1

Introduction to human–computer interaction

From fishing nets to drilling machines, tools are vital to human ability. They enhance our physical abilities and are central to many intellectual activities, such as writing, mathematics, and accounting. It is hardly surprising that great efforts have been made throughout history to innovate and refine them. In the long history of tools, the birth of the computer marks a watershed. The invention of transistors in the twentieth century, together with the theory of computation and the architecture of the digital computer, pushed humanity to transform itself.

At the core of this revolution was the ability to flexibly define and execute *computer programs*—sequences of logical operations executed by a computer. Transistors enable energy-efficient and noise-resilient logic gates, which, in turn, enable the implementation of Boolean operations, such as nand and nor. These logic gates are organized as logic circuits to implement the necessary functions of a computer, such as arithmetic logic units and registers, all the way up to a complete microprocessor. A computing system integrates a microprocessor with the memory and peripherals necessary to execute programs. An important enabler is the operating system, a special computer program that manages the hardware and software resources of a computer, such as access to memory and displays, and allocates such resources to programs. It also provides common services to programs, such as access to networks.

Programmability lends computers their power as tools. Computer programs can decompose complex activities into sequences of much simpler operations. When input is provided to a program, it is executed according to the program. For example, the task of sending a message to a contact is broken down into subtasks, such as sensing user input, committing letters, updating the display, selecting a recipient from a menu, or sending a message over a network. Each of these subtasks is implemented via numerous simpler operations performed by a computer. Computer programs also permit computation using different formalisms, from algebra to probability theory. They can be used to flexibly represent the types of input that are important. An input can represent virtually anything that can be defined using a binary system, such as images, text, and sound. This has been central to the wider appropriation of computing as a tool. Thus, the remarkable efficiency, flexibility, and scalability of computers as tools boil down to the concept of a programmable machine capable of interpreting computer programs.

However, computers would be useless as tools if they did not offer a way for people to control them. A special part of a computing system is the *user interface*. It is the part that the user can see and utilize to control the computer. Through the user interface, users can provide input and instructions to a computer and receive feedback from it. In short, the user interface enables *interaction* with a computer. The user interface simplifies the underlying technical system for the user: It allows users to command a computer without bothering with what it does under the hood. Instead of defining a long and incomprehensible binary code that commands a microprocessor to delete a file, the command can be provided with a few mouse movements and clicks. When a user drags an icon representing a file to the trash can to delete it, this instruction is propagated through many layers of instructions, trickling down from the file system managed by the operating system

Introduction to Human-Computer Interaction. Kasper Hornbæk, Per Ola Kristensson, and Antti Oulasvirta, Oxford University Press.
DOI: 10.1093/oso/9780192864543.003.0001

to low-level disk management handled by the physical storage device. In the end, the computer updates the display to communicate to the user what it has done.

The development of technology for *interactive computing systems* has been an important driver behind the widespread adoption of computing we have witnessed in the last 50 years. Figure 1.1 shows a few snapshots of the transformation of computing from machines used exclusively by trained professionals to tools suitable for everyone. The speed of this development has been mind-boggling. The authors of this book have experienced the adoption of the personal computer, the world wide web, the mobile phone, intelligent interactive systems, and virtual and augmented reality. These and many other technologies have impacted practically all areas of life during their lifetimes. Computing has had a pervasive impact on what we do, how we think, what we value, how we play, and even how we build and maintain social relationships, including romantic relationships. Computers have shaped our economies, leisure, transportation patterns, social networks, and even elections and wars.

Although computing systems can be autonomous, most systems are intended to be used by humans, specifically to assist them. The realization that this is not a trivial task has led to the field of research called *human–computer interaction*, or HCI for short. HCI emerged gradually through early efforts in other fields, including psychology, computer science, and electrical engineering. What we now know as the field of HCI emerged in the late 1970s. The birth of the field coincided with the revolution of the personal computer, which marked the transition of computers from expensive mainframes kept in organizations to computing technology suitable for homes and workplaces. Interest in HCI was fueled by the belief that computing technology will fail unless it is designed with serious consideration for its users.

Two fundamental questions characterized early research: *How* should all this computing power be used and *for what*? Around the emergence of the personal computer in the late 1970s, it became clear that software needs to be designed for end-users and not just for specialists, such as programmers. These and other observations prompted a search for both new visions of computing and fundamental knowledge about the nature of the human use of computers. Over the years, the focus in the HCI field shifted to the question we now see driving HCI research:

> How can people with different goals and capabilities, and in different contexts, be able to use computing productively, enjoyably, and safely?

In short, the noble goal of HCI research is to help create computing that supports the betterment of humankind.

The book provides an introduction to the field of HCI. HCI is primarily a research-driven field. It focuses not only on the scientific principles underpinning human–computer interaction but also on the very concrete goal of designing better computing systems. For this reason, while the starting point in this book is theoretical research that helps us understand interaction, we also cover various aspects of design. The pragmatic aims of HCI research cannot be overstated. Accordingly, the chapters in this book acknowledge the many challenges that practitioners face, and identify the solution principles that can be used to tackle them.

Since HCI is not typically taught in basic education, most HCI researchers to date have discovered the field during university studies. Everybody has a unique story to tell about how they fell in love with the area. We, the authors of this book, discovered HCI in very different ways. Kasper found HCI via scientific curiosity. As a student of computer science, Kasper realized during his third undergraduate year that HCI, and not lambda calculus as he thought at the time, unified his

