HUMAN COMPUTER INTERACTION 2

Chrisna L. Fucio
Numeriano B. Aguado

Bachelor of Science in Information Technology
College of Computing Studies

v2.01182024

LAGUNA UNIVERSITY

Vision

Laguna University shall be a socially responsive educational institution of choice providing holistically developed individuals in the Asia-Pacific Region.

Mission

Laguna University is committed to produce academically prepared and technically skilled individuals who are socially and morally upright.

Course Code: **PC 2211 – Human Computer Interaction 2**

Course Description:

This is an intermediate course of HCI which highlights the design, development, and evaluation of human-computer interfaces, with an emphasis on usability, interaction paradigms, computer-mediated human activities, and implications to society. These issues are studied from a number of perspectives including that of the engineer and end-user. A team-based project applies your knowledge and skills to the full life cycle of an interactive human-computer interface.

Course Intended Learning Outcomes (CILO):

At the end of this course, the students should be able to:

1. Enable student to evaluate an interactive product and explain what is good and bad about it in terms of the goals and core principles of interaction design.
2. Consider how interaction design activities can be integrated into other development lifecycles.
3. Translate paradigms, visions, theories, models, and frameworks in forming interaction design.
4. Evaluate which interface is best for a given application or activity.
5. Demonstrate how to plan and carry out an observation.
6. Demonstrate both physical computing kits and software development kits and their role in construction.
7. Construct and apply the key trends in practice related to interaction design.
8. Appreciate how to do field studies.

Course Requirements:

- **Class Standing - 60%**
- **Major Exams - 40%**

Periodic Grade 100%

Prelim Grade = 60% (Class standing) + 40% (Prelim exam)
Midterm Grade = 30% (Prelim Grade) + 70 % (Midterm Grade): [60% (Midterm Class standing) + 40% (Midterm exam)]
Final Grade = 30% (Midterm Grade) + 70 % (Final Grade): [60% (Final Class standing) + 40% (Final exam)]

Table of Contents

MODULE 1
FOUNDATION OF HUMAN COMPUTER INTERACTION

Introduction

What Is Human-Computer Interaction (HCI)?

Human-computer interaction (HCI) is the field of study that focuses on optimizing how users and computers interact by designing interactive computer interfaces that satisfy users' needs. It is a multidisciplinary subject covering computer science, behavioral sciences, cognitive science, ergonomics, psychology, and design principles (www.spiceworks.com).

The emergence of HCI dates back to the 1980s, when personal computing was on the rise. It was when **desktop computers** started appearing in households and corporate offices. HCI's journey began with video games, word processors, and numerical units (www.spiceworks.com).

However, with the advent of the internet and the explosion of mobile and diversified technologies such as voice-based and Internet of Things (IoT), computing became omnipresent and omnipotent. Technological competence further led to the evolution of user interactions. Consequently, the need for developing a tool that would make such man-machine interactions more human-like grew significantly. This established HCI as a technology, bringing different fields such as cognitive engineering, linguistics, neuroscience, and others under its realm (www.spiceworks.com).

Today, HCI focuses on designing, implementing, and evaluating **interactive interfaces** that enhance user experience using computing devices. This includes user **interface design**, user-centered design, and user experience design (www.spiceworks.com).

In this module, we introduce the fundamental components of an interactive system: the human user, the computer system itself and the nature of the interactive process. We then present a view of the history of interactive systems by looking at key interaction paradigms that have been significant.

Learning Outcomes

At the end of this module, students should be able to:

1. define the what is foundation of HCI;
2. provide us with a basic overview of the capabilities and limitations that affect our ability to use computer systems.
3. understand the user at this level that we can understand what makes for successful designs and describe what human's information processing capabilities.
4. distinguish the key components of HCI and even the goals, importance, principles and scope of HCI.

Lesson 1: What Is Human-Computer Interaction (HCI)?

Human-Computer Interaction is a multidisciplinary field that focuses on designing and evaluating computer systems and technologies that people interact with. It is concerned with understanding and improving the interaction between humans and computers to make technology more user-friendly, efficient, and enjoyable (www.simplilearn.com).

HCI specialists consider how to develop and deploy computer systems that satisfy human users. The majority of this research focuses on enhancing human-computer interaction by enhancing how people utilize and comprehend an interface.

Lesson 2: The Rise of HCI (www.simplilearn.com)

As a means of examining how and why computers might be made more user-friendly, HCI emerged in the early 1980s. In a short period of time, the research area expanded to practically encompass all IT.

It changed everything when Apple introduced the Macintosh in 1984. The usage of computers has become much more accessible, making communication simpler. Keyboard, mouse, and icon-based user interfaces were popular during this period.

Despite the fact that it has expanded significantly since its inception, the field of human-computer interaction will keep growing as more information about users and computers is learned.

Giving users additional options and adopting a more humane stance to accommodate people with various preferences, disabilities, and fears, new HCI methodologies are striving to personalize interaction tools and processes as much as feasible.

In addition to the exponential growth of HCI possibilities, sign systems are adopting technologies created for phones, homes, and other personal devices more swiftly.

Lesson 3: Importance of HCI

HCI is crucial in designing **intuitive interfaces** that people with different abilities and expertise usually access. Most importantly, human-computer interaction is helpful for communities lacking knowledge and formal training on interacting with specific computing systems. With efficient HCI designs, users need not consider the intricacies and complexities of using the computing system. **User-friendly interfaces** ensure that user interactions are clear, precise, and natural (www.spiceworks.com).

- **Enhancing User Experience**: HCI places a strong emphasis on creating technology that is user-friendly and intuitive. When users have positive experiences with technology, they are more likely to adopt it, use it more effectively, and express higher levels of satisfaction.
- **Increasing Productivity and Efficiency:** HCI principles are instrumental in designing systems and interfaces that streamline tasks and workflows. By reducing user effort and cognitive load, HCI contributes to increased productivity in various domains, from business applications to healthcare.
- **Reducing Errors and Frustration:** Through careful design and usability testing, HCI helps identify and mitigate potential sources of errors and user frustration. This can be especially critical in fields like aviation, healthcare, and finance, where errors can have serious consequences.
- **Enhancing Accessibility and Inclusivity:** HCI ensures that technology is accessible to individuals with disabilities. By considering diverse user needs, HCI promotes inclusivity and equal access to information and services.
- **Improving Decision-Making:** Effective data visualization and information presentation, key aspects of HCI, help users make informed decisions. In fields like data analytics and business intelligence, HCI plays a vital role in transforming complex data into actionable insights.

- **Enhancing Safety:** In critical systems such as self-driving cars, medical devices, and industrial equipment, HCI is crucial for designing interfaces that prioritize safety and usability, reducing the likelihood of accidents.
- **Driving Innovation:** HCI encourages creative thinking in technology design. As new interaction methods and technologies emerge (e.g., touch screens, voice assistants, augmented reality), HCI researchers and designers explore innovative ways to make them user-friendly and functional.
- **Adapting to Changing Technologies:** HCI researchers and practitioners continuously adapt to evolving technology trends. As technology advances, HCI professionals help users navigate and harness the potential of new tools and interfaces.
- **Ethical Considerations:** HCI is increasingly addressing ethical concerns related to technology design and use. This includes issues like data privacy, algorithmic bias, and the social impact of technology. HCI professionals work to ensure that technology respects ethical principles and societal values.
- **Competitive Advantage:** Businesses that prioritize HCI gain a competitive edge. User-friendly products and interfaces are more likely to attract and retain customers, resulting in increased market share and profitability.
- **User Satisfaction and Loyalty:** A positive user experience fosters user satisfaction and loyalty. Satisfied users are more likely to recommend products or services to others, contributing to brand reputation and growth.
- **Cost Savings:** Designing technology with HCI principles in mind can lead to cost savings in terms of reduced customer support needs, fewer errors and rework, and increased user efficiency.

Lesson 4: Components of Human-Computer Interaction (www.simplilearn.com)

HCI is primarily composed of four essential elements:

- ✓ **The User**
    - An individual or a group of individuals who work together on a project is referred to as the user component. HCI researches the needs, objectives, and interaction styles of users.
    - By "user", we may mean an individual user, a group of users working together. An appreciation of he way people's sensory systems (sight, hearing, touch) relay

information is vital. Also, different users form different conceptions or mental models about their interactions and have different ways of learning and keeping knowledge and. In addition, cultural and national differences play a part.

- ✓ **The Goal-Oriented Task**
  - When using a computer, a user always has a purpose or aim in mind. To achieve this, the computer presents a digital representation of things.
  - When we talk about the computer, we're referring to any technology ranging from desktop computers, to large scale computer systems. For example, if we were discussing the design of a Website, then the Website itself would be referred to as "the computer". Devices such as mobile phones or VCRs can also be considered to be “computers”.

- ✓ **The Interface**
  - An essential HCI element that can improve the quality of user interaction is the interface. Many interface-related factors need to be taken into account, including the type of interaction, screen resolution, display size, and even color contrast.
  - There are obvious differences between humans and machines. In spite of these, HCI attempts to ensure that they both get on with each other and interact successfully. In order to achieve a usable system, you need to apply what you know about humans and computers, and consult with likely users throughout the design process. In real systems, the schedule and the budget are important, and it is vital to find a balance between what would be ideal for the users and what is feasible in reality.

- ✓ **The Context**
  - HCI is not only about providing better communication between users and computers but also about factoring in the context and environment in which the system is accessed.

Lesson 5: Examples of HCI (www.spiceworks.com)

Technological development has brought to light several tools, gadgets, and devices such as wearable systems, voice assistants, health trackers, and smart TVs that have advanced human-computer interaction technology.

Let's look at some prominent examples of HCI that have accelerated its evolution.

1. IoT technology

IoT devices and applications have significantly impacted our daily lives. According to a May 2022 report by IoT Analytics, global IoT endpoints are expected to reach 14.4 billion in 2022 and grow to 27 billion (approx.) by 2025. As users interact with such devices, they tend to collect their data, which helps understand different user interaction patterns. IoT companies can make critical business decisions that can eventually drive their future revenues and profits.

A recent development in the field of HCI introduced the concept of '**pre-touch sensing**' through pre-touch phones. This means the phone can detect how the user holds the phone or which finger approaches the screen first for operation. Upon detecting the user's hand movements, the device immediately predicts the user's intentions and performs the task before the user gives any instructions.

Another HCI-related development is that of '**Paper ID**'. The paper acts as a touchscreen, senses the environment, detects gestures, and connects to other IoT devices. Fundamentally, it digitizes the paper and executes tasks based on gestures by focusing on man-machine interaction variables.

2. Eye-tracking technology

Eye-tracking is about detecting where a person is looking based on the gaze point. Eye-tracking devices use cameras to capture the user's gaze along with some embedded light sources for clarity. Moreover, these devices use machine learning algorithms and image processing capabilities for accurate **gaze detection**.

Businesses can use such eye-tracking systems to monitor their personnel's visual attention. It can help companies manage distractions that tend to trouble their employees, enhancing their focus on the task. In this manner, eye-tracking technology, along with HCI-enabled interactions, can help industries monitor the daily operations of their employees or workers.

Other applications include **'driver monitoring systems'** that ensure road security. Moreover, in the future, HCI-enabled eye-tracking systems may allow users to scroll through a computer screen just by rolling their eyeballs.

3. Speech recognition technology

Speech recognition technology interprets human language, derives meaning from it, and performs the task for the user. Recently, this technology has gained significant popularity with the emergence of chatbots and virtual assistants.

For example, products such as **Amazon's Alexa, Microsoft's Cortana, Google's Google Assistant,** and **Apple's Siri** employ speech recognition to enable user interaction with their devices, cars, etc. The combination of HCI and speech recognition further fine-tune man-machine interactions that allow the devices to interpret and respond to users' commands and questions with maximum accuracy. It has various applications, such as transcribing conference calls, training sessions, and interviews.

4. AR/VR technology

AR (augmented reality) and **VR** (virtual reality) are immersive technologies that allow humans to interact with the digital world and increase the productivity of their daily tasks. For example, smart glasses enable hands-free and seamless user interaction with computing systems. Consider an example of a chef who intends to learn a new recipe. With smart glass technology, the chef can learn and prepare the target dish simultaneously.

Moreover, the technology also reduces system downtime significantly. This implies that as smart AR/VR glasses such as 'Oculus Quest 2' are supported by apps, the faults or problems in the system can be resolved by maintenance teams in real-time. This enhances user experience in a minimum time span. Also, the glasses can detect the user's response to the interface and further optimize the interaction based on the user's personality, needs, and preferences.

Thus, AR/VR technology with the blend of HCI ensures that the task is accomplished with minimal errors and also achieves greater accuracy and quality. Currently, HCI research is targeting other fields of study, such as **brain-computer interfaces** and **sentiment analysis,** to boost the user's AR/VR experience.

A recent development in this regard has been enabled via '**Dexta Haptic Gloves**.' These VR gloves can sense and process touch parameters such as surface hardness, softness, etc. These gloves can memorize a user's finger movements by locking and unlocking the finger joints as they interact in the VR environment. Later, the gloves can replicate the recorded data of feelings across various degrees in real life.

5. Cloud computing

Today, companies across different fields are embracing **remote task forces**. According to a ‘Breaking Barriers 2020’ survey by Fuze (An 8×8 Company), around 83% of employees feel more productive working remotely. Considering the current trend, conventional workplaces will witness a massive rejig and transform entirely in a couple of decades. Thanks to **cloud computing a**nd human-computer interaction, such flexible offices have become a reality.

Moreover, an employee can access data on the cloud from any physical location by exploiting **cloud-based SaaS** services. Such virtual settings streamline workflows and support seamless collaboration with remote teams across industry verticals without impacting productivity. Thus, with time, the idea of traditional offices will cease to exist, mainly because of SaaS and HCI.

Lesson 6: Goals of HCI ((www.spiceworks.com)

The principal objective of HCI is to **develop functional systems** that are usable, safe, and efficient for end-users. The developer community can achieve this goal by fulfilling the following criteria:

- Have sound knowledge of how users use computing systems
- Design methods, techniques, and tools that allow users to access systems based on their needs
- Adjust, test, refine, validate, and ensure that users achieve effective communication or interaction with the systems
- Always give priority to end-users and lay the robust foundation of HCI

To realize the above points, developers must focus on two relevant areas: **usability** and **user experience**. Let’s look at each category in detail:

1. Usability

Usability is key to HCI as it ensures that users of all types can quickly learn and use computing systems. A practical and usable HCI system has the following characteristics:

- **How to use it:** This should be easy to learn and remember for new and infrequent users to learn and remember. For example, operating systems with a user-friendly interface are easier to understand than DOS operating systems that use a command-line interface.
- **Safe:** A safe system safeguards users from undesirable and dangerous situations. This may refer to users making mistakes and errors while using the system that may lead to severe consequences. Users can resolve this through HCI practices. For example, systems can be designed to prevent users from activating specific keys or

buttons accidentally. Another example could be to provide recovery plans once the user commits mistakes. This may give users the confidence to explore the system or interface further.

- **Efficient**: An efficient system defines how good the system is and whether it accomplishes the tasks that it is supposed to. Moreover, it illustrates how the system provides the necessary support to users to complete their tasks.
- **Effective**: A practical system provides high-quality performance. It describes whether the system can achieve the desired goals.
- **Utility**: Utility refers to the various functionalities and tools provided by the system to complete the intended task. For example, a sound utility system offers an integrated development environment (IDE) that provides intermittent help to programmers or users through suggestions.
- **Enjoyable**: Users find the computing system enjoyable to use when the interface is less complex to interpret and understand.

2. User experience

User experience is a subjective trait that focuses on how users feel about the computing system when interacting with it. Here, user feelings are studied individually so that developers and support teams can target particular users to evoke positive feelings while using the system.

