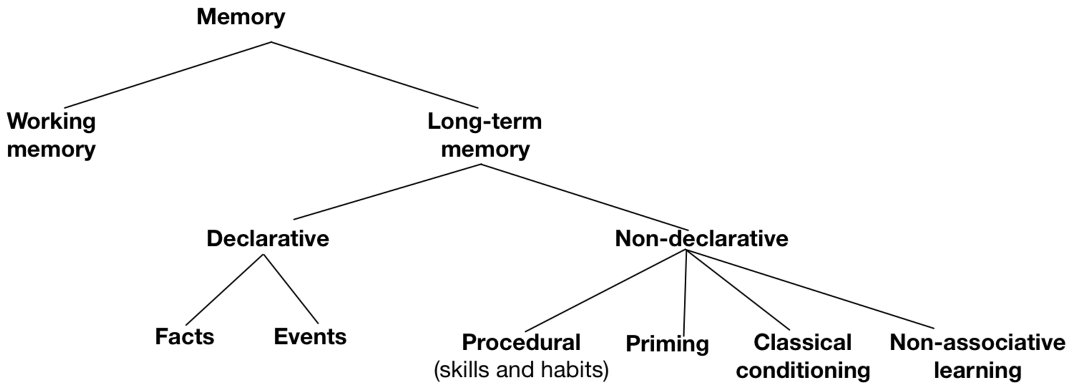


Figure 5.3.: Taxonomy of human memory systems [769]. See text for description of their roles in HCI.

as drawing the hand from a hot stove. *Procedural memory* refers to learned sequences of actions and thoughts. Procedures are like recipes. Instead of controlling actions one-by-one, the user can trigger them as a sequence. After extensive repetition, actions become associated to a larger and larger whole, which allows them to be executed without conscious awareness. For example, you may know ”automatically” the sequence with which you enter your name and password to a login.

Priming refers to an unconscious effect of previously seen stimuli on responses to a subsequent stimulus. Priming has a role in preparing us to respond in a manner appropriate to context. Contextual cues ’prime’ or make us readier to give certain responses. For example, asking a user to press left arrow button in a gaming context may lead to faster responses among gamers than doing that in a form-filling context.

Conditioning refers to learning of actions that are triggered (conditioned) by the environment. Whereas in priming only readiness to act is affected, in conditioning the probability of action selection changes. Conditioning is an unconscious association between the environment (cue) and the response. The strength of this association can be affected by reinforcement. *Positive reinforcement* is a reward given for successful behavior, and *negative reinforcement* a penalty or punishment for unwanted behavior. When reinforcement is removed, behavior can be inhibited and eventually removed. Conditioning is one mechanism that can be used for behavior-change (see Chapter 6). A positive behavior, like ceasing smoking or starting physical exercise, can be reinforced by introducing a positive reward. In gaming, positive reward can be, for example, a loot box or badge.

The multiplicity of memory systems is good for redundancy: if one system fails, the other can be used. And vice versa: Relying overly on a single system, while beneficial when circumstances stay the same, can be detrimental to our flexibility to adapt and recover from errors. Redundancy improves robustness.

5.3.3. A three-stage view to memory functioning

Time is a defining aspect of interaction with computers. Most things we do with computers take more than a few seconds, and some activities can take years. It is therefore natural to look at memory functioning from the perspective of time. Both declarative and non-declarative LTM can be understood in terms of three *stages* organized in time:

Encoding memory traces are formed during interaction;

Storage the traces are retained in-between encoding and retrieval, some are forgotten;

Retrieval the traces are retrieved at later stage, for example when using the same user interface again.

Encoding affects what we can – in principle – remember later on. On the other hand, the type of encoding depends on how much attention is paid and how deeply we process information. *The levels of processing (LOP) effect* is the finding that the ’depth’ of mental operations carried out during the task is associated with the strength of the encoded memory trace [172]. The deeper you process something that happens in the interface, the better you will remember it. Items in a user interface that are interesting are typically more deeply. Those that are just glanced at and rejected are encoded in a shallow way and more quickly forgotten. For example, when users were asked to navigate or read the content on a web page, they better remembered the corresponding elements, and almost entirely forgot the others [607]. An exciting application of the LOP effect in gesture-based interaction is given in the side box.

To later retrieve a memory trace, we need some *cue*. The cue can be something you see, feel, or do. It can (or can fail to) re-activate the original trace. Depending on the type of the cue available, three types of recall can be distinguished. First, *free recall* refers to an attempt to retrieve by self-generating cues; that is, without externally presented cues. For example, if you are deciding which programs can be used to do some task, you typically do that without having those programs visible. Second, *cued recall* refers to externally cued retrieval and is generally much easier than free recall. Success, however, depends on the cues. Third, *recognition* is an extreme version of cued recall: the whole object is presented. You need to decided whether it corresponds to a particular class or experience seen before.

Our visual recognition memory is amazingly good. You can recognize hundreds of faces you have seen during a football match, or many passwords you have used during your life. However, asking to recall them would be very difficult. This ability is exploited with *graphical user interfaces* (see Chapter 28). When we need to recall a command, such as in *command language interfaces*, we need to use free recall (see Chapter 27). This is effortful. In contrast, with GUIs, you can rely on strong visual cues and simply recognize familiar graphical elements.

There is an interesting interplay between cues and recall, and interface design affects this interplay. *Encoding–retrieval symmetry* refers to the similarity between conditions in the encoding of an item and those in retrieving it. High symmetry helps in retrieval. For example, it is harder to retrieve a password if the color of the login screen has changed.

Paper Example 5.3.1 : What makes a gesture memorable?

Gesture-based input holds great promise. However, memory is a limiting factor to gesture-based interaction. Unless gestures are demonstrated to users, users must actively recall them in order to use them.

Nacenta et al. [559] wanted to understand what makes a gesture memorable. They compared three types of gestures: 1) Gestures designed by designers for a particular application. The designers were informed about the goal of producing memorable gestures that are also easy to perform and recognizable by the recognizer. 2) Stock gestures: Generic gestures preloaded in applications. 3) User-defined gestures: Users were asked to generate their own gestures.

The experiment found that user-defined gestures are better than the other sets. User-defined gesture were up to 44 % better recalled and others. They were also perceived to be less effortful and time-consuming. Users also rated them higher in preference. The authors recommend allowing user-defined gestures when possible.

5.3.4. Forgetting

What happens between retrieval and encoding? Memories are lost. How does this happen?

According to *decay theory*, memory traces lose activation or strength over time. Events in the recent past usually have more pronounced importance for actions in the present. The more time that has elapsed since a memory was activated, the less probable it is that it can be retrieved. According to *interference theory*, memory traces become confused with each other. When we try to retrieve a memory trace, another (false) memory is activated at a similar level. Memories are not simply forgotten, they become mixed up.

However, memories are not forgotten in the same way. Encoding and forgetting adapt to the statistical structures of our environments. If all previous memories were competing for attention, we would have difficulties retrieving what we need. We need to forget, however, in a way that keeps the organismically most important memories available.

With practice, computer users become better at predicting what they will need to remember later. As we discuss in Chapter 21, memory adapts to the statistical distribution of the environment. For example, to recall a password, we need to retrieve it from long-term memory. But what determines how easily it can be retrieved? The ecological theory suggests that human long-term memory evolved to help survival by anticipating organismically important events. It is evolutionarily important to remember things that are important for survival and thriving. The expected value of remembering it in the future should be.

Paper Example 5.3.2 : Modeling how users remember where things are on a user interface

Theories of human LTM can explain how we remember, and when we fail to remember, interface elements over time. Consider, for example, the icons on your smartphone: over time, you learn to find them with increasing performance without putting conscious effort into it. How is this possible?

Long-term memory is an associative network in which nodes that encode UI elements are associated to other nodes [23]. An item in memory has a base level of activation that determines how likely it is to be recalled. It also has associations to other items, which can help or inhibit its retrieval. These associations have different strengths.

Consider a user who occasionally taps icons with some distribution. When an icon is selected, its association with its properties, such as its color and location, increases. However, afterwards, the association starts to decay. The farther in the past the previous accesses are, the less they contribute to the activation, resulting in the decay of memory. This can be expressed mathematically: The base activation of the icon B_i is

$$B_i = \ln(\sum_{j=1}^{n} t_j^{-d}), \tag{5.1}$$

where t_j is the time since the jth visitation of i and d is a decay parameter.

When retrieving the location of an item, the probability of retrieval is related to this base level. If $B_i > \tau$, where τ is some minimum threshold for retrievability, the item is retrievable. In case a cue is given that relates to the item, *source activation* B_{sa} can be added to B_i. This is called *spreading activation*: activation of whatever is in the focus of attention boosts the activation of relevant items and thereby helps their retrieval.

Finally, the time to retrieve the location of the item, T_i, depends on its activation. This is given by

$$T_i = Fe^{-fB_i}, \tag{5.2}$$

where F and f are individual-specific constants.

5.3.5. Learning over time

Every time you do something, subtle neural changes take place; These prepare you so that you can be better in the future when you encounter the task again. Over time, changes in performance are dramatic. What used to be hard and taxing becomes more automatic, and therefore more effortless. However, *how* you practice and whether you deliberately practice the task, or simply execute it affects how well you learn.

Perhaps the most dramatic improvements to performance occur early on. For example, after just a few tests with a new input devices, we typically see large changes, which then become proportionately smaller with more practice. Another way to say this is that increasing practice provides *diminishing returns.*

In motor learning, this relationship is captured in *power law of practice*:

$$RT = aP^{-b} + c \tag{5.3}$$

where RT is reaction time, P is the number of practice thus far, a, b, and c are free parameters; in other words, they obtain their value in a case-specific way. The law portrays a quantitative relationship between the best-effort performance and the amount of practice. It has been found to describe motor tasks, such as pointing as well as rolling cigars, but also to some extent mental performance, such as retrieving facts or solving arithmetic tasks. The free parameters (a, b, and c) are case-specific and control different aspects of the slope: intercept, rate of change. An example is given in Figure 5.4.

When performance improves, it is not only the expected reaction time that changes. Performance also becomes more *stable.* In other words, the variance in performance decreases. In some cases, this trend has been observed to follow power law of practice, too, however, with different free parameters. The moral is that early on when learning a skill, we struggle to keep performance stable and see large differences across trials. Large improvements are seen during the first trials, and performance stabilizes further, albeit with diminishing returns, with more practice. Stability is important in HCI, because it means that the user can better predict the outcomes of interaction, is less prone to errors, and has greater self-confidence.

With extensive practice, skills transition from controlled to automatic. *Automaticity* refers to fast and effortless performance of skills. For example, when using the camera application on a smartphone, you probably first struggled to find settings and even basic

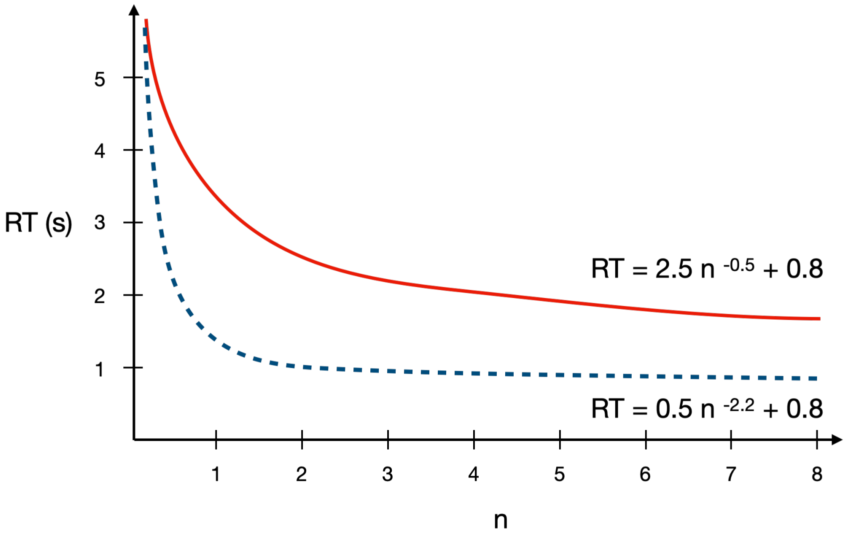


Figure 5.4.: The power law of practice states a relationship between reaction time and number of practice: $RT = aP^{-b} + c$, demonstrated with two parameter sets. Which curve shows faster learning? What do the three free parameters (a, b, c) mean?

functionality like recording a video. However, after a while, you can focus more on the object of photography than on the camera itself. The downside of automaticity is its inflexibility. Automaticity is often described as a ballistic skill: the skill is triggered by some cues and carried out to its completion with little opportunity to redirect it. This means that with increasing practice, users may become insensitive to changes in a design.

5.3.6. Designing practices

In many settings, we are not just using computers, but invest time and effort to practice their use. *How* we practice matters. The most inefficient way to learn a skill is to just do it, 'mindlessly'. Significant gains to performance can be obtained with some thought into training.

The basic parameters of any practice regime are:

1. Scheduling of practice: When practice happens
2. Selection of practices: Which practices are done
3. Feedback: How feedback helps inform change in strategy

Scheduling practices : How practice trials are scheduled over time has a significant impact on learning. The worst schedule is *massed*:

| | | | | | |

Here, all practice takes place in a compressed timeframe, which leaves very little time for consolidation of memories in the brain. Somewhat better is *equi-spaced* training:

| | | | | | |

Here, it is ensured that practices take place not too long after each other but not too shortly either. The best training schedule is *expanding practice*, here shown with an exponent of 2:

| | | | | |

This means that practice will take much longer. However, it gives much more time for slower learning processes in the brain.

Focused practice : Perhaps the worst way to learn a skill is to perform the activity. Focused practice refers to the selection of *isolated* practices with clear objectives and corrective feedback. These practices should not be any, but focus on those aspects that have room to improve. . But how to identify such practices? This can be done by a human mentor, but also by tracking the user's development and comparing against expected level of performance at that level of training.

Feedback : Feedback is important as it helps set goals and correct mistakes in computer use. Consider any computer game you have recently played: What kind of feedback does it provide about your performance? For example, it can provide a numerical score or indicate which level you have mastered. It could also be more specific, for example telling how many events of particular kind you have mastered. Feedback can be of three different kinds: (1) performance indicating, for example, a score or words per minute in text entry, (2) performance-correcting, providing qualitative feedback to help identify problematic areas of performance, and (3) strategy-enhancing.

5.3.7. Stages of skill development

How good do you think you are at using your mobile device? It is probably hard to tell. Skill development in HCI is often described in three stages: Novice, Intermediate, and Expert.

The novice stage is characterized by a struggle to complete tasks and high variability in performance. Performance is neither high in comparison to peers nor stable. However, while performance improvements can be obtained by repetition, in complex tasks, this may not suffice. After initial gains by repetition, the next improvements are achievable only by changes in *interaction strategies*. Learning is very specific to the strategy. For example, if you first use a menu for selecting commands, and later shift to keyboard shortcuts, there is only little if any transfer of the earlier skill to the new one. Any change in strategy may initially reverse previous advances in performance. Users may initially perform much worse than before the shift. However, performance will improve quickly, and in some cases according to power law of practice. Power law of practice can be used to estimate whether the new strategy will allow eventually obtaining superior level of performance.

Most of our computer skills are self-taught. Without deliberate practice, development thus arrests at the intermediate level of skill. This level is characterized by a level of performance that is acceptable for most regular activities. However, the acquired skill may not be robust, it may not generalize to new tasks. For example, in a study of mobile users, the skills obtained by intermediate-level users were found to be specific to the device [610]. They mainly learned their skills via early familiarization with a device, repeated use, and problem-solving situations. While experienced users exhibited superior performance for tasks, they failed to transfer that task due to lack of deeper conceptual representations of how the technology works. Instead, their performance was attributed to being better able to navigate and know where 'things are' in the interface.

The highest level of skill, the expert level, is only achievable with years of deliberate practice. That is, significant amount of practice, on the order of thousands of hours rather than hundreds, is required. This design of this practice matters and is normally done by coaches and other experts, for example in the case of video games. Practices should be isolated, focus on weak areas of performance, provide corrective feedback. Because such practice can be effortful, motivation for persistence is important: How to keep the user motivated even when practice is not fun?

The role of coaches, mentors, and exemplary users is important. In *apprenticeship* *learning*, an expert user guides the practice. Consider, for example, a medical doctor teaching each novice users how to use a new patient record system. An experienced user can involve the selection and scheduling of practices, provision of feedback, and motivation.

5.3.8. Knowledge in the world, knowledge in the head

Finally, users are not at the mercy of what they remember. They can flexibly *externalize* knowledge and use such externalized representations in service of recall. The essential distinction is between *knowledge-in-the-world* and *knowledge-in-the-head* [283]. For example, to avoid disclosing our passwords to others, we would like to store and maintain them in long-term memory, safe from peering eyes. This is knowledge-in-the-head. Alas, as everyone knows, we often store passwords on slips of paper, files on computers, etc.; knowledge-in-the-world. Our cognitive functioning has become dependent on us having access to that environment.

People also actively manipulate the environment to change the cognitive requirements of a task. The case of Tetris players is an eye-opening one. In a study of expert Tetris players, it was found that not all of them try to mentally rotate a piece to fit it to the landscape below [410]. Instead, what they do is to quickly flip the piece to *visually recognize* the best slot. This is beneficial because we are fast in recognizing solutions but much slower in mentally simulating events.

Distributed cognition can be very involved in complex task environments. Consider an airplane and a pilot and a co-pilot operating the cockpit to land the plane [357]. There are numerous computer devices showing the state of the airplane. There are standard operating procedures, checklists that show what to do and in what order. Moreover, there are two pilots, who by communication establish common ground on the status of the airplane. Speech acts, like telling the status of a checked meter, can update the state of the partner. These devices, practices, and communications are *together* a distributed cognition. The idea is that we cannot attribute cognition to a single brain, but its functioning is best understood in the joint operation of the pilot and its environment.

5.4. Reasoning and Decision-Making

We often face a situation in interaction where we do not have a direct solution to something that we need. We cannot see anything or recall anything that helps us proceed.

Reasoning is about thought processes that allow us to conclude something that we do not already know. Reasoning forms new beliefs from old rules via some rules or mechanisms. For example, if the browser window has not yet opened, you may reason that the application has not yet been launched, or that the tap was not registered. Another form of reasoning is *inference*, where we form a new belief based on observations. For example, we can – via inference – tell that an icon in a different operating system, say Mac OS X, will have the same function as the one we have seen in a Windows computer.

Prediction is a special kind of reasoning that is needed to act in a dynamic and changing environment. Predictions are reasoning about the future. The challenge there is to leap from observations (of the present) to say something about what *will* or *might* happen.

Reasoning is commonly needed when dealing with complex systems, for example:

- Reasoning about what a piece of software can do based on other knowledge about it, which may come from various sources such as advertisements, word-of-mouth, etc.
- Reasoning about what will happen if you issue a particular command to a software that you have not done before.
- Reasoning about risks involved with a decision related to use.

5.4.1. Mental models

Mental models are memory-based representations of interactive systems that are used for reasoning, inference, and prediction. They are representations of systems and the way users' inputs affect them. For example, a mental model of a thermostat may tell how the temperature set by the user affects the felt warmth in a room. By simulating in mind using a mental model, a user can reason about unobservable qualities of a system.

In a study of mental models, Mayer and Gallini [513] asked students to read descriptions of how a pump works. These descriptions had no illustrations, or they showed parts, operating steps, or both parts and steps, as shown in Figure 5.5. They found that the parts-and-steps illustration significantly helped the participants in both recall and problem-solving tasks. This finding indicates that mental representations of devices should allow users to simulate what happens to the internal state of a device when a certain input is given.

The current understanding is that regular untrained users are often unable to form coherent and complete mental models of the devices they use. Mental models are rarely complete models. Instead, users' knowledge tend to be multi-faceted and fragmented. For example, one may remember episodes of using a device, and use this to recall how to operate it. One may also remember unrelated facts about what a device can do. In practical problem-solving situations, users are often reluctant to engage in effortful reasoning and rather just try out things and see what happens.

5.4.2. Decision-Making

Decision-making refers to any situation in which a number of options is given and one or a subset must be chosen. Example: *Your computer is infected with a virus. Which virus removal application would you pick?*

- Software A: Removes the virus with 95 % probability, however slows down the computer significantly in 5 % of cases;
- Software B: Removes the virus with 70 % probability, but never slows down the computer

In the more general case, N options are given, each associated with gain g (e.g., removing the virus) that occurs with probability p and a loss (e.g., slowing down the computer) that occurs with probability $1 - p$. However, users may not be "rational" and always choose the option that is best for them.

Prospect theory posits that such choices are evaluated relative to a *reference point* [395]. For example, the status quo (most common choice), or the most recent choice may work as such a reference point. Such reference points guide how they reason about the gains

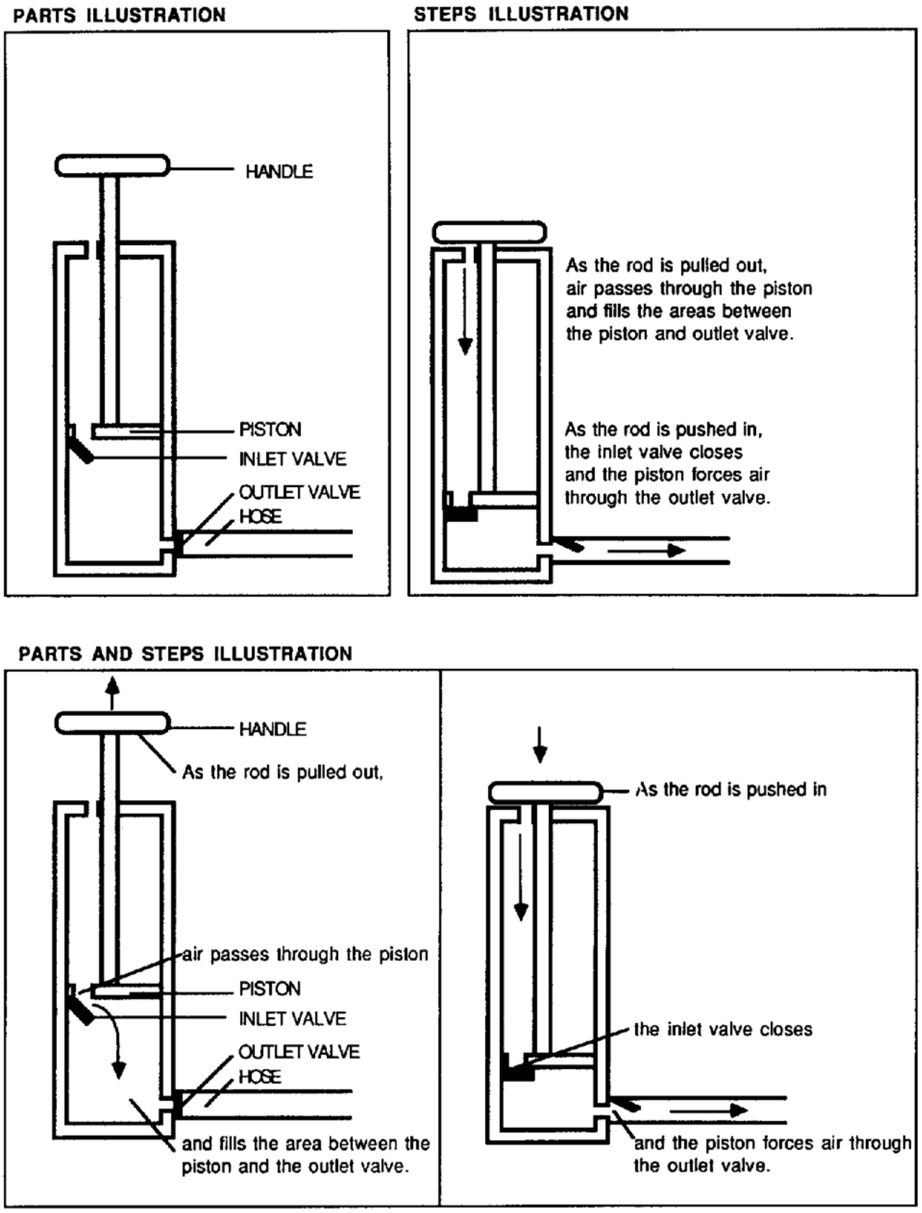


Figure 5.5.: Mental models of a pump system used in the experiment of Mayer and Gallini [513].

and losses involved. The second claim is that some people are *risk-averse* about gains – when compared against the reference point – but risk-seeking about losses. In other words, they do not want to lose the gains provided by the reference point, but they are willing to gamble with losses. The third claim is that some people are loss averse. In other words, losing something hurts more than gaining the same thing. For example, loss-averse gamers prefer an option that avoids losing a precious award obtained in a game, and weight this relatively more important than the possibility of gaining it.

This theory is perhaps the most successful theory of economic decision-making, applied in studies of bargaining, consumer choice, voting, and even politics. Also in HCI, users face decisions like this all the time: How should we choose clothes for our avatar, privacy settings, or the length of our passwords. These come with an aspect of expected gain, loss, and risk. Although many decisions in HCI, in principle could be analyzed like this, the challenge is how to estimate the gains and losses and their probabilities.

Multi-attribute choice refers to choice in the case where options are characterized by multiple attributes. By contrast, in the previous examples, there were only two attributes to consider: efficiency of virus removal and effect on the speed of the computer. We can generalize to more than two attributes. For example, if you want to buy a bicycle on an e-commerce site, you need to consider multiple attributes, such as price, type, size, location, etc. Multi-attribute problems are hard for users, especially novices [480]. Their eye movement patterns are scattered, they can spend a lot of time going back and forth inspecting options, and they may feel unsatisfied with their eventual choice. Why is multi-attribute choice hard?

Users need to move (limited) visual attention between the rows to examine the options. While doing that, they need to keep in mind the attributes they have seen so far, or at least those attributes important for making a decision. They also need to reason what was the best option and why. Few people can do this in a single pass, because keeping all these attributes in mind is hard. Rather, what users often do is ”satisficing”: They decide on what would be a good enough option and select that one as soon as they find it. To circumvent the limits of working memory, users may also write down options, for example to a spreadsheet, or seek external advice (reviews, opinions from friends). However, with experience, the problem becomes easier. Eye-movement strategies become less taxing and faster. Users learn to scan the options in more systematic way and directly reject options that are irrelevant.

5.4.3. Decision heuristics

Kahneman [393] popularized the finding in neuroscience that we have two systems of decision-making. ”System 1” is a fast system driven by intuition, emotion, imagery, and associative memory. ”System 2” is a slow system that can monitor System 1 and intervene if intuition is not sufficient for the task. For example, you may feel like you want to click on a social media application to check the latest posts (System 1). However, as you start doing that, your System 2 may intervene to stop that from happening, as you recently did.