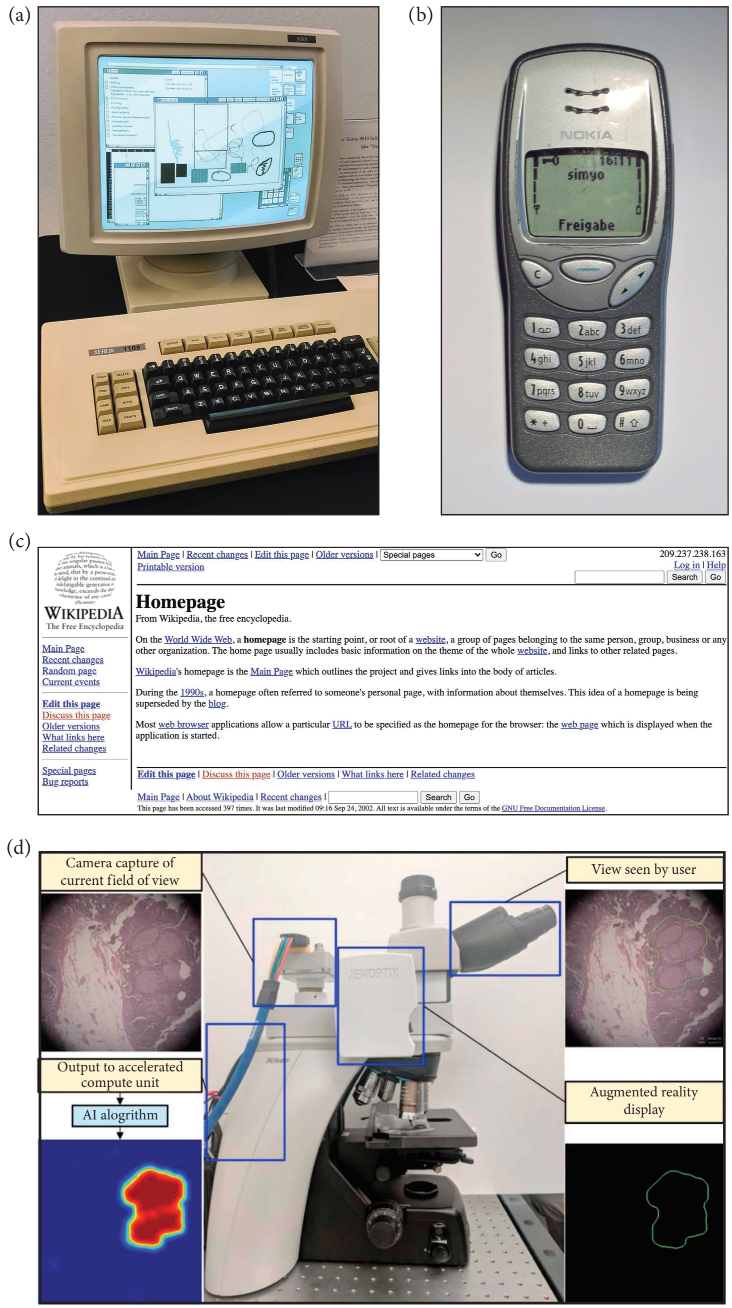


Figure 1.1 The evolution of interactive computing. The emergence of *personal computing*, exemplified by the Xerox Star from 1981 [391] (image by Leigh Klotz, reused under Creative Commons License). The emergence of *mobile computing*, exemplified by the popular Nokia 3210, around 1999 (image from Basic Master, reused under Creative Commons License). The emergence of *social computing*, exemplified by the collaborative encyclopedia Wikipedia, created in 2001 (image from archive.org). The emergence of *computing enhanced by artificial intelligence (AI)*, exemplified by an AI-assisted microscope from 2019 [146].

interests in technology, psychology, and philosophy. Another co-author, Per Ola, studied cognitive science and quickly became interested in applying it to the practical problem of how to help people enter text, a pursuit that eventually led to the invention of the gesture keyboard, a text-entry technique that is integrated into many mobile devices today. Finally, Antti was motivated by frustration while studying cognitive psychology for a master's degree. He felt that HCI is too complex for verbally expressed theories. Over the years, he has developed a view that scientific knowledge about users' behavior and thinking can be encapsulated into computational models, similar to other areas of applied science, which can then be used—with the help of algorithms—to explain and enhance interaction. We hope that this book can plant the seeds of new stories that will change the field in the future!

In the remainder of this chapter, we will expand on the defining characteristics of HCI. In particular, we will explain why HCI is a challenging field and why it needs to combine science, design, and engineering. We will also give a brief historical account of how HCI emerged as a field and became what it is now: a multidisciplinary melting pot. We strongly believe that HCI is of pivotal importance to society, which is why we will also dedicate some space to the key arguments supporting this belief. Finally, we will explain our approach to HCI and the structure of this book.

1.1 Why is HCI challenging?

What makes HCI worth studying, and why is a separate research field needed? First, humans are complex biological and social organisms. A human being is capable of complex thought and has developed language abilities, capabilities for fine motor movements, and generally excellent skills for learning to develop, use, and adapt tools to achieve goals. Several of these fundamental abilities are sufficiently well-understood to apply them in HCI, such as central aspects of human motor control. However, many open questions remain, such as those related to experience and language. Furthermore, HCI must account for individual differences. What may work for one user may be unsuitable for another. Individual differences in computer use are staggering, not only in terms of performance but also in terms of what is considered interesting or culturally appropriate. For example, an expert in HCI4D (HCI for development) described the challenges faced in non-Western contexts as follows [184, p. 2228]:

> We need to address the everyday problems of people. Most people don't know how to scroll, navigate. We need to do basic HCI work to make text larger. Also, time of day is the most prominent thing on [a phone's] screen. Let's replace that with the amount of airtime you have left. We need to improve upon what we built yesterday rather than doing novel interventions or focusing on the future.

Second, the computer is among the most complex tools humans have devised. The complexity of this tool stems from its elaborate stack, which ranges from transistors, logic gates, microcontrollers, and memory chips to operating systems, software drivers, software libraries and toolkits, applications, and user interfaces. Designing systems with this level of complexity is inherently challenging.

Third, HCI involves people interacting with computers in complex contexts where they try to accomplish a variety of goals. Thus, HCI requires some systems thinking. This means taking into account the full sociotechnical context in which specific HCI activities are situated. In HCI, even if we often focus on a subsystem (e.g., interaction with a graphical user interface), successful deployment relies on the ability to understand and relate observations to their wider, system-level context.

Fourth, design is hard. Ultimately, the purpose of HCI research is to influence the design of applications, interaction techniques, systems, or services so that users can achieve their goals in an effective, efficient, safe, and satisfactory manner. Why is doing such designing hard? One reason is that there is no perfect design. Design is about identifying and generating suitable solutions that trade off desirable characteristics, such as the user's speed for accuracy. In addition, designing requires creating new ideas. In design fixation, a designer maintains an early identified solution despite being inferior to other possible designs. It is hard to let go of old ideas and generate new ideas that are also valuable. Even experienced designers suffer from design fixation. Moreover, arriving at a design is a complex process where it is easy to introduce mistakes early in the process, such as accidentally injecting faulty user requirements based on misunderstandings, which are exceedingly difficult to correct later in the design process.

These reasons explain why HCI is hard. They indicate that HCI requires a unique combination of skills that almost no student receives from basic education. The skills cover software development, such as understanding software architectures and programming user interfaces; analytical skills, such as formal modeling of a user's performance; design practice, for example, interaction design, service design, or product design; and user research skills, such as carrying out and analyzing experiments, conducting interviews, and engaging in field studies. This book covers these basic skills.

1.2 Human–computer interaction as a field

The term *human–computer interaction* has been in use since the mid-1970s [e.g., 131]. Before then, the terms *man–computer interaction* and *man–machine interaction* had been in use since the early 1960s [e.g., 476, 588]. Until the 1970s, real-time interaction with a computer was impossible, mostly due to limited capabilities in processing power and displays. Computer requests were made in batches, sometimes in the clumsy form of punch cards. In addition, the time it took to get an answer from the computer was often in the order of hours. With the advent of interactive terminals and later graphical displays, the term *interaction* suddenly became relevant. Although computer users at that time were primarily professionals—such as managers, developers, and operators—researchers started to explore ways to provide more people with better access to computing. Consequently, the next decades saw the birth of a new scientific community with research groups, publication venues, and shared interests.