HCI systems classify **user interaction patterns** into the following categories and further refine the system based on the detected pattern:

- **Desirable traits** – satisfying, enjoyable, motivating, or surprising
- **Undesirable traits** – Frustrating, unpleasant, or annoying

Lesson 7: Principles of HCI (www.simplilearn.com)

Researchers and designers in the field of human-computer interaction have established numerous concepts. These regulations range from general norms and design guidelines to abstract design principles. Let's look at the most crucial HCI guidelines.

- Design for familiarity and learnability
- Make the elements readable and approachable.
- Tolerance for errors
- Flexibility

Lesson 8: Future Scope of HCI (www.simplilearn.com)

The most recent prototypes being created by businesses around the globe employing HCI theories include the following:

- Dexta haptic gloves
- Pre-touch sensing
- PaperID

MODULE
WHAT IS INTERACTION DESIGN?

Introduction

Why is it then that when computers are marketed as 'user friendly' and 'easy to use', mistakes in the operation can still occur? Did the designer of the applications actually try to use it with the trackball, or was it just that he was so expert with the system that the mistake never arose? We always assume that no one will. Criticisms are always pointed to the designers of computer software. More and more, our everyday lives involve programmed devices that do not sit on our desk, and these devices are just as unusable. Exactly how many Video cassette Recorder (VCR) designers understand the universal difficulty people have trying to set their machines to record a television program? Do car radio designers actually think it is safe to use so many knobs and displays that the driver has to divert attention away from the road completely in order to tune the radio or adjust the volume?

Computers and related devices have to be re-designed with an understanding that people with specific tasks in mind will want to use them smoothly as they use it. To do this, those who design these systems need to know how to think in terms of the eventual users' tasks and how to translate that knowledge into an executable system. But there is a problem with trying to teach the notion of designing computers for people. All designers *are* people and, most probably, they are users as well. Isn't it therefore natural to design for the user? Why does it need to be taught when we all know what a good interface looks like? As a result, the study of human–computer interaction (HCI) tends to come late in the designer's training, if at all. The scenario with which we started shows that this is a mistaken view; it is not at all natural or easy to design consistent, robust systems that will cope with all manner of user carelessness. The interface is not something that can be plugged in at the last minute; its design should be developed integrally with the rest of the system. It should not just present a 'pretty face', but should support the tasks that people actually want to do, and forgive the careless mistakes. We therefore need to consider how HCI fits into the design process.

Designing usable systems is not simply a matter for self and shared to other eventual user; it is increasingly a matter of law. National health and safety standards constrain employers to provide their workforce with usable computer systems: not just safe but usable and friendly.

Learning Outcomes

At the end of this lesson, the student should be able to:

1. Explain the difference between good and poor interaction design.
2. Describe what interaction design is and how it relates to human-computer interaction and other fields.
3. Explain the relationship between the user experience and usability.
4. Introduce what is meant by accessibility and inclusiveness in relation to human-computer interaction.
5. Describe what and who is involved in the process of interaction design.
6. Outline the different forms of guidance used in interaction design.
7. Enable you to evaluate an interactive product and explain what is good and bad about it in terms of the goals and core principles of interaction design.

Lesson 1: Good and Poor Design (Sharp, 2019)

A central concern of interaction design is to develop interactive products that are usable. By this we mean products that are generally easy to learn, effective to use, and provide an enjoy- able user experience. A good place to start thinking about how to design usable interactive products is to compare examples of well-designed and poorly designed ones. Through identifying the specific weaknesses and strengths of different interactive products, we can begin to understand what it means for something to be usable or not.

Here, we describe two examples of poorly designed products that have persisted over the years—a voice-mail system used in hotels and the ubiquitous remote control—and contrast these with two well-designed examples of the same products that perform the same function.

1.1 Voice-Mail System

Imagine the following scenario. You are staying at a hotel for a week while on a business trip. You see a blinking red light on the landline phone beside the bed. You are not sure what this means, so you pick up the handset. You listen to the tone and it goes “beep, beep, beep.” Maybe this means that there is a message for you. To find out how to access the message, you have to read a set of instructions next to the phone. You read and follow the first step:

1. Touch 41.

   The system responds: “You have reached the Sunny Hotel voice message center. Please enter the room number for which you would like to leave a message.”

   You wait to hear how to listen to a recorded message. But there are no further instructions from the phone. You look down at the instruction sheet again and read:

2. Touch*, your room number, and #.

   You do so and the system replies: “You have reached the mailbox for room 106. To leave a message, type in your password.”

You type in the room number again, and the system replies: “Please enter room number again and then your password.”

You don’t know what your password is. You thought it was the same as your room number, but clearly it is not. At this point, you give up and call the front desk for help. The person at the desk explains the correct procedure for listening to messages. This involves typing in, at the appropriate times, the room number and the extension number of the phone (the latter is the password, which is different from the room number). Moreover, it takes six steps to access a message. You give up.

What is problematic with this voice-mail system?

- It is infuriating.
- It is confusing.
- It is inefficient, requiring you to carry out a number of steps for basic tasks.
- It is difficult to use.
- It has no means of letting you know at a glance whether any messages have been left or how many there are. You have to pick up the handset to find out and then go through a series of steps to listen to them.
- It is not obvious what to do: The instructions are provided partially by the system and partially by a card beside the phone.

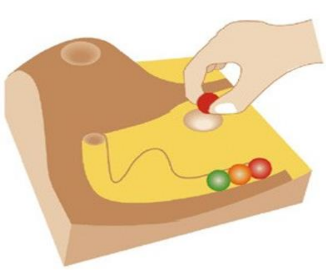


Figure 1.1 The marble answering machine
Source: Adapted from Crampton Smith (1995)

Now compare it to the phone answering shown at Figure 1.1. The illustration shows a small sketch of a phone answering machine. Incoming messages are represented using marbles. The number of marbles that have moved into the pinball-like chute indicates the number of messages. Placing one of these marbles into a dent on the machine causes the recorded message to play. Dropping the same marble into a different dent on the phone dials the caller who left the message.

How does the marble answering machine differ from the voice-mail system?

- It uses familiar physical objects that indicate visually at a glance how many messages have been left.
- It is aesthetically pleasing and enjoyable to use.
- It requires only one-step actions to perform core tasks.
- It is a simple but elegant design.
- It offers less functionality and allows anyone to listen to any of the messages.

The marble answering machine is considered a design classic. It was created by Durrell Bishop while he was a student at the Royal College of Art in London (described by Crampton Smith, 1995). One of his goals was to design a messaging system that represented its basic functionality in terms of the behavior of everyday objects. To do this, he capitalized on people's everyday knowledge of how the physical world works. In particular, he made use of the ubiquitous *(omnipresent)* everyday action of picking up a physical object and putting it down in another place.

This is an example of an interactive product designed with the users in mind. The focus is on providing them with a pleasurable experience but one that also makes efficient the activity of receiving messages. However, it is important to note that although the marble answering machine is an elegant and usable design, it would not be practical in a hotel setting. One of the main reasons is that it is not robust enough to be used in public places; for instance, the marbles could easily get lost or be taken as souvenirs. Also, the need to identify the user before allowing the messages to be played is essential in a hotel setting.

Therefore, when considering the design of an interactive product, it is important to consider where it is going to be used and who is going to use it. The marble answering machine would be more suitable in a home setting—provided that there were no children around who might be tempted to play with the marbles!

1.2 Remote Control

Every home entertainment system, be it the smart TV, set-top box, stereo system, and so forth, comes with its own remote control. Each one is different in terms of how it looks and works. Many have been designed with a dizzying array of small, multicolored, and double- labeled buttons (one on the button and one above or below it) that often seem arbitrarily positioned in relation to one another. Many viewers, especially when sitting in their living rooms, find it difficult to locate the right ones, even for the simplest of tasks, such as pausing or finding the main menu. It can be especially frustrating for those who need to put on their reading glasses each time to read the buttons. The remote control appears to have been put together very much as an afterthought.

In contrast, much effort and thought went into the design of the classic TiVo remote control with the user in mind (see Figure 1.2). TiVo is a digital video recorder that was originally developed to enable the viewer to record TV shows. The remote control was designed with large buttons that were clearly labeled and logically arranged, making them easy to locate and use in conjunction with the menu interface that appeared on the TV screen. In terms of its physical form, the remote device was designed to fit into the palm of a hand, having a peanut shape. It also has a playful look and feel about it: colorful buttons and cartoon icons are used that are distinctive, making it easy to identify them.

Figure 1.2 The TiVo remote control
Source: https://business.tivo.com/

How was it possible to create such a usable and appealing remote device where so many others have failed? The answer is simple: TiVo invested the time and effort to follow a user-centered design process. Specifically, TiVo's director of product design at the time involved potential users in the design process, getting their feedback on everything from the feel of the device in the hand to where best to place the batteries, making them easy to replace but not prone to falling out. He and his design team also resisted the trap of "buttonitis" to which so many other remote controls have fallen victim; that is one where buttons breed like rabbits— a button for every new function. They did this by restricting the number of control buttons embedded in the device to the essential ones. Other functions were then represented as part of the menu options and dialog boxes displayed on the TV screen, which could then be selected via the core set of physical control buttons. The result was a highly usable and pleasing device that has received much praise and numerous design awards.

Lesson 2: What to Design? (Sharp, 2019)

Designing interactive products requires considering who is going to be using them, how they are going to be used, and where they are going to be used. Another key concern is to understand the kind of activities people are doing when interacting with these products. The appropriateness of different kinds of interfaces and arrangements of input and output devices depends on what kinds of activities are to be supported. For example, if the activity is to enable people to bank online, then an interface that is secure, trustworthy, and easy to navigate is essential. In addition, an interface that allows the user to find out information about new services offered by the bank without it being intrusive would be useful.

The world is becoming suffused with technologies that support increasingly diverse activities. In a minute, think about what you can do using digital technology: send messages, gather information, write essays, control power plants, program, draw, plan, calculate, monitor others, and play games—just to name but a few. Now think about the types of interfaces and interactive devices that are available. They too are equally diverse: multitouch displays, speech-based systems, handheld devices, wearables, and large interactive displays—again, to name but a few. There are also many ways of designing how users can interact with a system, for instance, via the use of menus, commands, forms, icons, gestures, and so on. Furthermore, ever more innovative everyday artifacts are being created using novel materials, such as e-textiles and wearables

The Internet of Things (IoT) now means that many products and sensors can be connected to each other via the Internet, which enables them to talk to each other. Popular household IoT-enabled products include smart heating and lighting and home security systems where users can change the controls from an app on their phone or check out who is knocking on their door via a doorbell webcam. Other apps that are being developed are meant to make life easier for people, like finding a car parking space in busy areas.

The interfaces for everyday consumer items, such as cameras, microwave ovens, toasters, and washing machines, which used to be physical and the realm of product design, are now predominantly digitally based, requiring interaction design (called consumer electronics). The move toward transforming human-human transactions into solely interface-based ones has also introduced a new kind of customer interaction. Self-checkouts at grocery stores and libraries are now the norm where it is commonplace for customers to check out their own goods or books themselves, and at airports, where passengers check in their own luggage. While more cost-effective and efficient, it is impersonal and puts the onus on the person to interact with the system. Furthermore, accidentally pressing the wrong button or standing in the wrong place at a self-service checkout can result in a frustrating, and sometimes mortifying, experience.

What this all amounts to is a multitude of choices and decisions that interaction designers have to make for an ever-increasing range of products. A key question for interaction design is this: “How do you optimize the users’ interactions with a system, environment, or product so that they support the users’ activities in effective, useful, usable and pleasurable ways?” One could use intuition and hope for the best. Alternatively, one can be more principled in deciding which choices to make by basing them on an understanding of the users. This involves the following:

- Considering what people are good and bad at
- Considering what might help people with the way they currently do things
- Thinking through what might provide quality user experiences
- Listening to what people want and getting them involved in the design
- Using user-centered techniques during the design process

The aim of this book is to cover these aspects with the goal of showing you how to carry out interaction design. In particular, it focuses on how to identify users’ needs and the context of their activities. From this understanding, we move on to consider how to design usable, useful, and pleasurable interactive products.

Lesson 3: What Is Interaction Design? (Sharp, 2019)

By interaction design, we mean the following:

Designing interactive products to support the way people communicate and interact in their everyday and working lives

It is about creating user experiences that enhance and augment the way people work, communicate, and interact. More generally, Terry Winograd, 1997 originally described it as "designing spaces for human communication and interaction". John Thackara, 2001 viewed it as "the why as well as the how of our daily interactions using computers", while Dan Saffer, 2010 emphasized its artistic aspects: "the art of facilitating interactions between humans through products and services".

A number of terms have been used since to emphasize different aspects of what is being designed, including user interface design (UI), software design, user-centered design, product design, web design, user experience design, and interactive system design. Interaction design is generally used as the overarching term to describe the field, including its methods, theories, and approaches. User eXperience (UX) is used more widely in industry to refer to the profession. However, the terms can be used interchangeably. Also, it depends on their attribute and brand.

The Components of Interaction Design

We view interaction design as fundamental to many disciplines, fields, and approaches that are concerned with researching and designing computer-based systems for people. Figure 1.3 presents the core ones along with interdisciplinary fields that comprise one or more of these, such as cognitive ergonomics. It can be confusing to try to work out the differences between them as many overlap. The main differences between interaction design and the other approaches referred to in the figure come largely down to which methods, philosophies, and lenses they use to study, analyze, and design products. Another way they vary is in terms of the scope and problems they address. For example, information systems is concerned with the application of computing technology in domains such as business, health, and education, whereas ubiquitous *(always seen*) computing is concerned with the design, development, and deployment of pervasive *(spread throughout)* computing technologies (for example, IoT) and how they facilitate social inter-actions and human experiences.

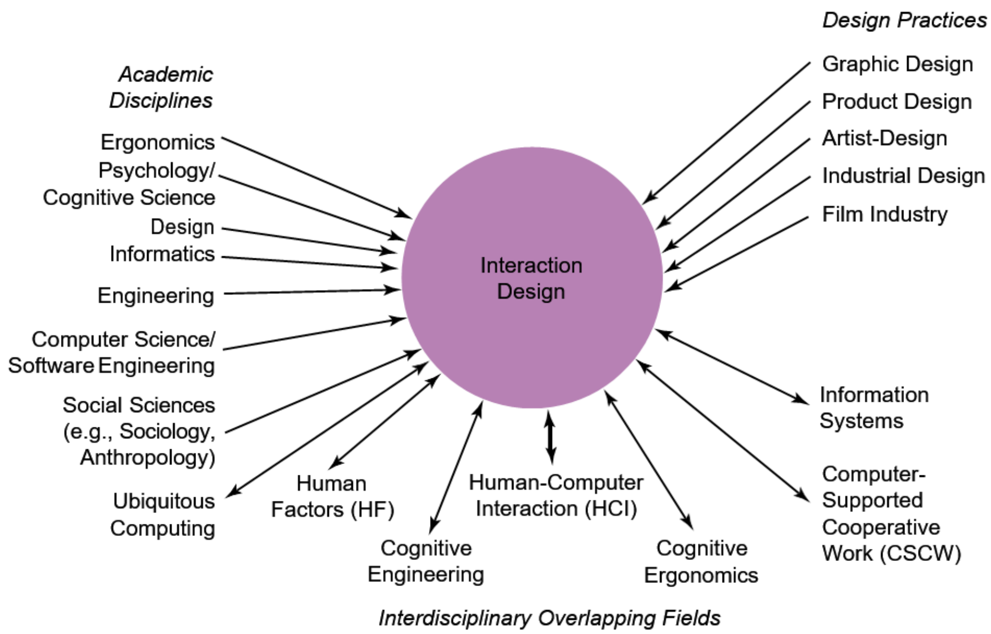


Figure 1.3 Relationship among contributing academic disciplines, design practices, and inter-disciplinary fields concerned with interaction design (double-headed arrows mean overlapping)

Who Is Involved in Interaction Design?

Figure 1.3 also shows that many people are involved in performing interaction design, ranging from social scientists to movie-makers. This is not surprising given that technology has become such a pervasive part of our lives. But it can all seem rather bewildering to the onlooker. How does the mix of players work together?