In many circumstances, we do not have the possibility to evaluate all options, or even if we did, the situation was too complex for full analysis. In such cases, we use System 1. A *cognitive heuristic* is a rule of thumb used to identify a quick solution to a complex problem. For example,

Anchoring occurs, when we center our choice around a known reference solution. For example, when choosing which application to use for some task, users often start from the application they already know.

Decoying occurs when a reference point we have prevents us from seeing another one 'behind' it. For example, a user who recalls 'ChatGPT' as an LLM (large language model) to ask a question from may fail to remember the names of other LLMs.

Availability heuristic refers to people's estimates of the probability of an event. Similar to the anchoring, the availability of a memory makes it more likely to be entered as a solution. Powerpoint, because of its prevalence as a slide editing software, may be the most likely option users consider even if there are many alternatives.

Status quo bias refers to the prevailing solution, such as a famous or popular option. For example, popular products like ChatGPT may first come to mind.

The bandwagon bias occurs when we see our peers following an option.

For example, if I ask you to name a great smartphone, you would be more likely to name something that quickly comes to mind, and the most recent design may be one of those. This heuristic is called the *availability heuristic.*

The catch with heuristics is that, while they allow us to generate a solution quickly, they limit the visibility of other solutions. They lead to *biases.* A bias, in this context, means a tendency to consider a skewed subsample in the space of options.

5.5. Simulating cognition in interactive tasks

Having described aspects of cognition relevant to HCI throughout this chapter, a natural question is how we would go about applying it to understand interactive tasks. *Cognitive models* are models that describe formally what happens in a person's mind during an interactive task. They provide an exact description of a cognitive mechanism, a way in which cognition mediates observable user behavior and the user's task.

A cognitive model receives some input that represents the task, and has some mechanism to link that input to predictions about behavior. A model can be expressed in different ways:

Rule systems Sets of rules or logical clauses that describe how information is processed with this in mind. For example, mental models are described as logical statements.

Mathematical models Statistical models that describe a relationship between factors relating to the task or design, the outcomes in the interaction, and the cognitive factors that mediate the two.

Simulations Step-wise executed computer programs. The best-known example is a class of cognitive architecture models, where different cognitive capacities, such as perception and memory, are program modules with their own inputs and outputs.

Data-driven models Models learned from training data, or via trial and error. For example, artificial neural networks can be used to understand aspects of visual saliency. Although such networks learn parameters from data, they also express architectural assumptions about the mind, such as the process of convolution.

In the early years of HCI, cognitive modeling had a central role in HCI [129]. One well-known example is Project Ernestine. Another example is GOMS (Goals, Operators, Methods, Selection Rules), which was an early cognitive model developed by Card et al. [129], which helped the development of cognitive architectures.

The development of cognitive architectures are inspired by computer architectures, which are descriptions of computer systems in terms of their underpinning components and their relationships. Cognition consists of serial and parallel information processing units. This terminology is familiar to anyone who has studied computer science or software engineering:

Peripherals These are sensors and actuators for interacting with the external environment. Examples include the oculomotor system, perception, and hearing.

Internal modules Such modules may operate independently, processing input, and producing outputs to other modules. They have capacity limitations, such as the number of items they can store or amount of time they take to process input, and they can maintain internal states and run complex programs.

Production rules These describe what is done and under which conditions. They are similar to computer programs that consists of a series of commands executed by modules.

A central processing unit The central executive is limited in capacity and can only process things serially. When a program is run, it produces a *trace* of human behavior. Such a trace may contain the time it takes to take an action or complete a task, the errors produced, and so on. Some models even predict blood oxygen levels in the modules.

One of the main motivations for cognitive architectures in early HCI was that they could be used to decrease the cost of empirical evaluations with human participants. In addition, they were also used to construct systems that adapt to the user, such as in cognitive tutoring, where cognitive models track a learner's cognitive development to select suitable study materials and interventions.

Why have cognitive models not been as widely adopted by practitioners as perhaps hoped at the outset of the field? First, applying a cognitive architecture model to an HCI problem requires a deep understanding of the user's task. Sometimes this understanding has to be so deep that it is easier to evaluate the design empirically than to build a model. For example, GOMS requires the researcher to specify rules, a production system, describing what the user does at the level of cognitive operations. Producing such descriptions can take weeks, or even months. Approaches aiming to lowering the barrier by allowing practitioners to demonstrate tasks were not widely adopted, perhaps because they still require a large amount of work [381].

Second, cognitive models tend to work only within a narrow range of applications. A production system that has been created for one task and one particular design cannot be readily adopted for another setting—it must be updated and revised. In other words, cognitive architecture-based models have a scaling problem.

However, the topic of cognitive modeling is experiencing a revival with machine learning based methods. New modeling approaches have been explored in HCI, most recently neural network models and reinforcement learning (Chapter 21). a key benefit of such approaches is that the researcher does not need to specify the production systems, or policies. Instead, such models can learn them.

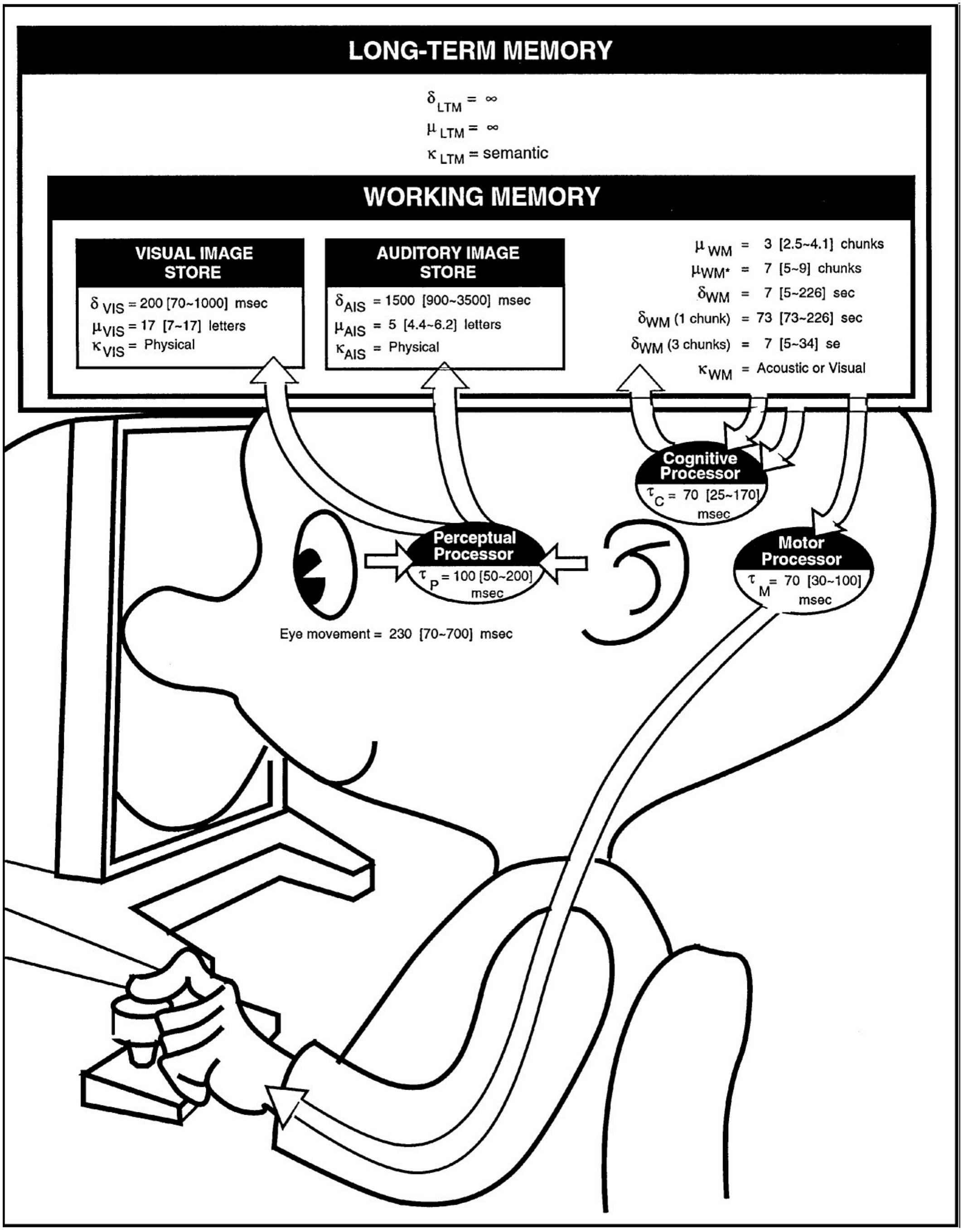


Figure 5.6.: GOMS is a cognitive architecture model of cognition. It simulates the processing of information in separate modules of mind, and the flow of information between them. Modules have internal capacity limits and processing times Card et al. [129].

Paper Example 5.5.1 : Project Ernestine

Project Ernestine is an example of cognitive modeling in HCI [284] that demonstrate the practical value of modeling cognition for HCI.

A telephone company was interested in improving the efficiency of their teleoperators. Teleoperators, a profession that does not exist any longer as switchboards are automated, were professionals who spoke with a caller and connected the call to its destination.

The company had designed a new workstation that they believed would improve task performance and help the company save millions of dollars. They hired cognitive scientists to understand if the new design is better than the old one and, if so, by how much. The practical goal was simple: to estimate task completion times for the two workstation designs.

The researchers came back with a startling argument: the new design is slower than the old one, despite the company's best attempt to improve it. The researchers pointed out that while the company had focused on the time that manual operations take, the new design changed the cognitive requirements, with non-obvious consequences on performance. The new design required the recall of information, which is slow, and this factor was neglected by the company.

To arrive at this conclusion, the authors used a variant of GOMS called CPM-GOMS. Figure 5.6 shows the cognitive architecture. CPM-GOMS predicted that the new workstation was 3 % *slower*, not faster, than the old one. They further predicted that performance varies across categories of call-handling tasks. Thus, they predicted, the company would lose money if they switched to the new workstation.

To ensure that the results were trustworthy, the authors validated the results against empirical data they collected after the deployment of the new workstation. Comparing model predictions against real data on calls collected over four months, they learned that the average prediction error was small, varying between 11 and 12 % in performance time, which corroborated their modeling approach.

Project Ernestine is an example of the special character of modeling in an applied field as HCI. Modeling in HCI is often by practical needs in evaluation and design of systems as opposed to a pursuit of knowledge for rigorous theory. Cognitive modeling can sometimes be used to evaluate designs, and provide insights into their cognitive requirements.

Summary

- The main function of cognition in interaction to help users deal with complex systems and situations.

- Cognition is limited, yet learning and adaptive.

- Theories of cognition help understand several key questions of information-rich task environments, for example what makes multitasking situations demanding, what happens when users learn to use a user interface, why they may fail to remember something they have seen before, and how they draw conclusions based on their beliefs about systems that are opaque to them.

- Many of HCI's guidelines and evaluative models are rooted on theories of cognition.

Exercises

1. Cognitive abilities. Measure your or your friend's cognitive abilities using PsyToolkit: `https://www.psytoolkit.org`. Carry out tests to assess working memory capacity (e.g., n-back), cognitive control (e.g., task switching), and reasoning (e.g., Hanoi tower). Repeat each test a few times to obtain a reliable measure. What are some tasks in computer use where these abilities are important?

2. Cognitive load. Cognition requires energy and effort, but how does design affect that? Take a computer game that you find mentally taxing. Ask a friend to play it and administer the NASA-TLX questionnaire after each level. Do measurements a few times for more robust estimates. Plot the components of NASA-TLX as a function of the game level. What were the aspects in the game that made each level less/more effortful?

3. Multiple resource theory. a) Name two common tasks in mobile interaction that, according to MRT (Multiple Resources Theory), would interfere each other when carried out simultaneously, and illustrate the conflict by annotating the MRT cube given in Figure 5.2. (b) Explain in your own words why interference occurs. (c) Explain how either of the tasks might be changed to reduce interference.

4. Observational study of human error. When was the last time you forgot your PIN code, missed your turn when driving, or forgot to unmute yourself in Zoom? Observational studies are an important part of the HCI toolbox. To prepare for this task, read more on error taxonomies based on human cognition: `http://www.errordiary.org/blog/wp-content/uploads/HumanErrorSGSWMarch2013.pdf`. Pay attention to the notions of "activation", "schema", and "cross-talk" to fully understand these concepts. Download the error-table Word file from the homepage of the book. It contains a table with these error types on rows: (1) Capture, (2) Double capture; (3) Omission; (4) Loss of activation; (5) Description slip; (6) Associative activation slip; (7) Repetition of action slip; (8) Cross-talk slip. Your task is to observe some everyday interaction taking place with technology for one hour and to systematically identify errors using this taxonomy. We recommend a public interface similar to a vending machine. Ensure that 1) you can safely observe users, for example, standing from a distance and taking auditory or written notes, and 2) the case is interesting from an error point-of-view; that is, you believe that more than a few of the different error types might appear.

5. Cognition and design. Pick a user interface design that you think is hard to use. Take screenshots of central screens. Analyze them using the design guidelines of Jeff Johnson listed in Table 41.1.

6. Memory. Human memory is important in interaction. Think about a concrete example of an interactive system that goes against the characteristics of memory that the chapter discusses. It could, for example, be about demanding too much of our short-term memory. Propose how the design could be improved.

6. Needs and Motivations

From the word processor to the Internet, interactive computer systems have changed the way we work, consume, and even lead our romantic lives. However, in addition to technical limitations, are there *human-related* limits to a wider adoption of computing?

What we *need* or desire as human beings is a defining factor in technology adoption and success. At a high level, technology makes inroads into society in two ways: *market-pull*, which means that there is a market need for a solution; and *technology-push*, which means that new technology creates a new need in the market. However, it is people and their needs that shape *both* pull and push factors.

The scientific study of human needs has become increasingly important in an attempt to understand the role that computing might play in our lives. In the early years of computing research, the question was what was *possible* to achieve with computers. This has now changed to identifying what is *desirable* for people. This chapter discusses human needs and motivations as they relate to interactive technology.

Needs are generally understood as requirements to live a fulfilling and healthy life. The satisfaction of needs is linked to positive experiences with the activities that mediate them. If it is possible to satisfy a need of fundamental importance, then it is likely that there are resulting positive effects. In that way, needs shape what we do and strive for, and the fulfillment of needs is our motive for engaging with technology.

These insights can be put to use in HCI. Let us consider a few examples.

- Hassenzahl et al. [318] noted that feeling related to other people, such as in love, closeness, and intimacy, is important for well-being. They then identified different user interface strategies that could be used to promote feeling related, including giving gifts, sharing memories, and being aware of one another. All of these considerations depart from the psychological need for relatedness.

- How to motivate people through interactive systems remains a challenge. Naqshbandi et al. [560] used self-determination theory (see subsection 6.2.3) to improve the motivation for volunteering. In particular, they attempted to use gratitude to increase motivation.

- How can interactive systems help people become the person they desire to be? This appears to be a basic need related to autonomy and meaning-making. Zimmerman [908] drew on product attachment theory to create design patterns to create such systems. For example, he created the Smart Bag, which helps parents transfer the responsibility to their children to pack clothes and equipment for athletics. In this way, parents are supported to increase the autonomy of their children.

First, let us look at the basic psychological needs that psychology research has identified and what they can be used for. Then we move on to discussing motivation, that is, what makes people act or not act. One important application of that HCI has been for changing people's behavior, which we discuss towards the end of this chapter.

6.1. Psychological Needs

A common lay concept defines a need as *a deficit*: People need something they do not have. According to this view, interactive systems must strive to create conditions that satisfy unfulfilled needs. However, this reasoning is actually circular. We need something because we do not have it. However, since we do not have it, we need it.

To make it possible to reason more usefully about needs, we need to *ground* these needs to something outside just the needs themselves, to factors that drive us as biological and social beings. This grounding has been a central theme in psychological science for almost a century. As a consequence, researchers have compiled and empirically verified several taxonomies of needs.

In this regard, computer use is not about satisfying deficits. However, satisfaction of needs positively contributes to wellbeing and positive experiences. For example, using social media can be both disturbing to social relationships and a way to satisfy various forms of inter-relatedness. Basic psychological needs are necessary for our thriving and well-being. However, as we learn in this chapter, they manifest themselves in complex ways as motivations, wants, and desires.

A central finding in psychological research on human motivation is that there are *basic psychological needs.* Such needs are experienced in many forms. Some may want to talk to their grandchildren, others like to help their friends move, and still others like to leave letters of appreciation. However, these needs are similar in that the aim of people is to feel *related* towards other people. Thus, we say that relatedness is a psychological need that is satisfied by these intentions or actions.

Perhaps the most well-known variant of this idea was Maslow's hierarchy of needs. As part of a larger humanistic psychology, Maslow posited five needs organized in the shape of a pyramid [508]. From lower to higher:

1. Physiological needs, such as food, water, warmth, rest;
2. Safety needs, such as security;
3. Belongingness and love needs, pursued via intimate relationships and friends;
4. Esteem needs, pursued via prestige and feeling of accomplishment;
5. Self-actualization, pursued by achieving one's felt potential and via creative activities.

The idea here is two-fold. First, the specifics of human needs can be grouped into exactly these five types. Second, the needs that a lower in the hierarchy (with a lower number) above must be satisfied before individuals can attend to the higher needs.

Psychological need	Explanation
Relatedness	Need for social relationships
Meaning	Need for purpose and direction
Stimulation	Need for novel sensations and thoughts
Competence	Need for the ability to perform well in important activities
Popularity	Need for recognition by others
Security	Need for protection of self from harm

Table 6.1.: Six psychological needs that are relevant for interactive systems [317]. They were selected from 10 needs developed in earlier work [743].

Maslow's needs have not received much attention in HCI. Two other frameworks have been influential: the ten psychological needs proposed by Sheldon et al. [743] and adapted for HCI by Hassenzahl et al. [317], and the three needs proposed by Self-determination Theory [181]. We will discuss those next.

6.1.1. Catalog of psychological needs

Sheldon et al. [743] worked to integrate and empirically validate a catalog of basic psychological needs. This work resulted in a list of ten needs. In the context of HCI, Hassenzahl et al. [317] has suggested that rich experiences with technology can be based on a much smaller set of universal psychological needs and that satisfaction of those needs is the reason why interaction with technology is generally a positive experience. This list of seven needs is shown in Table 6.1.

As an example, the need for meaning may be assessed by asking people to which extent they engaged in an experience to "becoming who I really am", or have "a sense of deeper purpose", or obtain "a deeper understanding of myself". The degree to which a person agrees reflects the degree to which the motivation to participate in the experience was a need for meaning. One may also ask about the extent to which an experience did, in fact, fulfill the need for meaning using the same questions.

6.1.2. The three needs in self-determination theory

Self-determination theory is the most prominent theory in HCI in terms of needs and motivations. It assumes three basic psychological needs:

1. **Autonomy**: the sense that actions are performed willingly, in alignment with one's self, and not directed by external forces;
2. **Competence**: the feeling of achieving mastery and controlling the outcomes of action;

3. **Relatedness**: the sense of reciprocal belonging in relation to other humans.

A *basic psychological need* refers to an ultimate biologically determined driver behind the behavior. Basic needs – as opposed to learned quasi-needs and motivations – are shared by all humans across cultural, economic, or societal circumstances. Any activity that satisfies a basic need can lead to intrinsic motivation and better well-being. Optimal psychological functioning requires that all three needs are met. To this end, people need the support and nurturing of the social environment.

What is the role of basic needs in the use of technology? The most important implication of the theory is that computers *per se* are not needed. A computer is a tool; it is a means to an end, and not an end in itself. However, to answer that question more deeply, we first need to understand how basic needs develop into motivations to act in a particular way.

6.1.3. Using Needs in HCI

How may HCI researchers and practitioners use an understanding of psychological needs? The primary reason is that need satisfaction gives energy and leads to positive affect. Need thwarting may lead to lack of energy or even negative experiences.

There are two basic uses for the needs in research and practice. Psychological needs may be used *for analysis*. That is, a model of needs may be used to analyze the results of user research. We discuss such research in Part III, but for now, imagine for instance interviews with or observations of users. The insights from such research may be related to the set of needs so that it is clear which basic needs have been discussed and which ones still need to be addressed (or at least considered) by designers of an application. Thus, needs, as an explanatory construct, enables us both to avoid solutions that do not satisfy any needs and to detect new opportunities for improving people's lives by fulfilling previously undiscovered needs.

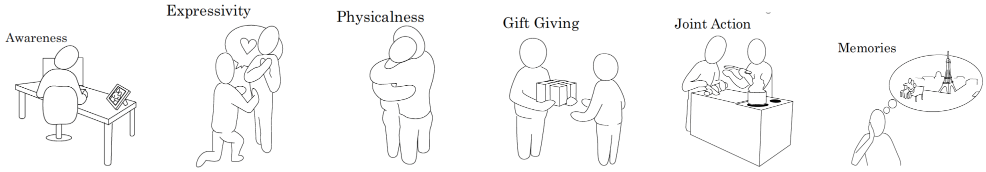
As an example, Kraus et al. [425] conducted semi-structured interviews (see Chapter 11) to understand users' motivations for doing privacy and security actions on their smartphone. The analysis showed that other needs were also determining which actions users took to be secure and private. For instance, people would switch off their wifi and data connections. This was not to be secure but is rather about the need for autonomy ("I want to be left alone") or saving money ("I can save some of my data contingent"). Thus, the specific actions taken can here be understood better from the perspective of a variety of needs, rather than being understood as all being about the need for security.

The second use of models of psychological needs is to *inspire design*. The idea here is that models of needs are assumed to be the source of positive experiences with technology. If that assumption is true, we can explore models of needs as a way to consider how to make experiences positive. The paper example below concerns an example where the need for relatedness is used to drive in design solutions.

Paper Example 6.1.1 : How to make couples apart feel related?

Due to work, pandemics, and other circumstances of life, partners are sometimes forced to spend longer periods of time physically away from each other. Although couples can send emojis and make phone calls, many researchers have been interested in supporting these situations better.

Hassenzahl et al. [318] focused on relatedness experiences as a fundamental psychological need that needs to be supported by technology in such situations. To figure out how to support this need, the authors surveyed more than 140 published designs. Among those, they identified six strategies for supporting relatedness (see below).



Each strategy is associated with examples of design that implement the strategy, with particular implementations of the strategy, and with a set of psychological principles that work for the strategy.

For example, awareness as a strategy for promoting relatedness means that an interactive system shares information about what one's partner is doing or feeling. This is a passive sharing of information but nevertheless one that promotes relatedness. The psychological processes that designers could further draw on in creating awareness are several. The self-disclosure through technology should be gradual. Several of the designs that Hassenzahl et al. [318] evaluated showed that such awareness can be intrusive and strange if it happens too quickly or to a too large degree. Further, being aware through technology works against an idealization of the distant partner. This has been found to happen in long-term relations because mundane details about the partners' lives are not available and because partners take extra effort when they are actually together. However, technology for supporting relatedness could support the building up and maintain of a realistic picture of one's partners.

The other strategies similarly contain details on psychological processes to help create technology that supports the need for feeling related.

6.1.4. Needs and Values

Some approaches to HCI focus on values rather than psychological needs. This is the case for value-sensitive design [254]. Let us briefly outline this approach and discuss its relation to research on needs.

In value-sensitive design the values of stakeholders are continuously taken into account over the course of a design process. For instance, value-sensitive design identifies autonomy as a central value. Friedman [253] discussed the concrete considerations in the design of a workstation for autonomy as follows.

> The workstation was designed to support speech input and multimedia, and thus included a built-in microphone. Nothing strange here. Except that the

> microphone automatically recorded audio information whenever the workstation was on. Now, imagine that you are in the middle of a video-conferencing session and a visitor comes into your office, or the phone rings. The only way to ensure audio privacy is to turn off the application, which is a cumbersome solution. Alternatively, a simple solution existed in the design process (ultimately vetoed by the design team): to install a hardware on/off switch on the microphone at the cost of 25 cents.

. This example shows how a focus on basic values, like autonomy, can inspire design considerations.

What is the difference between the view of needs just discussed and value-sensitive design? Values, as understood in value-sensitive design, are about what is important to people in their lives, in particular regarding how they should act (i.e., morality). Such values are numerous and include welfare, trust, informed consent, ownership, and many others. In that way, values are conscious choices about how to realize our needs. Nevertheless, values may be used in similar ways to needs in the evaluation and analysis of interactive systems.

6.2. Motivations

If behavior is driven by the same basic psychological needs, why do people differ in their behavior? How we choose to behave in a particular situation is often driven by an *anticipation* of the fulfillment of such needs. *Motivation* for using particular interactive systems is related to the needs we anticipate these systems will fulfill.

In general, *needs* are agreed to be universal, shared across cultures and people. On the contrary, motivations are individual and contextual expressions that drive behavioral choices. Although needs are more rigid and evolve slowly, motivations are more malleable. Motivations are the cogs that link the need to actual actions, since they define why we choose to do one thing and not something else. Needs explain why people universally reject certain types of technology —consider for example surveillance. Motivations explain why technology that becomes adopted in one user group in one context is rejected in other circumstances.

Such motivations may be affected by technology. For example, it is easy for a user to become distracted by an incoming notification or a tempting news feed on a social media website. However, technology may also help us stop smoking or start exercising by affecting our motivations. As we experience technology and hear others talk about it, our anticipations and motivations change. Over time, people and context diverge in the way they manifest the underlying basic needs.

6.2.1. Wants and Desires

Desires and *wants* are felt cravings for things that one may not already have. In the above, we have learned that needs are more basic; they may not be the same as our felt desires and wants. What people say they want may be very different from what they find satisfying to actually do. This discrepancy is essential in user research. A naive approach is simply to ask people what they want. if there is something a decade of research in psychology has taught us, it is that introspection does not offer special access to the causes behind behavior. Asking people simply does not work.

6.2.2. Quasi-needs

But how is it possible that wants and desires can be ”incorrect” or ”incomplete”? The reason is that they are learned. A *reinforcer* is an event that affects the satisfaction of a need. *Primary reinforcers* are basic psychological needs and biological necessities, like food, drink, and sexual pleasure. However, most of our reinforcers are *secondary*; that is, learned. For example, money or grades are reinforcers that are not directly satisfying a biological or psychological need. Instead, they are learned via *associative learning.* From experience, primary reinforces are linked to secondary reinforcers, which themselves are individually and culturally shaped.

Importantly, because motivations are based on learned associations, they can also be ”wrong”; that is, ultimately detrimental to well-being. A *quasi-need* refers to a statement of need that appears like a need, but is not one. For example, the urge to acquire a loot box in a computer game is not a need in a psychological sense, although it may appear very real and potent to the person who identifies it. Although reinforcers can be observed directly, they are not the same as motivations.

6.2.3. Motivation in self-determination Theory

Proposed by the psychologists Deci and Ryan, Self-determination theory (SDT) is one of the most successful broad theories in psychological science [181]. It is widely used in HCI, too. Unlike Maslow’s theory, which assumes that behavior is driven by minimization of deficits, SDT starts with the idea of people as active organisms pursuing self-growth, mastery, and fulfillment. The theory considers both practical as well as wider cultural and political conditions as they affect motivational dynamics (see below).