In the early 1990s, the international computer science organization—the Association for Computing Machinery (ACM)—set up a task force on education that defined HCI [339]. This has become the most widely used definition for characterizing the field. It reads as follows:

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

The definition has three noteworthy parts. First, HCI is about building interactive systems. It is concerned with design and engineering, including implementation, as highlighted in the quote. Ultimately, the goal of the field is to improve the interactive systems we use and to invent new ones. For example, if you attend an academic HCI conference, one presentation may be about the construction of super-resolution touchscreens [800], a second about how to design technologies for tracking the periods of adolescents [784], and a third about how to place content in the most comfortable places in mixed reality technology [233].

Second, people are the starting point for HCI. They also define what makes interactive computing ultimately good. Therefore, researchers and practitioners study people's needs and wants in depth and evaluate how people use systems. HCI's uncompromised focus on human aspects distinguishes it from areas in computer science that mostly focus on computation and algorithms and areas in other disciplines that typically focus on technology. In such areas, it is commonplace to make assumptions about users without always grounding them in empirical observations or theory. Conversely, HCI is based on justifiable and explicit knowledge about people and their abilities, needs, and wants. HCI goes beyond wishful thinking: It is about discovering solutions that actually make computer use more effective, efficient, and safe.

Third, HCI is concerned with investigating all phenomena relevant to interaction. In other words, its focus extends beyond technology to work and leisure and, more generally, to social, organizational, psychological, and many other factors. Wright and McCarthy [915] noted that "HCI is concerned with understanding the influence technology has on how people think, value, feel, and relate" and "using this understanding to inform technology design" [p. 644]. It is not a detour to understand people when developing technology—it is necessary to create technology that brings value to the world.

1.2.1 Research in HCI

According to the definition of HCI quoted in the previous section, research problems in HCI concern the human use of computing and how we might improve it. However, what does this mean in practice? What research problems are relevant to this goal?

Three main types of research problems can be distinguished in HCI [625]:

Empirical problems: These concern developing accounts of phenomena in interaction grounded in empirical data. An empirical research problem is often motivated by a lack of understanding of some aspect of interaction. For example, empirical research might focus on understanding how people discover and learn to use features in an interface or examining the effects of social media on relationships. As mentioned earlier, empirical problems may be about "all major phenomena surrounding interaction."

Conceptual problems: These are about explaining previously unconnected phenomena occurring in interaction through reference to theoretical constructs. Conceptual problems involve hypotheses, explanations, theories, and models. For example, a conceptual problem may concern reconciling different views of what privacy is or formulating an account of what it means for users to feel immersed in virtual reality.

Constructive problems: These tackle the knowledge needed for constructing interactive systems for some stated purpose concerning the human use of computing. This understanding does not need to be expressed formally in terms of models. Constructive problems may just as well concern visions for building brain–computer interfaces or guidelines that help designers create accessible user interfaces.

These three types of research problems complement each other. Addressing empirical problems can help identify new phenomena that HCI theory will need to be able to explain. The solutions to conceptual problems may help identify ideas for how to construct interactive systems more effectively. Engaging in the construction of user interfaces may create new forms of interaction that give rise to new empirical research problems. As such, the three types of research problems go hand in hand within the HCI domain. Empirical data are linked to knowledge and knowledge is linked to design.

Throughout this book, we will provide numerous examples of classic and contemporary research in HCI, as well as many examples of contributions from such research. Good venues for finding examples of ongoing research are conferences supported by the ACM Special Interest Group on Computer-Human Interaction, such as the flagship *ACM Conference on Human Factors in Computer Systems* (CHI), and academic journals, such as *ACM Transactions on Computer-Human Interaction*, *Human–Computer Interaction*, and the *International Journal on Human-Computer Studies*.

1.2.2 **The practice of HCI**

In addition to research, HCI is also practice. Hundreds of thousands of professionals around the world engage daily in designing, implementing, and evaluating interactive systems. Many of those systems have already found real-world applications or will do so in the near future. HCI practitioners hold different and continually evolving professional titles, such as *interaction designer*, *usability specialist*, *HCI specialist*, *user interface engineer*, *user researcher*, *behavioral analyst*, and *user experience designer*, among others.

What practitioners actually do has been a topic of research in HCI. Multiple studies have attempted to characterize their work and their views. A key takeaway from such studies is that practitioners' work spans four main activities:

- Many HCI practitioners work to create an understanding of users and their activities. For instance, this may involve the analysis of the tasks of users or the collection of empirical data that focus on the activities and work of users [862]. To inform practical decisions, practitioners determine user requirements, benchmark solutions against competitors, and conduct task analysis and user studies in the real world [43]. User research (Part III) forms the basis of their methodology.
- Most HCI practitioners actively contribute to and often drive constructive activities. For instance, they may design and prototype interactive systems using mockups or sketches [372]. A user interface designer, for example, may use digital tools to sketch wireframes of graphical user interfaces. The design of such tools has a significant impact on their work and, thereby, the world. They may use design techniques such as wireframing, sketching, and prototyping (Part VI).
- HCI practitioners evaluate interactive systems from the user's perspective, for instance, by testing the systems with users or by doing expert reviews [43, 862]. Many of these methods have roots in the behavioral and social sciences, especially psychology and sociology. We discuss the evaluation methods in Part VIII.
- HCI practitioners engage with other professionals, for example, from marketing and software development, as well as other stakeholders, including representatives of clients and end-users [613]. Occasionally, the structure of such engagements is systematized into a process model, which we discuss in Part VI. In such engagements, success often lies outside any defined process and depends on one's ability to communicate and persuade. Such soft skills are important in influencing decision-makers and opinion leaders in organizations. They can be decisive for the success of a project.

What is the relationship between academics and practitioners? Obviously, practitioners have been trained in HCI methods and theories such as the ones covered in this book. To apply such methods effectively in a situationally appropriate manner, it is desirable to know how these methods and theories have evolved, the explanatory mechanisms they rely on, the scientific and practical assumptions they make, and their advantages and disadvantages. Together, these aspects determine their suitability for different tasks and contexts.

Nevertheless, the relationship between research and practice is complex [164]. Researchers and practitioners continually debate the relevance of research for practice and the other way around. The intertwining of research and practice in HCI has been conceptualized as a three-stage process [767]. In the first step, thousands of researchers study a phenomenon. The results are eventually packaged into guidelines, demonstrations, software tools, and much more. In the second step, millions of developers who produce HCI applications are influenced by those outcomes. In the third and final step, these applications are used by billions of people. More generally, ideas and methods propagate through undergraduate and graduate students studying and researching HCI and postdocs and academic staff actively contributing to work in industry.