Designers need to know many different things about users, technologies, and the interactions among them to create effective user experiences. At the least, they need to understand how people act and react to events and how they communicate and interact with each other. To be able to create engaging user experiences, they also need to understand how emotions work, what is meant by aesthetics, desirability, and the role of narrative in human experience. They also need to understand the business side, technical side, manufacturing side, and marketing side. Clearly, it is difficult for one person to be well versed in all of these diverse areas and also know how to apply the different forms of knowledge to the process of interaction design.

Interaction design is ideally carried out by multidisciplinary teams, where the skill sets of engineers, designers, programmers, psychologists, anthropologists, sociologists, marketing people, artists, toy makers, product managers, and others are drawn upon. It is rarely the case, however, that a design team would have all of these professionals working together. Who to include in a team will depend on a number of factors, including a company's design philosophy, size, purpose, and product line.

One of the benefits of bringing together people with different backgrounds and training is the potential of many more ideas being generated, new methods developed, and more creative and original designs being produced. However, the downside is the costs involved. The more people there are with different backgrounds in a design team, the more difficult it can be to communicate and make progress with the designs being generated. Why? People with different backgrounds have different perspectives and ways of seeing and talking about the world. Similarly, a computer scientist's understanding of the term *representation* is often very different from that of a graphic designer or psychologist.

What this means in practice is that confusion, misunderstanding, and communication breakdowns can surface in a team. The various team members may have different ways of talking about design and may use the same terms to mean quite different things. Other problems can arise when a group of people who have not previously worked as a team are thrown together. It is founded that integration across different disciplines and expertise is difficult in many projects, especially when it comes to agreeing on and sharing tasks. The more disparate the team members—in terms of culture, background, and organizational structures—the more complex this is likely to be.

Lesson 4: The User Experience (Sharp, 2019)

The user experience refers to how a product behaves and is used by people in the real world. Jakob Nielsen and Don Norman (2014) define it as encompassing "all aspects of the end- user's interaction with the company, its services, and its products." As stressed by Jesse Garrett (2010), "Every product that is used by someone has a user experience: newspapers, ketchup bottles, reclining armchairs, cardigan sweaters." More specifically, it is about how people feel about a product and their pleasure and satisfaction when using it, looking at it, holding it, and opening or closing it. It includes their overall impression of how good it is to use, right down to the sensual effect small details have on them, such as how smoothly a switch rotates or the sound of a click and the touch of a button when pressing it. An important aspect is the quality of the experience someone has, be it a quick one, such as taking a photo; a leisurely one, such as playing with an interactive toy; or an integrated one, such as visiting a museum (Law et al., 2009).

It is important to point out that one cannot design a user experience, only design *for* a user experience. In particular, one cannot design a sensual experience, but only create the design features that can evoke it. For example, the outside case of a smartphone can be designed to be smooth, silky, and fit in the palm of a hand; when held, touched, looked at, and interacted with, that can provoke a sensual and satisfying user experience. Conversely, if it is designed to be heavy and awkward to hold, it is much more likely to end up providing a poor user experience—one that is uncomfortable and unpleasant.

Designers sometimes refer to UX as UXD. The addition of the *D* to UX is meant to encourage design thinking that focuses on the quality of the user experience rather than on the set of design methods to use (Allanwood and Beare, 2014).

It is not enough that we build products that function, that are understandable and usable, we also need to build joy and excitement, pleasure and fun, and yes, beauty to people's lives.

The first requirement for an exemplary user experience is to meet the exact needs of the customer, without fuss or bother. Next comes simplicity and elegance that produce products that are a joy to own, a joy to use. True user experience goes far beyond giving customers what they say they want, or providing checklist features. In order to achieve high-quality user experience in a company's offerings there must be a seamless merging of the services of multiple disciplines, including engineering, marketing, graphical and industrial design, and interface design.

It's important to distinguish the total user experience from the user interface (UI), even though the UI is obviously an extremely important part of the design. As an example, consider a website with movie reviews. Even if the UI for finding a film is perfect, the UX will be poor for a user who wants information about a small independent release if the underlying database only contains movies from the major studios.

We should also distinguish UX and usability: According to the definition of usability, it is a quality attribute of the UI, covering whether the system is easy to learn, efficient to use, pleasant, and so forth. Again, this is very important, and again total user experience is an even broader concept (Nielsen Norman Group, 2023).

There are many aspects of the user experience that can be considered and many ways of taking them into account when designing interactive products. Of central importance are the usability, functionality, aesthetics, content, look and feel, and emotional appeal. In addition, Jack Carroll (2004) stresses other wide-reaching aspects, including fun, health, social capital (the social resources that develop and are maintained through social networks, shared values, goals, and norms), and cultural identity, such as age, ethnicity, race, disability, family status, occupation, and education.

Understanding Users

A main reason for having a better understanding of people in the contexts in which they live, work, and learn is that it can help designers understand how to design interactive products that provide good user experiences or match a user's needs. A collaborative planning tool for a space mission, intended to be used by teams of scientists working in different parts of the world, will have quite different needs from one targeted at customer and sales agents, to be used in a furniture store to draw up kitchen layout plans. Understanding individual differences can also help designers appreciate that one size does not fit all; what works for one user group may be totally inappropriate for another. For example, children have different expectations than adults about how they want to learn or play. They may find having interactive quizzes and cartoon characters helping them along to be highly motivating, whereas most adults find them annoying. Conversely, adults often like talking-head discussions about topics, but children find them boring. Just as everyday objects like clothes, food, and games are designed differently for children, teenagers, and adults, so too should interactive products be designed for different kinds of users.

Learning more about people and what they do can also reveal incorrect assumptions that designers may have about particular user groups and what they need. For example, it is often assumed that because of deteriorating vision and dexterity, old people want things to be big—be it text or graphical elements appearing on a screen or the physical controls, like dials and switches, used to control devices. This may be true for some elderly people, but studies have shown that many people in their 70s, 80s, and older are perfectly capable of interacting with standard-size information and even small interfaces, for example, smartphones, just as well as those in their teens and 20s, even though, initially, some might think they will find it difficult (Siek et al., 2005). It is increasingly the case that as people get older, they do not like to consider themselves as lacking in cognitive and manual skills. Being aware of people's sensitivities, such as aging, is as important as knowing how to design for their capabilities (Johnson and Finn, 2017). In particular, while many older adults now feel comfortable with and use a range of technologies (for instance, email, online shop- ping, online games, or social media), they may resist adopting new technologies. This is not because they don't perceive them as being useful to their lives but because they don't want to waste their time getting caught up by the distractions that digital life brings (Knowles and Hanson, 2018), for example, not wanting to be "glued to one's mobile phone" like younger generations.

Being aware of cultural differences is also an important concern for interaction design, particularly for products intended for a diverse range of user groups from different countries. This can cause problems for designers when deciding on the format of online forms, especially if intended for global use. It is also a concern for products that have time as a function, such as operating systems, digital clocks, or car dashboards. To which cultural group do they give preference? How do they alert users to the format that is set as default? This raises the question of how easily an interface designed for one user group can be used and accepted by another. Why is it that certain products, like a fitness tracker, are universally accepted by people from all parts of the world, whereas websites are designed differently and reacted to differently by people from different cultures?

Designers have to understand more about users and know how they act and interact with one another, with information, and with various technologies, together with describing their abilities, emotions, needs, desires, and what causes them to get annoyed, frustrated, lose patience, and get bored. We draw upon relevant psychological theory and social science research. Such knowledge enables designers to determine which solutions to choose from the many design alternatives available and how to develop and test these further.

Accessibility and Inclusiveness

Accessibility refers to the extent to which an interactive product is accessible by as many people as possible. Companies like Google and Apple provide tools for their developers to promote this. The focus is on people with disabilities. For example, Android OS provides a range of tools for those with disabilities, such as hearing aid compatibility to a built-in screen reader, while Apple VoiceOver lets the user know what's happening on its devices, so they can easily navigate and even know who is in a selfie just taken, by listening to the phone. Inclusiveness means being fair, open, and equal to everyone. Inclusive design is an over-arching approach where designers strive to make their products and services accommodate the widest possible number of people. An example is ensuring that smartphones are being designed for all and made available to everyone—regardless of their disability, education, age, or income.

Whether or not a person is considered to be disabled changes over time with age, or as recovery from an accident progresses throughout their life. In addition, the severity and impact of an impairment can vary over the course of a day or in different environmental conditions. Disability can result because technologies are often designed in such a way as to necessitate a certain type of interaction that is impossible for someone with an impairment. Disability in this context is viewed as the result of poor interaction design between a user and the technology, not the impairment alone. Accessibility, on the other hand, opens up experiences so that they are accessible to all. Technologies that are now mainstream once started out as solutions to accessibility challenges. For example, SMS was designed for hearing-impaired people before it became a mainstream technology. Furthermore, designing for accessibility inherently results in inclusive design for all.

Accessibility can be achieved in two ways: first, through the inclusive design of technology, and second, through the design of assistive technology. When designing for accessibility, it is essential to understand the types of impairments that can lead to disability as they come in many forms. They are often classified by the type of impairment, for example:

- Sensory impairment (such as loss of vision or hearing)
- Physical impairment (having loss of functions to one or more parts of the body, for exam- ple, after a stroke or spinal cord injury)
- Cognitive (for instance, learning impairment or loss of memory/cognitive function due to old age or a condition such as Alzheimer's disease)

Within each type is a complex mix of people and capabilities. For example, a person might have only peripheral vision, be color blind, or have no light perception (and be registered blind). All are forms of visual impairment, and all require different design approaches. Color blindness can be overcome by an inclusive design approach. Designers can choose colors that will appear as separate colors to everyone. However, peripheral vision loss or complete blindness will often need an assistive technology to be designed.

Impairment can also be categorized as follows:

- Permanent (for example, long-term wheelchair user)
- Temporary (such as after an accident or illness)
- Situational (for instance, a noisy environment means a person can't hear)

Usability and User Experience Goals

Part of the process of understanding users is to be clear about the primary objective of developing an interactive product for them. Is it to design an efficient system that will allow them to be highly productive in their work? Is it to design a learning tool that will be challenging and motivating? Or, is it something else? To help identify the objectives, we suggest classifying them in terms of usability and user experience goals. Traditionally, usability goals are concerned with meeting specific usability criteria, such as efficiency, whereas user experience goals are concerned with explicating the nature of the user experience, for instance, to be aesthetically pleasing. It is important to note, however, that the distinction between the two types of goals is not clear-cut since usability is often fundamental to the quality of the user experience and, conversely, aspects of the user experience, such as how it feels and looks, are inextricably linked with how usable the product is. We distinguish between them here to help clarify their roles but stress the importance of considering them together when designing for a user experience. Also, historically HCI was concerned primarily with usability, but it has since become concerned with understanding, designing for, and evaluating a wider range of user experience aspects.

Usability Goals

Usability refers to ensuring that interactive products are easy to learn, effective to use, and enjoyable from the user's perspective. It involves optimizing the interactions people have with interactive products to enable them to carry out their activities at work, at school, and in their everyday lives. More specifically, usability is broken down into the following six goals:

- Effective to use (effectiveness)
- Efficient to use (efficiency)
- Safe to use (safety)
- Having good utility (utility)
- Easy to learn (learnability)
- Easy to remember how to use (memorability)

Usability goals are typically operationalized as questions. The purpose is to provide the interaction designer with a concrete means of assessing various aspects of an interactive product and the user experience. Through answering the questions, designers can be alerted very early on in the design process to potential design problems and conflicts that they might not have considered. However, simply asking "Is the system easy to learn?" is not going to be very helpful. Asking about the usability of a product in a more detailed way—for example, "How long will it take a user to figure out how to use the most basic functions for a new smartwatch; how much can they capitalize on from their prior experience; and how long would it take the user to learn the whole set of functions?"—will elicit far more information.

The following are descriptions of the usability goals and a question for each one:

a. *Effectiveness* is a general goal, and it refers to how good a product is at doing what it is supposed to do.

   Question: Is the product capable of allowing people to learn, carry out their work efficiently, access the information that they need, or buy the goods that they want?

b. *Efficiency* refers to the way a product supports users in carrying out their tasks. The marble answering machine described earlier in this chapter was considered efficient in that it let the user carry out common tasks, for example, listening to messages, through a minimal number of steps. In contrast, the voice-mail system was considered inefficient because it required the user to carry out many steps and learn an arbitrary set of sequences for the same common task. This implies that an efficient way of supporting common tasks is to let the user use single button or key presses. An example of where this kind of efficiency mechanism has been employed effectively is in online shopping. Once users have entered all of the necessary personal details in an online form to make a purchase, they can let the website save all of their personal details. Then, if they want to make another purchase at that site, they don't have to re-enter all of their personal details. A highly successful mechanism patented by Amazon.com is the one-click option, which requires users to click only a single button when they want to make another purchase.

   Question: Once users have learned how to use a product to carry out their tasks, can they sustain a high level of productivity?

c. *Safety* involves protecting the user from dangerous conditions and undesirable situations. In relation to the first ergonomic aspect, it refers to the external conditions where people work. For example, where there are hazardous conditions—such as X-ray machines or toxic chemicals—operators should be able to interact with and control computer-based systems remotely. The second aspect refers to helping any kind of user in any kind of situation to avoid the dangers of carrying out unwanted actions acciden- tally. It also refers to the perceived fears that users might have of the consequences of making errors and how this affects their behavior. Making interactive products safer in this sense involves (1) preventing the user from making serious errors by reducing the risk of wrong keys/buttons being mistakenly activated (an example is not placing the quit or delete-file command right next to the save command on a menu) and (2) providing users with various means of recovery should they make errors, such as an undo function. Safe interactive systems should engender confidence and allow the user the opportunity to explore the interface to try new operations (see Figure 1.9a). Another safety mechanism is confirming dialog boxes that give users another chance to consider their intentions (a well-known example is the appearance of a dialog box after issuing the command to delete everything in the trash, saying: “Are you sure you want to remove the items in the Trash permanently?”) (see Figure 1.9b).

Question: What is the range of errors that are possible using the product, and what measures are there to permit users to recover easily from them?

d. *Utility* refers to the extent to which the product provides the right kind of functionality so that users can do what they need or want to do. An example of a product with high utility is an accounting software package that provides a powerful computational tool that accountants can use to work out tax returns. An example of a product with low utility is a software drawing tool that does not allow users to draw freehand but forces them to use a mouse to create their drawings, using only polygon shapes.

Question: Does the product provide an appropriate set of functions that will enable users to carry out all of their tasks in the way they want to do them?

e. Learnability refers to how easy a system is to learn to use. It is well known that people don’t like spending a long time learning how to use a system. They want to get started right away and become competent at carrying out tasks without too much effort. This is especially true for interactive products intended for everyday use (for example social media, email, or a GPS) and those used only infrequently (for instance, online tax forms). To a certain extent, people are prepared to spend a longer time learning more complex systems that provide a wider range of functionality, such as web authoring tools. In these situations, pop-up tutorials can help by providing contextualized step-by-step material with hands-on exercises. A key concern is determining how much time users are prepared to spend learning a product. It seems like a waste if a product provides a range of functionality that the majority of users are unable or unprepared to spend the time learning how to use.

Question: Is it possible for the user to work out how to use the product by exploring the interface and trying certain actions? How hard will it be to learn the whole set of functions in this way?

f. *Memorability* refers to how easy a product is to remember how to use, once learned. This is especially important for interactive products that are used infrequently. If users haven’t used an operation for a few months or longer, they should be able to remember or at least rapidly be reminded how to use it. Users shouldn't have to keep relearning how to carryout tasks. Unfortunately, this tends to happen when the operations required to be learned are obscure, illogical, or poorly sequenced. Users need to be helped to remember how to do tasks. There are many ways of designing the interaction to support this. For example, users can be helped to remember the sequence of operations at different stages of a task through contextualized icons, meaningful command names, and menu options. Also, structuring options and icons so that they are placed in relevant categories of options, for example, placing all of the drawing tools in the same place on the screen, can help the user remember where to look to find a particular tool at a given stage of a task.