Equipped with new data and theories, psychological science has steered away from deficit needs and reinforcers and turned to *motivational dynamics* that underlie positive development, or self-growth. *Self-growth* refers to the motivational mechanisms that drive us to improve our ability to act socially and psychologically. People are seen as actively seeking new opportunities to master – as opposed to satisfying deficits. The other direction in theorizing has been linking needs to behavior. The concept of motivation and the processes shaping them, are much better understood today than during the era of Maslow. These processes explain via reference to environmental and developmental differences why people end up behaving so differently in respect to technology. Self-determination theory, discussed below, is one of the most widely studied theories in psychology currently and gaining increasing ground also in HCI applications.

Motivation concerns what moves people to action. According to SDT, motivations are ultimately rooted to a sense of self-determination. Two types of motivations are distinguished. *Intrinsic motivation* consists of activities that a person has integrated into sense of self. By contrast, *extrinsic motivation* refers to external regulation of motivation. Both controlled and autonomous motivation *energize* action. It gets people to actively pursue technology use. The opposite is *amotivation*: The lack of appeal for activity.

In the case of extrinsic motivation, it is not the self but external forces that determine the conditions for reward or punishment. An information worker may be motivated to learn to use a new information system just to avoid being punished or viewed negatively by colleagues. A computer gamer may seek approval or self-esteem by acquiring a reputation in the game. Extrinsic motivation is thus *instrumental*, aimed at outcomes distinct from the behavior itself. However, being controlled may lead to the experience of being pressured to behave or think in an externally defined way. This may ultimately lead to a thwarting of the activity.

Internalization is the process of transforming an extrinsic motive into personally approved activities. When given a chance to direct their own behavior, a user's actions will reflect the underlying values of self. The theory distinguishes the following continuum in internalization (see [815]):

- External regulation
- Introjected regulation
- Identified regulation
- Integrated regulation

In external regulation, the rewards that are motivating are controlled by the environment. This is the least self-determined type of extrinsic motivation and associated with negative drivers such as avoidance of punishment. Introjected regulation is a partially internalized motivation, but one that is not yet integrated as one's own. These can, for example, be about avoidance of guilt or shame. In identified motivation, in contrast, the activity is consciously valued as personally important. Integrated motivation, finally, the activity is congruent with personally valued goals and needs that are part of the self.

6.2.4. Applications of self-determination theory in HCI

There are many uses of SDT in HCI. Considering the needs from SDT systematically allows us to identify opportunities for technology development. A generalizable understanding of users' needs is a more solid foundation for innovation than observations of behaviors with no such explanation. The coupling to *motivation* is a key part of SDT. As needs are what motivates us, attention to them and to motivation help us to design and evaluate interactive systems.

According to STD, when we study ”user needs” we actually study motivations, the acquired drivers that make us pursue particular ends related to computers. For example, a programmer may have intrinsic motivation linked to the competence need, whereas someone else may not, and many manifest amotivation. A social media application can be argued to fulfil the need for relatedness by helping us to communicate with other people. The theory posits that individuals are oriented toward the three basic needs in different

Figure 6.1.: According to self-determination theory, motivations to act in a certain way develop over time in the interplay of environmental offerings and rewards and basic psychological needs.

proportions. One user may be pursuing autonomy more than relatedness or competence. These orientations are more stable, individual traits. These can be found out via surveys.

We can also use SDT to focus on developing technology that better helps people achieve their goals and sustains their motivations. Assisting people to do so has been a key goal of technology in health care, for example, where an important goal is to use models of motivation to change behavior. We will later in this chapter discuss behavior-change technologies, which draw on motivation theories to help people change behavior.

As another example, Peters et al. [632] discussed how different types of motivation can affect how we adopt technology. In particular, they argue that if our motivation to pick up a technology is autonomous ("I really want to try that app because I think it will help me engage with exercise more") it differs from situations where adoption is externally controlled ("my boss is forcing me to download this app").

SDT also allows us to *analyze* and classify user research by using models of needs and motivations to help see larger patterns in user reports and the essential drivers of what people want and do.

Finally, the paper example box following describes how we may obtain questionnaire information on motivations. In this way, SDT helps establish measures for use in empirical studies. Another example is games, where the sense of need satisfaction has been measured using PENS, the Player Experience of Need Satisfaction Scale [382]. It consists of five subscales partially motivated by SDT: competence, autonomy, relatedness, presence/immersion, and intuitive controls. Validation studies have supported this structure.

SDT has been applied in a number of particular domains. As an example, Tyack and Mekler [815] have charted the use of SDT within the context of HCI and games. In any application, three questions stand out.

- The first is whether the technological feature is linked to need satisfaction: *can* it, ultimately, serve as a vehicle for need satisfaction? Is it a potential for pursuing any of the three basic needs? If the answer is no, people may still be motivated (extrinsically) to use technology. However, intrinsic, integrated motivation will not appear.

- The second question is whether there are *feedback and rewards* in place to encourage behaviors that grow intrinsic motivation. Offering an extrinsic reward for an originally intrinsically motivated behavior can undermine intrinsic motivation, because behavior becomes controlled by external rewards, diminishing autonomy. Deadlines are a good example. Increasing options, in contrast, can increase intrinsic motivation. Competence, on the other hand, can be encouraged by giving positive feedback. Unexpected positive feedback on a task has been found to increase intrinsic motivation, arguably because it is fulfilling need for competence. Negative feedback can have the opposite effect.

- The third question is if the user is able to imagine or predict the effects of that feature on outcomes related to motivation. If the possibilities that the application offers are not visible, they are not tried out. The better the user can imagine (or knows) the outcomes, the more likely related action is.

Paper Example 6.2.1 : Questionnaire on Motivations

The point of this chapter is that the motivations that drive us to use interactive systems—the whys of interaction— are essential for HCI. However, they have received little attention. Therefore, Brühlmann et al. [103] developed a questionnaire to measure such motivations.

Questionnaire development is difficult, as we discuss in Chapter 13. The main idea of the questionnaire developed (the User Motivation Inventory, or UMI) was to draw on the motivation types of SDT. Six such types were distinguished and questionnaire items were developed from them. For instance, amotivation was gauged using questions such as ". I use [X], but I question why I continue to use it" and "I use [X], but I don't see why I should keep on bothering with it". Extrinsic motivation, in the form of external regulation, were gauged by questions such as "Other people will be upset if I don't use [X]". And finally, intrinsic motivation were gauged by questions such as "I think using [X] is an interesting activity" and "Using [X] is fun".

Brühlmann et al. [103] found that these questions indeed cluster into six groups, corresponding to the six motivation types. They also showed that the type of motivation could help distinguish participants who had considered quitting using a technology. Finally, more self-determinated types of motivation (in particular intrinsic motivation) were positively associated with need satisfaction measures and, interestingly, with so called vitality scores (e.g., "When I use [X], I feel alive and vital").

Altogether, this example shows that we can assess motivation with interactive systems. It also suggests that motivation that comes more from persons themselves are associated with valuable attitudes towards technology.

6.3. Behavior Change

It has long been realized that artifacts in our environment, including technology, can influence our motivations. This is the case, for example, for clocks, calendars, and memorabilia. They can help us remember what to do, what we have committed to and what type of person we would like to be. In short, they help us align our actions with our needs and motivations.

Interactive computer systems represent one of these types of artifacts. They can also help us to stay motivated and remind us of goals we have set ourselves. Such interactive systems are used for *behavior change*, and a large body of work has investigated how technology helps to change this [323, 640]. For instance, applications have been developed to help people exercise more, quit smoking, and drink more water. The rise of self tracking is similarly related to behavior change. The box below details one more example of using early mobile phone displays to change an individual's fitness regime. However, behavior change is more than about health and well-being. Whenever people have the choice *not* to start using a new service or system, we face a behavioral problem. Any new system may ask users to change their practices, and they may feel unwilling to do so.

One early view of how to impact how people behave was to liken it to persuasion. Following that line of thought, technology should simply mimic what we know about persuasion from other areas and apply them to the design of interactive systems. This was the view taken by the early work on so-called *persuasive design* []. [244] proposed an early framework for persuasive design. This framework separates three components of behavior: (1) motivation, (2) ability, and (3) triggers. It posits that all three components must be present for a behavior to occur. For instance, getting a user to consider using a stronger password, there must be motivation, ability (e.g., to use special characters fluently when typing), and a trigger (e.g., suggestion by colleague). Despite the intention of persuasive design, as originally formulated, not to include deceit [323], it might be misconstrued. Therefore, this section focuses on *behavior change* in a more general way.

The fundamental challenge in behavior change is that people commit to goals that they then fail to achieve. Thus, much work on behavior change has focused on helping people through stages of changing their behavior and on otherwise nudge them towards their desired behavior. We next discuss some key principles for doing so and some ethical considerations around behavior change.

Paper Example 6.3.1 : Getting healthy with a mobile phone app?

UbiFit Garden is a classic example of behavior change in an early mobile-phone app to promote a more healthy lifestyle [166]. The problem tackled by UbiFit Garden is the prevalent sedentary lifestyle and its impact on people's health and wellbeing. Would it be possible to use technology to encourage people to exercise more?

UbiFit Garden attempted to do so by three elements. A fitness element that infers fitness activities from sensor data. An application that allows users to see their workouts and the associated statistics. A glanceable display that shows the activity as flowers, butterflies, as recent goal attainment (see below). These mechanisms aimed to improve the contemplation, preparation, and action stages in the transtheoretical model.

A field trial of the system showed that the system was well received and that participants found the glanceable display motivating.

6.3.1. Stages in Behavior Change

One of the most influential models of behavior change is the *Transtheoretical Model*; called so because it is supposed to integrate several previous models of behavior change [658]. Its key idea is that behavior change occurs in five phases (see Table 6.2).

The model has been widely applied in HCI to reason about how interfaces help change behavior. [319] used the model to develop feedback mechanisms for energy consumption, aiming to make people's energy use more sustainable. For each of the five stages, they discuss goals, rationales, and recommendations for which information to present in tools for energy consumption.

For instance, in the precontemplation phase, a person is uninformed or unwilling to change behavior. The scenario presented by [319] concerns Mary, who is "36 years old, married, the mother of two school-age children (Logan and Sarah), and lives in Edsen Community". Mary is "somewhat aware of general environmental problems; she does not believe that her personal energy use (and in particular, her computer usage) has much

Stage	Explanation
Precontemplation	People are unaware of a need to change, of the benefits of doing so, and the drawbacks of their current behavior.
Contemplation	People are ambivalent about changing behavior but consider the pros and cons.
Preparation	People are ready to change and share this information with friends and family.
Action	The new behavior is trialed and evaluated
Maintenance	Circumstances are ensured for continuing the behavior in the future.

Table 6.2.: Overview of the stages in the Transtheoretical Model [658].

negative effect. In general, Mary does not believe she has the time or energy to make big energy changes". Based on the model, it is recommended to give personal feedback about the "benefits and consequences of the individual's non-sustainable energy behavior" and the small things that Mary could do which would influence the environment positively. Further, based on the theories, the authors suggest to "refer to social norms regarding sustainable energy behaviors by aligning the use of descriptive and injunctive normative messages". Similar advice is given for the other phases, making for a theory-driven set of recommendations to changing behavior in a particular domain.

6.3.2. Other Factors in Behavior Change

Several models of behavior change do not separate phases, but rather in terms of what determines what makes people act. Some determinants require people to reflect and be aware of their attitudes, the required changes, and the obstacles they face.

For instance, work on *goal setting* has been influential in behavior change. The key idea is that setting concrete goals has been shown to be instrumental in researching those goals and therefore changing behavior.

Self-determination theory may also be seen as a way to conceptualize behavior change. The idea is that if the needs and motivational processes outlined in Section 6.2.3, then supporting those needs and processes with technology might help maintain or bring about a certain behavior. Villalobos-Zúñiga and Cherubini [835] pursued this idea and related features in a selection of 208 applications to the basic ideas in self-determination theory. For instance, they identified five features of the apps that are related. One common application of self-determination theory is to justify the use of *gamification*, that is, the use of game elements in user interfaces for non-gaming contexts.

Another approach to behavior change has come from the field of behavioral economics, in particular the work by Thaler and Sunstein [799]. Their key idea, *nudging*, is to make subtle changes in the choices that people make with surprising large consequences for behavior. That is, nudging is “any aspect of the choice architecture that alters people’s behavior in a predictable way without forbidding any options or significantly changing their economic incentives” [799, p. 8]. Nudging has been widely used in user interfaces [126].

Finally, *dual-process theories* have been influential in understanding behavior change (see Chapter 5). In the context of behavior change, they separate a mostly fast, automatic process of thinking about behaviors from a slow, reflective process. The former is associated with habits and holds great power in human behavior. However, tapping that process might rely on what [10] called mindless behavior changes. Several of these ideas have been used in HCI to understand and design technologies for behavior change. For example, [485] used dual-process theory to understand current digital tools for self-control.

6.3.3. Ethics of Behavior Change

Changing people’s behaviors is not to be taken lightly. We might lead them to believe things that are not correct to have them act in a particular way. When we change what people do, we may also cause their actions to be incongruent with their needs. Thus, messing with users’ needs, wants, and motivations caries ethical implications.

Another area where ethical discussions have amassed is the role of technology design in creating addictive or otherwise problematic relations to technology. The concern here is that the profound understanding of behavior that we have just discussed might be used to manipulate users rather than to help them achieve their goals and needs. The classic example is the design of gambling environments [727]. [explain] In HCI, similar concerns have been voiced for the use of social media and for mobile phones in particular.

One area where ethical discussions have become prominent is in so-called dark patterns for user interfaces. A dark pattern comprises user interface elements “that benefit an online service by coercing, steering, or deceiving users into making unintended and potentially harmful decisions” [510]. Such patterns are not in the best interest of users, by definition. Yet, it remains a thorny question whether it is unethical for HCI researchers to contribute to them or for software developers to implement them. In the context of nudging, [126] discussed how *transparency* of a nudge is related to its ethics. More generally, when behavior change technologies are clear to people, they are less ethically problematic than when they are not.

6.4. Gamification

It is clear from the discussion in this chapter that intrinsic motivation can be forceful. Thus, many people have thought that we should add user interface elements to interactive systems that promote intrinsic motivation. The idea is that if we could make different aspects of computer tasks more intrinsically motivating, then we could boost performance and satisfaction.

Gamification is the idea of adding game mechanics to engage users to solve a problem or perform a task. A range of gamification strategies have been considered in the past. One approach is to add reward strategies, such as awarding points or giving badges based on users' performance. A variation of this approach is to make it a competition by making rewards visible to other users through the use of high score tables and other means. Another strategy is to make the task itself feel more like a game. This can be achieved, for example, by adding storylines as users progress or direct game mechanics, such as increasing the difficulty as users become more proficient with the task.

One example of gamification in HCI is to help label images. Labeling images is very laborious. One way to enable labeling at a large scale is by letting users play a game that simultaneously results in images being correctly labeled by users [842]. Two remote players are shown an image and are instructed to guess the word that best describes the image. A player cannot see the other player's guesses. Once the two players agree on a common word, the game moves on to the next image. The game has "taboo words" (very common words for images). The players cannot use "taboo words" describe the image. This makes the game harder. The list of "taboo words" grows dynamically as the game is played. In that way, very large sets of images (e.g,. all the images indexed by Google) can be labeled relatively quickly.

The act of gamification is an instance of incentive-based design in which users are encouraged to perform certain functions by the system that rewards particular user behavior. Such strategies can be very effective, but they are also often criticized for manipulating users. Therefore, if they are used, they must be used with care.

According to SDT, through cognitive evaluation, people associate *external rewards* (like success in a game) with satisfaction of basic needs. However, this external reward can undermine intrinsic motivation. For example, success in a game can transform into a negative, a controlling, and amotivating experience. Thus, gamification remains a contentious strategy because it might undermine intrinsic motivation and its associated benefits.

Summary

- Self-determination theory proposes universal psychological needs that are manifested in behavior via motivations.
- Besides the identification of user needs, the main use of needs theories lies in behavior change applications, which is an ethically contested application.
- The theory of motivations discussed in this chapter warns against the naive view that experiences can be designed. Because of the very complex relationship between internal and external processes in behavioral regulation, there is no deterministic relationship between design and experience. Only by thoroughly understanding both the prevailing state and the psychological processes in a particular case, one can hope to positively influence the formation of experiences.

Exercises

1. Psychological needs. Pick an application you use frequently. Which of the six psychological needs of Hassenzahl (see Table 6.1) are relevant and why? How about

the basic needs of Deci and Ryan?

2. Do users know what they need? In a group discussion, find a technology that you all use. Could you have foreseen what would be useful to you before you started using it? Could other users? Could a contextual study?
3. Quasi-needs vs. needs. Here are statements by users: which are potential needs and which are quasi-needs? ”I need to check email often.”; ”I want to see if there are new messages from my son.”; ”I must update my Facebook application.”; ”I want to avoid anyone stealing my account.”
4. Technology push vs. technology pull. What makes people desire some technology according to the SDT? Think about a hypothetical technological feature using AI. Answer the following questions: 1) Will the technological feature be linked to need satisfaction? 2) Will there be feedback and rewards in place that grow intrinsic motivation? 3) Will users be able to imagine or predict the effects that feature on outcomes related to motivation?
5. Behavior change. Consider a hypothetical application where a large language model (e.g., ChatGPT) was used to motivate people to start using public transportation. Develop a scenario for the application and analyze it using the transtheoretical model.

7. Experience

A key part of human life and hence of interacting with computers is our experience. Consider the three examples in Figure 7.1. They include software and hardware errors, as well as carefully designed functionality. But they also include people who experience despair, frustration, and anger. And users who recall interaction and associated experiences of joy and competence. All of these encounters describe people's *experience*, the topic of this chapter.

On the surface, the concept of experience seems straightforward. We all know what experience is from our wake movements. The full picture, however, is much deeper

A: When I first purchased my phone, I had problems with the use of the GPS application and Google Maps and usually had difficult in reaching a destination on time or even reaching one at all. So one day I decided that it would be in my best interest to look into the iPhone's capabilities and see if I could rectify the situation myself. I've always been sub-par with technology, and this was certainly a step up from my blackberry that I had for a number of years. I had found out that you could pinpoint the exact location of your phone and use that as a starting point for directions. Needless to say, when I found out about this, I was ecstatic knowing that the directions I were receiving were not only accurate but also the most efficient route available. I never stopped using that feature after that day.

B: I was on the computer with my son who lives in Las Vegas, and we were skyping for the first time. My granddaughter was born 4 days before, I wasn't able to get to Vegas to see her, so we decided this was the next best thing. We turned our Skype on and there she was ...... 4 days old, he showed me all her toes and fingers, she was the prettiest thing I have ever saw. The Skype experience was great. I would never imagine in all my years we could do this and be so far away but having the feeling that I could touch her through the computer screen. We have skyped every day for 2 weeks now. I have seen my granddaughter cry, sleep, I even watched them change her diaper and take a bottle. It was like I was right there doing it myself.

C: I had a bad experience with a presentation at my sister's wedding. I was trying to present a slideshow of pictures of the bride and groom as a tribute, and I had spent weeks preparing it on my computer. When the day finally came, I gave a little speech, announced my slideshow, and then attempted to start it. All I got was an error message!

Figure 7.1.: Examples of positive and negative experiences [811]. Some participants answered the question "Bring to mind a single outstanding positive experience you have had recently with interactive technology", others the same question but about a negative experience.

and more complex. What we all share is *experiencing*—we all the time witness the totality of our mental life play out as pictures, emotions, internal speech, and sensations. This experience is ongoing and constantly changing. When typing on a laptop, we are experiencing the feel of the keyboard, the background buzz from the fridge, the breeze of air from an open door, the nagging feeling of not knowing where a function is in a pull-down menu, and much more.

Our *experience*, on the contrary, is what we tell ourselves and others. We might feel satisfied after typing up a text, or we might tell about the hassle of forgetting the location of the comment feature in the menu hierarchy. But what we remember and how we recount our experiences are only indirectly linked to our experience. They result from a context-dependent process of creating experiences. These processes and the broad categories of experience content are the topics of this chapter.

Why is understanding experience and the processes that create it important to HCI? There are at least three reasons: First, during the past 20 years, HCI has expanded from interactive systems, or products, to also include experiences. The area of *user experience* has developed theories and methods to help achieve this extension [316, 451, 43]. One argument behind this area is that people derive more value and more positive emotions from experiences than from products. Although the personal computer of the 1980s was focused on 'getting things done'—and, ergo, usability was important—the computer of the 2020s offers many diverse opportunities to consumers, who *choose* these opportunities much based on their experiences of using them. The principles of how people experience help to understand the mechanisms behind choice and the extent to which we may influence them via design.

Second, experiences matter. When we move, think, and collaborate, we do so not only as described in the respective chapters, we also *experience* it. Ethically, positive experience constitutes an objective as important in design as anything else. Thus, independently of our other understanding of people, we need to understand their experienced perspective.

Third, experiences, despite their incomparable quality, can be categorized and measured, and obtaining such knowledge can be informative. For example, we may separate aspects of experience that are about affect, the stimulation of curiosity, novelty, or aesthetics. We can also evaluate some of these aspects empirically using appropriate methods, helping to set design goals, conducting research studies, and iterative development. For example, [133] made an early argument about the importance of fun when engaging with interactive systems. Fun being a particular type of experience that we all know. Their work has inspired much subsequent work on how to design for fun and how to measure it.

Next, we discuss why experience is important, some different types of experience, the processes that shape them, and how to assess experiences in HCI.

7.1. What is Experience?

Think back to your last vacation or longer trip. What was it like? Did you enjoy yourself? What were you expecting before going on vacation? What was the highlight of the vacation? These questions are all about experiences. In studies of people's responses to such questions, the answers are surprising.

In a study of students' experiences before, during, and after spring break, among other things, the researchers asked about students' overall subjective experience [882]. They asked them to rate the statement "I will be satisfied with this vacation" on a 1–5 scale with the endpoints being disagreeing and agreeing.

To the surprise of the researchers, the satisfaction with the vacation was higher *after* the vacation than during the vacation. Satisfaction before the vacation influenced not only the experience during the vacation, but also how the experience afterwards. Furthermore, when the researchers asked the students if they wanted to repeat the vacation, their ratings of experience during the vacation did not influence their answers. Only the ratings made after the vacation influenced their desire to repeat the vacation. What is the explanation for this surprising result?

7.1.1. Experiencing and experiences

As mentioned, it is useful to distinguish between *experiencing* and *experiences*. The former refers to the ongoing, moment-to-moment experiencing, for instance, of a computer system when using it. The latter refers to aggregated accounts of an experience, for instance, as you would share them with a friend.

Each construct is associated with different methods of assessment, principles of development, and implications for design. This distinction is grounded in research on psychology, in particular the notion of the experiencing and remembering own experiences [394]. In the vacation example, experiencing influences experiences less directly than might be expected. Furthermore, experiences seem to influence the desire to redo the vacation more directly than experiencing.

In HCI, experiencing has been defined as "momentary, mainly evaluative feeling (good-bad) while interacting with a product or service" [314, p. 12]. Thus, experiencing encompasses the totality of feelings, memories, and thoughts, as available to us through introspection. Frequently we are mainly interested in reactions to the product or service and therefore mainly focus on a simple evaluative feeling.

In HCI, experience is commonly defined as "a person's perceptions and responses that result from the use or anticipated use of a product, system or service" [369]. Thus, experiences are accounts of episodes of interaction, they are aggregates or summaries of a series of experiencing. Usually, they are constructed as wholes that can be named, and they have a beginning and an end. As seen in the vacation example, the relation between experiences and experiencing is complex. Psychologists have detailed many surprising dynamics in how momentary experiencing is turned into experiences, some of those relations are given by names such as sequencing effects, duration neglect, and the peak-end rule.

Sequencing effects mean that the order in which we are experiencing things is important for our experience. Whether something happens recently or some time ago, and how distinctive an experience is, affect the way experiences are remembered.

The peak-end rule captures the finding that people tend to anchor their assessment of experience to ends and peaks. 'Ends' are the most recent experiences, say the last

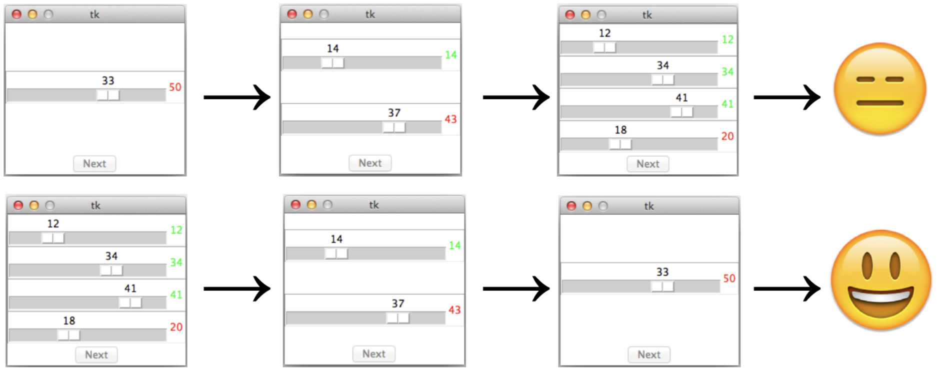


Figure 7.2.: Two sequences of screenshots used in an experiment on peak-end effects [158]. Going through the three screens in the order of the top row decreases satisfaction compared to the order of the bottom row.

time you used a particular information system. Peak experiences, on the other hand, are the best or worst experiences associated with the system. For example, if you experienced something surprisingly delightful with a service, you are prone to rate the overall experience higher than what that singular event might warrant.

Duration neglect is the finding that pleasure ratings are insensitive to the duration of those sequences. Even very short experiences can have a strong influence on the stated experience.

These effects matter in HCI. For example, Figure 7.2 shows two screenshots. If experiences were just summaries of experiencing, there should be no difference between working through the top row of screens and the bottom row. Cockburn et al. [158], however, suggested that there would be a difference based on the peak-end rule. They showed that a combination of different peak-and-end workloads (here: the number of sliders to manipulate) changed which sequence that participants in their study experienced as the best, which was measured by asking for their preference.

More generally, researchers currently understand that user experiences are a result of *inference*. Experiences are created as a result of an active interpretative process that is shaped by mood, context, attention, goals, and attitudes, among others [198]. At any given moment, there are many facets of experience that could be used to form a view about experience. For example, you could infer it based on emotions, previous experiences, peak-end experiences, socially shared beliefs, or any combination thereof. When inferring, we form a representation of what our experience is. This inference process is largely unconscious, unless we ask users to report their experience. The inference itself may happen in at least two ways [826]. First, inference can happen as a specific-to-general inference, such as creating an overall experience from its parts. For example, users may infer how great an application is based on momentary feelings of joy or disappointment they happen to recall. Second, inference can also happen by a general assessment that spills over into specific parts of an experience. For example, users can make an overall evaluation of an experience of an interactive system, which may ”spill over” into evaluation of its components. This is called the *halo effect*. The ”halo” of the product affects inferences of its parts.