This book does not cover tools and issues tailored for practitioners; for such content, we refer the reader to the fast-moving frontier of practitioner books. In this book, we focus on more enduring HCI principles and skills.

1.2.3 Relation of HCI to other fields and disciplines

HCI is a field that brings together researchers and practitioners from different disciplines. As will become evident throughout this book, HCI also draws on methods and insights from other fields, including those from psychology, sociology, cognitive science, anthropology, computer science, design, art, economics, health science, media studies, organizational studies, and many more.

Let us give a more in-depth example. Suchman [804] drew on methods from anthropology and sociology to critique early attempts to design interactive systems based on an assumption about people that was then prominent in cognitive science. They suggested that users interact based on *plans*, that is, sequences of actions they imagine before actually doing something with a computer. Her studies of photocopies showed that plans often fail and need to be changed. Users need to improvise actions and recover from errors and unplanned events. Moreover, plans are often not construed in advance. Such observations called for research on users' actual practices and a change in the way user interfaces are designed, providing more room for different structures of action and, importantly, for improvisation. Throughout the book, we will encounter many similar results from other fields.

Sometimes, other phrases are used in place of human–computer interaction. They include *usability*, which emphasizes whether users can use systems to achieve goals effectively, efficiently, and safely; *user experience*, which emphasizes the motivations and experiences of users; *interaction design*, which emphasizes the design of user interfaces and the conceptualization of novel interactions; and *human factors and ergonomics*, which emphasize human capabilities as the basis of interaction. In this book, we prefer the term HCI when describing this field.

1.3 Fundamental concepts

The book is centered on a few fundamental HCI concepts. Next, we will briefly outline them, explain why they are important, and highlight how they contribute to HCI. Figure 1.2 shows these concepts.

1.3.1 **Human-centered approach**

HCI focuses on people who use an interactive system or are affected by its use. This focus is often called being *user-centered* or *human-centered* to contrast it with a focus on the technology itself [423, 604]. Human-centeredness distinguishes HCI from many other technical disciplines.

Being human-centered has three immediate and important implications: (1) a requirement to understand users, including their needs and motivations; (2) a requirement to engage with people as part of research and design; and (3) a requirement to conduct an ethical evaluation of how an interactive system may directly or indirectly affect people.

First, being human-centered means that you seek to understand the people who will use the interactive system. This understanding encompasses everything that depends on computer use, from basic theories of human perception to how people organize work and communicate with each other. Part II of the book summarizes key theories for understanding people. This knowledge has implications for how we design the system to *match people* as opposed to requiring people to match the system. Being human-centered also means striving for the best possible understanding of people. In this book, we cover vastly different points of view and theories, ranging from theories on how humans process information (Chapter 5) to theories on motivation (Chapter 6) and collaboration (Chapter 8). A rich set of theories can be brought to bear on any use situation. Figure 1.3 shows an example.

Second, being human-centered means engaging with people as part of any research and design activity to understand their *specific* concerns and practices. In 1985, Gould and Lewis [285, p. 300] noted that

> designers must understand who the users will be. This understanding is reached in part by directly studying their cognitive, behavioral, anthropometric, and attitudinal characteristics, and in part by studying the nature of the work expected to be accomplished.

This is as true now as it was then. HCI researchers and practitioners aim to create interactive systems that support humans and their goals; therefore, they need to start from those goals. Part III presents methods for empirically studying users, what they want, what they do, and what they value.

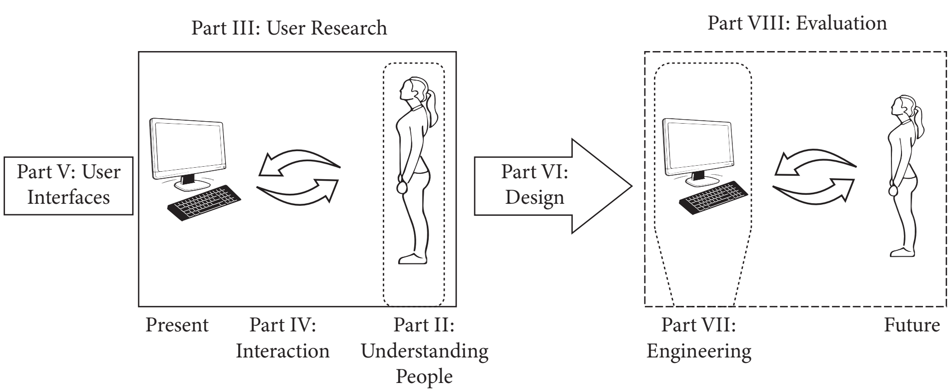


Figure 1.2 An overview of the central areas of concern in HCI. The left-hand side of the figure is about the present; the right-hand side of the figure is about the future as envisaged through design, realized through engineering, and assessed through evaluation. The dark areas indicate the corresponding parts of the book.

Figure 1.3 Theories of HCI shed light on what happens in interaction: how people perceive, experience, and behave. This figure indicates some of the core theories covered in this book, including the ones about perception, movement, motivation, and cognition.

Third, being human-centered implies a particular ethical stance toward people. This stance means that the primary rationale for any practical decision should be rooted in understanding the people who use or are affected by the system. There is a responsibility, even if it is not often stated aloud, to avoid harm and try to find the best possible solution for people. For instance, the ACM code of ethics states that "a computing professional should contribute to society and human well-being, acknowledging that all people are stakeholders in computing."[1] Thus, being human-centered means ensuring users' privacy, security, fair treatment, good work conditions, and much more. This is a theme we will revisit throughout this book.

1.3.2 Interaction

Interaction is a concept that is fundamental in HCI and specific to this field [357]. Intuitively, it refers to the reciprocal influence between people and an interactive system that takes place through the user interface. Interaction is, in other words, not a property of the system design or the user but something that emerges when they influence each other. Therefore, we need to understand the interaction to understand how users can be supported in their tasks. Interactive systems are tools that help users achieve their goals. Interaction also concerns how users operate or manage the user interface, how they interpret messages, and how they decide what to do next.

The interactive relationship between humans and computers can be complex because it is not singularly defined by the user or the computer. Additionally, interaction is often complex because it is also affected by specific activities and the context of use. This complexity is a key aspect of HCI that affects even the most mundane considerations. For instance, our expectations about the response time of a game may affect how we experience the game; our experience might fundamentally change when there is also a lag in the game's response time. However, our expectations concerning lag may differ depending on our location (on the move, at home). User expectations may be completely different for other applications and depend on factors such as social norms or what is considered an acceptable lag.

Finally, interaction often involves co-adaptation between people and computers [646], meaning that both the user and the system learn and adapt to each other during interactions. Such co-adaption can happen without hands-on contact, through changing work practices, or by fundamentally changing values or habits. Part IV outlines the concept of interaction and the distinct ways of thinking about computer use it affords.