<u>Question</u>: What types of interface support have been provided to help users remember how to carry out tasks, especially for products and operations they use infrequently?

In addition to couching usability goals in terms of specific questions, they are turned into usability criteria. These are specific objectives that enable the usability of a product to be assessed in terms of how it can improve (or not improve) a user's performance. Examples of commonly used usability criteria are time to complete a task (efficiency), time to learn a task (learnability), and the number of errors made when carrying out a given task over time (memorability). These can provide quantitative indicators of the extent to which productivity has increased, or how work, training, or learning have been improved. They are also useful for measuring the extent to which personal, public, and home-based products support leisure and information gathering activities. However, they do not address the overall quality of the user experience, which is where user experience goals come into play.

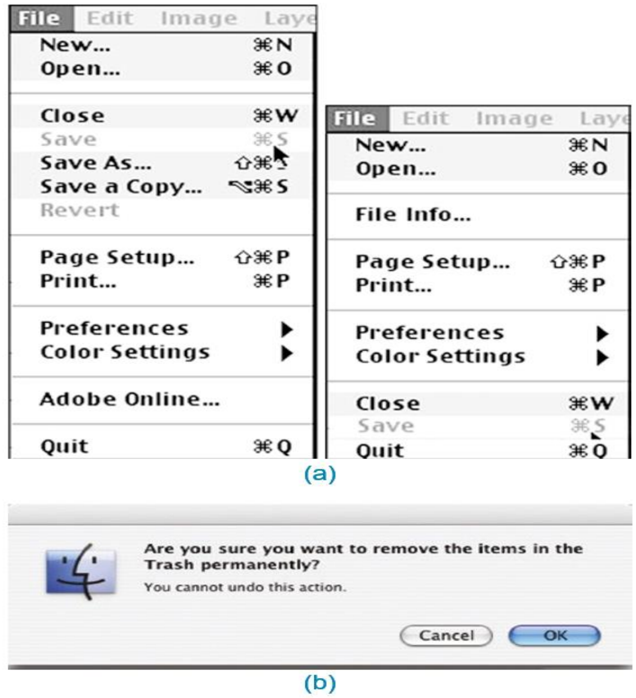


Figure 1.4 (a) A safe and unsafe menu. Which is which and why? (b) A warning dialog box for Mac OS X

User Experience Goals

A diversity of user experience goals has been articulated in interaction design, which covers a range of emotions and felt experiences. These include desirable and undesirable ones, as shown in Table 1.1.

Desirable aspects		
Satisfying	Helpful	Fun
Enjoyable	Motivating	Provocative
Engaging	Challenging	Surprising
Pleasurable	Enhancing sociability	Rewarding
Exciting	Supporting creativity	Emotionally fulfilling
Entertaining	Cognitively stimulating	Experiencing flow
Undesirable aspects		
Boring	Unpleasant	
Frustrating	Patronizing	
Making one feel guilty	Making one feel stupid	
Annoying	Cutesy	
Childish	Gimmicky	

Table 1.1 Desirable and undesirable aspects of the user experience

Many of these are subjective qualities and are concerned with how a system feels to a user. They differ from the more objective usability goals in that they are concerned with how users experience an interactive product from their perspective, rather than assessing how useful or productive a system is from its own perspective. Whereas the terms used to describe usability goals comprise a small distinct set, many more terms are used to describe the multifaceted nature of the user experience. They also overlap with what they are referring to. In so doing, they offer subtly different options for expressing the way an experience varies for the same activity over time, technology, and place. For example, we may describe listening to music in the shower as highly pleasurable, but consider it more apt to describe listening to music in the car as enjoyable. Similarly, listening to music on a high-end powerful music system may invoke exciting and emotionally fulfilling feelings, while listening to it on a smartphone that has a shuffle mode may be serendipitously enjoyable, especially not knowing what tune is next. The process of selecting terms that best convey a user's feelings, state of being, emotions, sensations, and so forth when using or interacting with a product at a given time and place can help designers understand the multifaceted and changing nature of the user experience.

The concepts can be further defined in terms of elements that contribute to making a user experience pleasurable, fun, exciting, and so on. They include attention, pace, play, interactivity, conscious and unconscious control, style of narrative, and flow. The concept of flow is popular in interaction design for informing the design of user experiences for websites, video games, and other interactive products. It refers to a state of intense emotional involvement that comes from being completely involved in an activity, like playing music, and where time flies. Instead of designing web interfaces to cater to visitors who know what they want, they can be designed to induce a state of flow, leading the visitor to some unexpected place, where they become completely absorbed.

The quality of the user experience may also be affected by single actions performed at an interface. For example, people can get much pleasure from turning a knob that has the perfect level of gliding resistance; they may enjoy flicking their finger from the bottom of a smartphone screen to reveal a new menu, with the effect that it appears by magic, or enjoy the sound of trash being emptied from the trashcan on a screen. These one-off actions can be performed seldom or several times a day—which the user never tires of doing. Dan Saffer (2014) has described these as *micro-interactions* and argues that designing these moments of interaction at the interface—despite being small—can have a big impact on the user experience.

Lesson 5: Design Principles (Sharp, 2019)

Design principles are used by interaction designers to aid their thinking when designing for the user experience. These are generalizable abstractions intended to orient designers toward thinking about different aspects of their designs. A well-known example is ***feedback***: Products should be designed to provide adequate feedback to the users that informs them about what has already been done so that they know what to do next in the interface. Another one that is important is ***findability*** (Morville, 2005). This refers to the degree to which a particular object is easy to discover or locate—be it navigating a website, moving through a building, or finding the delete image option on a digital camera. Related to this is the principle of ***navigability***: Is it obvious what to do and where to go in an interface; are the menus structured in a way that allows the user to move smoothly through them to reach the option they want?

Design principles are derived from a mix of theory-based knowledge, experience, and common sense. They tend to be written in a prescriptive manner, suggesting to designers what to provide and what to avoid at the interface—if you like, the dos and don'ts of interaction design. More specifically, they are intended to help designers explain and improve their designs. However, they are not intended to specify how to design an actual interface, for instance, telling the designer how to design a particular icon or how to structure a web portal, but to act more like triggers for designers, ensuring that they provide certain features in an interface. A number of design principles have been promoted. The best known are concerned with how to determine what users should see and do when carrying out their tasks using an interactive product. Here we briefly describe the most common ones: visibility, feedback, constraints, consistency, and affordance.

- ***Visibility***

    The importance of visibility is exemplified by our contrasting examples at the beginning of the chapter. The voice-mail system made the presence and number of waiting messages invisible, while the answering machine made both aspects highly visible. The more visible functions are, the more likely it is that users will be able to know what to do next. Don Norman (1988) describes the controls of a car to emphasize this point. The controls for different operations are clearly visible, such as indicators, headlights, horn, and hazard warning lights, indicating what can be done. The relationship between the way the controls have been positioned in the car and what they do makes it easy for the driver to find the appropriate control for the task at hand.

In contrast, when functions are out of sight, it makes them more difficult to find and to know how to use. For example, devices and environments that have become automated through the use of sensor technology (usually for hygiene and energy-saving reasons)—like faucets, elevators, and lights—can sometimes be more difficult for people to know how to control, especially how to activate or deactivate them. This can result in people getting caught short and frustrated. Figure 1.5 shows a sign that explains how to use the automatically controlled faucet for what is normally an everyday and well-learned activity. It also states that the faucets cannot be operated if wearing black clothing. It does not explain, however, what to do if you are wearing black clothing! Increasingly, highly visible controlling devices, like knobs, buttons, and switches, which are intuitive to use, have been replaced by invisible and ambiguous activating zones where people have to guess where to move their hands, bodies, or feet—on, into, or in front of—to make them work.

Figure 1.5 A sign in the restrooms at the Cincinnati airport

Source: http://www.baddesigns.com

- ***Feedback***

Related to the concept of visibility is feedback. This is best illustrated by an analogy to what everyday life would be like without it. Imagine trying to play a guitar, slice bread using a knife, or write using a pen if none of the actions produced any effect for several seconds.

There would be an unbearable delay before the music was produced, the bread was cut, or the words appeared on the paper, making it almost impossible for the person to continue with the next strum, cut, or stroke.

Feedback involves sending back information about what action has been done and what has been accomplished, allowing the person to continue with the activity. Various kinds of feedback are available for interaction design—audio, tactile, verbal, visual, and combinations of these. Deciding which combinations are appropriate for different types of activities and interactivities is central. Using feedback in the right way can also provide the necessary visibility for user interaction.

- ***Constraints***

The design concept of *constraining* refers to determining ways of restricting the kinds of user inter- action that can take place at a given moment. There are various ways that this can be achieved. A common design practice in graphical user interfaces is to deactivate certain menu options by shading them gray, thereby restricting the user only to actions permissible at that stage of the activity (see Figure 1.6). One of the advantages of this form of constraining is that it prevents the user from selecting incorrect options and thereby reduces the chance of making a mistake.

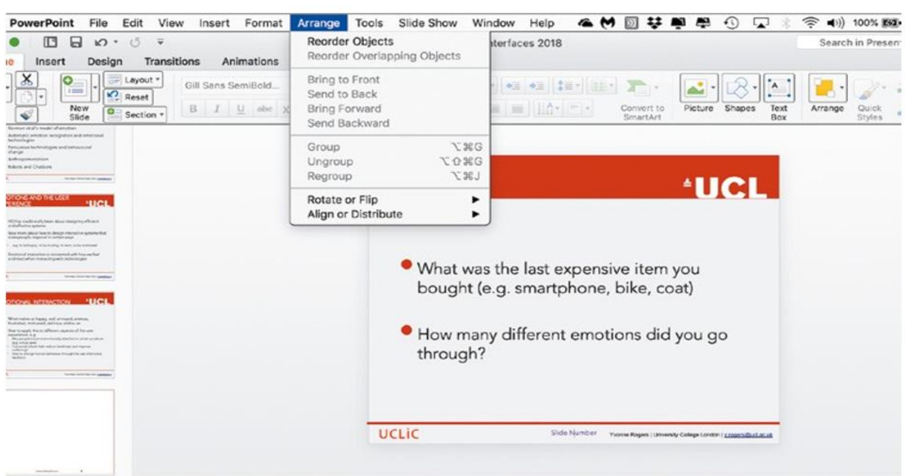
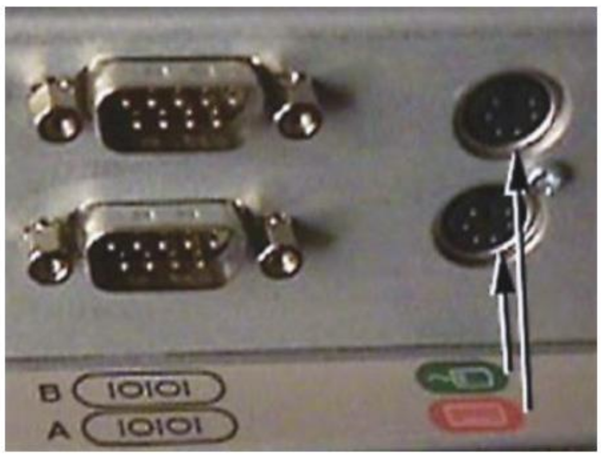


Figure 1.6 A menu showing restricted availability of options as an example of logical constraining. Gray text indicates deactivated options.

Source: https://www.ucl.ac.uk

The use of different kinds of graphical representations can also constrain a person's interpretation of a problem or information space. For example, flow chart diagrams show which objects are related to which, thereby constraining the way that the information can be perceived. The physical design of a device can also constrain how it is used; for example, the external slots in a computer have been designed to allow a cable or card to be inserted in a certain way only. Sometimes, however, the physical constraint is ambiguous, as shown in Figure 1.7. The figure shows part of the back of a computer. There are two sets of connectors; the two on the right are for a mouse and a keyboard. They look identical and are physically constrained in the



same way. How do you know which is which? Do the labels help?

Figure 1.7 Ambiguous constraints on the back of a computer
Source: http://www.baddesigns.com

- ***Consistency***

This refers to designing interfaces to have similar operations and use similar elements for achieving similar tasks. In particular, a consistent interface is one that follows rules, such as using the same operation to select all objects. For example, a consistent operation is using the same input action to highlight any graphical object on the interface, such as always clicking the left mouse button. Inconsistent interfaces, on the other hand, allow exceptions to a rule. An example is where certain graphical objects (for example, email messages presented in a table) can be high- lighted only by using the right mouse button, while all other operations are highlighted using the left mouse button. The problem with this kind of inconsistency is that it is quite arbitrary, making it difficult for users to remember and making its use more prone to mistakes.

One of the benefits of consistent interfaces, therefore, is that they are easier to learn and use. Users have to learn only a single mode of operation that is applicable to all objects. This principle works well for simple interfaces with limited operations, such as a portable radio with a small number of operations mapped onto separate buttons. Here, all the user has to do is to learn what each button represents and select accordingly. However, it can be more problematic to apply the concept of consistency to more complex interfaces, especially when many different operations need to be designed. For example, consider how to design an inter- face for an application that offers hundreds of operations, such as a word-processing appli- cation. There is simply not enough space for a thousand buttons, each of which maps to an individual operation. Even if there were, it would be extremely difficult and time-consuming for the user to search through all of them to find the desired operation. A much more effective design solution is to create categories of commands that can be mapped into subsets of operations that can be displayed at the interface, for instance, via menus.

- ***Affordance***

This is a term used to refer to an attribute of an object that allows people to know how to use it. For example, a mouse button invites pushing (in so doing, activating clicking) by the way it is physically constrained in its plastic shell. At a simple level, to afford means “to give a clue”' (Norman, 1988). When the affordances of a physical object are perceptually obvious, it is easy to know how to interact with it. For example, a door handle affords pulling, a cup handle affords grasping, and a mouse button affords pushing. The term has since been much popularized in interaction design, being used to describe how interfaces should make it obvi- ous as to what can be done when using them. For example, graphical elements like buttons, icons, links, and scrollbars are discussed with respect to how to make it appear obvious how they should be used: icons should be designed to afford clicking, scrollbars to afford moving up and down, and buttons to afford pushing.

Don Norman (1999) suggests that there are two kinds of affordance: perceived and real. Physical objects are said to have real affordances, like grasping, that are perceptually obvious and do not have to be learned. In contrast, user interfaces that are screen-based are virtual and do not have these kinds of real affordances. Using this distinction, he argues that it does not make sense to try to design for real affordances at the interface, except when designing physical devices, like control consoles, where affordances like pulling and pressing are helpful in guiding the user to know what to do. Alternatively, screen-based interfaces are better conceptualized as perceived affordances, which are essentially learned conventions. However, watching a one-year-old swiping smartphone screens, zooming in and out on images with their finger and thumb, and touching menu options suggests that kind of learning comes naturally.

Lesson 6: Applying Design Principles in Practice

One of the challenges of applying more than one of the design principles in interaction design is that trade-offs can arise among them. For example, the more you try to constrain an interface, the less visible information becomes. The same can also happen when trying to apply a single design principle. For example, the more an interface is designed to afford through trying to resemble the way physical objects look, the more it can become cluttered and difficult to use. It can also be the case that the more an interface is designed to be aesthetic, the less usable it becomes. Consistency can be a problematic design principle; trying to design an interface to be consistent with something can make it inconsistent with something else. Furthermore, sometimes inconsistent interfaces are actually easier to use than consistent interfaces. This is illustrated by Jonathan Grudin's classic (1989) use of the analogy of where knives are stored in a house. Knives come in a variety of forms, including butter knives, steak knives, table knives, and fish knives. An easy place to put them all and subsequently locate them is in the top drawer by the sink. This makes it easy for everyone to find them and follows a simple consistent rule. But what about the knives that don't fit or are too sharp to put in the drawer, like carving knives and bread knives? They are placed in a wooden block. And what about the best knives kept only for special occasions? They are placed in the cabinet in another room for safekeeping. And what about other knives like putty knives and paint-scraping knives used in home improvement projects (kept in the garage) and jack-knives (kept in one's pockets or backpack)? Very quickly, the consistency rule begins to break down.