It is important to understand that this inferential process means that design does not directly determine experience. Although designers may talk about *user experience design*, experience is not literally designed. Rather, design can influence what users consider as input to this inferential process.

7.1.2. Consequences of Experiences

User experiences are not without consequences. One cause is that well-being is more closely associated with experiences than with the use or ownership of products. So, rather than the interactive system itself, it is the experiences that facilitate what matters to people. For example, purchases made to provide users with an experience were valued more than those made for material possessions [821]. The researchers asked students to rate the question “How happy does thinking about it make you?” with respect to a material purchase and a life experience. Ratings were markedly different along this and other dimensions—experiences made students happier. Other papers have reported similar findings, and many think that this is a key argument for why experiences are an important part of HCI [315].

Another part of the argument is that experiences are important for several higher-level decisions. They include decisions to purchase a product, whether to continue using a computer system, and whether to recommend a digital service to a friend. It is not only each moment of use that matters, nor the actual outcome. Our experiences as we summarize them to ourselves and tell others are also critical. For example, Norman [586] raises an interesting riddle about the memory of things—that is, our experiences of things—instead of the actual state of the world:

> The argument starts with a simple thought experiment. Suppose in some task, using a product or getting a service from a company, you had some perfectly horrid experience along with some positive ones. Now, just suppose you had no memory of the horrid experience. Would you go back and repeat the experience? Most people would repeat something they remembered as enjoyable. Of course, the premise is suspicious: If the experience were truly that horrible, I would maintain a memory of the negative parts. Yes, but memories for bad experiences dissipate differently than those for good ones. The negative emotions associated with the bad parts fade away more quickly than the cognitive evaluation does. So although I remember the events, the emotions have dissipated. Notice the delight with which the writer of the email shared her story of the negative experience with me. Yes, the bad things were horrible. But yes, she would go back.

This is similar to the vacation example; willingness to go back to the vacation destination is not strongly related to the actual vacation experiences. Norman’s argument is that our memories and experiences are what matter in such decisions.

A third reason is that experiences provide material for sharing stories with others. Outside HCI, this aspect has been shown to be important. For example, people talk more about their experiences, compared to material purchases [434]. They also find more pleasure in talking about their experiences. Thus, experiences have distinct social consequences, including their potential for storytelling. This consequence is about experiences that occur to an individual. Forlizzi and Battarbee [249] suggested the term *co-experience* to capture such situations, but also situations where the experience was initially shared.

7.1.3. Processes in Experience

Experiences are complex, constructed phenomena that develop over time. McCarthy and Wright [515] formulated a temporal model that combines these insights, see Table 7.1. They separated several processes that together shape experiences. The processes occur in a particular order. This account makes clear that experience encompasses all parts of our mental lives. It also means that attention to experience covers moments that do not include the actual use of a system. It also covers the full spectrum of psychological phenomena around system use, including perceptions of the system, but also of oneself and other users.

Some processes in Table 7.1 are of particular importance. Anticipation has shown to shape experiences, and thereby also experiencing. Some even define satisfaction as the fulfillment of expectations. Most definitions emphasize that all aspects of product use are in focus, and some include anticipated use of products and experiences following the use situation.

As suggested by the model, experiences are also shaped well after they have happened. Reflection shapes experiences after they happen. Isaacs et al. [367] targeted reflection head on by developing a mobile system that would help support reflection by allowing the capture of everyday experiences and return to reflect on them at a later time. In a system deployment with 44 users, the researchers documented that the app indeed supports reflection and through that improves psychological well-being. Finally, recounting emphasizes the role of experiences relative to other people. People tell about their experiences to other and derive value from that. Note that McCarthy and Wright [515] also consider the stories we tell ourselves about interactive systems for reounting.

What might we use such a framework for? Grönvall et al. [288] were interested in how people experience a shape-changing bench. The bench was 2.5 meters long with a rectangular form. It holds up to six people and can change shape through eight linear actuators embedded in the upholstery. The researchers wanted to use it to explore how people experience shape-change as part of their daily lives. In addition, the intention of the study was to study how colocated strangers might interact around a shape-changing bench.

Based on deployment of the bench in an airport, mall, and a concert hall foyer, Grönvall et al. [288] analyzed video recordings and brief interviews with 129 users of the shape-changing bench. The resulting interviews were analyzed using the McCarthy and Wright model. Let us consider a few insights from that analysis. The anticipation process was shaped about whether the bench is actually a bench or something else: Participants

Process	Explanation
Anticipation	Expectations for an interactive system and its use. Having an idea of how a mobile phone should feel is an example.
Connecting	The immediate response to a system, before talking about it or attempting to put it into words. The sense of spaciousness and wonder the first time one don a virtual-reality headset would be an example.
Interpreting	Making sense of the experience, its structure, and valence. For instance, one might try to figure out where on a website one has ended up.
Reflecting	This is about making judgments of the experience as a whole and figuring out why. An example of reflecting concerns whether an experience meets our expectations and how that makes us feel.
Appropriating	Making an experience our own and relating it to ourselves, our history, and our planned future. Figuring out when and where one feels comfortably texting using a new smartwatch is about appropriating.
Recounting	Recalling past events for personal reminiscence or social sharing.

Table 7.1.: Six processes involved in experiences, based on McCarthy and Wright [515].

would say things like “I went off right away when it began to move. I thought, oh... this is not for seating”. Later, participants would note that as the bench moved, it impacted them not only their senses, but gave rise to “fright, to confusion, surprise, dizziness, and amusement”.

Grönvall et al. [288] also found an active process that did not fit the model; they called it *exploring.* That process consisted of trying out things to make sense of the bench. Overall, however, the model helped structure and analyze the empirical data.

7.2. Types of Experience

Work on experiences in HCI originated in a dissatisfaction with the types of experience that the early work had focused on. In particular, early work focused on avoiding errors, interface features that annoyed people, and frustrating parts of the user interface. This can be called the *deficit approach* to experiences. On the contrary, the work on HCI and user experience has focused on *positive experiences*, including those of enjoyment, meaning, and stimulation.

Another feature of the work on experience in HCI is that experience is considered to be *multi-faceted.* This means that researchers are not only focusing on the value of a product to accomplish tasks; they also focus on symbolic and aesthetic value.

So far, we have emphasized that experience is holistic, covering all of our momentary mental lives. How does this work with the idea that there are *types* of experience, which suggests division and reduction? We believe that it is possible and useful to distinguish types of experiences. Our rationale follows that of Göritz and colleagues [317].

> Due to experiences’ highly situated, unique and inseparable character [...] they lend themselves to description, but not to any type of categorization or reduction to a set of underlying principles. [e]xperiences can be described in retrospect. However, in the moment of description, they are gone and will never occur again. This actually would be the end of story for experience in HCI, because designing for bygone and unrepeatable experiences is futile. [...] although two experiences may never be alike, we may nevertheless be able to categorize them. [...] To give an example: the positive experience from arc-welding is a consequence of challenge, skills and mastery – in short: competence. This competence experience differs clearly from the experience of an ”I love you” message. Here the positive experience stems from feeling related to other people and, thus, maybe thought of as relatedness experience.

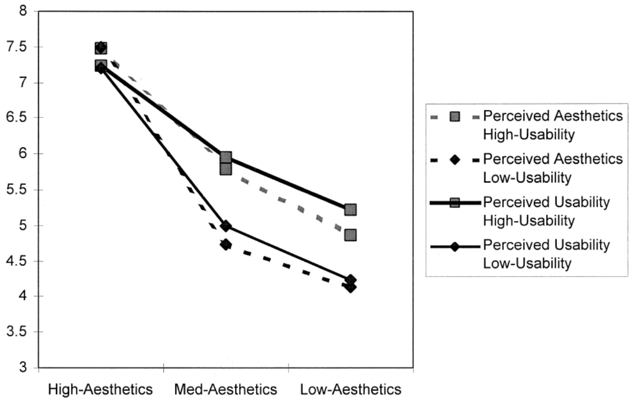
Thus, experiences can be classified without disregarding or denying the fleeting and complex quality of experiencing. On this background, a couple of classes of experiences may be distinguished. We cover those next.

Paper Example 7.2.1 : Is that which is beautiful usable?

Form versus function, or beauty versus ease-of-use, is a classic tension in art and design. Is that which beautiful indeed also useful, does useful artifacts become beautiful to users, or are these concepts perhaps simply orthogonal? Tractinsky et al. [806] were interested in this question because it is significant for HCI—it determines which aspects of experiences we should focus on.

Tractinsky and colleagues managed to turn this question into something that could be explored experimentally. They did so by creating different variations of the interface for an automatic teller machine (ATM), appearing high, medium, or low in aesthetics. At the same time, some variants of the ATM were easy to use and others were difficult to use. The comparison of these variations should reveal whether form and function, or aesthetics and ease-of-use, are indeed unrelated. The figure below shows the ratings of perceived usability and perceived aesthetics across three levels of aesthetics and two levels of usability (square versus circle marks in the graph). Note that here usability and aesthetics are both independent variables (something that is manipulated, see Chapter 43) and dependent variables (something that is measured, in this case with questionnaires).

The key insight is that while there is a difference between low and high usability (the gap between the two groups of lines) the effect of aesthetics (along the x-axis) is much larger, both on measures of perceived aesthetics and on measures of perceived usability.



Initially, participants rated their perception of the aesthetics of the interface and their perceived ease-of-use of the ATM interface. This showed that aesthetics and ease-of-use were correlated. Next, participants got to experience the actual usability of the ATM by using it to complete tasks. Surprisingly, this showed that the aesthetics influence the rating of usability *after* participants had used the system. In contrast, the actual usability of the system did not influence ratings of usability and aesthetics. Perhaps therefore, the authors entitled their paper "What is beautiful is usable". It is important to realize that this is merely a correlation, not evidence that one caused the other. Still, it indicates that some simple aspects of experience (such as perception of aesthetics) may fundamentally impact the use of interactive computing systems.

7.2.1. The Pragmatic and the Hedonic

There is a saying that the journey is more important than the destination. It reflects a particular type of travel experience that emphasizes novelty, experience, and time to observe and reflect. In contrast, many trips are about getting from A to B; they are about being efficient and hassle-free. This difference is reflected in the distinction between the quality of *pragmatic* and *hedonic*, in particular, regarding types of experience and perception of interactive systems.

Pragmatic experiences are those that concern practical matters and the achievement of goals. When interactive systems are simple, clear, or understandable, people describe pragmatic types of experiences. They are sometimes also called utilitarian or instrumental. For most of the history of interactive systems, researchers and practitioners have considered the user's experience as a side-effect compared to other parts of the interaction or to the functionality of an interactive system. They have typically focused on the instrumental value of the interactive system as the most important, including, for instance, users' performance when solving tasks or the utility of the interactive system. This is no longer the case.

In addition, the hedonic aspects of interactive systems and experiences in a more general sense are about stimulation, novelty, curiosity, aesthetics, and pleasure. They are not about what is being achieved as getting to the right destination quickly, but about the pleasure involved in getting there. In a frequently used questionnaire, AttrakDiff, on the hedonic aspects of interactive systems, some central items are Stylish, Professional, Inventive, Creative, Challenging, and Captivation. Examples of pragmatic quality include Simple, Practical, and Straightforward.

In HCI, this distinction was empirically investigated around the turn of the millennium and later turned into a model for user experience [313]. Figure 7.3 shows the model. Three key ideas emerge. A designer designs product features that are associated with an intended product character (top row). Whether the user sees that intended product character is another question (bottom row).

Moreover, among the apparent product characters, Hassenzahl separated pragmatic attributes from hedonic ones. Hedonic attributes are divided into three kinds.

- Stimulation, which is the extent to which interactive systems provide new impressions or opportunities, and thereby make us pay attention, feel curious, or be motivated.
- Identification is the way in which products help us express ourselves to others.
- Evocation is the extent to which a product reminds us about past events, thoughts, or relationships. Though important, this aspect has not been included in much work on the hedonic and the pragmatic.

Finally, according to the model, the perception of hedonic and pragmatic attitudes together drives overall evaluation of a product, such as its goodness. This is a form of inference, where individuals weight together the different types of perception of a interactive system.

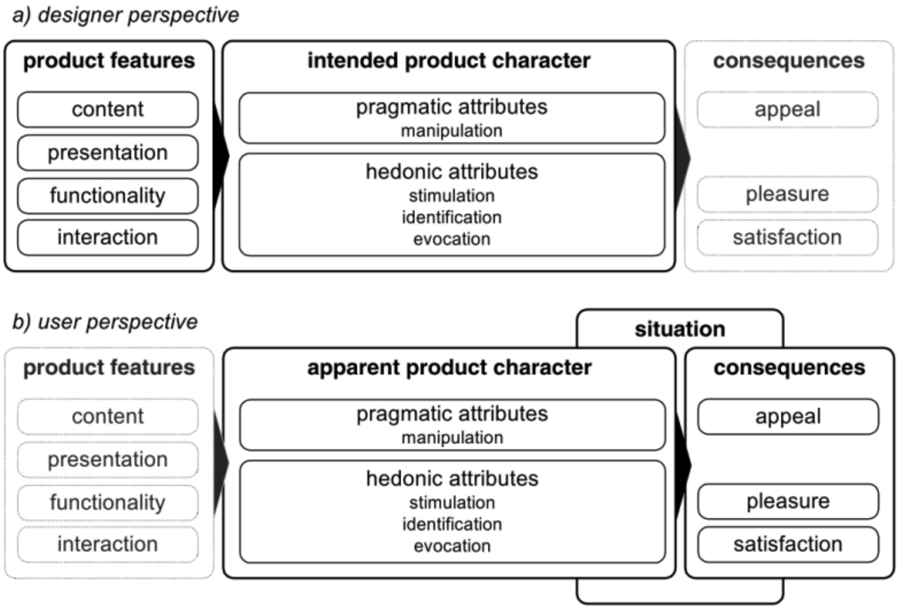


Figure 7.3.: A model of user experiences by Hassenzahl [313]. The top part shows the designer's perspective: you control product features and have intentions for the product. The bottom part shows the user perspective. The product as it appears to the user have certain experiential consequences when used in a particular situation.

Diefenbach et al. [190] showed that across many studies of AttrakDiff, the pragmatic and the hedonic dimension of technology perception are correlated. They found a correlation coefficient r of .62. The coefficient quantifies the strength of the relation between two variables. In this case, it was unexpectedly high and suggests that the current instruments for separating hedonic and pragmatic quality need more work. They are not discriminative enough when it comes to the two types. Nevertheless, AttrakDiff remains a widely used instrument.

7.2.2. Pleasure

The experience of pleasure is a fundamental and positive emotion. It is related to other terms such as enjoyment, delight, and fun. For simplicity, let us consider pleasure as a sensation that is good or desirable. Although we can all name a multitude of such pleasures, would it be possible to distinguish some general types?

Jordan [388] presented an influential model of different kinds of pleasure in the context of interactive systems. He distinguished four kinds of pleasure.

- *Physio-pleasure* concerns the pleasures derived from the senses, including the tactile quality of a keyboard or the smell of a car.
- *Socio-pleasure* concerns the pleasure that comes from our relationships with other people and groups.
- *Psycho-pleasure* is about the pleasure of thinking and feeling.
- *Ideo-pleasure* is about the pleasure derived from the values they embody or help us embody.

Jordan's model is handy because it lends itself to a more systematic understanding of pleasure in evaluation and design. For instance, one might design for particular types of pleasure or evaluate the pleasures an interactive system delivers by categorizing users' statements using the model above.

7.2.3. Emotions

It has been said that emotion is at the heart of any human experience [249, p. 264]. The feeling of whether something is good or bad is a fundamental evaluation that is central to our reactions to our environment, the objects in it, and the people who surround us. It is also fundamental to our experiences with interactive systems.

But what is emotion? Before answering, let us note that the study of emotion is highly complex [47]. It contains a host of theories, which fundamentally differ in their conceptions of what emotions and related phenomena are. Some theories hold that there is a set of basic, fundamental emotions, such as anger and disgust. Other theories hold that emotions are a result of appraisal, that is, of an individual's active processing of internal and external stimuli. The mechanisms and measures of emotion that are derived from differ markedly; Ekkekakis [217] provided a useful first step to understanding emotion. With that said, two approaches have been prominent in HCI to understanding emotions.

One prominent model of *core affect* separates valence and arousal [700]. The general idea of this model is that there are two fundamental dimensions.

1. Valence, which describes positive vs. negative emotions. Positive emotions are energizing emotions, such as being happy or proud.
2. Arousal, which describes the level of energy involved or the degree of activation. High-arousal states are related to being alert or attentive.

These dimensions capture what is called core affect, the essence of how we feel at any moment. Core affect is not directed at anything, but is a fundamental evaluative orientation. This is why the term affect is used and not emotion in this model: emotion can be directed at something or someone. Sometimes other dimensions are included in core affect, in particular dominance (how controlling a stimuli is).

Core affect has been used in numerous studies in HCI as a way of capturing some basic aspects of emotional reactions. Gatti et al. [267] collected ratings of valence and arousal for a selection of auditory, haptic, and visual stimuli. What can such ratings be used for? One idea is to use stimuli to influence affect—for instance, we may play stimuli high in valence and arousal to influence people to feel high valence and arousal.

Another approach is to focus on changes in core affect directed towards an object or internal antecedent; somewhat confusingly, this is often referred to as an emotion. For instance, I may experience fear if a shape-changing interface moves towards me, or I may experience bliss when an odor-creating device makes me recall my childhood. Emotions in this sense are often called basic emotions, and considered to include fear, anger, surprise, disgust, happiness, and love.

Desmet [185] identified 25 positive emotions that people experience with physical and interactive products. The intention was to find emotions that are specific to such products. The emotions were identified and it was checked that regular users do indeed experience them with products. The list of products include emotions already mentioned but also (a) Dreaminess, "To be dreamy is to enjoy a calm state of introspection and thoughtfulness", (b) Pride, "the experience of an enjoyable sense of self-worth or achievement", or (c) Worship, "Worship is the experience of an urge to idolize and honor someone (or something)". These emotions can, for instance, be used as design goals or to evaluate whether an interactive system supports them.

7.3. Assessing and Measuring Experience

Given that it is useful to know about experiences in HCI, how do we assess or measure them? When we consider our own experiences, it is immediately clear that they are highly complex and elusive; indeed, there are aspects that only we can likely ever know in their full richness. However, several methods have been developed that capture the dynamics of experience. They focus on self-reports and first-person methods. We will discuss such methods in depth in Part III but give a sample next.

At the same time, we believe that it is possible to capture dimensions of experiences. Being able to do so is important to develop an understanding of how people experience computers in general, as well as when we try to evaluate the experiences with a particular interactive system (the topic of Part VIII). Thus, we also discuss commonly used measures of experience.

7.3.1. Rich reports on experiences

One method is to ask people to report their experiences. This can be through open-ended questioning, for example, in an interview session (see Chapter **??**). For instance, Woo and Lim [893] were interested in understanding people's experiences with using sensors and actuators to create their smart homes. To get rich accounts on this, they used semi-structured interviews.

Another approach is *the narrative method*, illustrated in the vignettes in Figure 7.1. Narratives are first-person accounts of events organized in the form of a story. They can describe events from a remembered perspective. As they rely on memory, narrative accounts tend to be selective and focus on the narrator's viewpoint. They can help designers feel empathy with another person's viewpoint. *The critical incident method* builds on narrative accounts, but focuses on certain definitive types of experience. Critical incidents can be elicited by asking participants, for example, to ”Bring to mind a single outstanding positive experience you have had recently with interactive technology. Please retell the experience as accurately and detailed as you remember and try to be as concrete as possible. You can use as many words as you like, so that outsiders can easily understand your experience.” [811].

Finally, micro-phonemonological interviews are used as a method to obtain rich reports on experiences (see Chapter 11). The idea is to go deep into the participants' experience. For instance, Obrist et al. [593] were interested in understanding how the experience of particular tactile stimuli felt. To obtain such data, the authors first stimulate participants with different tactile stimuli and then asked open-ended questions such as "What words would you use to describe how it felt on your hand, if at all?". That gives rich data on the experience of the stimuli and its development over time.

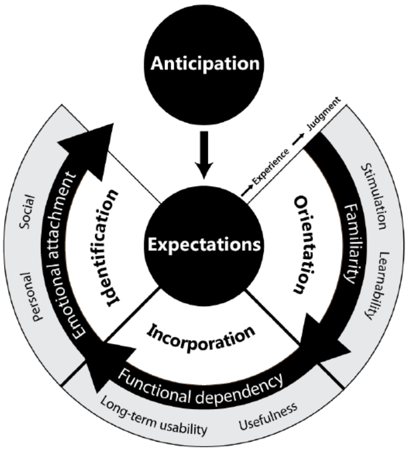
Paper Example 7.3.1 : The Experience of a new phone

Imagine that you get a new smartphone. Obviously, you have expectations about how it will work. Once you begin using it, your view of the phone will likely develop as you discover features, bugs, and unexpected uses. But what exactly are people experiencing with a new phone, and how do their experiences develop over time?

This is what [396] decided to study. They followed six new users of iPhones over a period of four weeks, as well as a week before the phone was purchased. After each day of use, the participants reconstructed their experiences throughout the day, writing more details about the three most impactful of these experiences and rating the product in general using a variant of AttrakDiff. This resulted in 482 reports on impactful experiences. For instance, one participant noted the following.

> [Day 8] "...I had the chance to show off my iPhone to some of my colleagues. I showed them some functions that are rather difficult to operate in other phones... I felt good having a BETTER device. I still have some cards to show which I will in do due time to surprise them even more"

The authors used content analysis to analyze these reports and develop the model shown below.



The main point of the model is that there is a phase of orientation, where gaining familiarity is central. In that phase, the stimulation of the device as well as its learnability is central. That phase at some stage turns into a phase of incorporation into the users' lives where they come to depend on the functionality. That changes, in turn, into identification with the device, where, for instance social aspects of the device, becomes more important for participants.

7.3.2. Measures

An alternative way to characterize the experiences that users of interactive systems have is through measures, that is, quantitative indicators of dimensions of the experience. For instance, we may measure pragmatic aspects of a software system or the emotion towards a system. Measures of experience are important for being able to quantify and work systematically with experiences in development. They have also helped establish many of the phenomena discussed in this chapter.

These measures may be obtained in several ways. We can use rating scales at one time that contain specific statements about experience (see Chapter 13). We may also ask self-reports. The *UX Curve* is a method in which users are given a timeline on a paper (e.g., starting from day 0 when starting to use a service) and asked to draw a curve that describes the level of experience over time. Users are then interviewed to elaborate on the experiences, factors behind the increase/decrease, and what happened in the peaks and ends [433]. This method, however, is suspectible to bias caused by the act of drawing a curve. The threat is that forcing experiences to a curve may omit and embellish memories of experiences.

Another way to obtain repeated self-reports during use. The method *Experience Sampling Method* (ESM) originates from flow theory, but its use in HCI is broader. In an ESM study, users install a mobile application or use a digital beeper that notifies them when to provide experience reports. Such reports are most often ratings but can also be richer

Table 7.2 shows a list of some frequently discussed dimensions of experience, as well as their corresponding measures (synthesized from [43, 636]). These models show that the fun or practical aspects of experiences are not all there is. Researchers increasingly discuss that the *meaning* people experience (or derive from interactive systems) is central. One way of thinking about this is ancient; it is called eudaimonia, which refers to how an experience is related to personal happiness. It is presently thought that meaning stems from events that have long-term value to self.

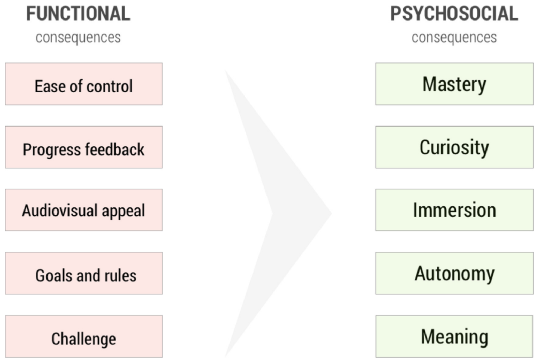
Concept	Explanation	Examples of measures
Affect	Fundamental assessment of experiences in terms of their valence, arousal, and dominance.	The self-assessment mannequin requires participants to rate the valence, arousal, and dominance using drawings of themselves [92].
Enjoyment	The extent to which users experience positive emotions and pleasure.	Perceived enjoyment has been conceptualized as "the extent to which the activity of using the computer is perceived to be enjoyable in it's own right, apart from any performance consequences that may be anticipated" and measured based on this idea [180].
Aesthetics	The extent to which users experience attraction and beauty, for instance in a user interface	Lavie and Tractinsky [450] separated classical aesthetics and expressive aesthetics of websites. The former is about whether the site is clear, symmetric, and pleasant; the latter about whether the site is creative, fascinating, and original.
Engagement	The extent to which something is attractive and drawing interests	Doherty and Doherty [199] reviewed different theories and measures for engagement.
Burden	The extent to which users experience burdens in using technology	Suh et al. [787] developed a questionnaire that assesses users' experience of burden. The questionnaire separates (1) difficulty of use, (2) physical burdens, such as physical discomfort, (3) negative impact on time and social factors, (4) mental and emotional burdens, (5) issues about privacy, and (6) financial burdens, such as costs.

Table 7.2.: Measures of users' experience that are often employed in HCI.

Paper Example 7.3.2 : Measures of the gamer experience?

Computer gamers often play games for the experience of playing the games. The challenge of mastering the game, the fun of winning, the arousal of opening a loot. For the reasons discussed in this section, researchers studying games have been interested in being able to ask questions about games across levels, different storylines, or different types of players to learn about the experiences.

Abeele et al. [4] developed and validated a way of asking about player experiences. Their idea was to discover the fundamental dimensions in such experiences and develop a questionnaire to assess them. The dimensions were discovered through discussions with experts in games user research and from evaluations from players' salient game experiences. From that work, a model of five functional dimensions and five psychosocial aspects were developed. These dimensions appear mostly unrelated, yet important to gaming experiences.



Behind each dimension are three questions that are used in practical assessment for that dimension. For instance, for ease of control, respondents are asked "I thought the game was easy to control", "The actions to control the game were clear to me", and "It was easy to know how to perform actions in the game". For mastery, respondents are asked "I felt capable while playing the game", "I felt I was good at playing this game", and "I felt a sense of mastery playing this game".

7.4. Can experiences be designed?

Given that HCI is about building interactive systems, you may reasonably wonder if experiences of the types we have discussed so far may be designed? For instance, would it be possible to design an interactive system that makes users feel pride? Can we design something that makes users feel happy?