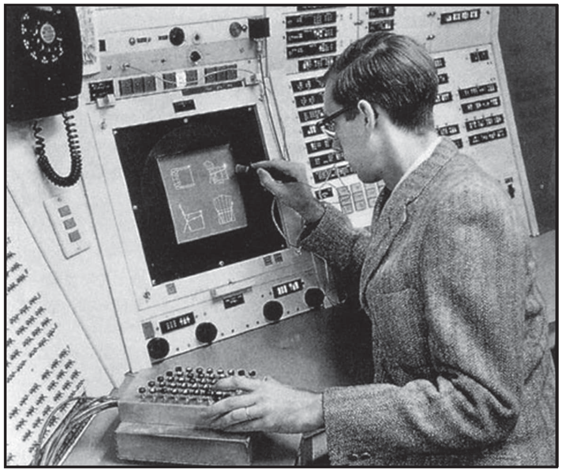
1.3.3 User interface

A key technical construct in HCI is the user interface. It refers to the parts of an interactive system that the user comes into contact with or that in other ways shape the user's perception of the system. Because the behavior of the system that the user experiences is affected by the system and the interface, design is almost invariably concerned with both the user interface and the broader system. Interactive systems include digital products, online services, appliances, web sites, smartphones, and so on. These are the primary objects of design and what practitioners and researchers try to envision differently.

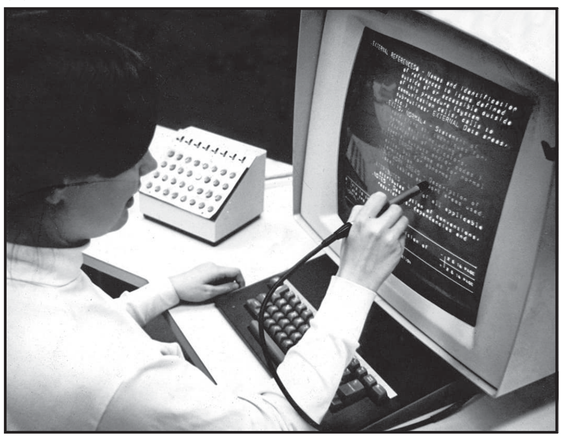
The user interfaces studied in HCI research are continuously changing (Figure 1.4). The 1970s and the early 1980s saw a lot of work on command-line interfaces, for example, research on the naming of commands and the syntax of command-line arguments. The 1980s saw pioneering work on the Xerox Star personal desktop computer.

[1] https://ethics.acm.org.

(a)



(b)



(c)

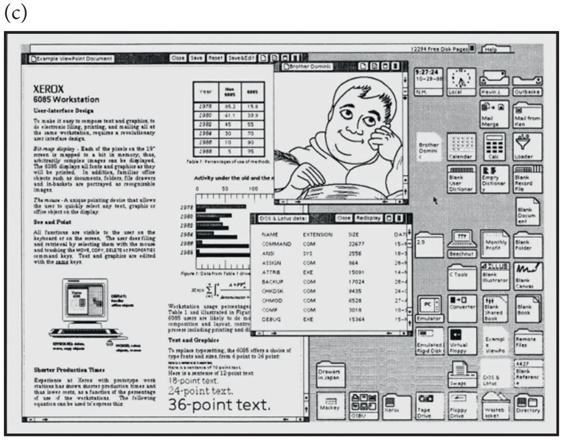


Figure 1.4 Three revolutionary user interfaces. Top: The Sketchpad interactive system was unveiled in 1964 [807]. Middle: The Hypertext Editing System, developed around 1969 [843], provided hypertext links that the user could follow by selecting them with a pen (image by Gregory Lloyd under Creative Commons License). Bottom: The Xerox Star pioneered the graphical user interface around 1981.

In addition to recent developments, research on user interfaces can draw lessons from older works. Many insights from studies on earlier systems remain relevant. For example, spoken-dialogue systems build on work on command-line interfaces, and augmented reality interfaces contain menus that are similar to those of graphical user interfaces. Part V covers a variety of user interfaces and the current understanding of what makes them work and how to design them.

1.3.4 **Design**

HCI aims to change the world by designing human-centered systems. Designing is the process of arriving at a plan, specification, prototype, system, or service—a *design*. In HCI, this often means designing a user interface and relevant parts of the underlying interactive system. However, it can also be about designing other elements, for example, services.

In general, design is about envisioning things as they could be. Although design processes have always been present in HCI, since the mid-1990s, they have been intensively developed and reflected upon [551, 899]. The book edited by Winograd [899], titled *Bringing Design to Software*, reflects this ethos. A key sentiment in that book is as follows.

> The education of computer professionals has often concentrated on the understanding of computational mechanisms, and on engineering methods that are intended to ensure that the mechanisms behave as the programmer intends. The focus is on the objects being designed: the hardware and software. The primary concern is to implement a specified functionality efficiently. When software engineers or programmers say that a piece of software works, they typically mean that it is robust, is reliable, and meets its functional specification. These concerns are indeed important. Any designer who ignores them does so at the risk of disaster.
>
> But this inward-looking perspective, with its focus on function and construction, is one-sided. To design software that really works, we need to move from a constructor's-eye view to a designer's-eye view, taking the system, the users, and the context all together as a starting point. When a designer says that something works (for example, a layout for a book cover or a design for a housing complex), the term reflects a broader meaning. Good design produces an object that works for people in a context of values and needs, to produce quality results and a satisfying experience.

The bottom line is that all interactive systems are designed. This gives HCI practitioners and researchers immense importance but also immense responsibility. Design is discussed in Part VI of the book.

1.3.5 **Engineering and computing**

Engineering refers to the use of technical principles, such as mathematics, science, and technical know-how, to realize a design that best meets a given set of expectations, which are typically captured in a requirements specification. Several emergent qualities of an implementation are important to users, including the performance, safety, robustness, and explainability of a system. When HCI researchers work on such engineering problems, they need to integrate their understanding with the theory and practice of engineering and computer science.

While there are many engineering and technical challenges in creating interactive systems, HCI focuses on the challenges that matter to people. A well-engineered interactive system is shaped and refined throughout the development process. As a consequence, HCI spills over into many areas that are not traditionally associated with it. In particular, HCI intersects many areas of computer science. For example, computer graphics considers questions on people's perception of graphical displays, scenes, and objects. Moreover, many outputs from machine learning and natural language processing research, such as speech recognition, automatic text summaries, recommendations, and search, are relevant only when implemented in interactive computers. HCI has always been central to software engineering. Early HCI research discovered that software design that assumes users will make errors tends to work more reliably [597]. This had deep implications for the design of software architectures. For example, allowing users to reverse their actions—providing an *undo* feature—can be surprisingly difficult to implement well, demonstrating that such concerns are not simply minor features to be added later on. Instead, they should be considered in the early stages of software architecture development [64, 514]. HCI is also relevant in research on computing networks and systems; for instance, researchers are making efforts to understand how people perceive network latency in networked applications, such as games [396].