Jonathan Grudin notes how, in extending the number of places where knives are kept, inconsistency is introduced, which in turn increases the time needed to learn where they are all stored. However, the placement of the knives in different places often makes it easier to find them because they are at hand for the context in which they are used and are also next to the other objects used for a specific task; for instance, all of the home improvement project tools are stored together in a box in the garage. The same is true when designing interfaces: introducing inconsistency can make it more difficult to learn an interface, but in the long run it can make it easier to use.

Assessment Task

In writings, answer the following activities (1.1 to 1.3) in accordance to what you have read and learn.

ACTIVITY 1.1

In practice, the makeup of a given design team depends on the kind of interactive product being built. Who do you think should be involved in developing a public kiosk providing information about the exhibits available in a science museum?

ACTIVITY 1.2

There are more desirable than undesirable aspects of the user experience listed in Table 1.1. Why do you think this is so? Should you consider all of these when designing a product?

ACTIVITY 1.3

This activity is intended for you to put into practice what you have studied in this chapter. Specifically, the objective is to enable you to define usability and user experience goals and to transform these and other design principles into specific questions to help evaluate an interactive product.

Find an everyday handheld device, for example, a remote control, digital camera, or smartphone and examine how it has been designed, paying particular attention to how the user is meant to interact with it.

a. From your first impressions, write down what is good and bad about the way the device works.
b. Give a description of the user experience resulting from interacting with it.

c. Outline some of the core micro-interactions that are supported by it. Are they pleasurable, easy, and obvious?
d. Based on your reading of this chapter and any other material you have come across about interaction design, compile a set of usability and user experience goals that you think will be most relevant in evaluating the device. Decide which are the most important ones and explain why.
e. Translate each of your sets of usability and user experience goals into two or three specific questions. Then use them to assess how well your device fares.
f. Repeat steps (c) and (d), but this time use the design principles outlined in the chapter.
g. Finally, discuss possible improvements to the interface based on the answers obtained in steps (d) and (e).

Hands-on Activity 1.4

One of the main design principles for website design is simplicity.

Create your own website following the design principles

Summary

In this module, we have looked at what interaction design is and its importance when developing apps, products, services, and systems. To begin, a number of good and bad designs were presented to illustrate how interaction design can make a difference. We described who and what is involved in interaction design and the need to understand accessibility and inclusiveness. We explained in detail what usability and user experience are, how they have been characterized, and how to operationalize them to assess the quality of a user experience resulting from interact- ing with an interactive product. The increasing emphasis on designing for the user experience and not just products that are usable was stressed. A number of core design principles were also introduced that provide guidance for helping to inform the interaction design process.

Key Points

- Interaction design is concerned with designing interactive products to support the way people communicate and interact in their everyday and working lives.
- Interaction design is multidisciplinary, involving many inputs from wide-ranging disciplines and fields.
- The notion of the user experience is central to interaction design.
- Optimizing the interaction between users and interactive products requires consideration of a number of interdependent factors, including context of use, types of activity, UX goals, accessibility, cultural differences, and user groups.
- Identifying and specifying relevant usability and user experience goals can help lead to the design of good interactive products.
- Design principles, such as feedback and simplicity, are useful heuristics for informing, analyzing, and evaluating aspects of an interactive product.

References

A. Biele, Cezary, "*Human movements in human-computer interaction (HCI*", Springer, 2022

B. Li, Yang and Hilliges, Otmar, *ed.by Artificial intelligence for human computer interaction: a modern approach*, Springer, 2021

C. Thakur, Nirmalya, and B.D., Parameshachari, ed.by" *Human-computer interaction and beyond : advances towards smart and interconnected environments (part I)",* 2021

D. Jennifer Preece, Helen Sharp, Yvonne Rogers, "*Interaction Design: Beyond Human-Computer Interaction", 5th Edition*, Wiley 2019

E. HCI (human-computer interaction) from https://www.cs.bham.ac.uk/~rxb/Teaching/HCI%20II/intro.html

F. HCI (human-computer interface) from www.Tutorialspoint.com

G. Vijay Kanade, *What Is HCI (Human-Computer Interaction)? Meaning, Importance, Examples, and Goals*: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-hci/

What is Human Computer Interaction? By Simplilearn Last updated on Nov 7, 2023: https://www.simplilearn.com/what-is-human-computer-interaction-article

MODULE 3

THE PROCESS OF INTERACTION DESIGN

Introduction

Imagine that you have been asked to design a cloud-based service to enable people to share and curate their photos, movies, music, chats, documents, and so on, in an efficient, safe, and enjoyable way. What would you do? How would you start? Would you begin by sketching how the interface might look, work out how the system architecture should be structured, or just start coding? Or, would you start by asking users about their current experiences with sharing files and examine the existing tools, for example, Dropbox and Google Drive, and based on this begin thinking about how you were going to design the new service? What would you do next? This chapter discusses the process of interaction design, that is, how to design an interactive product.

There are many fields of design, such as graphic design, architectural design, industrial design, and software design. Although each discipline has its own approach to design, there are commonalities. This approach has four phases which are iterated:

- *Discover*: Designers try to gather insights about the problem.
- *Define*: Designers develop a clear brief that frames the design challenge.
- *Develop*: Solutions or concepts are created, prototyped, tested, and iterated.
- *Deliver*: The resulting project is finalized, produced, and launched.

Interaction design also follows these phases, and it is underpinned by the philosophy of user-centered design, that is, involving users throughout development. Traditionally, interaction designers begin by doing user research and then sketching their ideas. But who are the users to be researched, and how can they be involved in development? Will they know what they want or need if we just ask them? From where do interaction designers get their ideas, and how do they generate designs?

In this module, we raise and answer these kinds of questions, discuss user-centered design, and explore the four basic activities of the interaction design process. We also introduce a lifecycle model of interaction design that captures these activities and the relationships among them.

Learning Outcomes

The main goals of this chapter are to accomplish the following:

- Reflect on what interaction design involves.
- Explain some of the advantages of involving users in development.
- Explain the main principles of a user-centered approach.
- Introduce the four basic activities of interaction design and how they are related in a simple lifecycle model.
- Ask some important questions about the interaction design process and provide the answers.
- Consider how interaction design activities can be integrated into other development lifecycles.

Lesson 1. What Is Involved in Interaction Design? (Sharp, 2019)

Interaction design has specific activities focused on discovering requirements for the product, designing something to fulfill those requirements, and producing prototypes that are then evaluated. In addition, interaction design focuses attention on users and their goals.

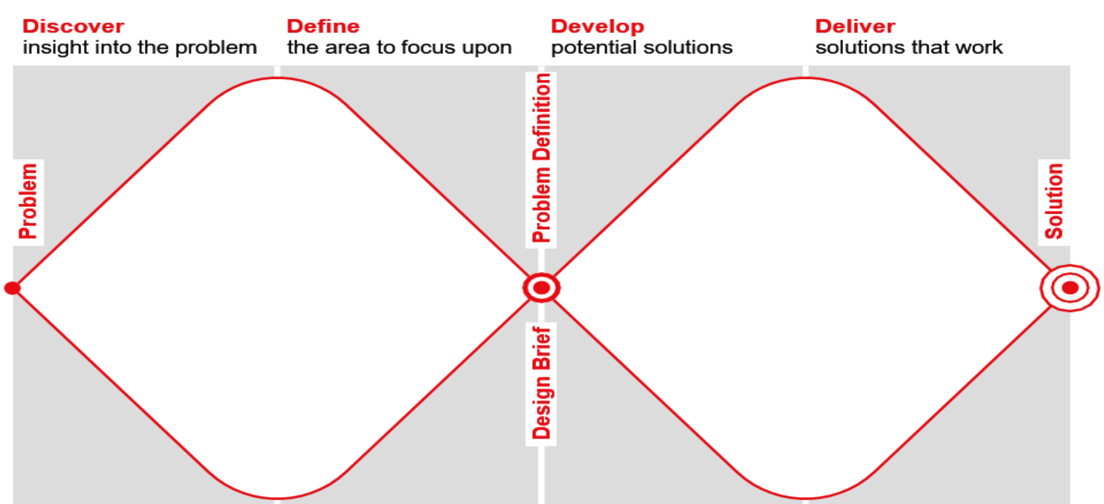


Figure 2.1 The double diamond of design

Source: Adapted from https://www.designcouncil.org.uk/news-opinion/design-process-what-double-diamond

For example, the artifact's use and target domain are investigated by taking a user-centered approach to development, users' opinions and reactions to early designs are sought, and users are involved appropriately in the development process itself. This means that users' concerns direct the development rather than just technical concerns.

Design is also about trade-offs—about balancing conflicting requirements. One common form of trade-off when developing a system to offer advice, for example, is deciding how much choice will be given to the user and how much direction the system should offer. Often, the division will depend on the purpose of the system, for example, whether it is for playing music tracks or for controlling traffic flow. Getting the balance right requires experience, but it also requires the development and evaluation of alternative solutions.

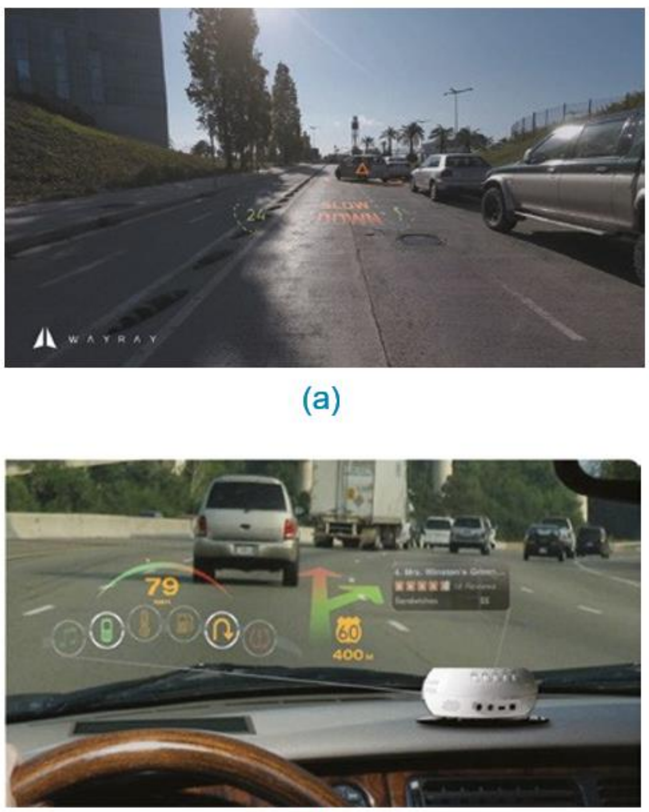
Generating alternatives is a key principle in most design disciplines and one that is also central to interaction design. Linus Pauling, twice a Nobel Prize winner, once said, "The best way to get a good idea is to get lots of ideas." Generating lots of ideas is not necessarily hard, but choosing which of them to pursue is more difficult. For example, Tom Kelley (2016) describes seven secrets for successful brainstorms, including sharpening the focus (having a well-honed problem statement), having playful rules (to encourage ideas), and getting physical (using visual props).

Involving users and others in the design process means that the designs and potential solutions will need to be communicated to people other than the original designer. This requires the design to be captured and expressed in a form that allows review, revision, and improvement. There are many ways of doing this, one of the simplest being to produce a series of sketches. Other common approaches are to write a description in natural language, to draw a series of diagrams, and to build a prototype, that is, a limited version of the final product. A combination of these techniques is likely to be the most effective. When users are involved, capturing and expressing a design in a suitable format is especially important since they are unlikely to understand jargon or specialist notations. In fact, a form with which users can interact is most effective, so building prototypes is an extremely powerful approach

- **Understanding the Problem Space**

Deciding what to design is key, and exploring the problem space is one way in which to decide. This is the first phase in the double diamond, but it can be overlooked by those new to interaction design. In the process of creating an interactive product, it can be tempting to begin at the nuts and bolts level of design. By this we mean working out how to design the physical interface and what technologies and interaction styles to use, for example, whether to use multitouch, voice, graphical user interface, heads-up display, augmented reality, gesture-based, and so forth. The problem with starting here is that potential users and their context can be misunderstood, and usability and user experience goals can be overlooked, both of which were discussed in Module1, “What Is Interaction Design?”

For example, consider the augmented reality displays and holographic navigation systems that are available in some cars nowadays (see Figure 2.3). They are the result of decades of research into human factors of information displays (for instance, Campbell et al., 2016), the driving experience itself (Perterer et al., 2013), and the suitability of different technologies as well as improvements in technology. Understanding the problem space has been critical in arriving at workable solutions that are safe and trusted. Having said that, some people may not be comfortable using a holographic navigation system and choose not to have one installed.



(a)

(b)

Figure 2.3 (a) Example of the holographic navigation display from WayRay which overlays GPS navigation instructions onto the road ahead and gathers and shares driver statistics (b) an augmented reality navigation system available in some cars today

Sources: (a) Used courtesy of WayRay, (b) Used courtesy of Muhammad Saad

While it is certainly necessary at some point to choose which technology to employ and decide how to design the physical aspects, it is better to make these decisions after articulating the nature of the problem space. By this we mean understanding what is currently the user experience or the product, why a change is needed, and how this change will improve the user experience. In the previous example, this involves finding out what is problem- atic with existing support for navigating while driving. An example is ensuring that drivers can continue to drive safely without being distracted when looking at a small GPS display mounted on the dashboard to figure out on which road it is asking them to “turn left.” Even when designing for a new user experience, it still requires understanding the context for which it will be used and the possible current user expectations.

The process of articulating the problem space is typically done as a team effort. Invariably, team members will have differing perspectives on it. For example, a project manager is likely to be concerned about a proposed solution in terms of budgets, timelines, and staffing costs, whereas a software engineer will be thinking about breaking it down into specific technical concepts. The implications of pursuing each perspective need to be considered in relation to one another. Although time-consuming and sometimes resulting in disagreements among the design team, the benefits of this process can far outweigh the associated costs: there will be much less chance of incorrect assumptions and unsupported claims creeping into a design solution that later turn out to be unusable or unwanted. Spending time enumerating and reflecting upon ideas during the early stages of the design process enables more options and possibilities to be considered. Furthermore, designers are increasingly expected to justify their choice of problems and to be able to present clearly and convincingly their rationale in business as well as design language. Being able to think and analyze, present, and argue is valued as much as the ability to create a product.

- **The Importance of Involving Users**

Module 1 stressed the importance of understanding users, and the previous description emphasizes the need to involve users in interaction design. Involving users in development is important because it’s the best way to ensure that the end product is usable and that it indeed will be used. In the past, it was common for developers to talk only to managers, experts, or proxy users, or even to use their own judgment without reference to anyone else. While others involved in designing the product can provide useful information, they will not have the same perspective as the target user who performs the activity every day or who will use the intended product on a regular basis.

In commercial projects, a role called the *product owner* is common. The product owner’s job is to filter user and customer input to the development cycle and to prioritize requirements or features. This person is usually someone with business and technical knowledge, but not interaction design knowledge, and they are rarely (if ever) a direct user of the product. Although the product owner may be called upon to assess designs, they are a proxy user at best, and their involvement does not avoid the need for user involvement.

The best way to ensure that developers gain a good understanding of users' goals, leading to a more appropriate, more usable product, is to involve target users throughout development. However, two other aspects unrelated to functionality are equally as important if the product is to be usable and used: expectation management and ownership.