One answer to this question is negative. Designers have no control over experiences, they are individual and idiosyncratic. In particular, for more complex experiences, like those discussed above; it is difficult to find scientific papers where such experiences are determined through changes in an interactive system. From this viewpoint, the notion of ”user experience design” is a misnomer.

Another answer is positive. Experiences are intentionally created all the time, from dining over movies to arts. Why should HCI be different?

The correct answer to the question in the section heading is still being worked out. It is clear that the formation of an experience depends on a complex way on the individual and what happens in interaction. While design affects that process, it does not determine it. This chapter has covered general principles of experiences. Those may be drawn upon in design. For instance, the peak-end-rule suggest that we should pay particular attention to the end of experiences to achieve the impact we want. One promising direction has been to create chart designs that produce certain emotions. This helps to determine which interface design choices to make. For instance, Lim et al. [469] suggested a set of attributes to describe interactivity, one of the components of interactive systems that design can affect. For instance, attributes included response speed (fast, slow), expectedness (expected, unexpected), or movement speed (fast, slow). Through a study with users, Lim et al. [469] could link such attributes to emotional qualities. For example, movements that happen sequentially, continuously, and slowly are perceived as sympathetic.

Summary

- Experiences are created through inference from ongoing experiencing.
- Experience is not a single monolithic thing but comprises, among other things, pragmatic and hedonic aspects.
- Affect is a special aspect of experience characterized by positive and negative dimensions—valence.
- Experiences are indirectly influenced by the designer—they cannot be directly designed.

Exercises

1. Understanding experience. Return to the three examples at the beginning of this chapter. Pick a framework for types of experience that has been discussed in this chapter (e.g., hedonic/pragmatic, core affect). Then analyze the three narratives using that perspective. What do the aspects help you notice?
2. Measuring experience. In this chapter, we discussed several ways of assessing experience through questionnaires. Find an online version of AttrakDiff, Self-Assessment Mannequin, the Burden scale, the Player Experience questionnaire by Abeele et al. [4] or another of the questionnaires. Use it to assess a user interface or website that you use.
3. Exceptional experiences. Think about an exceptional experience you had with computers. Try to bring to light the details of how it came about, what happened,

and how it affected you later. You may write a description or narrative, in the form of a story, of the experience. Then go through each of the six processes identified by McCarthy and Wright [515] and shown in Table 7.1. If nothing matches a particular process, think about why. If the processes help think about new facets of experience, consider why you did not initially notice those facets.

4. In 1974, the philosopher Nozick discussed the experience machine [589]. He imagined the machine to work as follows.

   > What matters other than how people's experiences feel "from the inside"? Suppose there were an experience machine that would give you any experience that you desired. Superduper neuropsychologists could stimulate your brain so that you would think and feel you were writing a great novel, or making a friend, or reading an interesting book. All the time you would be floating in a tank, with electrodes attached to your brain. Should you plug into this machine for life, preprogramming your life's experiences?

   It is possible to imagine virtual reality or brain-computer interfaces that will be able to work as described by Nozick. Please consider the following questions: (1) should we build such a machine? (2) should we regulate the use of such a machine? (3) would you want to plug into such a machine? (4) How does the idea of this machine relate to Norman's discussion of memory and actuality?

5. Pick a model of experience. Consider the design of a new fitness app and discuss how this model might enhance the experience of fitness. Detail at least three different ways in which the design might be shaped by the model and discuss them with others, if possible.

6. Narrative experiences. Below are a number of stories about good and bad experiences with interactive systems (from [811]). The participants answered the question: ”Bring to mind a single outstanding positive experience you have had recently with interactive technology”. Your task is to sort these statements into clusters, which are about different ways in which a system can be good. Bad systems are put in the same clusters based on what they lack or do not fulfill for the users. Name the clusters. Discuss if anything is missing. What aspects of usability and user experience are the clusters about? What aspects are not mentioned?

   > When I first purchased my iPhone 4, I had problems with the use of the GPS application and Google Maps and usually had difficulty in reaching a destination on time or even reaching one at all. So one day I decided that it would be in my best interest to look into the iPhone's capabilities and see if I could rectify the situation myself. I've always been sub-par with technology and this was certainly a step up from my blackberry that I had for a number of years. I had found out that you could pinpoint the exact location of your phone and use that as a starting point for

directions. Needless to say, when I found out about this, I was ecstatic knowing that the directions I were receiving were not only accurate but also the most efficient route available. I never stopped using that feature after that day.

I was recently able to assist a foreign student in contacting her family and boyfriend through the use of Skype on my laptop. As soon as the program was up and running, she was in touch with her family for around 30 minutes. It was very nice to know that her stay was made easier by being able to see and talk to her loved ones on Skype and it made me feel good to be able to let her do that. It was also nice to know that it didn't cost her any money. This experience made me more likely to use Skype myself, even though I don't have friends and family overseas.

An outstanding positive experience that I had recently with my iPhone happened yesterday. My dog had a toy hanging out of his mouth and I used my camera on my iPhone to capture that image. If I had tried to grab my Nikon from the other room, I would have missed this photo. These are the types of moments that you have to capture as they happen. It's so handy to have a good camera on my phone, which is always on me.

The specific experience I had was with Dropbox. I was working on an important document for work. And then I needed to access that document but did not have my work computer with me. But, because I had the document stored in my Dropbox, I was able to simply load Dropbox onto my home computer, have it sync, and start working on my document at home. Then, when I saved it, I didn't have to email it to myself or do anything like that – I knew that it would simply appear updated at work the next day.

I was on the computer with my son who lives in Las Vegas, and we were skyping for the first time. My granddaughter was born 4 days before, I wasn't able to get to Vegas to see her so we decided this was the next best thing. We turned our Skype on and there she was ...... 4 days old, he showed me all her toes and fingers, she was the prettiest thing I have ever saw. The skype experience was great. I would never imagine in all my years we could do this and be so far away but having the feeling that I could touch her through the computer screen. We have skyped every day for 2 weeks now. I have seen my granddaughter cry, sleep, I even watched them change her diaper and take a bottle. It was like I was right there doing it myself.

8. Collaboration

When we interact with computer systems we frequently collaborate with other people. It is challenging to identify any complex activity that does *not* at some point involve other people. There are numerous systems for collaboration and communication, such as messaging services, videoconferencing, shared calendars, and so on. However, there are many challenges in designing, building, deploying, and supporting collaborative systems and, as a consequence, many such systems have failed in the past [297]. In this chapter, we learn about what is known about people collaborating through the use of interactive technology, which is an important piece of the puzzle in realizing successful collaborative systems. It turns out that there are substantial changes in interaction depending on the particular purpose and context of a collaborative activity, as illustrated in the following three examples:

- In the London Underground operators have learned to collaborate without constantly announcing their actions or requesting information [321]. The practices that create this notion of shared awareness is tacit and cannot easily be taught using formal procedures and rules. Nonetheless, it is a kind of knowledge that new operators learn as apprentices in their teams.

- Workers do not 'just collaborate' but they actively need to coordinate *how* to collaborate. This is called articulation work. Figure 8.1 shows a setup adopted by medical team carrying out vascular surgery. Displays are actively oriented in a way so that other people in the room can see and refer to them, thus helping establish common ground [523].

- In some computer games, such as *League of Legends*, players collaborate with strangers for less than an hour. Even in such settings, collaboration is rich and complex. For example, players exercise self-discipline to ensure that collaboration between players remains frictionless and players make conscious efforts to create a pleasant working atmosphere [423].

Collaboration is not only the act of collaborating. It also involves activities that *support* collaboration, such as planning collaborative work, being aware of other collaborators' goals and actions, and determining what to do next. These aspects give rise to an underpinning question, which we will continually investigate in this chapter: What is *special* about human–computer interaction when it involves multiple people?

As we have seen, collaboration generally involves *sharing information and objects*. For example, coworkers might need to share a document they are writing, and standard operating procedures might govern the information flows within a team. Such sharing

Figure 8.1.: Articulation work refers to communication that helps establish conditions for collaboration. Surgeons, for example, need to ensure displays are oriented in a way that others in the room can see and refer to them, thus helping establish common ground [523].

is inevitable, in particular, when work is distributed and mediated through interactive systems. Thus, understanding how such sharing takes place in collaboration and how to rethink such sharing structures for collaborative interactive systems is central.

Collaboration is increasingly important in modern society, and this realization has propagated through to the design of interactive systems. Yet it is only recently that computers' roles in cooperative work have been studied at a large scale. Schmidt [721] attributes the usefulness of computer systems to their ability to allow *horizontal coordination*. In contrast to vertical coordination, which occurs with top-down management from a higher level of an organization to a lower level, computers allow workers to coordinate work among themselves in a horizontal direction. This results in one-to-one and one-to-many communication among arbitrary subgroups within an organization.

Since collaboration is a central human activity it is important to be aware of the many facets of collaboration and collaborative behavior in order to design interactive systems that allow users to collaborate to achieve their goals. Failure to do so results in systems that do not support people as they should because they induce friction when users seek natural and effective ways of collaboration. Ackerman [9, p. 179] expressed the problem of meeting this *social-technical gap* as follows:

> ...human activity is highly flexible, nuanced, and contextualized and that computational entities such as information sharing, roles, and social norms need to be similarly flexible, nuanced, and contextualized. However, current systems cannot fully support the social world uncovered by these findings.

> [...] The social-technical gap is the divide between what we know we must support socially and what we can support technically.

As a consequence, our understanding of interactive systems for collaboration has expanded and has given rise to a subfield of human-computer interaction called Computer-Supported Cooperative Work (CSCW for short). This subfield studies "the ways in which software, developed to support groups, affects individuals and is adapted to different organizational contexts: and systems developed to support organizational goals as they act through individuals, groups, and projects" [296, p. 21]. An example of such a collaborative system is shown in Figure 8.1. In this chapter we will call such, and other systems that facilitate collaboration, *collaborative systems*.

Using knowledge from HCI research it is possible to design various interactive systems that facilitate collaboration, including so-called *groupware*, which are "computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment " [219, p. 40].

In the rest of this chapter, we will introduce concepts and theories rooted in social sciences to understand collaboration, coordination, and common information spaces. This will serve as a knowledge foundation for building interactive systems that allow users to collaborate in desirable ways.

8.1. Understanding collaboration

As a motivation to understand the value of *some* form of collaboration let us first consider a very simple situation that would benefit from cooperation. Game theory is a field of mathematics that studies how rational agents interact by constructing mathematical models. A well-known problem, *the prisoner's dilemma*, studies the optimal strategies of two agents, two prisoners, investigated for a crime. Each prisoner is told that they both face two years in prison on a lesser charge unless one of them betrays their partner. In this case the prisoner that betrays their partner receives no prison sentence, while the betrayed partner receives 10 years in prison. However, if both prisoners confess they each receive five years in prison. The prisoners are held in isolation and have no means to communicate with each other or anyone else except the police. In addition, it is assumed that the prisoners have no means of retaliation as a result of being betrayed.

In this problem, each prisoner faces a choice: *cooperate* with the other prisoner or *betray* them. The optimal rational choice is for each prisoner to betray their partner. This can be realized by considering the choices for one of the prisoners. If their partner cooperates then the rational choice is for the prisoner to betray them. This is because the prisoner then receives no years in prison while their betrayed partner receives 10 years. On the other hand, if their partner betrays them then the rational choice is for the prisoner to also betray their partner. This is because then both prisoners receive five years in prison. The alternative, to cooperate, would result in the prisoner receiving 10 years. The same reasoning holds for the other prisoner as well.

Therefore the rational strategy is for each prisoner to betray their partner. However, the dilemma is that if both prisoners cooperate, they would both receive only two years, which would be a collectively better outcome. Hence, while mutual cooperation results in the optimal outcome for the group as a whole, this choice is irrational from the point of view of an individual prisoner. This is an example of a system that does not allow individuals to achieve the optimal outcome for the group as a whole because the system does not facilitate cooperation.

8.1.1. Collaboration and cooperation

The word *collaboration* has its origin in Latin; 'col' means 'together' and 'laborare' to work. Collaboration is

> a mutually beneficial relationship between two or more parties who work toward common goals by sharing responsibility, authority, and accountability for achieving results' [152].

In contrast, *cooperation*, refers to activities where there does not need to be a shared responsibility. In cooperation, a *division of labor* is in place, each person is responsible for some part of problem solving. However, in cooperation, this division can be imposed to a particular participant, and thus there may not be a need to negotiate or establish a division of labor during the activity. In other words, collaboration emphasizes the joint construction of goals, understandings, and division of labor.

Cooperation and collaboration often occur within the context of work. In fact, *work* is typically cooperative—the success of work frequently depends on other people. However, it is not necessarily collaborative, as workers do not need to be autonomous or share goals and may not even share information. For this reason, administrators, scientists, and engineers can sometimes work at a relative far distance from colleagues and still carry out productive work.

A defining aspect of cooperation and collaboration is the *distribution of work*. That is, different aspect of work are distributed among the actors and their contexts. The communicative actions workers take to support cooperation and collaboration in a team are jointly called *articulation work*. These can include anything from a digital work hour scheduling system in a hospital to materials and communicative practices.

These concepts are important, because collaboration changes the focus in design. Schmidt [722, p.4] writes:

> In order to develop computer-based system that support the articulation of cooperative work in terms of making articulation work more flexible, efficient, and effective, the very issue of how multiple users work together and coordinate and mesh their individual activities has become the focal issue.

Thus, a central question in the design of collaborative systems is how technology contributes to the organization of work.

8.1.2. Size of Collaboration

There are different types of collaboration and, consequently, different collaborative systems. An aspect of any collaboration is the size of the collaboration, which is typically divided

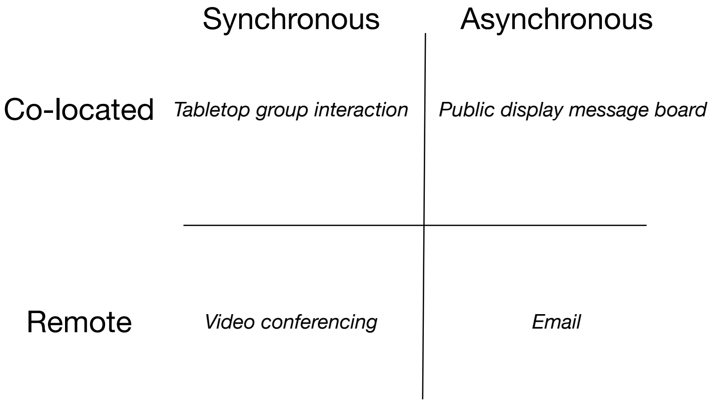


Figure 8.2.: Collaborative systems and activities can be viewed according to a two-axis model: synchronous–asynchronous and co-located–remote.

into four different units of analysis: (1) small groups; (2) project organizations; (3) organizations; and (4) social networks.

Small groups consist of 2–20 group members. A group is here a unit that perceives itself as having a shared identify, a 'we'. An example of small group collaboration is three people working together on a shared tabletop display. *Project organizations* can have hundreds of participants, in any combination of members of the local or distributed group. An example is the organization of a large scientific conference, which may involve 40–80 people managing various aspects, such as paper reviewing, and a small local committee of 5–10 people who are in the same geographical location and ensures local arrangements are appropriate. *Organizations* may include tens of people, but may sometimes involve millions of members. Examples of such an organizations are universities, government agencies, or large firms. Finally, *social networks* consist of people participating in larger cooperative undertakings, an example is the Wikipedia online encyclopedia project. Social networks vary in size but be very large.

These units differ in the types of practices, power structures, and so on that are in place or emerge. Therefore, they tend to require different approaches, technology support, and research methods.

8.1.3. Types of Collaboration

As a starting point for analysis, a model of collaborative technology is the two-axis model (see Olson and Olson [601]), which is shown in Figure 8.2. Fundamentally the two-axis model captures two important aspects of collaboration: (1) *time*—how people collaborative with respect to the timing of collaborative events and actions; and (2) *space*—how users are distributed in space when they collaborate.

The *synchronous–asynchronous* axis describes how close in time the collaboration is taking place. A completely asynchronous collaborative system would not enable the sender of, say, a message to know when the receiver would receive the message or whether the receiver would even see the message at all. A completely synchronous collaborative system means that collaboration is taking place at the same time.

The *co-located–remote* axis captures where users are located with respect to each other. A collaborative system that only allows remote collaboration means that users have to necessarily be in the different locations and have no means of using local communication facilities, such as face-to-face conversations. On the other hand, a completely co-located collaborative system demands that users are present at the same location. The two-axis model is useful at a high-level as time and space considerations have substantial impact on how people can collaborate and, for example, coordinate their work. To get a feel for the two-axis model we now discuss an example for each cell in the model.

An example of a co-located synchronous collaborative activity is a group of users collaborating by interacting with a tabletop display. Such collaboration raises unique considerations, such as how people decide to share the tabletop display among group members.

An example of an asynchronous co-located activity is a public interactive display that allows users to attach notices to other people. Another, more provocative, example is the "snatcher catcher" [484], a prototype fridge that maintains a record of which user took what food and when, thus potentially enforcing a sharing protocol among the users of the fridge.

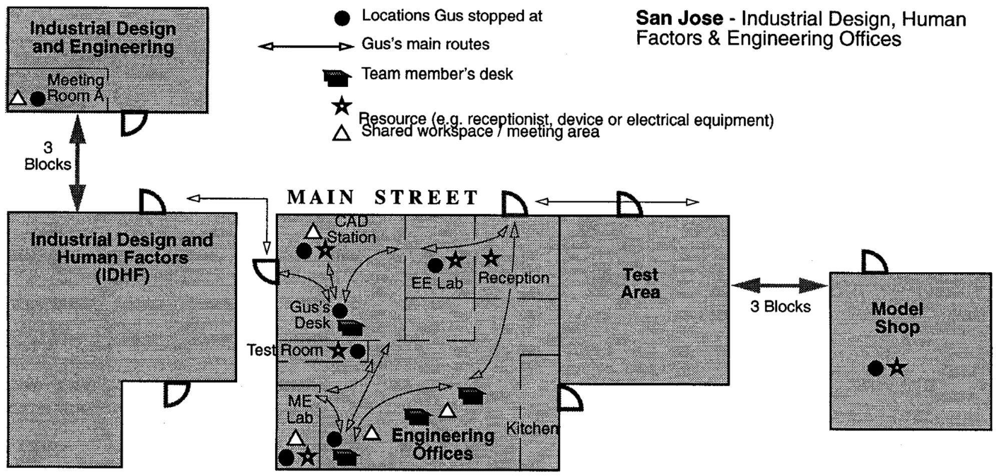
A common example of a remote synchronous collaboration is videoconferencing which enables people to have virtual meetings. However, virtual meetings and collaborative activities can also take place in a virtual world, for example, inside a game.

Finally, asynchronous and remote collaboration allows people to work together at remote locations without being present at the same time. A common example is email, which is in fact an extreme example as there is no reliable notification that an email has ever been read by a recipient. Other examples are instant messaging services and chat applications, some which will inform the sender of a message that it is actively being read, or has been read.

The two-axis model can be used to explore new designs. For example, Stewart et al. [775] explored single-display groupware, which means collaboration is co-located and synchronous. The system enabled high school students to interact on the same display with multiple mice. Usability testing revealed 85% of the children felt using two mice was easier and 98% felt it was more fun. Comments revealed the children felt it enabled peer-teaching ("if [my partner was stuck and] I wanted to help there's another mouse") or just provide agency ("you could do whatever you want") as the system did not enforce collaboration.

Paper Example 8.1.1 : Local mobility in the workplace.

What do people do in the workplace? Do workers mostly sit at their desk or do they move around? Bellotti and Bly [59] studied the movement of workers in a product design team. By observing the movement of workers they found that they spent a considerable amount of time not at their desk, instead sharing resources, communicating, and creating possibilities for information sharing. This contributed to a greater awareness of what was going on in the organization. The schematic below shows the movement of an engineer ('Gus') in the workplace over a single four-hour morning.



However, this distributing working style of local mobility also raised challenges. It was difficult to locate people as people were moving around, which was especially an issue when trying to locate workers by phone. The way of working also meant that remote workers had a lack of awareness as they were not physically present and could therefore not benefit from people walking around in the workplace. It was also observed that communication was difficult between remote workers and that most coordination activities occurred between people working near each other. Finally, coordination was in general more difficult when workers were not co-located. It was observed that while local mobility benefits local collaboration it disadvantages remote collaboration.

Based on the observations and analysis Bellotti and Bly [59] suggest two design goals. First, replicate some of the advantages of local mobility, such as coordination and opportunities for enhancing awareness and taking part of informal communication. Second, reduce friction and make it easier for remote collaborators to communicate, coordinate, and collaborate.

While the two-axis model is useful as a starting point, there are additional factors that govern collaborative systems. Lee and Paine [457] propose a set of such additional factors, which we elaborate on below:

Scale The number of participants involved in a collaboration is important as the larger the group, the more coordination is required. With up to a thousand members in a project or organization, negotiation, grounding, and other collaborative tasks become much harder, as we will see later in this chapter. This means that different interfaces may be needed as systems suddenly have to handle more complex social arrangements and practices. What works for a small group will not necessarily work for a large organization.

Communities of practice *Communities of practice* "...are groups of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly" [864]. The term refers to a concept of learning that describes how beginners learn the practices, tools, and norms of a community. A beginner assumes a role in the periphery of interaction and gradually grows increasingly competent, and thereby close to the center of gravity in the community. Since forming a new community takes time newcomers must be exposed to each other, taught, and mentored, which means communicative resources must be allocated for their benefit.

Nascence The degree of coordination actions that are under development by the participants is called *nascence*. Consider a bulletin board used for bureaucratically defined messages: a highly regulated platform with little nascence. After an initial set up of the information space, it will rarely require any modifications. Now contrast this system with social media apps, which have become sites of continuous adaptation and appropriation. Their interfaces must support creative expression, different media, and other ways of evolving the practices of users. An important part of a designer's work becomes following such evolving practices and reacting to them by continually offering better interfaces to support them.

Planned permanence The intended stability of a collaborative arrangement is another factor. In arrangements that are intentionally short-term, participants may be continuously negotiating collaborative practices. If they fail to establish such practices they may struggle in their collaborative efforts. Consider a project or course work: because of the short-term nature of the group, the practices that are set up may fail to support healthy group functioning, say, safe expression of differing opinions, which may hamper the group's effort.

Turnover The stability of the groups of participants is called *turnover*. For example, in events attended by a large number of people, such as scientific conferences, participants frequently enter and leave rooms, which provides a barrier for the disclosure of private information. Moreover, overhead may increase as practices need to be communicated frequently. Imagine having to organize a continuously changing crowd versus managing a stable team: the challenges will likely be very different.

8.1.4. Collaborative tasks

Just as there is no single ”interaction task” there is no single ”collaborative task”. Research on collaborative systems always starts by trying to understand what is being done. In the case of collaboration, this understanding is very different from the individual-centric interaction that we have focused on so far in this book.

McGrath [517] provides a comprehensive and well-known typology for group tasks. The typology, shown in Figure 8.3, has two primary dimensions: *conflict–cooperation* and *conceptual–behavioral*.

The conflict–cooperation axis is defined by a transition from tasks where members have conflicting interest to those where they are shared. The conceptual–behavioral dimension distinguishes between tasks where thoughts and beliefs, which are conceptual, dominates versus tasks where action dominates.

The typology is further divided into four quadrants:

Quadrant I: Generate This quadrant contains tasks where ideas and plans are generated for cooperation.

Quadrant II: Negotiate This is the quadrant where conflicts are resolved.

Quadrant III: Choose This quadrant shows tasks where an option is to be chosen from a set.

Quadrant IV: Execute This quadrant contains tasks relating to executing tasks and resolving conflicts of power.

Creativity tasks require participants to create novel ideas. The defining goal is the novelty and informativeness of the result that arise. For example, brainstorming is commonly studied as a collaborative task in the context of creativity-support systems.

Planning tasks require the formation of a plan to reach some desired state in some environment. The goal is to produce a plan that meets the stated criteria, such as outcomes related to its efficacy.

Decision making tasks require choosing a desirable alternative among multiple options. Decision making tasks can involve multiple attributes to consider, uncertainty, and even conflicting interests.

Conflict resolution tasks seek acceptable compromises among members with conflicting interests. The goal here is in resolving some existing policy or conflict. The former involves understanding how to act and the latter how to think.

Performances are tasks carried out in light of some objective or standard. Goals here range from meeting a stated goal, to excel, or to win.

To support collaboration by design, the first challenge is to recognize the type of task at hand. The kind of support that, say, a decision task requires is very different from, for example, a performance task.

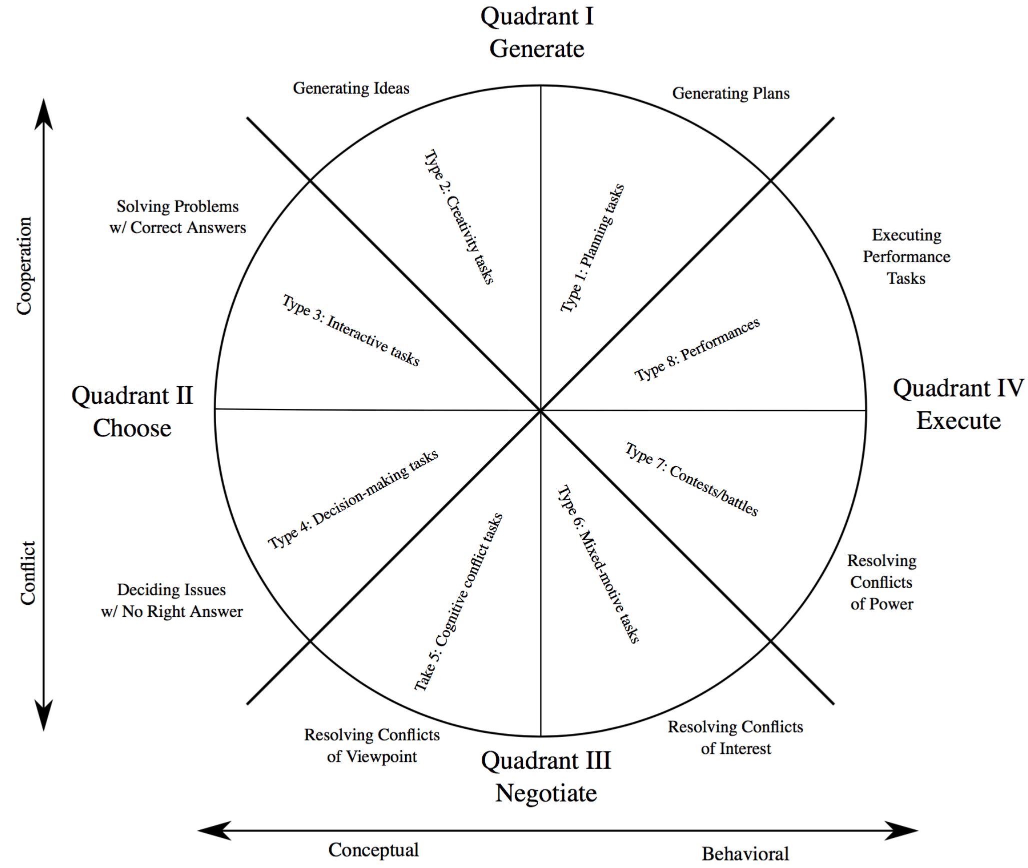


Figure 8.3.: The group task circumplex [517].