In summary, HCI is not merely about the surface properties of interactive systems, but it also has deep implications for other fields regarding the development of interactive systems. Engineering is discussed in Part VII of the book.

1.3.6 **Evaluation**

In HCI, *evaluation* refers to the application of some systematic methodology to attribute human-related values to an artifact, prototype, system, or process. Examples of such attributes include performance, experience, safety, and ethical aspects, such as the avoidance of bias or harm. HCI research places a relatively strong emphasis on carrying out evaluations with users, which contrasts with some technically oriented disciplines that often measure technical properties without the involvement of users.

Verification, validation, and testing are variants of evaluation. *Verification* means ensuring that the design meets all the requirements and constraints imposed on the design task. For example, we review requirements specifications in Part VII. *Validation* means ensuring that the design is suitable for its intended purpose. As such, a design can succeed in verification but fail in validation, for example, if the design is used in some way that was unanticipated during the design process. In this sense, HCI is no different from other design fields. This book emphasizes the need to evaluate both assumptions and designs. Finally, *evaluation* refers to getting information about how well interactive systems fulfill their goals. For example, in usability testing (Chapter 43), the evaluation is performed by asking users to perform the tasks assigned to them. Several usability metrics are used to gauge how well they succeeded.

HCI evaluations almost always use multiple methods. In HCI, we need to arrive at robust, generalizable, and reproducible findings that practical decisions can be based on. It is often naive to expect a single or just a few evaluation methods to be sufficient; these methods, despite some overlaps, complement each other. As a result, practitioners often employ multiple evaluation methods, ranging from ethnographic studies and interviews to studying log files, conducting controlled experiments, and analyzing computational models.

In HCI practice, evaluation is closely coupled with iterative design. Most design processes incorporate evaluation as an ongoing activity throughout all design, building, and deployment activities. Early HCI pioneers [285] argued that "design must be iterative: There must be a cycle of design, test and measure, and redesign, repeated as often as necessary" [p. 300].

In Part VIII, we discuss principles and methods for evaluating user interfaces.

1.4 **Why HCI matters**

Understanding HCI is crucial when creating interactive systems for human use to ensure that such systems have a positive impact on their users and the world. Despite popular belief, this understanding is not trivial and cannot be achieved by amateurs; it requires knowledge and discipline. Next, we outline the main justifications for this argument. The intention is to make clear *why* research, practice, and education in HCI matter.

1.4.1 Interactive systems are difficult to use

HCI was born out of the realization that interactive systems are difficult to use. As interactive systems began to be used by nonexperts in computers, it became increasingly clear that many users had difficulties understanding how to provide input and how to interpret output. We have all used interactive systems that exhibit unexpected behavior and difficult-to-understand instructions, requiring us to work around stumbling blocks or give up using the system altogether. Paper Example 1.4.1 shows how even door handles can be difficult to use; throughout the book, many similar examples from papers are presented.

Paper Example 1.4.1: The design of everyday things

The Design of Everyday Things is a book written by Norman, one of the pioneers of HCI. Norman's book introduced ideas from cognitive psychology to bear on the design of everyday things and interactive systems. The book changed the general perception of why everyday things need to be designed with users in mind [600]. Consider the three doors below, in particular, how to open them.[2] The doors are difficult for most users (search the World Wide Web for "Norman doors" for similar examples).

continued

[2] Photographs from Don Norman, with permission to reprint. These were also used in Figures 1.2 and 1.3 in Norman's book [600].

Paper Example 1.4.1: The design of everyday things *(continued)*

The flat handlebar on the door to the left signals the door should be pushed; however, the left door needs to be pulled. The door in the middle does not offer any hint of how it can be opened. The door to the right is a sliding door, but this is so unclear to people that some friendly person has added an explanation. Whenever you see explanations or labels added to things, think: Why was this not designed to make the explanation/label unnecessary?

This example makes three points. First, even simple things are hard to design right; for interactive systems, the challenge to match users' abilities is magnitudes harder. Second, all objects, including interactive systems, have been designed; sometimes, designers have not paid sufficient attention to the people who will use their designs. Third, designers have the opportunity and even a responsibility to improve the world through their designs. Does this responsibility also apply to you?

Data suggest that such frustrations are a common experience. A study found that frustrating episodes with computers occur frequently due to error messages, faulty network connections, long download times, and difficult-to-find features [141, 334]. Shockingly, about half of the time spent on a computer can be spent dealing with such incidents. These challenges may seem minor, but over time, they compound and can have unexpected and unfortunate downstream consequences. For example, consider how poorly some user interfaces support elderly users, some of whom may struggle to use computers for even basic tasks. Such struggles are known to negatively affect users' willingness to use computers and thus their ability to benefit from computers, which can have negative effects on their ability to participate in society. Therefore, as computing spreads to broader user groups and new contexts of use, HCI is needed to make interactive systems usable and useful.

1.4.2 The egocentric fallacy

People are complex beyond ordinary intuition, and not all people are alike. It is an *egocentric fallacy* to assume that others are like us—to attempt to explain other people by reference to one's own experience. You are often NOT the user [452]. As a consequence, intuition only goes so far in helping practitioners understand interaction and design interactive systems—it can even be misleading.

A good example is HCI research on mental models. A mental model captures how people understand something. For instance, people have vastly different beliefs about how calculators work [598]. These beliefs can explain the errors and the issues they face when using calculators. For instance, many people think they need to press "clear" (or the equivalent) multiple times when using a calculator. More generally, many constructs that we use in HCI help us go beyond our everyday observations and avoid the egocentric fallacy. The implication is that we must use time and energy effectively to understand users.

1.4.3 HCI is right

As we have argued above, being human-centered is a value of its own. We have a responsibility to take the needs and abilities of users seriously when designing technology. In the words of designer Rams, "Indifference towards people and the reality they live in is the one and only cardinal sin in design." HCI has the potential to have a positive impact on the world, and HCI research can cause less bias, less frustration, and greater well-being [649].

In a classic book, Landauer [454] argued that computers are difficult to use and that this is why, at the time, computers did not improve the productivity of companies that invested in them. Landauer argued that the duty of HCI was to improve computers. We find that this argument is still valid.

1.4.4 **HCI pays off**

HCI has financial value when done correctly, as it can help open up and conquer new markets, increase productivity, and lower the costs of completing work. Investments in HCI pay off. Research on this topic has found several-fold returns on the investments in HCI made by a company [71].