Expectation management is the process of making sure that the users' expectations of the new product are realistic. Its purpose is to ensure that there are no surprises for users when the product arrives. If users feel they have been cheated by promises that have not been fulfilled, then this will cause resistance and even rejection. Marketing of the new arrival must be careful not to misrepresent the product, although it may be particularly difficult to achieve with a large and complex system (Nevo and Wade, 2007). How many times have you seen an advertisement for something that you thought would be really good to have, but when you actually see one, you discover that the marketing hype was a little exaggerated? We expect that you felt quite disappointed and let down. This is the kind of feeling that expectation management tries to avoid.

Involving users throughout development helps with expectation management because they can see the product's capabilities from an early stage. They will also understand better how it will affect their jobs and lives and why the features are designed that way. Adequate and timely training is another technique for managing expectations. If users have the chance to work with the product before it is released through training or hands-on demonstrations of a prerelease version, then they will understand better what to expect when the final product is available.

A second reason for user involvement is ownership. Users who are involved and feel that they have contributed to a product's development are more likely to feel a sense of ownership toward it and support its use (Bano et al., 2017).

How to involve users, in what roles, and for how long, needs careful planning.

- **Degrees of User Involvement**

Different degrees of user involvement are possible, ranging from fully engaged throughout all iterations of the development process to targeted participation in specific activities and from small groups of individual users in face-to-face contexts to hundreds of thousands of potential users and stakeholders online. Where available, individual users may be co-opted onto the design team so that they are major contributors to the development. This has pros and cons. On the downside, full-time involvement may mean that they become out of touch with their user community, while part-time involvement might result in a high workload for them. On the positive side, having a user engaged full or part-time does mean that the input is available continually throughout development. On the other hand, users may take part in specific activities to inform the development or to evaluate designs once they are available. This is a valuable form of involvement, but the users' input is limited to that particular activity. Where the circumstances around a project limit user involvement in this way, there are techniques to keep users' concerns uppermost in developers' minds, such as through personas.

Initially, user involvement took the form of small groups or individuals taking part in face-to-face information-gathering design, or evaluation sessions, but increasing online connectivity has led to a situation in which many thousands of potential users can contribute to product development. There is still a place for face-to-face user involvement and *in in-place* studies, but the range of possibilities for user involvement is now much wider. One example of this is online feedback exchange (OFE) systems, which are increasingly used to test design concepts with millions of target users before going to market (Foong et al., 2017).

In fact, design is becoming increasingly participative through crowdsourcing design ideas and examples, for instance (Yu et al., 2016). Where crowdsourcing is used, a range of different people are encouraged to contribute, and this can include any and all of the stakeholders. This wide participation helps to bring different perspectives to the process, which enhances the design itself, produces more user satisfaction with the final product, and engenders a sense of ownership. Another example of involving users at scale is citizen engagement, the goal of which is to engage a population—civic or otherwise—with the aim of promoting empowerment through technology. The underlying aim is to involve members of the public in helping them make a change in their lives where technology is often viewed as an integral part of the process.

Participatory design, also sometimes referred to as *cooperative design* or *co-design*, is an overarching design philosophy that places those for whom systems, technologies, and services are being designed, as central actors in creation activities. The idea is that instead of being passive receivers of new technological or industrial artifacts, end users and stake-holders are active participants in the design process.

The individual circumstances of the project affect what is realistic and appropriate. If the end-user groups are identifiable, for example, the product is for a particular company, then it is easier to involve them. If, however, the product is intended for the open market, it is unlikely that users will be available to join the design team. In this case, targeted activities and online feedback systems may be employed.

Lesson 2: What Is a User-Centered Approach? (Sharp, 2019)

We emphasize the need for a user-centered approach to development. By this we mean that the real users and their goals, not just technology, are the driving force behind product development. As a consequence, a well-designed system will make the most of human skill and judgment, will be directly relevant to the activity in hand, and will sup- port rather than constrain the user. This is less of a technique and more of a philosophy.

When the field of HCI was being established, John Gould and Clayton Lewis (1985) laid down three principles that they believed would lead to a “useful and easy to use computer system.” These principles are as follows:

1. ***Early focus on users and tasks***. This means first understanding who the users will be by directly studying their cognitive, behavioral, anthropomorphic, and attitudinal character- istics. This requires observing users doing their normal tasks, studying the nature of those tasks, and then involving users in the design process.
2. ***Empirical measurement***. Early in development, the reactions and performance of intended users to printed scenarios, manuals, and so forth, are observed and measured. Later, users interact with simulations and prototypes, and their performance and reactions are observed, recorded, and analyzed.
3. ***Iterative design***. When problems are found in user testing, they are fixed, and then more tests and observations are carried out to see the effects of the fixes. This means that design and development are iterative, with cycles of design-test-measure-redesign being repeated as often as necessary.

These three principles are now generally accepted as the basis for a user-centered approach. When this paper was written, however, they were not accepted by most developers. We discuss these principles in more detail in the following sections.

Early Focus on Users and Tasks

This principle can be expanded and clarified through the following five further principles:

1. *Users’ tasks and goals are the driving force behind the development*.

   While technology will inform design options and choices, it is not the driving force. Instead of saying “Where can we deploy this new technology?” say “What technologies are available to provide better support for users’ goals?”

2. *Users' behavior and context of use are studied, and the system is designed to support them*.
   This is not just about capturing users' tasks and goals. How people perform their tasks is also significant. Understanding behavior highlights priorities, preferences, and implicit intentions.
3. *Users' characteristics are captured and designed for*.
   When things go wrong with technology, people often think it is their fault. People are prone to making errors and have certain limitations, both cognitive and physical. Products designed to support people should take these limitations into account and try to prevent mistakes from being made. Cognitive aspects, such as attention, memory, and perception issues are introduced later, "Cognitive Aspects." Physical aspects include height, mobility, and strength. Some characteristics are general, such as color blindness, which affects about 4.5 percent of the population, but some characteristics are associated with a particular job or task. In addition to general characteristics, those traits specific to the intended user group also need to be captured.
4. *Users are consulted throughout development from earliest phases to the latest*.
   As discussed earlier, there are different levels of user involvement, and there are different ways in which to consult users.
5. *All design decisions are taken within the context of the users, their activities, and their environment*.
   This does not necessarily mean that users are actively involved in design decisions, but that is one option.

Empirical Measurement

Where possible, specific usability and user experience goals should be identified, clearly documented, and agreed upon at the beginning of the project. They can help designers choose between alternative designs and check on progress as the product is developed. Identifying specific goals up front means that the product can be empirically evaluated at regular stages throughout development.

Iterative Design

Iteration allows designs to be refined based on feedback. As users and designers engage with the domain and start to discuss requirements, needs, hopes, and aspirations, then different insights into what is needed, what will help, and what is feasible will emerge. This leads to a need for iteration—for the activities to inform each other and to be repeated. No matter how good the designers are and however clear the users may think their vision is of the required artifact, ideas will need to be revised in light of feedback, likely several times. This is particu- larly true when trying to innovate. Innovation rarely emerges whole and ready to go. It takes time, evolution, trial and error, and a great deal of patience. Iteration is inevitable because designers never get the solution right the first time (Gould and Lewis, 1985).

Four Basic Activities of Interaction Design

The four basic activities for interaction design are as follows:

1. Discovering requirements for the interactive product.
2. Designing alternatives that meet those requirements.
3. Prototyping the alternative designs so that they can be communicated and assessed.
4. Evaluating the product and the user experience it offers throughout the process.

Discovering Requirements

This activity covers the left side of the double diamond of design, and it is focused on discovering something new about the world and defining what will be developed. In the case of interaction design, this includes understanding the target users and the support an interactive product could usefully provide. This understanding is gleaned through data gathering and analysis. It forms the basis of the product's requirements and underpins subsequent design and development.

Designing Alternatives

This is the core activity of designing and is part of the Develop phase of the double diamond: proposing ideas for meeting the requirements. For interaction design, this activity can be viewed as two sub-activities: conceptual design and concrete design. ***Conceptual design*** involves producing the conceptual model for the product, and a conceptual model describes an abstraction outlining what people can do with a product and what concepts are needed to understand how to interact with it. ***Concrete design*** considers the detail of the product including the colors, sounds, and images to use, menu design, and icon design. Alternatives are considered at every point.

Prototyping

Prototyping is also part of the Develop phase of the double diamond. Interaction design involves designing the behavior of interactive products as well as their look and feel. The most effective way for users to evaluate such designs is to interact with them, and this can be achieved through prototyping. This does not necessarily mean that a piece of software is required. There are different prototyping techniques, not all of which require a working piece of software. For example, paper-based prototypes are quick and cheap to build and are effective for identifying problems in the early stages of design, and through role-playing users can get a real sense of what it will be like to interact with the product.

Evaluating

Evaluating is also part of the Develop phase of the double diamond. It is the process of deter- mining the usability and acceptability of the product or design measured in terms of a variety of usability and user-experience criteria. Evaluation does not replace activities concerned with quality assurance and testing to make sure that the final product is fit for its intended purpose, but it complements and enhances them.

The activities to discover requirements, design alternatives, build prototypes, and evaluate them are intertwined: alternatives are evaluated through the prototypes, and the results are fed back into further design or to identify alternative requirements.

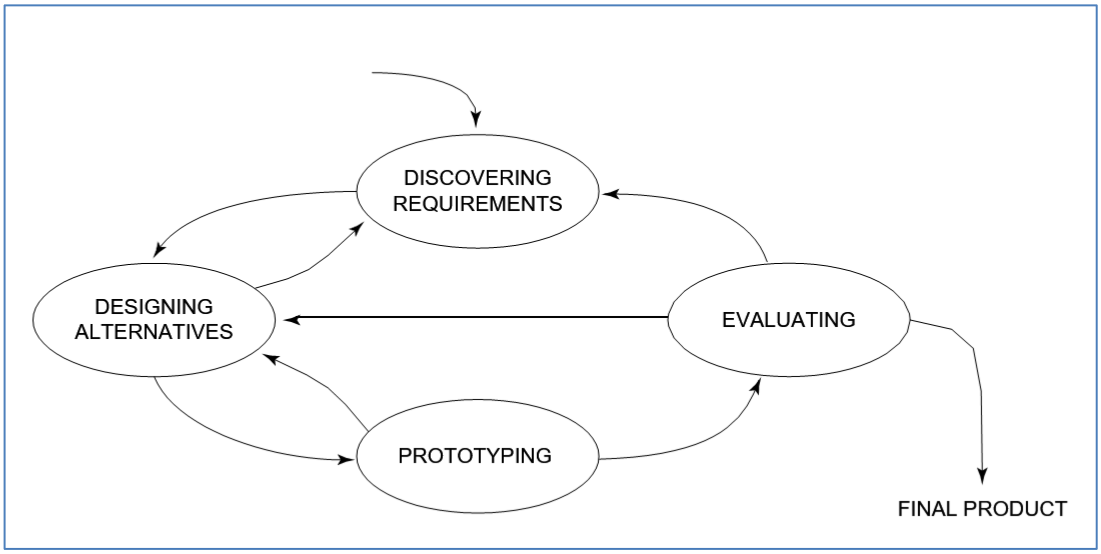
Lesson 3: A Simple Lifecycle Model for Interaction Design
(Sharp, 2019)

Understanding what activities are involved in interaction design is the first step to being able to do it, but it is also important to consider how the activities are related to one another. The term *lifecycle model* (or *process model*) is used to represent a model that captures a set of activities and how they are related. Existing models have varying levels of sophistication and complexity and are often not prescriptive. For projects involving only a few experienced developers, a simple process is adequate. However, for larger systems involving tens or hundreds of developers with hundreds or thousands of users, a simple process just isn’t enough to provide the management structure and discipline necessary to engineer a usable product.

Many lifecycle models have been proposed in fields related to interaction design. For example, software engineering lifecycle models include the waterfall, spiral, and V models.

Lets focus on the classic lifecycle model shown in Figure 2.4. This model shows how the four activities of interaction design are related, and it incorporates the three principles of user-centered design discussed earlier.

Many projects start by discovering requirements from which alternative designs are generated. Prototype versions of the designs are developed and then evaluated. During prototyping or based on feedback from evaluations, the team may need to refine the requirements or to redesign. One or more alternative designs may follow this iterative cycle in parallel. Implicit in this cycle is that the final product will emerge in an evolutionary



fashion from an initial idea through to the finished product or from limited functionality to sophisticated functionality. Exactly how this evolution happens varies from project to project. However, many times through the cycle the product goes, development ends with an evaluation activity that ensures that the final product meets the prescribed user experience and usability criteria. This evolutionary production is part of the Delivery phase of the double diamond.

Figure 2.4 A simple interaction design lifecycle model

Some Practical Issues

The discussion so far has highlighted some issues about the practical application of user- centered design and the simple lifecycle of interaction design introduced earlier. These issues are listed here:

- Who are the users?
- What are the users' needs?
- How to generate alternative designs

- How to choose among alternatives
- How to integrate interaction design activities with other lifecycle models

- **Who Are the Users?**

Identifying users may seem like a straightforward activity, but it can be harder than you think. There is a wide collection of people who all have a stake in the development of a successful product. These people are called ***stakeholders***. Stakeholders are the individuals or groups that can influence or be influenced by the success or failure of a project. Some experts observed that is pertinent to a user-centered view of development: "

The group of stakeholders for a particular product will be larger than the group of users. It will include customers who pay for it; users who interact with it; developers who design, build, and maintain it; legislators who impose rules on the development and operation of it; people who may lose their jobs because of its introduction; and so on (Sharp et al., 1999).

Identifying the stakeholders for a project helps to decide who to involve as users and to what degree, but identifying relevant stakeholders can be tricky. Ian Alexander and Suzanne Robertson (2004) suggest using an onion diagram to model stakeholders and their involvement.

- **What Are the Users' Needs?**

It may be tempting for designers simply to design what they would like to use themselves, but their ideas would not necessarily coincide with those of the target user group, because they have different experiences and expectations. Several practitioners and commentators have observed that it's an "eye-opening experience" when developers or designers see a user struggling to complete a task that seemed so clear to them.

Focusing on people's goals, usability goals and user experience goals is a more promising approach to interaction design than simply expecting stakeholders to be able to articulate the requirements for a product.

- **How to Generate Alternative Designs**

A common human tendency is to stick with something that works. While recognizing that a better solution may exist, it is easy to accept the one that works as being "good enough." Settling for a solution that is good enough may be undesirable because better alternatives may never be considered, and considering alternative solutions is a crucial step in the process of design. But where do these alternative ideas come from?

One answer to this question is that they come from the individual designer's flair and creativity. Creativity and invention are often wrapped in mystique, but a lot has been uncovered about the process and of how creativity can be enhanced or inspired.

A more pragmatic answer to this question, then, is that alternatives come from seeking different perspectives and looking at other designs. The process of inspiration and creativity can be enhanced by prompting a designer's own experience and studying others' ideas and suggestions. Deliberately seeking out suitable sources of inspiration is a valuable step in any design process. These sources may be very close to the intended new product, such as competitors' products; they may be earlier versions of similar systems; or they may be from a completely different domain.

Under some circumstances, the scope to consider alternative designs is limited. Design is a process of balancing constraints and trading off one set of requirements with another, and the constraints may mean that there are few viable alternatives available. For example, when designing software to run under the Windows operating system, the design must conform to the Windows look and feel and to other constraints intended to make Windows programs consistent for the user. When producing an upgrade to an existing system, keeping familiar elements of it to retain the same user experience may be prioritized.

- **How to Choose Among Alternative Designs**

Choosing among alternatives is mostly about making design decisions: Will the device use keyboard entry or a touch screen? Will the product provide an automatic memory function or not? These decisions will be informed by the information gathered about users and their tasks and by the technical feasibility of an idea. Broadly speaking, though, the decisions fall into two categories: those that are about externally visible and measurable features and those that are about characteristics internal to the system that cannot be observed or measured without dissecting it. For example, in a photocopier, externally visible and measurable factors include the physical size of the machine, the speed and quality of copying, the different sizes of paper it can use, and so on. Underlying each of these factors are other considerations that cannot be observed or studied without dissecting the machine. For example, the choice of materials used in a photocopier may depend on its friction rating and how much it deforms under certain conditions. In interaction design, the user experience is the driving force behind the design and so externally visible and measurable behavior is the main focus. Detailed internal workings are still important to the extent that they affect external behavior or features.