8.1.5. Group interactions

The way a group functions is affected by a number of factors, some external and some internal to the group itself. [517] proposes a model of group interactions that outlines the main classes of factors and their inter-relationships. Some of these factors are latent: they are not directly visible in apparent actions. Consider for example the effect of personality or power structures. Complicating this further is the fact that many factors interact with other factors. Consider for example if the group has learned a particular way of acting which does not work in a new setting.

The model by [517] outlines five main classes of factors:

Task Groups perform together to complete something. The objectives, constraints, and other properties of the task affect the behavioral patterns and interactions that emerge. The typology of tasks introduced by the group circumplex earlier in this chapter is helpful here.

Environment Action takes place in some context, with particular resources, constraints, and so on.

Individual attributes The group members' individual properties, such as personality traits, beliefs, and so on affect group interaction, behavioral patterns, and group structure.

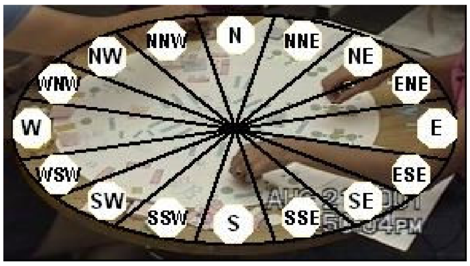
Group structure and relations This refers to facilitatory and impeding factors governing inter-relationships among group members, such as affection, power, and so on.

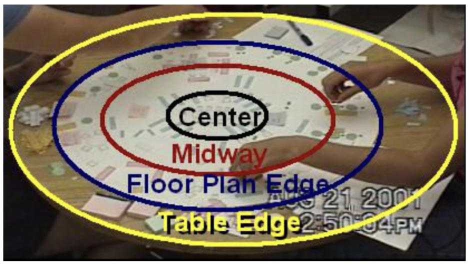
Behavioral patterns Constructs that shape the expected or routine way of interaction are called *behavioral patterns.* Examples of such patterns are the assumed roles and divisions of labor within a group.

Paper Example 8.1.2 : Territoriality when collaborating using tabletop displays.

Tabletop displays allow new forms of collaboration. However, it also involves many design decisions, such as whether these systems should enforce ownership of digital content on the workspace, or automatically reorient items, for example, by allowing users to define regions for this purpose in the workspace.

Scott et al. [730] investigated users' territoriality on regular tabletops in two studies. The first study examined users who solved puzzles collaboratively in a casual setting. Among other things, the study revealed that the participants partitioned the space into three different interaction areas which the authors call personal, group, and storage. The second study investigated users who collaborated on a furniture layout planning activity in a more controlled laboratory setting. In this study all tabletop activity was recorded. To assist the analysis, the tabletop region was divided into 16 directional zones, which are shown in the figure below to the left, and four radial zones, as illustrated to the right.





The results show that the users define territories on the tabletop. The personal territory enable users to reserve a certain area and certain set of resources of a shared tabletop space for their own use while group territories allow users spaces to collaborate, such as collectively solving a puzzle together. Storage territories are used to store task resources that are not currently in use, such as tools, and non-task items (such as food and drink). Knowledge of such territorial behavior allow the design of digital tabletops that better support users natural collaborative instincts.

For example, consider a hypothetical case of a distributed software team. This team is using email to decide between features for a next product update. How might the factors that govern group interactions be present within this context?

First, assuming that some options for features are already known, the task would be a decision making task. The challenge in this type of task is to understand the value of various options and being able to compare them in light of some set of objectives. It may be then expected that some of the group exchanges relate to trying to define what the objectives are and what makes a desirable feature for this update. One can also expect exchanges about the options and their perceived positive and negative qualities Second, the environment consists of the email clients, the organizational practices, and the artefacts. However, the environment is open-ended, since anything on the Internet can be used to achieve the task. Third, the researchers would look at the individual properties of the team: introverts and extroverts, skeptics, promoters, and so on. Fourth, group relations can affect those who may feel responsible for taking initiative or giving orders to others. Fifth, the group may have some patterns on how software updates are led, for example, a particular member initiating discussions, or someone in the group stepping into a particular role.

8.1.6. Communication acts

Interactions within a team, especially when we talk about computer-mediated collaboration, are *communication acts*. They can be one-to-one, as when member A communicates with member B. However, they can also be one-to-many, such as when member C communicates with members A and B.

These communications can be about the task, for example, they can be about choices, negotiations, or preferences. However, in group interactions, task-focused messaging may be insufficient. The group process itself then needs to be organized so that it can function efficiently. To this end, communication acts may have *interpersonal purposes*. This means that these acts may seek to attract or convince another member about competence. They may also attempt to directly influence the group process, for example, by dividing up labor or by establishing a process or form for collaboration.

Effective performance of a group boils down to communication acts that establish the task and the processes that the group should follow. What we observe as a functional or dysfunctional performance is a result of such efforts.

McGrath [517] distinguishes three group problems affecting group's performance: (1) consensus on underlying values and goals; (2) ensuring or developing resources and abilities to achieve those goals; and (3) developing norms to guide the group's work.

8.2. Coordination

Coordination refers to the "construction and maintenance of a shared conception of a problem" [688]. Outside of HCI, everybody knows mechanisms for coordination. Sharing a project plan, glancing at what fellow puzzle players do, shouting to a biking friend to slow down, these are all examples of coordination.

So why do we need to coordinate? One answer is due to dependencies of different tasks we wish to carry out. A user working on one task might require the results from another task that another user is engaged in. Another, related, answer is that these task dependencies are not static, that is, they tend to change. If a task dependency changes, people affected by the changes need to be informed in some way.

So how do users coordinate? One answer is through artifacts [724, p. 162]:

> Artifacts have been in use for coordination purposes in cooperative settings for centuries, of course – in the form of time tables, checklists, routing schemes, catalogues, classification schemes for large repositories, and so on. Now, given the infinite versatility of computer systems, it is our contention that such artifacts in the form of computational coordination mechanisms can provide a degree of perspicuity and flexibility to artifactually supported articulation work that was unthinkable with previous technologies, typically based on inscriptions on paper or cardboard.

Another coordination mechanism is a *protocol*. Protocols define socially dictated distributions and constraints, such as a checklist that tells a pilot and a co-pilot who does what, and in what order. However, these protocols are frequently not prescriptive. This means that after users have learned the protocols, they tend to deviate from them when they have reasons to do so.

8.2.1. Articulation work

One finding in HCI is that one cannot 'just' work. The concept of *articulation work* describes activities that are extraneous to the work itself. From a productivity perspective, it is *additional work*. However, this is a limited view.

Articulation work is about getting things done in work in a way that is situationally more appropriate. Its purpose is to maintain and define work with others. Articulation work is about deciding how tasks are carried out, for example, how tasks are scheduled, divided, managed, aligned, and organized into larger clusters.

Even a strong organizational procedure does not predetermine the way work actually happens. Work emerges through articulation work among members of a group. Articulation work is required to decide how the work is carried out. In practice, this means that articulation work refers to tasks that mediate individuals in work. It contains task allocation, task scheduling, and practices of work—how work is carried out.

As an example, Nardi et al. [563] use the term *outeraction* for instant messaging that aims to create conditions for social exchange with others at the workplace: "Outeraction is a set of communicative processes outside of information exchange, in which people reach out to others in patently social ways to enable information exchange." [563, p.79]. While instant messaging is normally considered as a channel for quick clarification of questions about ongoing work and a way to keep in touch with friends and colleagues, it is also used to set conditions for collaboration at work. For example, workers used it to see who is available for communication [563, p.83]:

> First thing this morning I opened it up [the buddy list] and looked to see who was online. My boss was online and I saw that people in Commerce were online. Other designers were online and I knew that there was a certain person that I wanted to contact and she wasn't there so I knew that I could check later.

Workers also sometimes used instant messaging to 'probe' other users. For example, "Suzi?" was a message that, when answered to, formed an almost attentional contract: for some time period after sending the response, the respondent would be available for messaging. Workers also used instant messaging to move conversations to some other form of media, which might be more suitable for communication.

Articulation work is important not only in teams but also in multi-site distributed work. For example, Neang et al. [567] report on articulation work among data scientists. They interviewed 43 ocean science researchers in 22 laboratories in three countries and discovered that open science does not simply mean that a data set is released to others in a 'dead drop'. Data scientists do not only provide a dataset for others to use, waiting for others to arrive and use it. Instead they work with other groups to define if, and how, other people's datasets can be transferred, and what the benefits would be. Such conversations bring together those who collect datasets and those who use them to, for example, build computational models. Researchers engage in discussions on how datasets can be used across sites using different software and following different practices.

Articulation work is important to be aware of in HCI since the design of an information space for work is not only about supporting the work itself but also about enabling the articulations needed to make such work possible. The interface required to support the work that makes work possible is an aspect which HCI can help address.

8.2.2. Awareness

A collaborator's ability to follow what others are doing, how their subtasks are progressing, and what they attend to is called *awareness*. Maintaining awareness is easy when others are co-located, or next to you. However, when collaboration is distributed in space and time, maintaining such awareness is more challenging. When awareness breaks down, coordination of work becomes hard and prone to errors, and even conflict.

Awareness systems are interactive systems that sense and mediate *cues* about distant collaborators [207]. For example, an awareness system may show a live video stream from a remote office or it can be a implemented as a small symbol in a user interface that denotes that a particular worker is at their desk.

Awareness systems help collaborators coordinate their work. Simply knowing where a person is becomes an indication of what can be done with that person. Awareness systems, and articulation work in general, can also help non-work related psychological functions. It can be about friendship, emotional support, and so on. These awareness systems can have a surprisingly strong effect on the coherence and performance of a group.

8.2.3. Boundary objects

Another prominent way of understanding the sharing of information and artifacts has been through the concept of *boundary objects*. A boundary object is an object that is shared among collaborators and helps them coordinate, share information, or coordinate. Consider, for example, a map that is shared in a dispatch center. Good boundary objects are hard to design. Star [772] writes

> both plastic enough to adapt to local needs and constraints of the various parties employing them, yet robust enough to maintain a common identify across sites. They are weakly structured in common use, and become strongly structured in individual site-use. Like a blackboard, a boundary object "sits in the middle" of a group of actors with divergent points of view.

8.3. Group cognition

The process of learning and negotiating a shared representation of a problem is known as *group cognition* [771]. Research on group cognition comes from studies of small group learning and ideation. A central tenet is that shared knowledge is constructed in a process of mutual learning and negotiation. The diverse views of members are learned, and a joint understanding is established via reciprocal action. Compared to the theories discussed so far, it provides an alternative view of collaboration. It emphasizes the formation of joint beliefs, or common ground, as a central aspect for a group to function.

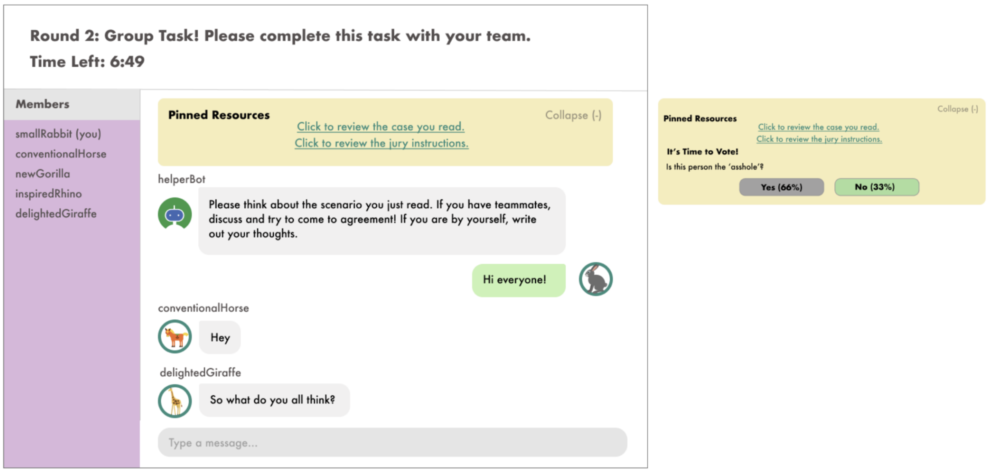
Group cognition emphasizes that the basis is mutuality: members construct not only their own interpretations but also interpretations of each others' beliefs. That all members engage in learning and negotiation assumes that all are assumed to be legitimate partners in collaboration. This has been argued to improve the experience of ownership over a joint activity. Achieving group cognition can require that participants with very different viewpoints search for new understandings that may challenge their earlier beliefs. This may be difficult to support through computer-mediated activities.

Paper Example 8.3.1 : Do online juries work?

A jury is a common method to make decisions in many judicial systems and frequently jury decisions are made using an online jury. However, for a jury to be effective it has to be consistent. That is, it has to make similar decisions in cases that share similar characteristics.

Hu et al. [354] studied whether online juries make repeatable, consistent decisions in an online experiment involving 1121 workers on the online crowdsourcing platform Amazon Mechanical Turk. The research question was whether individuals or groups make consistent decisions.

The participants were presented with a 'case' and jury instructions and asked to deliberate via an online chat system, as shown in the figure below.



Round 2: Group Task! Please complete this task with your team.
Time Left: 6:49

Members
smallRabbit (you)
conventionalHorse
newGorilla
inspiredRhino
delightedGiraffe

Pinned Resources
Click to review the case you read.
Click to review the jury instructions.

helperBot: Please think about the scenario you just read. If you have teammates, discuss and try to come to agreement! If you are by yourself, write out your thoughts.

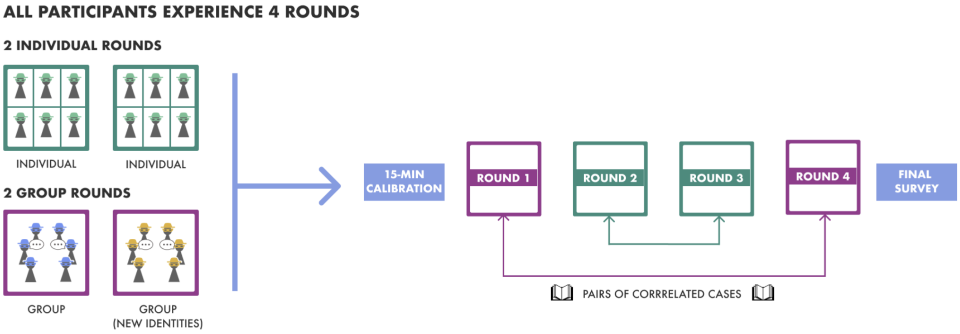
Hi everyone!

conventionalHorse: Hey

delightedGiraffe: So what do you all think?

Pinned Resources
Click to review the case you read.
Click to review the jury instructions.
It's Time to Vote!
Is this person the 'asshole'?
Yes (66%) No (33%)

Hu et al. [354] used a within-subjects design to investigate this research question. Each participant participated in four jury deliberations. In the first deliberation participants deliberated as part of a group, as shown in the figure above. Then there were two successive deliberations where the individual juror considered a case on their own. Finally, each participant was part of a group deliberation for the fourth case. Importantly, the group deliberations involved the same individual participants but with different identities. This process is illustrated in the figure below.



Overall, Hu et al. [354] demonstrate that both individuals and groups can make repeatable, consistent decisions when participating in online juries.

8.3.1. Intersubjectivity

Consider the following situation. You and two friends are messaging about evening plans, discussing where to go. One of you suggests going ‘to the usual place’, and the rest of you agree with thumbs up. The amazing brevity of this interaction was possible because of co-participated shared knowledge—all members of the group knew what ’the usual place’ is and why it is a good choice.

Now consider what would change if one of the three members of the group were changed to an outsider. How much more effort would have been required to achieve the same outcome? This example illustrates an important point for HCI: collaboration is impossible unless there is some shared understanding of ends and means. This shared understanding must be achieved somehow, and the design of technology provides context and constraints to achieve that.

This topic has gained significant attention in philosophy, social sciences, and management sciences, where this type of shared understanding is called *intersubjectivity.* More precisely, it refers to how two or more individuals interrelate. That is, understand each other with the purpose of acting together. For design, the implication is that the target of design lies beyond the interface itself. A question a designer must ask is not which features or contents are available, but how those features affect the joint construction of intersubjectivity. In the rest of this section, we examine different activities involving intersubjectivity.

8.3.2. Grounding

Grounding refers to the active creation of shared knowledge related to a collaboration. Grounding is one mechanism for intersubjectivity. A participant does something in order to update the knowledge of another participant. Grounding has a role that is more prominent in collaborations on a smaller scale. Groundings can express, for example:

Objectives What needs to be done.

Focus of attention What is being done now.

Beliefs Beliefs about the task or the environment.

Two types can be distinguished based on modality. In *embodied grounding*, bodies are used for grounding. An *indexical gesture* refers to a gesture that specifies an object of interest for the partner. For example, pointing or looking at an object can index it. For example, a lecturer may move the cursor on display during a slideshow in order to index an object during a talk. The flick of the cursor on top of an element reduces the ambiguity about what the talk refers to. *Linguistic grounding* achieves the same purpose but uses language. A participant can then directly state what others should think or believe.

As an example, in a study of information-sharing in an emergency department [627], a registration associate (RA2) encounters two patient records for the same patient. This ambiguity must be resolved in some way. The RA2 asks EM2, a member of a medical service team (EM2). EM2 suggests that the patient was treated at another hospital, which RA2 then confirms using an information system: [627, p.325]

> RA2, a registration associate, is registering a patient and finds that the patient has two different medical record numbers, one that she found in Eclipsys and the other provided by the patient. RA2 is looking in Eclipsys, trying to figure out which one is correct. The EMS member, EM2, who brought in the patient, is standing next to her. RA2 asks EM2 whether he knows why the patient has two different medical record numbers.
>
> EM2: "Is that because she went to [another hospital] for a while?"
>
> RA2 (looks through Eclipsys): "Ah, I found it. It seems she hasn't come to us in a while. What is the address you picked her up at?"
>
> EM2 tells her where he picked up the patient. RA uses the address information to verify that she has the correct record number and makes a note of it in Eclypsis.

So far we have discussed grounding as a one-way activity: a kind of declaration to co-present to other members of a group. However, what if the other group members fail to notice a grounding? This can, and will, happen. It may, for example, occur due to users having an inability to perceive it, or due to a misunderstanding of the grounding act. This is why participants often need to confirm the effect of grounding. This can be achieved by directly asking people. However, more frequently there are cues such as 'hmm' or a nodding that help infer reception. In some more experienced teams, collaborative practices have developed to offer a "receipt". In social media, this can be a thumbs-up emoji or simply a 'k'.

An intriguing trade-off has been noted in grounding behavior. A participant can choose to invest effort into being very clear, and thereby avoiding effort in repairing misunderstandings. However, there is also another possibility. The participant can choose to be more sloppy in the initial act, but be ready to engage in repair. *Least effort* groundings are specific and simple responses and conversational agents may use such tactics to balance this tradeoff. If an actor misunderstands an utterance, it may need to ask for more information to help grounding. If this happens frequently, such requests should be kept as quick as possible to avoid encumbering the user.

8.3.3. Theory of mind

In the Sally–Anne experiments a young child sees an adult insert an object into one of two drawers. After the adult leaves the room, another adult takes the object and puts it into the other drawer while the child is watching. The first adult then returns and the child is asked which drawer they should open to retrieve the object.

The problem may appear trivial but it is not. The ability to attribute mental states and infer their contents on others is called *theory of mind*. It develops relatively late in human development. Knowing what the collaborator knows is important for predicting what a collaborator may do. Another aspect of theory of mind is awareness of one's own beliefs and their relevance to the collaboration. If you know that the other partner may have a different belief than you, you may invest effort into communicating or harmonizing that understanding.

Theory of mind is also an aspect of human collaboration that computers struggle with. People may at times expect capabilities from computers that they do not possess, especially if the computer appears human-like. Although a computer can receive an explicit command from the user, there are presently no methods that allow computers to infer what the user may be thinking or believing. This makes collaboration more effortful. In the absence of this capability, interfaces must resort to hand-crafted rules to ask for confirmations, such as ”Are you sure you want to do this?” With a theory of mind, this might not be needed.

Some recent research on artificial intelligence has looked at the possibility of theory of mind in computers. They have found that in cases where the human partner's beliefs can be easily inferred, this can be exploited to improve recommendations. Interestingly, if the user knows this, this capability can be exploited. Theory of mind thus relaxes communication requirements.

8.4. Why collaborative systems fail

Understanding collaboration is important in HCI to design successful *collaborative systems*, such as messaging services, videoconferencing systems, and shared calendars. The terminology for such systems vary. In the past, such systems were often called *groupware* or *computer-supported cooperative work (CSCW) applications*.

Early research on collaborative systems focused on understanding *why* such systems failed. A review [297] lists eight challenges emerging from detailed investigations of failed collaborative systems:

Disparity between work and benefit Users have their own experiences, preferences, and goals. To maximize uptake each user should benefit, although perhaps not equally. However, collaborative systems tend to not provide the same benefits to all members of the group, nor do they tend to require the same amount of work. Such disparities may lead to limited uptake of systems. This is problematic as collaborative systems tend to only be successful if a critical mass of users choose to use the system.

Critical mass and prisoner's dilemma For collaborative software to provide benefits to group members, it needs to be used by many, perhaps all, members of a group. Moreover, some systems can induce situations where pursuing personal benefits leaves all members, and the group as a whole, worse of. This is analogous to the problem in game theory known as *the prisoner's dilemma*, which we discussed in the beginning of this chapter.

Disruption of social processes Collaborative systems sometimes promotes activities that clash with social, political, and motivational factors. Outside of collaborative systems, such factors may be implicit, changing, or depending on subtle negotiation. For example, a joint calendar system may suggest meeting times that appear to be unscheduled among participants. However, the absence of a calendar entry does not necessarily mean that the time is available. As another example, a joint decision making system that records each user's position may be deemed politically unacceptable as having a record of individuals' opposing opinions at the time may be to these users' disadvantage at a later time.

Exception handling Grudin [297] notes that work processes can be thought of as either being (1) the way things are *supposed* to work; or (2) the way they *do* work. In reality users engage in a range of activities, *exception handling*, to make things work when the supposed work processes are not fit for purpose. When collaborative systems assumes work is solely organized as how things are supposed work the systems inevitably fail to support contingencies, improvisations, and odd cases that characterize established work practices. This lack of support for how work is actually organized in practice may lead to systems not being used or accepted.

Infrequently used features In practice, individual work is frequent and most organizations try to minimize communication and communication overhead to enable workers to more efficiently achieve their tasks. As a consequence, collaborative systems that overemphasize collaboration and frequently cause individual tasks to be harder to accomplish may fail. Infrequently used features in collaborative systems should therefore not obstruct frequently used features that workers require to carry out their individual work.

Difficulty of evaluation Evaluating collaborative systems is more difficult than typical single-user systems. First, the benefits that a collaborative system bring are difficult to measure. Which metric can you put in place to demonstrate to managers that System A is better than System B? Second, lab studies cannot capture organizational complexities which means collaborative systems have to be evaluated when they are deployed into organizations. Third, evaluation of collaborative systems is likely to take time as substantial benefits may only materialize after extensive usage. Fourth, collaborative systems typically have to be evaluated with imprecise qualitative methods, which demand expertise in their application. Fifth, it is difficult to generalize the results as some groups of users may find ways to use a flawed collaborative system while other groups of users may struggle with a poorly installed but otherwise excellent collaborative system. These factors make it difficult for designers and researchers of collaborative systems to learn from experience.

Failure of intuition Users rarely have a full picture of the systemic effects of a collaborative system, yet they can form strong opinions based on their individual experiences and expectations. While intuition may sometimes serve as a guide in single-user applications, it fails when there is a need to anticipate the effects of extra work required by people, which will likely result in resistance and neglect.

The adoption process Collaborative systems are complex and when they are introduced they require proper training of workers and technical support to achieve widespread adoption. Further, the different stakeholders affected by the collaborative system should be engaged as active stakeholders in the process of designing and installing the system to motivate workers and encourage a sense of ownership of the system.

The eight challenges make it clear that there are substantial difficulties to overcome in designing successful collaborative systems. However, they also highlight that the key to a successful outcome is understanding how people choose to collaborate, coordinate, and communicate as a group of users.

Summary

- Collaboration is a mutually beneficial relationship between two or more parties who work toward common goals by sharing responsibility, authority, and accountability for achieving results

- The central challenge in supporting collaboration with interactive systems is that they are rigid, scripted, and rule abiding while social activity is flexible, improvisational, and negotiated.

- Articulation work are activities that are extraneous to the work itself to get things done in work in a way that is situationally appropriate.

- The process of learning and negotiating a shared representation of a problem is known as group cognition. Intersubjectivity is a shared understanding between group members. Grounding and theory of mind are two mechanisms to achieve intersubjectivity.

Exercises

1. Social gap. Ackerman [9] describes the social gap, that is, the gap between what we can support technically in current systems and what human activity requires (see page 8). Choose a specific activity that interests you, such as participating in an online forum, playing with other people in a computer game, or co-authoring using a collaborative document editor. Describe the social gap between this activity and the technology platform that you are using.

2. The matrix model of collaboration. Consider four collaborative activities in which you are participating using some form of collaborative system. Map these four activities to the synchronous–asynchronous, co-located–remote matrix model. Now, describe how the nature of these activities would change if they are moved to another location in the matrix. Would it still be possible to allow users to achieve their goals? What would be the issues that would arise, and how could they be resolved?

3. Analyzing collaborations. Use the factors suggested by Lee and Paine [457] to describe the following collaborative activities: (1) volunteer editors determining editing policies on Wikipedia; (2) a small firm consisting of experienced employees and new hires that are using a shared spreadsheet to calculate a budget for a business proposal; and (3) online learners working on a joint lab report.

4. Why collaborative systems fail. The eight challenges explaining why collaborative systems fail [297] was published in 1994. Since then, many collaborative systems have been deployed and some of them are widely used. Reflecting on the systems you use yourself, which of the eight challenges are the most relevant? Are any of these eight challenges now fully addressed?

9. Communication

Communication technology is a strong candidate for the most influential technology in human history. The communication technologies that we use started as disruptions but gradually became woven into the fabric of our daily lives. Email, social media, video conferencing, and instant messaging have transformed the way we work, learn, plan, stay in touch with others, and even lead our romantic lives. It is remarkable how diverse these technologies are. Instant messaging is an asynchronous one-to-one textual channel. Email is a textual, asynchronous channel with a small, definite audience size. Video conferencing, in contrast, is a multimodal, but synchronous channel. Social media is a multi-modal asynchronous channel from many to many. Each of these has numerous variants.