User interfaces and interaction design constitute a significant part of nearly all technology projects. In 1992, Myers and Rosson [571] found that 48% of the software code was related to the user interface. This means that getting anything related to the user interface right early is important, and that necessary fixes to the user interface that are identified after the interactive system is deployed are very costly to implement. This is because the implementation, training, and documentation for the parts that need fixing may have already been completed. In summary, instead of asking if HCI pays off, it is better to ask if one can afford to do without.

1.4.5 **HCI shapes the future**

HCI is well-positioned to shape the future; see Figure 1.5 for an example of how this has occurred in the past. Systems, products, and services go through constant lifecycles of introduction, growth, maturity, and decline in the market. Competition and changing demand create a need for existing systems, products, and services to evolve or be replaced with new alternatives. User interfaces

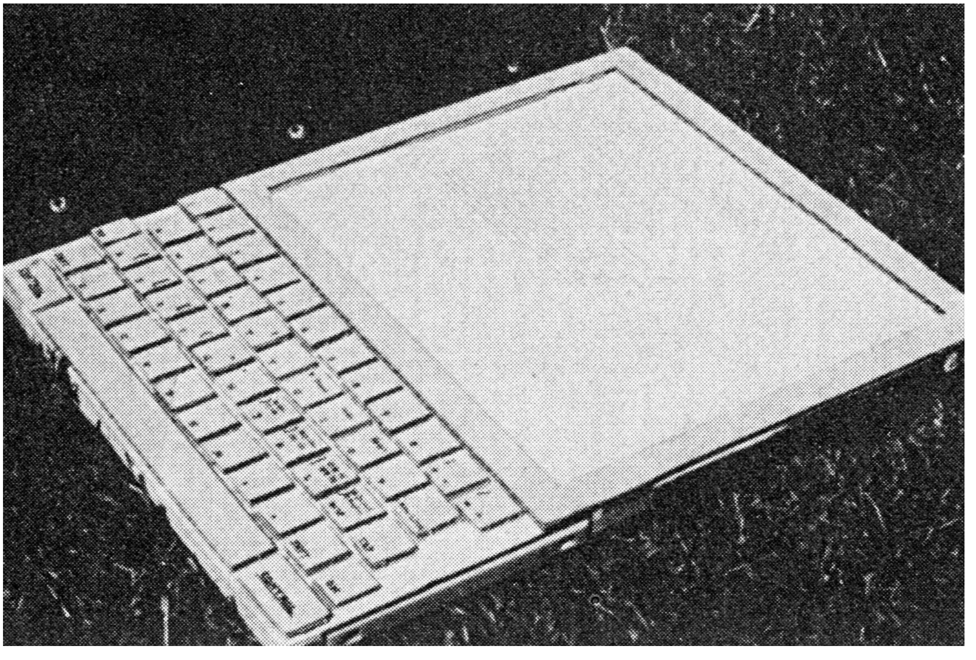


Figure 1.5 Dynabook, created in the late 1970s by Kay and Goldberg [407]. This vision predated tablets and contained many ideas for the creative and programmable use of media.

are integral parts of people's experiences when using mobile phones, apps, web sites, and so on. HCI is critical to discovering new ways to use new products, enjoy new services, and manage new systems.

Two common strategies for bringing something new to the market are *market pull* and *technology push*. Market pull refers to sensing that there is market demand for a new offering. HCI, with its rich plurality of human-centered research methods, is eminently suitable for capturing users' needs and wants and framing such findings as actionable design know-how. HCI is also exceptional at facilitating technology push. Technology push refers to the introduction of new technology into the market, which generates demand for something that was previously unavailable. Many visions and research discoveries in HCI have changed how we view computer use, such as ubiquitous computing, tangible interfaces, and virtual and augmented reality.

1.5 Our approach to HCI

The approach we follow in this book differs from those of many other textbooks. In this section, we outline the key tenets and the philosophy that have shaped the book.

1.5.1 A focus on principles

The book focuses on *elementary* HCI principles and skills. By "principle" we refer to a foundational idea or rule that explains or controls how something happens or works in HCI. For example, *direct manipulation* is a principle for organizing the interaction with graphical user interfaces (Chapter 28). The principle states that computational objects must be presented on display and acted on by the user through direct, reversible, and incremental actions with immediate feedback. There is substantive empirical evidence on the benefits of direct manipulation; hence, we find it an important principle to teach. Throughout the chapters, we introduce several principles, and we summarize them in a table in Chapter 46.

1.5.2 Pluralism in methods and theories

All methods are limited [532]. Consequently, we will cover a range of different research approaches and methodologies and carefully account for their strengths and weaknesses. We will cover the views of science, design, and engineering as ways of knowing in HCI. For example, we will explain the principles of design thinking alongside concepts of validity for experimental studies. We consider them all useful, even if they are rarely utilized together. The value of a viewpoint does not lie in the authority of who stated it but in how it helps to achieve better HCI.

At the same time, we believe that HCI phenomena can be studied at a variety of levels, all of which are relevant. This is a form of theoretical pluralism [713]. HCI phenomena span eye movements, emotional reactions, aesthetic experiences, social interactions, and organizational structures; they also span behaviors from the millisecond level to changes in the use of interactive systems over decades as well as the individual, group, and societal levels. The book covers many of these phenomena and the associated theories but makes no argument that any of them is more important than others, nor that a single theory can be used for all HCI phenomena.

1.5.3 **Essential insights supported by research**

This book focuses on some essential insights into HCI rather than analyzing current user interfaces or cataloging trends in contemporary HCI research. We have distilled the principled insights that we believe will hold and be valuable to academics and practitioners for decades. Therefore, while the reader (including instructors) may apply these essential insights to their HCI problems, it is hoped that the insights presented in this book will outlive particular interaction problems, fashions, and fads.

We have picked essential insights we believe are supported by evidence, including from academic areas other than HCI. We have prioritized this evidence-based approach over allegiance to any particular method, philosophical stance, or set of beliefs. To provide supporting evidence, we often cite relevant empirical studies. For example, consider the Gestalt principles of visual perception (Chapter 3), a set of ideas about how people group visual stimuli. Our reason for including them in this book is the evidence that they are useful in HCI, for instance, the study by Parush et al. [642]. Many parts of the book present empirical findings related to HCI principles, insights, and methods.

1.5.4 **Optimism in solving HCI problems**

Our attitude toward HCI is that many of the issues that HCI practitioners and researchers try to address are solvable. We base this attitude on the historical observation that many of the world's most successful systems are based on HCI research [e.g., 570, 767]. They include mobile devices, Wikipedia, search engines, extended realities, and computer games. Many key technologies, such as e-commerce systems, social media, augmented reality, text input technology, and editing software, have been shaped by HCI. Figure 1.6 shows an overview of computing innovations on which HCI has had a substantial, if not pivotal, impact. For us, this type of analysis shows that with the right combination of innovation, strong theories about people, and careful evaluation, HCI problems *are* solvable.