- **How to Integrate Interaction Design Activities Within Other Lifecycle Models**

As illustrated previously, many other disciplines contribute to interaction design, and some of these disciplines have lifecycles of their own. Prominent among them are those associated with software development, and integrating interaction design activities within software development has been discussed for many years.

The latest attempts to integrate these practices focus on agile software development. Agile methods began to emerge in the late 1990s. The most well-known of these are eXtreme Programming (Beck and Andres, 2005), Scrum (Schwaber and Beedle, 2002), and Kanban (Anderson, 2010). The Dynamic Systems Development Method (DSDM) (DSDM, 2014), although established before the current agile movement, also belongs to the agile family as it adheres to the agile manifesto. These methods differ, but they all stress the importance of iteration, early and repeated user feedback, being able to handle emergent requirements, and striking a good balance between flexibility and structure. They also all emphasize collaboration, face-to-face communication, streamlined processes to avoid unnecessary activities, and the importance of practice over process, that is, of getting work done.

Assessment Task

ACTIVITY 2.1 – eCommerce Webpage for Plantitas

Assume you are involved in developing a novel online experience for buying garden plants. Although many websites exist for buying plants online, you want to produce a distinct experience to increase the organization's market share. Suggest ways of applying the previous principles in this task.

Comments on what to do:

To address the first three principles, you would need to find out (re-search) about the tasks and goals, behavior, and characteristics of potential customers of the new experience, together with any different contexts of use.

Study some current users of existing online plant shops will provide some information, and it will also identify some challenges to be addressed in the new experience. However, as you want to increase the organization's market share, consulting existing users alone would not be enough. Alternative avenues of investigation include physical shopping situations—for example, shopping at the market, in the local corner shop, and so on, and local gardening clubs, radio programs, or podcasts. These alternatives will help you find the advantages and disadvantages of buying plants in different settings, and you will observe different behaviors. By looking at these options, a new set of potential users and contexts can be identified.

For the fourth principle, the set of new users will emerge as investigations progress, but people who are representative of the user group may be accessible from the beginning. Work- shops or evaluation sessions could be run with them, possibly in one of the alternative shop- ping environments such as the market. The last principle could be supported through the creation of a design room that houses all of the data collected, and it is a place where the development team can go to find out more about the users and the product goals.

ACTIVITY 2.2 – Plan for a digital Timepiece for Basketball Game

These days, timepieces (such as clocks, wristwatches, and so on) have a variety of functions. Not only do they tell the time and date, but they can speak to you, remind you when it's time to do something, and record your exercise habits among other things. The interface for these devices, however, shows the time in one of two basic ways: as a digital number such as 11:40 or through an analog display with two or three hands—one to represent the hour, one for the minutes, and one for the seconds.

This in-depth activity is to design an innovative timepiece. This could be in the form of a wristwatch, a mantelpiece clock, a sculpture for a garden or balcony, or any other kind of timepiece you prefer. The goal is to be inventive and exploratory by following these steps:

(a) Think about the interactive product that you are designing: What do you want it to do? Find three to five potential users, and ask them what they would want. Write a list of requirements for the clock, together with some usability criteria and user experience criteria based on the definitions in Module1.

(b) Look around for similar devices and seek out other sources of inspiration that you might find helpful. Make a note of any findings that are interesting, useful, or insightful.

(c) Sketch some initial designs for the timepiece. Try to develop at least two distinct alternatives that meet your set of requirements.

(d) Evaluate the two designs by using your usability criteria and by role-playing an interaction with your sketches. Involve potential users in the evaluation, if possible. Does it do what you want? Is the time or other information being displayed always clear? Design is iterative, so you may want to return to earlier elements of the process before you choose one of your alternatives.

Summary

In this module, we looked at user-centered design and the process of interaction design. That is, what is user-centered design, what activities are required in order to design an interactive product, and how are these activities related? A simple interaction design lifecycle model consisting of four activities was introduced, and issues surrounding the involvement and identification of users, generating alternative designs, evaluating designs, and integrating user- centered concerns with other lifecycles were discussed.

Key Points

- Different design disciplines follow different approaches, but they have commonalities that are captured in the double diamond of design.
- It is important to have a good understanding of the problem space before trying to build anything.
- The interaction design process consists of four basic activities: discover requirements, design alternatives that meet those requirements, prototype the designs so that they can be communicated and assessed, and evaluate them.
- User-centered design rests on three principles: early focus on users and tasks, empirical measurement, and iterative design. These principles are also key for interaction design.
- Involving users in the design process assists with expectation management and feelings of ownership, but how and when to involve users requires careful planning.
- There are many ways to understand who users are and what their goals are in using a product, including rapid iterations of working prototypes.
- Looking at others' designs and involving other people in design provides useful inspiration and encourages designers to consider alternative design solutions, which is key to effective design.
- Usability criteria, technical feasibility, and users' feedback on prototypes can all be used to choose among alternatives.
- Prototyping is a useful technique for facilitating user feedback on designs at all stages.
- Interaction design activities are becoming better integrated with lifecycle models from other related disciplines such as software engineering.

References

a. Biele, Cezary, "*Human movements in human-computer interaction (HCI*", Springer, 2022

b. Li, Yang and Hilliges, Otmar, *ed.by Artificial intelligence for human computer interaction: a modern approach*, Springer, 2021

c. Thakur, Nirmalya, and B.D., Parameshachari, ed.by" *Human-computer interaction and beyond : advances towards smart and interconnected environments (part I)",* 2021

d. Jennifer Preece, Helen Sharp, Yvonne Rogers, "*Interaction Design: Beyond Human-Computer Interaction", 5th Edition*, Wiley 2019

e. HCI (human-computer interaction) from https://www.cs.bham.ac.uk/~rxb/Teaching/HCI%20II/intro.html

f. HCI (human-computer interface) from www.Tutorialspoint.com

g. Vijay Kanade, *What Is HCI (Human-Computer Interaction)? Meaning, Importance, Examples, and Goals*: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-hci/

h. What is Human Computer Interaction? By Simplilearn Last updated on Nov 7, 2023: https://www.simplilearn.com/what-is-human-computer-interaction-article

MODULE 4
CONCEPTUALIZING INTERACTION

Introduction

When coming up with new ideas as part of a design project, it is important to conceptualize them in terms of what the proposed product will do. Sometimes, this is referred to as creating a *proof of concept*. In relation to the double diamond framework, it can be viewed as an initial pass to help define the area and also when exploring solutions. One reason for needing to do this is as a reality check where fuzzy ideas and assumptions about the benefits of the proposed product are scrutinized in terms of their feasibility: How realistic is it to develop what they have suggested, and how desirable and useful will it actually be? Another reason is to enable designers to begin articulating what the basic building blocks will be when developing the product. From a user experience (UX) perspective, it can lead to better clarity, forcing designers to explain how users will understand, learn about, and interact with the product.

Learning Outcomes

At the end of this lesson, the student should be able to:

- Explain how to conceptualize interaction.
- Describe what a conceptual model is and how to begin to formulate one.
- Discuss the use of interface metaphors as part of a conceptual model.
- Outline the core interaction types for informing the development of a conceptual model.
- Introduce paradigms, visions, theories, models, and frameworks informing interaction design.

Lesson 1. Conceptualizing Interaction (Sharp, 2019)

When beginning a design project, it is important to be clear about the underlying assumptions and claims. By an assumption, we mean taking something for granted that requires fur- ther investigation; for example, people now want an entertainment and navigation system in their cars. By a claim, we mean stating something to be true when it is still open to question. For instance, a multimodal style of interaction for controlling this system—one that involves speaking or gesturing while driving—is perfectly safe.

Writing down your assumptions and claims and then trying to defend and support them can highlight those that are vague or wanting. In so doing, poorly constructed design ideas can be reformulated. In many projects, this process involves identifying human activities and interactivities that are problematic and working out how they might be improved through being supported with a different set of functions. In others, it can be more speculative, requiring thinking through how to design for an engaging user experience that does not exist.

Explaining people's assumptions and claims about why they think something might be a good idea (or not) enables the design team as a whole to view multiple perspectives on the problem space and, in so doing, reveals conflicting and problematic ones. The following framework is intended to provide a set of core questions to aid design teams in this process:

- Are there problems with an existing product or user experience? If so, what are they?
- Why do you think there are problems?
- What evidence do you have to support the existence of these problems?
- How do you think your proposed design ideas might overcome these problems?

Making clear what one's assumptions are about a problem and the claims being made about potential solutions should be carried out early on and throughout a project. Design teams also need to work out how best to conceptualize the design space. Primarily, this involves articulating the proposed solution as a conceptual model with respect to the user experience. The benefits of conceptualizing the design space in this way are as follows:

- **Orientation** Enabling the design team to ask specific kinds of questions about how the conceptual model will be understood by the targeted users.
- **Open-Mindedness** Allowing the team to explore a range of different ideas to address the problems identified.
- **Common Ground** Allowing the design team to establish a set of common terms

that all can understand and agree upon, reducing the chance of misunderstandings and confusion arising later.

Once formulated and agreed upon, a conceptual model can then become a shared blueprint leading to a testable proof of concept. It can be represented as a textual description and/ or in a diagrammatic form, depending on the preferred *common language* used by the design team. It can be used not just by user experience designers but also to communicate ideas to business, engineering, finance, product, and marketing units. The conceptual model is used by the design team as the basis from which they can develop more detailed and concrete aspects of the design. In doing so, design teams can produce simpler designs that match up with users' tasks, allow for faster development time, result in improved customer uptake, and need less training and customer support (Johnson and Henderson, 2012).

• Conceptual Models

A *model* is a simplified description of a system or process that helps describe how it works. In this section, we look at a particular kind of model used in interaction design intended to articulate the problem and design space—the *conceptual model*. We describe more generally how models have been developed to explain phenomena in human-computer interaction.

Jeff Johnson and Austin Henderson (2002) define a conceptual model as "a high-level description of how a system is organized and operates". In this sense, it is an abstraction outlining what people can do with a product and what concepts are needed to understand how to interact with it. A key benefit of conceptualizing a design at this level is that it enables "designers to straighten out their thinking before they start laying out their widgets".

In a nutshell, a conceptual model provides a working strategy and a framework of general concepts and their interrelations. The core components are as follows:

- Metaphors and analogies that convey to people how to understand what a product is used for and how to use it for an activity (for example browsing and bookmarking).
- The concepts to which people are exposed through the product, including the task-domain objects they create and manipulate, their attributes, and the operations that can be per- formed on them (such as saving, revisiting, and

organizing).

- The relationships between those concepts (for instance, whether one object contains another).
- The mappings between the concepts and the user experience the product is designed to support or invoke (for example, one can revisit a page through looking at a list of visited sites, most-frequently visited, or saved websites).

How the various metaphors, concepts, and their relationships are organized determines the user experience. By explaining these, the design team can debate the merits of providing different methods and how they support the main concepts, for example, saving, revisiting, categorizing, reorganizing, and their mapping to the task domain. They can also begin discussing whether a new overall metaphor may be preferable that combines the activities of browsing, searching, and revisiting. In turn, this can lead the design team to articulate the kinds of relationships between them, such as containership. For example, what is the best way to sort and revisit saved pages, and how many and what types of containers should be used (for example, folders, bars, or panes)? The same enumeration of concepts can be repeated for other functions of the web browser—both current and new. In so doing, the design team can begin to work out systematically what will be the simplest and most effective and memorable way of supporting users while browsing the Internet.

The best conceptual models are often those that appear obvious and simple; that is, the operations they support are intuitive to use. However, sometimes applications can end up being based on overly complex conceptual models, especially if they are the result of a series of upgrades, where more and more functions and ways of doing something are added to the original conceptual model. While tech companies often provide videos showing what new features are included in an upgrade, users may not pay much attention to them or skip them entirely. Furthermore, many people prefer to stick to the methods they have always used and trusted and, not surprisingly, become annoyed when they find one or more have been removed or changed. For example, when Facebook rolled out its revised newsfeed a few years back, many users were unhappy, as evidenced by their post- ings and tweets, preferring the old interface that they had gotten used to. A challenge for software companies, therefore, is how best to introduce new features that they have added to an upgrade—and explain their assumed benefits to users—while also justifying why they removed others.

Most interface applications are actually based on well-established conceptual models. For example, a conceptual model based on the core aspects of the customer experience when at a shopping mall underlies most online shopping websites. These include the placement of items that a customer wants to purchase into a shopping cart or basket and proceeding to checkout when they're ready to make the purchase. Collections of patterns are now readily available to help design the interface for these core transactional processes, together with many other aspects of a user experience, meaning interaction designers do not have to start from scratch every time they design or redesign an application. Examples include patterns for online forms and navigation on mobile phones.

It is rare for completely new conceptual models to emerge that transform the way daily and work activities are carried out at an interface. All of innovations made what was previously limited to a few skilled people accessible to all, while greatly expanding what is possible. The graphical desktop dramatically changed how office tasks could be performed (including creating, editing, and printing documents). Performing these tasks using the computers prevalent at the time was significantly more arduous, having to learn and use a command language like DOS or UNIX. Digital spreadsheets made accounting highly flexible and easier to accomplish, enabling a diversity of new computations to be performed simply through filling in interactive boxes. The World Wide Web allowed anyone to browse a network of information remotely. Since then, e-readers and digital authoring tools have introduced new ways of reading documents and books online, supporting associated activities such as annotating, highlighting, linking, commenting, copying, and tracking. The web has also enabled and made many other kinds of activities easier, such as browsing for news, weather, sports, and financial information, as well as banking, shop- ping, and learning online among other tasks. Importantly, all of these conceptual models were based on familiar activities.

- ## Interface Metaphors

Metaphors are considered to be a central component of a conceptual model. They provide a structure that is similar in some way to aspects of a familiar entity (or entities), but they also have their own behaviors and properties. More specifically, an *interface metaphor* is one that is instantiated in some way as part of the user interface, such as the desktop metaphor. Another well-known one is the *search engine*, originally coined in the early 1990s to refer to a software tool that indexed and retrieved files remotely from the Internet using various algorithms to match terms selected by the user. The metaphor invites comparisons between a mechanical engine, which has several working parts, and the everyday action of looking in different places to find something. The functions supported by a search engine also include other features besides those belonging to an engine that searches, such as listing and prioritizing the results of a search. It also does these actions in quite different ways from how a mechanical engine works or how a human being might search a library for books on a given topic. The similarities implied by the use of the term *search engine*, therefore, are at a general level. They are meant to conjure up the essence of the process of finding relevant information, enabling the user to link these to less familiar aspects of the functionality provided.

Interface metaphors are intended to provide familiar entities that enable people readily to understand the underlying conceptual model and know what to do at the interface. However, they can also contravene people's expectations about how things should be, such as the recycle bin (trash can) that sits on the desktop. Logically and culturally (meaning, in the real world), it should be placed under the desk. But users would not have been able to see it because it would have been hidden by the desktop surface. So, it needed to go on the desk- top. While some users found this irksome, most did not find it to be a problem. Once they understood why the recycle bin icon was on the desktop, they simply accepted it being there.

- ## Interaction Types

Another way of conceptualizing the design space is in terms of the interaction types that will underlie the user experience. Essentially, these are the ways a person interacts with a product or application. Originally, we identified four main types: instructing, conversing, manipulating, and exploring (Preece et al., 2018). A fifth type has since been proposed by Christopher Lueg et al. (2019) that we have added to ours, which they call *responding*. This refers to proactive systems that initiate a request in situations to which a user can respond, for example, when Netflix pauses a person's viewing to ask them whether they would like to continue watching. Deciding upon which of the interaction types to use, and why, can help designers formulate a conceptual model before committing to a particular interface in which to implement them, such as speech-based, gesture-based, touch-based, menu-based, and so on. Note that we are distinguishing here between interaction types (which we discuss in this section) and interface types. While cost and other product constraints will often dictate which interface style can be used for a given application, considering the interaction type that will best support a user experience can highlight the potential trade-offs, dilemmas, and pros and cons.