The field of *computer-mediated communication*, or CMC for short, employs a variety of methodologies and theories to understand what is special about communication when it is mediated by technology. CMC studies the following questions: When conversation partners are unable to experience each other in the 'usual' way, in person, what happens to communication and, in the longer term, to our social relationships? To answer this question, researchers use log analysis and large-scale surveys, charting what people do online and why. They interview users, record conversations, and perform controlled experiments to study how social relationships develop.

Although most of us use communication technologies, our intuitions may not reflect the general tendencies or causes behind our behaviors. When communication acts are digital, they can be separated in time and space in surprising ways and adopt formats that would not occur in face-to-face interaction. Many things change, from how conversations are structured to how people are presented to each other (Figure 9.1). This means that lay expectations may simply not hold. Indeed, research on CMC has led to some surprising findings.

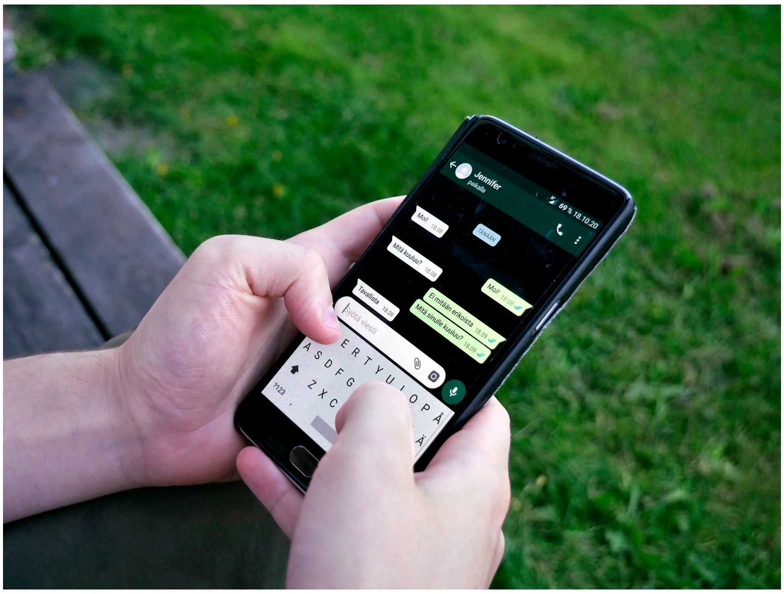
- In 1985, Kiesler et al. [407] compared how people communicate in face-to-face exchanges when they were in separate locations, and when they write text to each other on a computer. They thought that participants in the computer condition would be less inhibited in their communication than those in the face-to-face condition. To investigate this hypothesis, they counted the times the participants wrote something impolite, how often they swore, and how much they flirted. The computer-mediated communication channel led to more uninhibited communication. Although we now know about trolling and flaming on social media, the result was surprising in 1985 and foreshadowed the phenomena we now see.

- Bos et al. [89] studied if trust formed in communication with others is affected by the type of communication technology used. This was studied using a social dilemma game. Participants could invest tokens as individuals or as a group. If the players invested as a group, the reward was greater. They played the game while communicating via face-to-face, video, audio, or text chat. The authors found that text chat performed the worst in terms of investment, while the video and audio conditions were almost as good as face-to-face communication. Still, the latter channels took longer for full trust to build and were more vulnerable to opportunistic behavior. To explain this, we need to understand how the cues that a channel allows affect the formation of trust.

- In virtual reality, people may experience that a virtual body feels like its own. When the virtual body moves in synchronicity with the real body, such body ownership may be felt strongly. Embodying a virtual body that speaks makes participants misattribute the speaking to themselves and, surprisingly, shifts the fundamental frequency of their voice when they subsequently speak toward the voice of the virtual body [39]. Therefore, body ownership can fundamentally affect communication.

The results of research on CMC have helped develop new systems to facilitate communication; see Figure 9.2.

In the next section, we start our look at communication by contrasting face-to-face conversations with how we might communicate through interactive systems. The main lesson there is that computer-mediated communication is fundamentally different in



Moi! 18.08
Mitä kuuluu? 18.08
Moi! 18.08
Ei mitään erikoista 18.09
Mitä sinulle kuuluu? 18.09
Tavallista 18.09

Figure 9.1.: In computer-mediated communication (CMC), many things are different from unmediated communication: the modalities we use for communication, how conversations are structured, the synchronicity of communication (how real-time it happens), how other people are presented etc. Such differences affect the content, style, and consequences of communication. Picture by Santeri Viinamäki, shared under CC BY-SA 4.0.

many aspects of communication. We then dive into the structure of conversations, which is a key to understanding so that we understand how to mimic or replace them in interactive systems. Communication may not take place only between two persons. Thus, we look at distributed groups and networks of people communicating synchronously or asynchronously, with social network sites as a prime example. Here, the focus shifts from individual acts of communication (e.g., messaging) to the formation and maintenance of social relationships. This changes communication and is related to collaboration (see Chapter 8). Finally, the computer itself can be thought of as a conversation partner. Recent advances in AI have made it possible to converse with artificial agents, such as intelligent agents. We have intelligent assistants at home and in our cars with whom we can talk. The question is not whether they are special as communication partners, but how.

9.1. Beyond face-to-face communication

Human-to-human communication is diverse and has been shaped through millennia, first through face-to-face communication and later via written communication. Although communication technology is a relatively recent phenomenon, its evolution has been fast and dramatic. Many systems that were once popular have been later overtaken by others. For instance, Usenet was created around 1980. Therein, users could read and post messages in a list of categories; they could also reply to messages. This technology extended earlier bulletin-board systems, and the idea of threaded messages, which was used in some clients, is still widely used today, for example in email clients. However,

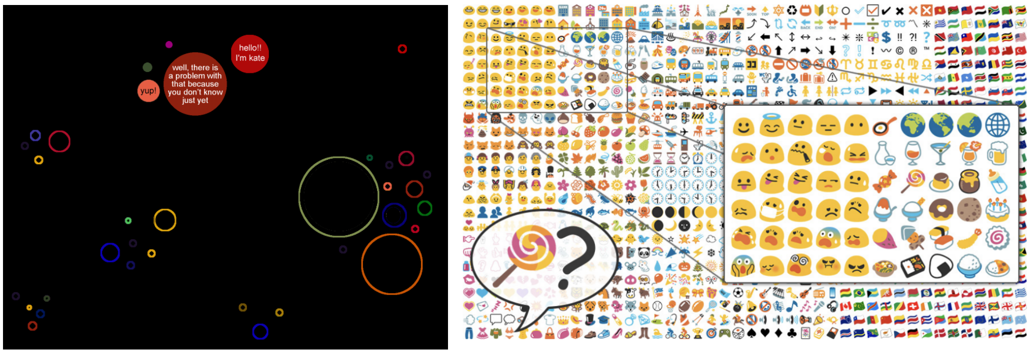


Figure 9.2.: HCI studies interfaces and systems for human-human communication. To the left is shown an interface ChatCircles [833]. Rather than the typical list of messages ordered by time, users and their chat messages are shown as colored circles. Circles fade out over time and can be moved around the display, allowing nearby to be read. To the right is shown an interface for picking emojis [649]. Emojis has transformed communication, but are still cumbersome to input, slowing chats and limiting expressivity. In this interface, users zoom to select emojis to improve communication.

Characteristic	Human–human conversation	Computer-mediated communication
Modalities	Speech, hearing, vision, touching	Potentially any human modality
Cues	Speech, gestural, facial	Potentially anything including speech, gestural, facial, textual, paralinguistic
Structure	Conversation	Conversation, but also many other forms
Synchronicity	Synchronous	Synchronous or asynchronous
Audience	Others within a few meters	Potentially any group of people anywhere using the system
Identity	Negotiated	Designed; constrained by system
Power	Socially defined	Socially defined but also enforced by system
Norms	Existing norms and etiquette for conversations	Emerging and negotiated

Table 9.1.: This table shows eight characteristics of human to human conversations as they take place face to face. Those characteristics are affected when we communicate through computers.

we suspect that most of our readers have never heard about or used Usenet. Besides Usenet, many other systems captured the interest of researchers, including Internet Relay Chat, Short Message Service (SMS), e-mail, wikis, blogs, instant messaging, video conferencing, and different branded social media sites. Currently, an increasing proportion of communication occurs through interactive systems (see, e.g., Olson and Olson [601]).

However, little is gained by enumerating such systems. One must understand the characteristics that define them as communication channels. Only then can we begin to understand how those characteristics affect the communication behaviors that emerge. Unique to HCI's research on this topic is the attempt to expose the relationship between the designed features of a system and patterns that emerge in communication.

Table 9.1 lists seven defining characteristics that may be compared with typical face-to-face conversations. Communication technology offers different *modalities* for communication. In the context of communication, the term modality refers to the presentation method of a communication act. We presently have, among others, textual, auditory, pictorial, and video-based modalities. However, there is experimental research in HCI with other modalities, such as touch. For instance, Samani et al. [706] created an interface that allows people to kiss each other at a distance, transferring the kiss from a receiving device and transmitting it to a distant conversation partner.

Mediation also affects *cues* available for communication. In everyday conversations, communication is not just about what we say: body language, facial gestures, prosodic features like pauses, and direction of gaze all affect the way the other person can interpret the utterance. Compare this to the following exchange with a friend:

> You: Can you pick me up at 3pm from the school?
> Friend: Fine.

What do you think the friend may feel, given this response? Is the friend positive about the favor, or does the period signify dissatisfaction? The non-availability of cues has been a major topic of research in computer-mediated communication. Paper Example 9.1.1 shows how the lack of cues from the gaze of the eye can affect communication.

The *structure* of communication refers to the way that communication acts are organized and represented in the system. Computer-mediated communication supports the traditional conversational structure, where verbal expressions are taken in turns. Consider, for example, the way group messaging is organized in popular applications: in reverse chronological order with the latest message on the bottom. Constraints are often imposed. For example, posts in social media often have a character limit. Text messaging only ten years ago was limited to 140 characters. But technology has also enabled quite radically different structures; for example, computer vision -based overlays on faces shown on video communications.

Synchronicity denotes the simultaneity of communication (see also the taxonomy presented in Chapter 8). In asynchronous communication, the receiver controls when to respond to a message. It can be now, in near real time, or it can be much later, even weeks. In synchronous communication, by contrast, no such control is available, but the receiver must attend to the message as soon as it is sent. This seemingly small feature has significant consequences. Most importantly, it makes it possible to maintain several communications simultaneously since a message does not need to be attended at the time of receiving it.

The traditional boundaries that define who we address and how are redefined in computer-mediated communication. *Audience* is no longer defined by those within shouting distance, but can be anyone who has access to a particular communication system. While one-to-one communication is closed to two defined individuals, communication can occur with arbitrarily defined, and even unspecified audiences. Consider a group messaging service where messages can only be read by named individuals, or social media applications where a post using a public profile can be ready by any user of the system.

Another factor is the *power relationship* among the participants. The system can, by design, nominate a subset of participants with special access or powers. Consider, for example, discussion moderators or group owners, who can decide the audience and permissible topics in a forum. Finally, *identities* can be redefined in ways allowed by the channel. Users can take up on names, avatars, and even arbitrary histories available to others. This is captured in the New Yorker comic subtitled ”On the Internet, nobody knows you are a dog.” Thereby, users get new opportunities for self-expression, but also for subversion and exploitation. However, *norms* place restrictions on what is socially and culturally appropriate. While norms are not a designed aspect of a channel, they influence their use so much that ignoring them would be a mistake.

This section has explored some differences between technologies. Next, we turn to how these affect what actually goes on in face-to-face conversations between people. This turns out to be complicated and, yet, necessary to understand.

Paper Example 9.1.1 : Eye gaze in video conversations

While we have mostly discussed verbal communication, embodied cues like gaze and gestures have a significant role especially in video-based communication. Video is popular as a communication medium between pairs or small groups of people. Video communication typically uses a camera (e.g., on top of a laptop) that is placed in a different place from the representation of the person who is being talked to. However, as you may have noticed when using applications like Zoom or Teams, this creates a mismatch between where a person is looking and the camera image of that person.

Studies of face-to-face communication have shown that even slight displacement of where you are looking significantly impact conversations. To illustrate this, in your next conversation, look down about 1 centimeter (0.4 inch) from your conversation partner's eyes towards their nose. In video conferences, it is hard to say to whom a person is talking to and we must use other means to infer this, which may increase workload and make it harder to control turn-taking.

Vertegaal et al. [832] studied how the movement of camera can affect. Their findings suggest that camera shifts can be distracting when they are large. In their study, larger than 8 degrees of visual angle were particularly distracting. To help create more natural video conferencing, researchers need to solve how to communicate via eye gaze via video.

9.2. Conversations

Let us go a little deeper into conversations. Their importance is obvious: We engage in conversation to transmit knowledge, coordinate activities, form beliefs, show identities, and conduct social transactions. In other words, conversation is the main means of social interaction. Although we routinely participate in conversations as participants, we may be blind to their underlying structures.

These structures become visible when they break down. Consider the following exchange, for example:

> A: How are you doing?
> B: I love cheese.

Clearly, this is not an adequate response. It would lead to disruption and the need to repair or stop communication. But *why* is it not an adequate response? After all, it is a linguistically correct phrase.

Conversation analysis (CA for short) refers to the study of how *order* is interactionally achieved in conversation. It become a popular topic in sociology, when Goodwin and Heritage [279] asked a foundational question: How is social interaction possible? Their emphasis shifted from mental constructs to social and cultural constructs, especially rules, procedures, and conventions. Although these constructs are present, they are not overt in communication. Participants in a conversation have an expectation on what those rules are and what they mean, in other words, how one ought to behave in a particular circumstance. However, these constructs can be exposed with CA.

Before Schegloff and colleagues' work, regular talk among people was seen as a degenerate form of communication. The prevailing idea was that linguists should focus on idealized sentences. Schegloff and colleagues challenged this view. They asked how *casual* conversations are organized. In their view, the answer centers on choice. When given the freedom to choose what to respond, participants choose to respond in a particular way. Why? By contrast, heavily scripted conversations, for example in the courtroom, restrict the choice of the participants and therefore diminish the evidence one can collect on the involved conversational practices.

Conversation analysis subscribes to an interactional view of communication. That is, conversation is seen as a dynamic process where the ultimate outcome depends on actions taken by both parties. What a participant 'intends' at the outset does not predetermine the actual outcomes. Instead, outcomes are shaped as a conversation unfolds. The participants, via their choices, steer and modify the conversation. Each action or utterance is understood within the context of the conversation, including its outcome.

Symbol	Feature of conversation
(.)	Pause of less than 2/10 s
(0.3)	Pause of specific length (s)
:	Extension of the preceeding vowel
you know	Stress or emphasis
↑ ↓	Rising/falling in intonation
[ ]	Overlapping talk
delete-	Cut-off sound at
><, <>	Faster, slower speech
°don't°	Quieter speech
(chillin')	Unclear speech (guess)
DON'T	Louder speech
=	Continuous talk between speakers

Figure 9.3.: Conversation analysis uses several symbols to present turntaking and temporal structure in conversations (based on [872])

Unique to conversation analysis is its focus on ”real” or ”ordinary” communication events. This makes the approach suitable for studying everyday computer-mediated communication. The recording and transcribing of talk is the starting point for conversation analysis. Conversation analysis does not stop at transcription. It offers a *transcription system*, the goal of which is to expose techniques that participants use to achieve order. The system consists of symbols that denote pauses, stress, intonation, overlap, cut-off, tempo-changes, and loudness in conversations. A list of basic symbols in transcriptions is shown in Figure 9.3. The way communication partners take turns, sequence and time their responses, and how they engage in guiding the other partners become visible via the system. The transcription system is generic in the sense that almost any verbal exchange among two or more participants can be transcribed with it.

The transcription system may appear rigid and heavy. Whereas an audio tape can be transcribed at a rate of 4:1 (e.g., four hours to transcribe one hour recording), conversation analysis takes an order of magnitude more. However, the advantage of CA is that it allows a rigorous and reproducible approach to understanding conversations via turn-taking. It also offers a systematic presentation of empirical evidence. The transcripts that are produced can be examined by peers, facilitating the reproducibility of science.

But why bother with pauses and 'uhms'? It turns out that these 'empty' actions can be an important mechanism to make conversation work. As part of the repertoire of communication acts, they contain information, even if not in the strict linguistic sense, then pragmatically in that conversational setting. For example, an 'uhm' in the middle of another's utterance can signal a desire to speak, hesitancy, or criticism. There is information in pauses and silences. For example, we learn later in this chapter that smartwatch users can exploit small pauses to check notifications on their watches.

After transcription, conversation analysis is carried out with a variety of analytical devices that can expose the reasoning and inference behind social interactions. Next, we look at some of them.

9.2.1. Turn-taking

Turn-taking refers to the organization of conversations into a sequence of utterances (turns) between conversation partners. Each turn consists of an action (utterance) by one participant, the nature of which affects the possible turns that can follow. Turns are mostly non-overlapping with, but overlapping can be used as a cue for changing the turn.

Sacks et al. [702] proposed 'simplest systematics' to explain turn-taking . In *turn construction*, a turn is put together, including signals about its start and end, and opportunities for others to transition and start talking. The way this is done defines what *others* can do.

The three central techniques people use for coordinating turn-tasking, as described by Sacks and colleagues, are:

1. Turn constructions, which guide the possibilities for the next turn. Example: signaling with a pause that a sentence has come to a conclusion;

2. Speaker specification, which define who is expected to take the next turn, which can also be the current speaker;

3. Rule sets for ordering options for actions.

Another insight from conversation analysis is that not all utterances are communicative. A speaker may take a *strategic* action that directs turn-taking in the conversation. Consider, for example, completing an utterance and stopping speaking, however, realizing that no-one replies. What can you do: continue and extend the original turn or ask for select a speaker to respond? This example illustrates that the speaker can influence the organization of the conversation by choosing who speaks next. *Recipient design* refers to the selection of the next speaker. These designs can be subtle additions to an utterance, or explicit requests made verbally or non-verbally, for example, by directing gaze.

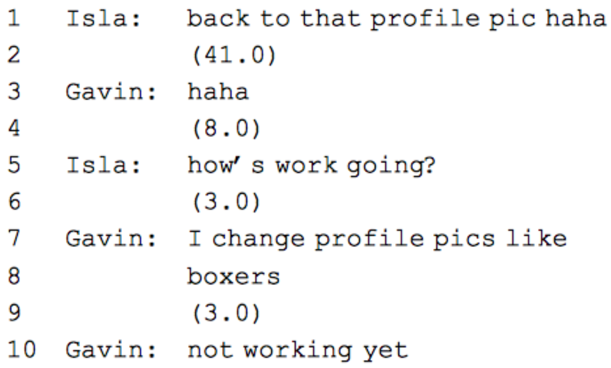
?] used the concept of turntaking to inform the design of an activity-tracking application. In contrast to normal activity trackers, where one logs on an individual basis, the application designed restricted tracking to one member. This changed tracking practice but made users of the app to communicate and interaction with each other.

9.2.2. Interactional sequences

The concept of *interactional sequence* elaborates understanding of how turns can be ordered. The focal point of analysis becomes the "here and now", or the current conversational action [279].

Turn adjacency is the simplest type of interactional sequences: a turn in communication is understood as related to the one immediately preceding it. The first pair part, for example a question, requires a reciprocal action (the second pair part). In other words, the first pair part limits the options of the conversation partner. An everyday adjacency pair is a terminal exchange, like ”Bye” and ”Good bye”.

If the second pair part is omitted, the initiator may engage in remedial efforts, such as repeating the question. If the other produces an *acknowledgement token*, such as ”hmm”, the response can be delayed or even avoided. If this adjacency is disrupted (adjacency disruption), other simultaneous threads interrupt it, as this example from Facebook conversations shows:



Isla: back to that profile pic haha
(41.0)
Gavin: haha
(8.0)
Isla: how' s work going?
(3.0)
Gavin: I change profile pics like
boxers
(3.0)
Gavin: not working yet

In this excerpt, taken from logged Facebook conversations [524], Isla mentions Gavin’s profile picture. However, Gavin self-selects to take a next turn and breaks up this take. This leads to several turns taken consecutively. Such a disruption can occur deliberately or because participants do not monitor each other’s turns.

The concept of turn adjacency highlights the interactive character of conversation. An appropriate conversational action is underpinned by an understanding of what precedes it. The preceding action also limits the options of the following. In this way, conversations are seen as interactional and context-bound. As we learn later, conversational agents struggle with adjacency and other features of interactional sequences.

9.2.3. Repair

Like everything in human behavior, conversations are prone to error and breakdowns. *Repair* refers to mechanisms to correct misunderstandings. A repair may be first initiated to signal that it is starting. It is often the same person who initiates the repair that then does the repair. However, the initiation offers the opportunity for the other to intervene and repair instead. The other may also request a repair, for example, by expressing uncertainty or asking a question. It is important for computer-mediated communications to allow repair. This does not necessarily mean editing or undoing messages, but offering a feature that helps explain a misunderstanding in its context.

Sharples [742] studied breakdowns and repair in a synchronous application for tutoring students. The delays in connection led students to adopt strategies in which they waited to ensure that their tutor was still listening. Although this sometimes worked, it also led to awkward periods of silence. As in many other studies, students and tutors developed new conventions to deal with the limitations of the application.

9.2.4. Common ground

Common ground refers to shared beliefs and goals related to a shared activity. Often in a joint task, communicators must first establish common ground before talking about the substance topic. For example, consider gamers talking about which objectives are available on a map, or which roles they can take, before actually embarking on a quest together. When there is no common ground, repairs and compensations may be needed. Participants must correct misunderstandings or take time to re-establish common ground.

Sometimes, common ground can become a topic of a discussion: 'What are we talking about?' Users may also talk about how to achieve common ground when the channel does not support that: 'How should we talk about it?'

It is interesting to note that common ground is difficult for artificial agents to achieve. When conversing with an intelligent agent, it cannot use embodied means to facilitate the communication of meaning. Moreover, while they are reasonable in speech recognition, they do not yet master 'the simplest systematics' of conversations. In Chapter 22 we provide an example of communication breakdowns in family use of Alexa [61]. Paper Example 9.2.1 contains an example where technology enters the conversation as a topic.

9.2.5. Compensational mechanisms

Although the original work on CA focused on face-to-face interactions, it has been successfully applied in a variety of settings, from doctor–patients discussions to computer-mediated communication. It can be used to understand digitally mediated conversations. However, present-day communication is increasingly multimodal, and requires going beyond conventional transcription-based methods.

Consider emoticons and emojis. Emoticons, like :) and : (use special characters to convey facial expressions [524]. They pre-date emojis, which use graphical icons and expand the repertoire. Meredith [524] argued that these "smilies" can not only convey facial expressions, but they have a conversational function. A smilie inserted to the beginning of a text can indicate irony, or some other stance toward one's own turn, hinting how it should be interpreted. Similarly, special combinations of punctuation have emerged that do not have a correspondent in face-to-face communication. For example, a double question mark (??) may indicate surprise, and capital letters to indicate excitement or anger. Lacking physical bodies, users have also developed remedial practices, such as leaving a room, to achieve similar functions.

Paper Example 9.2.1 : Smartwatch in everyday social interactions

The smartwatch is not just a small smartphone for notifications and health tracking. One difference to a smartphone that we rarely think about is that smartwatches are mostly not visible to others, at least unless purposefully displayed to them. At the same time, notifications and other information is more readily available.

To understand how this affects social interactions with others, Pizza et al. [646] instrumented 12 participants with 'the stalk camera' (see the figure below) and recorded them for 34 days. The authors found that the smartwatch is emerging as a complement to the smartphone, allowing users a quick, unobtrusive, and less disruptive access to communications.

To understand smartwatch's effect on social interaction, the authors first looked how its notifications may cause disruption. They learned, however, that there were no incidents of talk with someone else being disturbed. The authors looked at how responses (to the other person) would be delayed, if hesitations would occur, or if repairs in conversations would occur, but, surprisingly, found none.

Instead, they found that smartwatch users can skillfully intertwine micro-interactions with notifications without disrupting the structures of turntaking. One can look at a notification while talking and preparing a meal, or postpone glances to a later moment. Although notifications had no visible effect on the temporal structure of conversations, the participants were nonetheless aware about their disruptiveness [p. 5464]:

> It's like, something vibrated, I know something happened. I'm curious now, but I don't want to be rude so I'm not going to look. There are two processes, trying to listen to somebody and trying to not listen to your watch.

The smartwatch was thought to offer more control and much less disruption to social interactions than a smartphone. This requires more attention and more time to pick up and check. In conversational-analytical terms, it affords richer turntaking strategies. For example, users sometimes check their watches while doing something else, even with hands or under the table.

The authors also found that smartwatches are not just disruptions. They can be appropriated as communicative resources. The information that notifications bring can contribute to social interactions. The authors found that users would, for example, read aloud topics from their notifications, bringing them into a conversation.

9.3. Online behavior

Online behavior refers to the tendency of a communicator to send specific message contents. Online behavior is affected by the design of the communication channel. Consider receiving the following message for a compliment you send to a close person:

> Thank you.

Is this irony, a retort, or a genuine expression of thankfulness? Answering that question would be much easier in face-to-face interaction, where facial and other cues are available. It turns out that the availability of cues has a large impact on the way we behave online.

According to the *cues-filtered-out hypothesis*, communication applications that do not allow nonverbal cues can hamper social functions that involve those cues [848]. The study of this idea started already in the 1970s, when several experiments found a lack of nonverbal cues limited the warmth and involvement experienced in communication. The theory suggests that users cannot assess the characteristics of their communication partners, such as demographic, personality, or interpersonal characteristics. Consequently, they are relying on verbal persuasion. Although newer theories have superseded the cues-filtered-out theory, the core finding has remained the same: a change in cue systems affects the way social interaction and thereby relationships form.

9.3.1. Social presence

Virtual environments make it possible to radically change the way we are presented to each other. Whereas in situated interactions we rely on cues about the place, social context, and bodies when communicating, these can be practically anything in virtual spaces.

An *avatar* is a virtual character that is used in virtual environments to represent a person. An avatar may give the impression of a real body. However, it has both limits when compared to real bodies, but it can also have extraordinary characteristics that do not correspond in the real world. The avatar can take up almost anything that can be graphically presented, from an animal to a human-like character. Users, however, do not choose avatars arbitrarily. They tend to pick avatars that are aligned with their desired selves [810]. Users who are unhappy with themselves tend to pick avatars that are dissimilar. They may pick avatars that are more attractive or competent.