Furthermore, HCI insights are currently helping to design and improve numerous interactive systems. It does not mean that there is necessarily one complete solution for a given HCI issue, that all situations are straightforward to design for, or that HCI is easy. However, for a surprisingly large number of problems in HCI, we actually *have* suitable methods or actionable insights. As we can apply known principles and findings to solve HCI problems, the conclusion "it depends" is used less often in HCI than one might think. In this book, we emphasize that there are known answers in HCI. The reader will find that this book takes a critical position toward many popular beliefs in the field. The book emphasizes how the field has developed its capacity to solve problems. We want to give the reader actionable access to the collective insights that underpin this statement.

Do not mistake this optimism for the belief that all problems are solvable with HCI. We do not subscribe to the idea that any societal or individual problem is solvable by technology, such as by an app. We believe that in some situations, the best design decision is not to implement a system. Such decisions, just as the decision to design, should be based on a principled understanding of users and other stakeholders.

Summary

- Computing is a powerful tool; interactive systems and user interfaces help control and tame it.

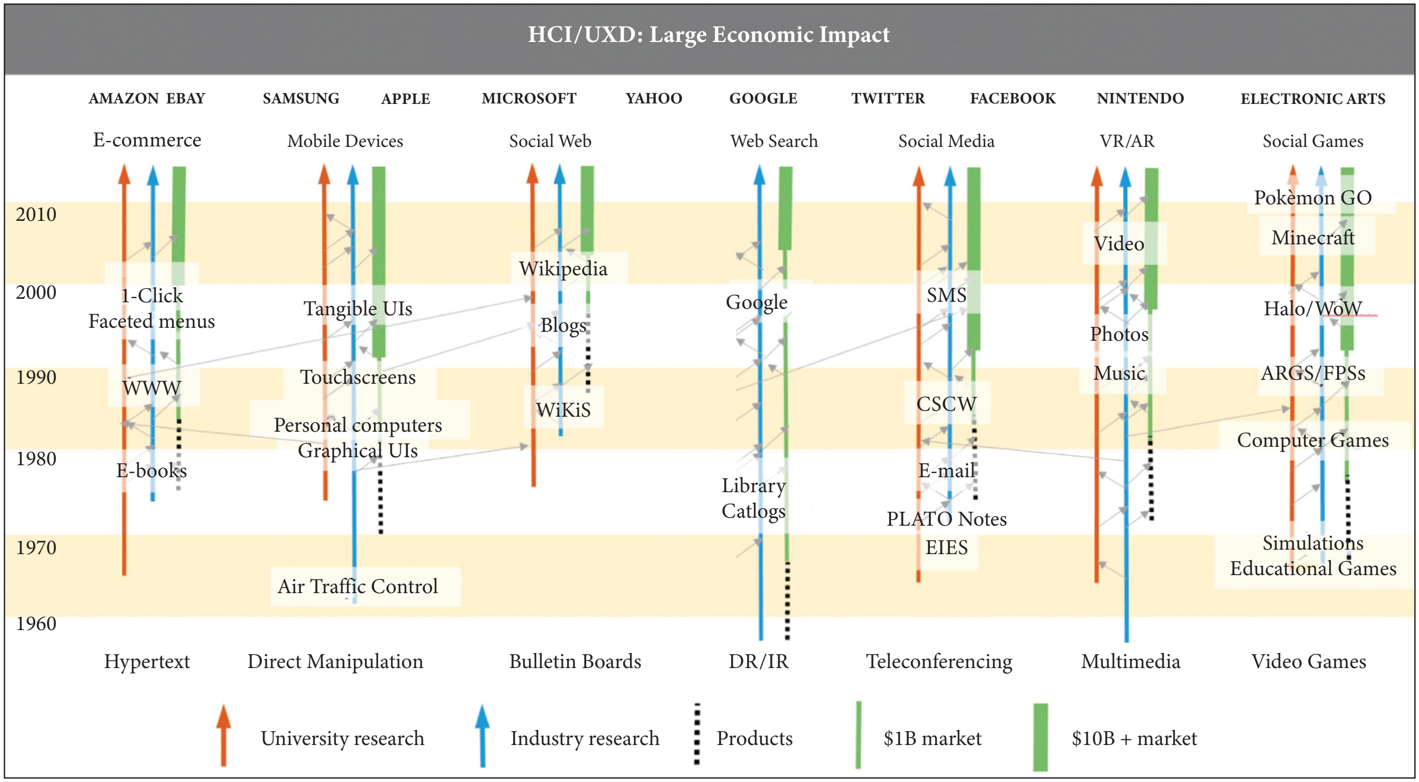


Figure 1.6 The economic impact of HCI [767]. The figure shows how well-known successes in interactive computing are related to research on HCI in academia and industry.

- HCI is concerned with people, developing technology, and understanding interaction.
- Essential activities in HCI concern understanding people, studying what users need and want, designing and engineering interactive systems, and evaluating their benefits to users.
- HCI is important because you are not the user. HCI offers a disciplined approach for tackling some of the tough problems faced in design and innovation where intuition is insufficient.

Exercises

1. Can you identify the user? Identify three relevant user groups for the following systems: (a) a mobile phone app allowing users to view timetables for buses in their city; (b) an online banking web site; (c) an educational web site that teaches children early-stage mathematics; and (d) a C++ compiler (a piece of software that translates program code into machine code).
2. Identify user-centered design (or the lack thereof) in everyday objects. As you go through your day, take note of examples of everyday things that are *not* designed well. Try to adopt the mindset that made Norman [597] see so many difficult doors; see Paper Example 1.4.1. Why are they designed or not designed well? What do you think their designers had in mind? How could they be improved?
3. Analyze the bottleneck. Think about a product or service you frequently use and consider a feature or function you find annoying or frustrating. Why do you find this feature or function annoying? What are you trying to achieve? Why is it difficult to achieve? What are the assumptions that the product or service is making that cause the bottleneck? Can you think of all the relevant factors that determine this function or feature? Can you think of an alternative design that might work better? Why do you think that your design would solve the problem? How would you test your assumptions?
4. Approaches to interactive computing. HCI is pluralistic, but what does this mean in practice? Consider being responsible for a multidisciplinary effort to improve the usefulness and usability of a social media application. Consider how the following disciplines might contribute to such effort: (a) psychology, (b) computer science, and (c) design.
5. Investigating frustrations in computer use. Choose an hour during which you will track the frustrations you encounter during computer use. Do not fuss over a definition of frustration—simply log whatever you experience as frustrating. For each frustration, write down what it was about, how frustrated you became, and how much time you lost on the episode. You can use a spreadsheet or a piece of paper. When you are done, look at your frustrations or those of fellow learners. Are they about the user interface or something else? Could they be avoided through better design? Are they frequent or rare? You can compare your findings on these frustrations to, for example, Ceaparu et al. [141] or Hertzum and Hornbæk [334].