One of the most significant findings of CMC research is that avatars have consequences. The avatar you choose affects the social interactions and relationships that form. One role that the physical body has is that it makes the communicator present. By altering the virtual body, the person controlling it is misrepresented. How does this affect the experience of the conversation partner?

Social presence refers to the experience of being together with another actor in a virtual world. In a landmark experiment, Nowak and Biocca [588] asked undergraduate students to get to know a partner in order to compete together in a competition with a 100 USD prize. The experiment varied the level of anthropomorphism of the image of the interactants; that is, how human-like they appear. In the high-anthropomorphism condition the image was a graphical model of the face, in the low-anthropomorphism condition it consisted of only lips and eyes, and in the no-image condition no image was shown. Moreover, in half of the conditions, the participants were told that the partner is a computer agent, and in the other half that it is a real human.

All interactions took place in a 3D environment resembling a meeting room where discussions took place by speaking. The virtual confederate would first introduce himself or herself, after which the participant provided introductory information. Everything was scripted; however, the virtual confederate was pre-recorded. After the trial, the participants completed the social presence questionnaire:

- To what extent did you feel able to assess your partner's reactions to what you said?–Able to assess reactions, not able to assess reactions.
- To what extent was this like a face-to-face meeting?–A lot like face to face, not like face to face at all.
- To what extent was this like you were in the same room with your partner?–A lot like being in the same room, not like being in the same room at all.
- To what extent did your partner seem 'real'?– Very real, not real at all.
- How likely is it that you would choose to use this system of interaction for a meeting in which you wanted to persuade others of something? –Very likely, not likely at all.
- To what extent did you feel you could get to know someone that you met only through this system?–Very well, not at all.

The results were surprising. The image with higher anthropomorphism did not lead to higher social presence. Having an almost cartoon-like image led to higher felt presence. Yet, having no image at all had the lowest presence. The authors proposed that the high anthropomorphic image led to the highest expectations of the social skills of the conversant. When the partner was unable to react to the participant's turns, it may have led to a harsher evaluation. The results show that the way the conversation partner is presented affects users' expectations on conversational competence and behavior. When this expectation is violated, they feel less present in the social situation.

In a follow-up experiment, the level of anthropomorphism was systematically manipulated at 12 levels. At one extreme, the images were photos of human faces. At the other, they were cartoons. Participants were asked to do social dilemma tasks, where success is affected by how trustworthy the other interactant is perceived. In a social dilemma task, a user is put in the role of giving advice to a hypothetical scenario with risk, such as related to careers, relationships, or personal finance. The options that are given can be risky but suggest more rewards, or safer with less rewards. The results of the experiment suggest that, in this task, a higher level of anthropomorphism has a positive linear effect on social responses, especially trustworthiness. When the representation of a partner is more human-like, it is perceived as more reliable.

These and other recent results expose that the sense of presence is more complicated as a psychological construct than initially thought. Many factors affect the experience and the full picture is still emerging. For example, people evaluate the competence of the conversation partner based on cues such as style, attractiveness, gender, age, and race.

9.3.2. Media richness

Media richness theory is a candidate for explaining *how* social behavior adapts as a function of communication cues (see [848]). The theory has two constructs: media richness and equivocality. Media richness is defined via four dimensions: (1) the number of cue systems supported by a medium, (2) the immediacy of feedback it provides, (3) the potential for natural expression and language (as opposed to formal, structured expression), and (4) the specificity of audience addressing it allows. Face-to-face communication is the richest medium, because it has most cue systems, simultaneity, natural language, and high level fo control in addressing desired audience. Text messaging, by contrast, would be lower in the two first dimensions. Equivocality, the second construct, is the degree to which information in a social situation is subject to multiple interpretations.

The theory states that media richness should match equivocality. When the medium is leaner than the social situation requires, communication is less effective. Suboptimality may have negative consequences. However, the theory does not tell that those consequences can be.

In one study, managers in a company were asked to rate which channels would be most suitable for which communication purpose. Their responses were taken to support the theory, as they showed sensitivity to optimality, as suggested by the theory. Further studies, however suggested that managers may not actually behave as they told [848]. When they were observed in their workplaces, they would often choose a suboptimal channel and use it with no discernable consequences on communication effectiveness. Later research has suggested that instead of effectiveness, media richness may affect the efficiency of communication as predicted. That is, groups using a richer channel may be able to complete their tasks faster.

Testing the theory has turned out to be difficult, because it is hard to vary the four dimensions independently. Instead, what happens is that many dimensions change together. Moreover, when people are given a choice of communication medium, they do not only consider its optimality for the communication purpose. They may consider factors like expense or convenience in using a medium.

9.3.3. Signaling and Reciprocity

The decline of trust in social media has been one of the most significant socio-political phenomena of recent times. Trust is a desirable factor in most social interactions. A higher level of trust can promote deeper interactions and engagements that would not happen otherwise. Why is it that establishing trust is so difficult?

Early theories suggested that trust follows from *reciprocity*. When interactants match their levels of self-disclosure matches, trust will follow. For example, if you disclose your age and gender, but the other party deliberately not, the principle of reciprocity is broken. But how to understand situations in social media where posters can be anonymous or even take up on deceptive identities? Such settings are associated with increasing negative behavior, such as flaming.

Signaling theory is about text-based social media, Twitter being the most well-known example. According to the theory, people engage in assessing which signals of each other are reliable [201]. They are active in thinking how to best use social cues in communication. They weigh the costs of deceiving against the benefits achievable. For example, in text-based discussions online, participants can easily craft any sort of self-descriptions. In other words, the costs are low. Anyone can easily deceive others about their age: you can type ”24” in your profile as you can type ”42”. This is drastically different to face-to-face situations. Here, manufacturing false signals is more difficult. For example, if you intend to claim that you are wealthy, but you are not, you should first acquire external signals of wealth. Thus, the signals in face-to-face communication are more trustworthy and, according to the theory, are more trustworthy. There are ways users can also improve the reliability of their signal. For example, they can present auxiliary information as proofs of authenticity, or ask other contacts to comment on credibility.

9.3.4. Social information processing

The *social information processing* theory differs from the other theories by proposing that the lack of nonverbal cues does not impede social interaction [847]. Communicators are willing to develop interpersonal affinity regardless of the medium, and they try to do that by adapting their own behavior to compensate for the lack of cues. They do that by expressing more using the medium itself, changing their styles, and finding alternative ways to express themselves.

Such techniques take time to develop, however. Once developed, however, they can become culturally shared and assist in the use of the medium. For example, emoticons and emojis can compensate for the lack of emotional cues in textual communication. Their evolution has taken more than a decade and has still not stopped. However, the theory is not limited to emoticons.

The theory makes a surprising prediction: because communicating nonverbal cues verbally is inefficient, establishing sufficient relationships takes more time. The slower rate of nonverbal communication means that constructing representations of the other partner takes more time. These two predictions – compensation of lack of cues and longer time-spans for building relationships – have received evidence from experimental studies. In these studies, the cue systems are experimentally removed or added, and the effect on other cue systems and the style is recorded. It has also been found that on multimodal platforms, users select which messages to convey in which modes. For example, profile images in social media can be designed to convey certain traits, while text provides others.

9.3.5. Self-presentation

So far, we have discussed how users process information about each other. Given that such inferential processes happen, it is not a surprise that users may want to control how others perceive them.

Self-presentation refers to actions taken to present oneself in some fashion that is desirable to oneself. Self-presentation can take any communication modality: text, video and audio clips, streaming video etc. Some video- and photo-based social network sites, for example, have become 'repositories' of digital self-presentations [473]. The point of self-presentation is that users are not passively showing who they are or what they do, but are *construeing* how they are seen by others. Users not only curate what they post, but also stage events to take favorable posts, and engage with technical means like filters and settings to control who sees what. An example case about a popular online game is given in the side box.

Paper Example 9.3.1 : Self-presentation in an online game

An illustrative point in case is the study of Ducheneaut et al. [211]. They discussed social behavior in World of Warcraft (WoW), a massive multiplayer online game. They analyzed how players' networks were structured, finding that self-presentation is a key factor. They found that while WoW is a social environment, the role of others is mainly that of an audience. That is, instead of playing with others or against them, players rely on them as an audience. Gamers tell playing the game for the social factor; however, many gamers stay outside of groups (e.g., guilds). Instead of forming groups, the players of that game perform and perform for others, and gather artifacts that they chitchat about with them. The authors describe this as being ‘'alone together”: gamers are surrounded by others, but do not interact with them with the idea of forming relationships. Why does such behavior emerge? The authors point out that WoW, like many other multiplayer games, has incentives based on reputation. An avatar showing rare items is part of the player's identity. An audience that rewards such displays, rewards the behavior, and thereby reinforces it. The game is addictive not because of other gamers, but because of the self-image players are able to build with their help.

Another role the others have is via social presence. The game offers a general communication channel, which the gamers can use to communicate with players who are not close by in the virtual world. These channels offer constant chatter, which creates an experience of being with others similar to being in a crowded cafeteria. This, and other designed communication features, ensure that the players feel being with others during a game.

Self-presentation in virtual worlds can be tricky, since posts cannot be controlled after initial sharing [473]:

> Consider the example of Alex tagging a Facebook photo of Bill at a party. Facebook's interface provides Alex with little information about Bill's privacy settings including the visibility of tagged photos, or how many and which of Bill's Facebook friends might see the photo.

The issue is that other users may not care about the original poster's self-presentation goals. Shares, likes, and comments may be seen as threats to self-presentation, even if these were not done with a malevolent purpose. This issue is harder to handle for those users who have diverse audience groups. The corresponding issue for HCI research is how to allow better control and visibility of audience [473]. *Audience cues* are visualizations and controls that help posters understand who sees posts. The dilemma is that limiting the audience more than a norm (expectation) may work against the poster, throttling dissemination or attracting negative views about the poster.

9.4. Social Networks and Online Communities

It is clear that computer-mediated communication has changed the very way people create and maintain relationships with others. In particular, the communication that are one to many and many to many has changed dramatically in the past decades. From Facebook to Tinder, social networks and online communities are not 'virtual' and therefore 'less real' but important and consequential in our lifes. But why is it that some social networks become unsustainable while others flourish? To design sustainable networks and communities, it is important to understand the mechanisms that underpin social relationships.

According to Boyd and Ellison [91], *social network sites* (SNSs) are

> web-based services that allow individuals to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the system.

LinkedIn, Facebook, and Couchsurfer are SNSs according to this definition, and other services such as Whatsapp include only elements of a social network. For example, Whatsapp does not allow traversing other connections' connections, while for example Facebook does.

An *online community* is another relevant concept. It is an interest-based forum of connections and interactions, such as Stack Overflow for programming. Online communities often have some features shared with SNSs, such as the creation of profiles and networks.

When social networks started to gain popularity, they attracted lots of attention from research. Initially, it was assumed that these services would be used to create new networks and meet new people. However, this is not always the case. Studies of Facebook found that it is used primarily to maintain existing (offline) relationships or solidify offline connections, rather than meeting new people [220]. It was also surprising how many social networks never attracted enough use to be sustainable, and how some turned into ill behaviors like flaming.

One of the attractions of social networks and online communities is that they may be subjected to a host of statistical analyses. These methods can precisely describe the network's structure, with people as nodes and different types of connections as their links. The benefit of such analysis is that it can help illuminate larger-scale patterns that would be difficult or outright impossible with other methods. Analyses of social networks have illuminated, among others, that social relationship can be described in terms of the strength of connection [281]. Weak ties are connections to strangers and ad hoc acquintances, who may be met online or in person. Strong ties are relatives, friends, and other close-knit groups. One observed issue in social networks is that users would like to communicate differently with these two types, which may require putting extra work from them, or if that fails, inhibit self-presentation.

While the macro-level of social networks can be studied using methods like this, to understand the role that design has on the emergence of social ties, we need to understand basic social psychological mechanisms in online relationships.

9.4.1. Online social relationships

Baym [53] points social network sites (SNSs) as a breakthrough which led to an increase in *networked individualism.* Each user is at the center of a personal community. While online communities like bulletin board systems existed before the now-popular SNSs like Facebook and TikTok and Mastodon, what has changed is the ease with which users can define who they include in their personal networks. Most SNSs allow users to “follow", “invite", or otherwise define whose posts they attend. Other users’ networks can be traversed. SNSs, however, have not replaced other types of relationships. Instead, they have created a new platform for social relationships that stand in-between blogs, which focus on the individual, and online communities, where the focus is on a group or community.

Design matters in the emergence of social networks. Comparing two SNSs of the 2011, Baym writes [p. 38?],

> ”The two sites differ in their affordances. Neither allows much flexibility in page design, as MySpace and LiveJournal do, but Facebook allows users more breadth in shaping their profile. Facebook users can add applications (including several from Last.fm) in order to shape their self-presentation, play games with their friends, and promote causes they find important. They can maintain photo albums, import blog posts, share items and videos from elsewhere on the net. Last.fm users can do very few of these things, but they can display the music they listen to in real time, create radio streams for others to hear, tag music and to bands, author band wiki entries, and see personalized charts of their own and others’ listening habits, which cannot be done on Facebook. Both sites allow users to create groups, and both recommend people with whom one might connect – Facebook by calculating the number of shared friends, Last.fm by calculating the number of shared listens. Not surprisingly, the two sites result in differing social contexts. While Facebook is seen as a space in which to socialize playfully with peers, Last.fm is all and only about music – one may socialize, but it’s most likely going to be about music. Some of its users do not use its social features, friending no one, yet still have satisfying interaction with the site, a situation that would be unimaginable on Facebook.

Design affects social relationships by affecting social psychological mechanisms underlying the formation of social relationships [53]. Social relationships need to be created. You need to add someone to your network. Social relationships also need to be maintained: via messaging, we create the content of a social relationship. Design affects how this happens:

Identity and authenticity: As we discuss elsewhere in this chapter, early online interactions were often anonymous, increasing the 'flaming'. Modern SNSs, by contrast, allow users to link to organizations they belong to (e.g., companies, sport clubs). They can also share rich media like video and photos. These are important for the authenticity of contacts. Authenticity, on the other hand, is important for reciprocity: you are more likely to share personally relevant information to a stranger whose authenticity you trust. Users are also more likely to connect to people with whom they share acquaintances. Shared connections increase trust on authenticity.

Audience and self-presentation: Every communication act in an SNS is visible to a specific audience. This is a critical consideration in social interaction: who sees the message and what kind of effect will it have on them? In the case of Twitter, for example, the audience is anyone using Twitter, but more likely the people who follow you. In the case of Whatsapp, the audience whoever belongs to the group you post in. Knowing the audience is important for self-presentation: we need to control how we appear to others.

Affordances for communication: Some SNSs make visible which contacts are available and how. For example, you may see a friend appear online and start chatting. Visibility is important, because users can now engage in different types of communication: synchronous or asynchronous. But SNSs can also make visible other information, for example other people's comments on one's posts, or one's profile image, email addresses etc. These all provide different degrees of freedom for the audience to engage.

Privacy: The concept of an audience is also bound to the concept of privacy. In SNSs, users can control *what* they disclose to others. But others can also affect what is supposed to be disclosed: they can not only explicitly ask for disclosure but create an expectation for reciprocity. Privacy, in other words, is *constructed* via social interactions [618].

A personalized network can become unsustainable over time. It is known that in larger networks like Facebook, pairwise interactions can be very rare [53]. However, because contacts tend to be added rather than removed, it means that the audience also grows, making it harder to control self-presentation and privacy. Concepts like 'a friend' are becoming harder to define in the era of SNSs. However, users may also creatively utilize the *ambiguity* of such concepts. They can situationally decide who is invited, for example, to some group or event, with the ambiguity of the relationship providing plausible deniability.

9.4.2. Online communities

Online communities are often interest-based: the motivation for joining is a shared interest in something outside of the online service itself, for example particular types of culture (e.g., music, cartoons) or local influence (e.g., community shelters, sport clubs). Online communities, unlike SNSes which are individual-centric, are group.- or community-centric.

Group- or community-centricity matters. Unlike SNSs, online communities develop *social norms* for desirable behavior [53]. They may provide instructions on how to post, comment, or on power structures like moderators. SNSs, by contrast, often have implicit norms: 'ways of the house that everyone knows', but which may not be written anywhere. For example, Stack Overflow users may have different norms on when to post and how to use emojis than those of Tinder.

An important question is how to foster longevity and respectful behavior in social communities. *Silent majority* is a typical phenomenon in social networks. That is, there are users who mainly follow others but do not engage with original content. Activating users to contribute boils down not only to the designed features, as discussed above, but to *incentives* for participation. Among the techniques studied in HCI are *reputation systems*, which may for example inform who has most followers, most likes, etc., thus allowing social comparison and social motivation. A key challenge, then, is how to create grow communities based on *legitimate, peripheral participation.* That is, how to allow users to develop their participation from a 'lurker'[1] toward an active member.

Research has shown that being connected may have a dark side. Online communities and social networks suffer from harassment, trolling, and they have been misused to distribute misinformation and manipulate political opinions. The increasing use of AI for selecting contents has created *filter bubbles* (echo chambers). Because AI tends to pick posts based on predicted liking and sharing, users tend to be exposed to confirmatory content. Filter bubbles thus feed the polarization of social networks. It can reinforce biases that are inherent in a community. Addressing these issues is a topic of extensive research across boundaries.

9.4.3. Communication affects social relationships

This chapter has reviewed social mechanisms affecting computer-mediated communication. Perhaps the most obvious takeaway for HCI is this: Human-human communication poses requirements for the design of communication technology. The modalities, cues, and structures of communication, and the features offered for the specification of audience and the negotiation of power structures affect what kind of communication emerges. Communication services that ignore these factors are bound to see little participation and unwanted behaviors like flaming. Yet, behind these factors is a slower, dynamic process that shapes users and societies. The way people use communication technology affects their social relationships, well-being, and culture. For example, the mere presence of communication technology can affect the way people hold conversations and what they are ready to disclose. An early study looked at how the presence of mobile devices affects

---

[1] An oft-used term we do not condone. Instead of a lurker, we recommend using the term passive user.

conversations between collocated people [661]. In two experiments, the authors found that in one-to-one (dyadic) communication, the presence of mobile devices changes the discussion of personally meaningful topics. The authors conclude [p. 244]:

> [. . .] the mere presence of mobile phones inhibited the development of interpersonal closeness and trust, and reduced the extent to which individuals felt empathy and understanding from their partners. [. . .] these effects might happen outside of conscious awareness.

This effect has been replicated [214]. However, different communication media can have widely different effects, as we saw in the case of online forums [407].

While the causal mechanisms behind long-term changes are still being studied, findings like these suggest that communication technology is not an inert outlet for communication. Rather, it is shaped by and shapes the formation of social relationships. For instance, it has been found that people addicted to smartphones are more likely to 'phubb' others in a social situation [151]. The term refers to the act of snubbing copresent people by attending one's phone. These kinds of effects can have complex, population-level consequences. Emerging research looking at population-level statistics suggests that changes in social relationships caused by increased use of digital media may have made mental health issues among youth, such as depression, self-harm, and suicide, worse [814]. This dynamic process is still evolving and its consequences are hard to study, and even harder to predict and control. More research is needed to understand the mechanisms that drive these dynamic processes.

9.5. Computers as Communication Partners

So far in this chapter, we have mostly discussed analytic machinery for understanding computer-mediated human–human communication. The reader may wonder if this machinery also applies to conversations with computers? Such a system could, for instance, be an animated agent that writes back when the user writes something, or an intelligent agent that speaks back when the user asks. We can use analyses of turn taking, for example, to see how conversations are structured with intelligent agents. We can also look at the cues that are conveyed and how they contribute to efficiency of communication. But there is a more fundamental question that we need to look at: Are computers perceived as competent social actors?

Nass et al. [565] have argued that people largely treat computers as social actors almost like they treat humans. Rather than treating computers as inanimate tools, people rely on expectations and perceptions that they transfer from everyday social life with humans.

One representative experiment asked if people are polite to computers like they are to other humans [564]. Nass and colleagues tested this in a study where a computer tutored participants. After tutoring, participants rated their impression of the computer tutor. This was done in three ways: on the same computer, using paper, or on another computer. Take a moment to consider which would be rated the best.

Surprisingly, evaluations on the same computer were rated better than on paper. Nass et al. [564] interpreted this as showing that participants were more polite to the computer.

Crucially, the evaluations done on another computer are also less positive than those done on the same computer. Thus, participants seem to apply social norms to computers. The same pattern has been shown to hold for other individual and social patterns in communication. For instance, if interactive systems are perceived to have a personality of a particular type, its users respond along the lines with which they would respond to a computer; Nass and colleagues have also shown that gender stereotyping has influence computers. Moreover, the methodologies presented in this chapter have also been applied to study conversations. Porcheron et al. [654]

Anthropomorphic systems try to deliberately make the system's looks and actions resemble those of humans. The interactive system Rea, for instance, presented the user with [140]. Figure 9.4 shows two additional examples of interactive anthropomorphic systems.

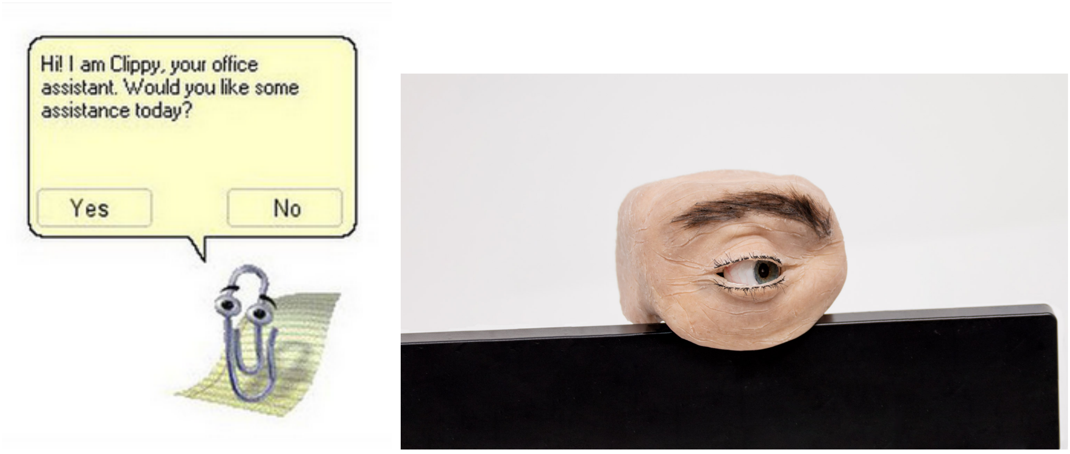


Figure 9.4.: To the left, a commercial example of an early agent, Clippy. It was part of Microsoft Office for years, although it was widely critiziced. To the right, a web camera designed by Teyssier et al. [798] to appear like a human eye. It even moves to follow the user (find the video online, it is interesting).

At the same time, many researchers have been critical about anthropomorphic systems. The argument has been that even though people *may* treat computers as social actors, it does not mean that we *should* design computers with this goal in mind and, in particular, it does not mean that we should make computers look and behave as people. There are three main arguments for this view [750]. First, new technologies are often designed to resemble older ones. This is an ineffective strategy that limits innovation. As an example, the first cars contained reins; this did not facilitate inventing a better way of control, the steering wheel. Second, anthropomorphic systems may make users anxious and in some variants undermine users' feeling of control and responsibility. Third, and finally, anthropomorphic systems lead to a conflating of the abilities of people and machines.

Summary

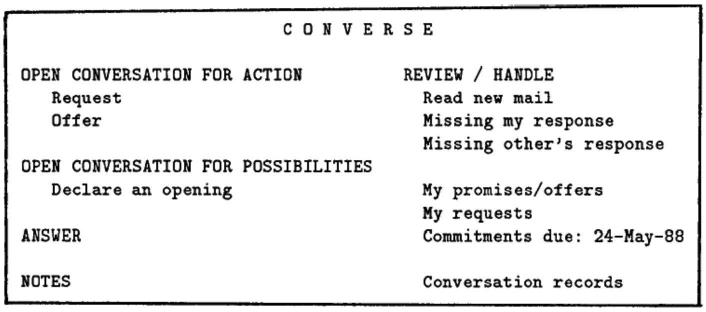
- Communication mediated by interactive systems differs from face to face communication.
- Computer-mediated communication may be discussed by the features of the channels it uses or by its impact on the individual and social functions of the communication.
- Conversation analysis is used to understand the structure (turntaking, adjacency, repairs, ...) of both online and human-to-human communications.
- Social networks are associated with both positive (social capital) and negative phenomena (e.g., trolling, biases, polarization).
- Design affects authenticity, audiences, privacy, serenpiditious communication, and the maintenance of social networks. These in turn affect the type of social relationships that emerge. However, it is an open question how to design mediated interactions that promote fair and unbiased participation.

Exercises

1. Analyzing computer-medicated communication. Consider an interactive system that you use for communication. It may be a computer game where you need to coordinate with a team, an electronic whiteboard where you leave messages for others, a chat client, or something fourth. Use the features in Table 9.1 to analyze the different between the interactive system and face-to-face conversations. If you have experienced break-downs in communication with the interactive system, select a couple of examples and analyze them using the table.
2. Self-presentation. Choose a social media application and look at its posts. What are the means of self-presentation it allows? Which audience cues does it allow?
3. Repair and compensation in conversations with artificial agents. Pick an everyday discussion topic and an interactive language model (e.g,. ChatGPT). Run a few conversations on the topic with the purpose of causing some misunderstanding. 1) Which interactional sequence strategies does the agent utilize (e.g,. acknowledgement tokens)? 2) When conversation breaks down, how does the agent try to recover from it? Does it notice it breaking down? How does it afford you to recover from it?
4. Real vs. artificial communication partners. Many interactive systems mimic conversations among people, for example, by creating the impression that you are conversing with a person. Such agents or avatars often resemble people in their physical appearance, manner of wording, or body language. Try to identify such human-like or antrophomorphic communication systems among the systems you use. According to this chapter, what might be the benefits of using such systems for communication with a computer? What are the drawbacks?

5. Designing systems for communication. Consider interactive systems for communicating with people in your neighborhood. How could one design such systems to support communication, as described by Table 9.1? What would the concepts in this chapter about online communities teach us for this question?

6. The Coordinator. Winograd and Flores [881] presented a famous view on computer-based communication, rooted in the view that language is action. As part of working with that view, a tool called The Coordinator was created. Rather than just sending an unspecified message, the tool supported structured requests like those shown in the figure below. For instance, your message could request or offer something.



```
                              C O N V E R S E

OPEN CONVERSATION FOR ACTION          REVIEW / HANDLE
   Request                              Read new mail
   Offer                                Missing my response
                                        Missing other's response
OPEN CONVERSATION FOR POSSIBILITIES
   Declare an opening                   My promises/offers
                                        My requests
ANSWER                                  Commitments due: 24-May-88

NOTES                                   Conversation records
```

Discuss how such a system might influence communication. Does it make it easier or harder to request anything? What to do if you are not clear about whether you make a request or an offer?