Here, we describe in more detail each of the five types of interaction. It should be noted that they are not meant to be mutually exclusive (for example, someone can interact with a system based on different kinds of activities); nor are they meant to be definitive. Also, the label used for each type refers to the user's action even though the system may be the active partner in initiating the interaction.

- *Instructing*: Where users issue instructions to a system. This can be done in a number of ways, including typing in commands, selecting options from menus in a windows environment or on a multitouch screen, speaking aloud commands, gesturing, pressing buttons, or using a combination of function keys.
- *Conversing*: Where users have a dialog with a system. Users can speak via an interface or type in questions to which the system replies via text or speech output.
- *Manipulating*: Where users interact with objects in a virtual or physical space by manipu- lating them (for instance, opening, holding, closing, and placing). Users can hone their familiar knowledge of how to interact with objects.
- *Exploring*: Where users move through a virtual environment or a physical space. Virtual environments include 3D worlds and augmented and virtual reality systems. They enable users to hone their familiar knowledge by physically moving around. Physical spaces that use sensor-based technologies include smart rooms and ambient environments, also ena- bling people to capitalize on familiarity.
- *Responding*: Where the system initiates the interaction and the user chooses whether to respond. For example, proactive mobile location-based technology can alert people to points of interest. They can choose to look at the information popping up on their phone or ignore it. An example is the Google Now Card, shown in Figure 3.5, which pops up a restaurant recommendation for the user to contemplate when they are walking nearby.

- ***Instructing***

This type of interaction describes how users carry out their tasks by telling the system what to do. Examples include giving instructions to a system to perform operations such as tell the time, print a file, and remind the user of an appointment. A diverse range of products has been designed based on this model, including home entertainment systems, consumer electronics, and computers. The way in which the user issues instructions can vary from pressing buttons to typing in strings of characters. Many activities are readily supported by giving instructions. In Windows and other *graphical user interfaces (GUIs)*, control keys or the selection of menu options via a mouse, touch pad, or touch screen are used. Typically, a wide range of functions are provided from which users have to select when they want to do something to the object on which they are working. For example, a user writing a report using a word processor will want to format the document, count the number of words typed, and check the spelling. The user instructs the system to do these operations by issuing appropriate commands. Typically, commands are carried out in a sequence, with the system responding appropriately (or not) as instructed.

One of the main benefits of designing an interaction based on issuing instructions is that the interaction is quick and efficient. It is particularly fitting where there is a frequent need to repeat actions performed on multiple objects. Examples include the repetitive actions of saving, deleting, and organizing files.

- **Conversing**

This form of interaction is based on the idea of a person having a conversation with a system, where the system acts as a dialogue partner. In particular, the system is designed to respond in a way that another human being might when having a conversation. It differs from the activity of instructing in so far as it encompasses a two-way communication process, with the system acting like a partner rather than a machine that obeys orders. It has been most commonly used for applications where the user needs to find out specific kinds of information or wants to discuss issues. Examples include advisory systems, help facilities, chatbots, and robots.

The kinds of conversation that are currently supported range from simple voice-recognition, menu-driven systems, to more complex natural language–based systems that involve the system parsing and responding to queries typed in or spoken by the user. Examples of the former include banking, ticket booking, and train-time inquiries, where the user talks to the system in single-word phrases and numbers, that is, yes, no, three, and so on, in response to prompts from the system. Examples of the latter include help systems, where the user types in a specific query, such as “How do I change the margin widths?” to which the system responds by giving various answers. Advances in AI during the last few years have resulted in a significant improvement in speech recognition to the extent that many companies now routinely employ speech-based and chatbot-based interaction for their customer queries.

A main benefit of developing a conceptual model that uses a conversational style of inter- action is that it allows people to interact with a system in a way that is familiar to them. For example, Apple’s speech system, Siri, lets you talk to it as if it were another person. You can ask it to do tasks for you, such as make a phone call, schedule a meeting, or send a message. You can also ask it indirect questions that it knows how to answer, such as “Do I need an umbrella today?” It will look up the weather for where you are and then answer with some- thing like, “I don’t believe it’s raining” while also providing a weather forecast (see Figure 3.1).

Figure 3.1 Siri’s response to the question “Do I need an umbrella today?”

A problem that can arise from using a conversational-based interaction type is that certain kinds of tasks are transformed into cumbersome and one-sided interactions. This is especially true for automated phone-based systems that use auditory menus to advance the interaction. Users have to listen to a voice providing several options, then make-a-selection, and repeat through further layers of menus before accomplishing their goal, for example, reaching a real human or paying a bill. Here is the beginning of a dialogue between a user who wants to find out about car insurance and an insurance company's phone reception system:

> *<user dials an insurance company>*
> ***"Welcome to St. Paul's Insurance Company. Press 1 if you are a new customer; 2 if you are an existing customer."***
> *<user presses 1>*
> ***"Thank you for calling St. Paul's Insurance Company. If you require house insurance, say 1; car insurance, say 2; travel insurance, say 3; health insurance, say 4; other, say 5."***
> *<user says 2>*
> ***"You have reached the car insurance division. If you require information about fully comprehensive insurance, say 1; third-party insurance, say 2. .... "***

- ***Manipulating***

This form of interaction involves manipulating objects, and it capitalizes on users' knowledge of how they do so in the physical world. For example, digital objects can be manipulated by moving, selecting, opening, and closing. Extensions to these actions include zooming in and out, stretching, and shrinking—actions that are not possible with objects in the real world. Human actions can be imitated through the use of physical controllers or gestures made in the air, such as the gesture control technology now used in some cars. Physical toys and robots have also been embedded with technology that enable them to act and react in ways depending on whether they are squeezed, touched, or moved. Tagged physical objects (such as balls, bricks, or blocks) that are manipulated in a physical world (for example, placed on a surface) can result in other physical and digital events occurring, such as a lever moving or a sound or animation being played.

A framework that has been highly influential (originating from the early days of HCI) in guiding the design of GUI applications is direct manipulation. It proposes that digital objects be designed at the interface level so that they can be interacted with in ways that are analogous to how physical objects in the physical world are manipulated.

In so doing, direct manipulation interfaces are assumed to enable users to feel that they are directly controlling the digital objects represented by the computer. The three core principles are as follows:

- Continuous representation of the objects and actions of interest
- Rapid reversible incremental actions with immediate feedback about the object of interest
- Physical actions and button pressing instead of issuing commands with complex syntax

According to these principles, an object on the screen remains visible while a user performs physical actions on it, and any actions performed on it are immediately visible. For example, a user can move a file by dragging an icon that represents it from one part of the desktop to another. The benefits of direct manipulation include the following:

- Helping beginners learn basic functionality rapidly
- Enabling experienced users to work rapidly on a wide range of tasks
- Allowing infrequent users to remember how to carry out operations over time
- Preventing the need for error messages, except rarely
- Showing users immediately how their actions are furthering their goals
- Reducing users' experiences of anxiety
- Helping users gain confidence and mastery and feel in control

Many apps have been developed based on some form of direct manipulation, including word processors, video games, learning tools, and image editing tools. However, while direct manipulation interfaces provide a versatile mode of interaction, they do have their drawbacks. In particular, not all tasks can be described by objects, and not all actions can be undertaken directly. Some tasks are also better achieved through issuing commands. For example, consider how you edit a report using a word processor. This would be tedious, and it would be easy to miss one or two. By contrast, this operation is relatively effortless and also likely to be more accurate when using a command-based interaction. This can be done by selecting a menu option or using a combination of command keys and then typing the changes required into the dialog box that pops up.

- *Exploring*

This mode of interaction involves users moving through virtual or physical environments. For example, users can explore aspects of a virtual 3D environment, such as the interior of a building. Physical environments can also be embedded with sensing technologies that, when they detect the presence of someone or certain body movements, respond by triggering certain digital or physical events. The basic idea is to enable people to explore and interact with an environment, be it physical or digital, by exploiting their knowledge of how they move and navigate through existing spaces.

- ***Responding***

  This mode of interaction involves the system taking the initiative to alert, describe, or show the user something that it "thinks" is of interest or relevance to the context the user is presently in. It can do this through detecting the location and/or presence of someone in a vicinity (for instance, a nearby coffee bar where friends are meeting) and notifying them about it on their phone or watch. Smartphones and wearable devices are becoming increasingly proactive in initiating user interaction in this way, rather than waiting for the user to ask, command, explore, or manipulate. An example is a fitness tracker that notifies the user of a milestone they have reached for a given activity, for example, having walked 10,000 steps in a day. The fitness tracker does this automatically without any requests made by the user; the user responds by looking at the notification on their screen or listening to an audio announcement that is made. Another example is when the system automatically provides some funny or useful information for the user, based on what it has learned from their repeated behaviors when carrying out particular actions in a given context. For example, after taking a photo of a friend's cute dog in the park, Google Lens will automatically pop up information that identifies the breed of the dog.

  For some people, this kind of system-initiated interaction—where additional information is provided which has not been requested—might get a bit tiresome or frustrating, especially if the system gets it wrong. The challenge is knowing when the user will find it useful and interesting and how much and what kind of contextual information to provide without overwhelming or annoying them. Also, it needs to know what to do when it gets it wrong. For example, if it thinks the dog is a teddy bear, will it apologize? Will the user be able to correct it and tell it what the photo actually is? Or will the system be given a second chance?

Lesson 2: Paradigms, Visions, Theories, Models, and Frameworks

(Sharp, 2019)

Other sources of conceptual inspiration and knowledge that are used to inform design and guide research are paradigms, visions, theories, models, and frameworks (Carroll, 2003). These vary in terms of their scale and specificity to a particular problem space. A *paradigm* refers to a general approach that has been adopted by a community of researchers and designers for carrying out their work in terms of shared assumptions, concepts, values, and practices. A *vision* is a future scenario that frames research and development in interaction design— often depicted in the form of a film or a narrative. A *theory* is a well-substantiated explanation of some aspect of a phenomenon; for example, the theory of information processing that explains how the mind, or some aspect of it, is assumed to work. A *model* is a simplification of some aspect of human-computer interaction intended to make it easier for designers to predict and evaluate alternative designs. A *framework* is a set of interrelated concepts and/or a set of specific questions that are intended to inform a particular domain area.

- ***Paradigms***

Following a particular paradigm means adopting a set of practices upon which a community has agreed. These include the following:

- The questions to be asked and how they should be framed
- The phenomena to be observed
- The way in which findings from studies are to be analyzed and interpreted (Kuhn, 1972)

In the 1980s, the prevailing paradigm in human-computer interaction was how to design user-centered applications for the desktop computer. Questions about what and how to design were framed in terms of specifying the requirements for a single user interacting with a screen-based interface. Task analytic and usability methods were developed based on an individual user's cognitive capabilities. Windows, Icons, Menus, and Pointers (WIMP) was used as a way of characterizing the core features of an interface for a single user. This was later superseded by the graphical user interface (GUI). Now many interfaces have touch screens that users tap, press and hold, pinch, swipe, slide, and stretch.

A big influence on the paradigm shift that took place in HCI in the 1990s was Mark Weiser's (1991) vision of ubiquitous technology. He proposed that computers would become part of the environment, embedded in a variety of everyday objects, devices, and displays. He envisioned a world of serenity, comfort, and awareness, where people were kept perpetually informed of what was happening around them, what was going to happen, and what had just happened. *Ubiquitous computing* devices would enter a person's center of attention when needed and move to the periphery of their attention when not, enabling the person to switch calmly and effortlessly between activities without having to figure out how to use a computer when performing their tasks. In essence, the technology would be unobtrusive and largely disappear into the background. People would be able to get on with their everyday and working lives, interacting with information and communicating and collaborating with others without being distracted or becoming frustrated with technology.

- ***Visions***

  Visions of the future, like Mark Weiser's vision of ubiquitous technology, provide a powerful driving force that can lead to a paradigm shift in terms of what research and development is carried out in companies and universities. A number of tech companies have produced videos about the future of technology and society, inviting audiences to imagine what life will be like in 10, 15, or 20 years' time. One of the earliest was Apple's 1987 Knowledge Navigator, which presented a scenario of a professor using a touchscreen tablet with a speech-based intelligent assistant reminding him of what he needed to do that day while answering the phone and helping him prepare his lectures. It was 25 years ahead of its time—set in 2011— the actual year that Apple launched its speech system, Siri. It was much viewed and discussed, inspiring widespread research into and development of future interfaces.

  A current vision that has become pervasive is AI. Both utopian and dystopian visions are being bandied about on how AI will make our lives easier on the one hand and how it will take our jobs away on the other. This time, it is not just computer scientists who are extolling the benefits or dangers of AI advances for society but also journalists, social commentators, policy-makers, and bloggers. AI is now replacing the user interface for an increasing number of applications where the user had to make choices, for example, smart- phones learning your music preferences and home heating systems deciding when to turn the heating on and off and what temperature you prefer. One objective is to reduce the stress of people having to make decisions; another is to improve upon what they would choose. For example, in the future instead of having to agonize over which clothes to buy, or vacation to select, a personal assistant will be able to choose on your behalf. Another example depicts what a driverless car will be like in a few years, where the focus is not so much on current concerns with safety and convenience but more on improving comfort and life quality in terms of the ultimate personalized passenger experience. More and more everyday tasks will be transformed through AI learning what choices are best in a given situation.

  While there are many benefits of letting machines make decisions for us, we may feel a loss of control. Moreover, we may not understand why the AI system chose to drive the car along a particular route or why our voice-assisted home robot keeps ordering too much milk. There are increasing expectations that AI researchers find ways of explaining the rationale behind the decisions that AI systems make on the user's behalf. This need is often referred to as *transparency* and *accountability.* It is an area that is of central concern to interaction design researchers, who have started conducting user studies on transparency and developing explanations that are meaningful and reassuring to the user.

Another challenge is to develop new kinds of interfaces and conceptual models that can support the synergy of humans and AI systems, which will amplify and extend what they can do currently. This could include novel ways of enhancing group collaboration, creative problem-solving, forward planning, policy-making, and other areas that can become intractable, complex, and messy, such as divorce settlements.

The different kinds of future visions provide concrete scenarios of how society can use the next generation of imagined technologies to make their lives more comfortable, safe, informative, and efficient. Furthermore, they also raise many questions concerning privacy, trust, and what we want as a society. They provide much food for thought for researchers, policy-makers, and developers, challenging them to consider both positive and negative implications.

Many new challenges, themes, and questions have been articulated through such visions including the following:

- How to enable people to access and interact with information in their work, social, and everyday lives using an assortment of technologies
- How to design user experiences for people using interfaces that are part of the environment but where there are no obvious controlling devices
- How and in what form to provide contextually relevant information to people at appropriate times and places to support them while on the move
- How to ensure that information that is passed around via interconnected displays, devices, and objects is secure and trustworthy

- ***Theories***

Over the past 30 years, numerous theories have been imported into human-computer interaction, providing a means of analyzing and predicting the performance of users carrying out tasks for specific types of computer interfaces and systems (Rogers, 2012). These have been primarily cognitive, social, affective, and organizational in origin. For example, cognitive theories about human memory were used in the 1980s to determine the best ways of representing operations, given people's memory limitations. One of the main benefits of applying such theories in interaction design is to help identify factors (cognitive, social, and affective) relevant to the design and evaluation of interactive products. Some of the most influential theories in HCI, including distributed cognition, will be covered in the next chapter.

- ***Models***

We discussed earlier why a conceptual model is important and how to generate one when designing a new product. The term *model* has also been used more generally in interaction design to describe, in a simplified way, some aspect of human behavior or human-computer interaction. Typically, it depicts how the core features and processes underlying a phenomenon are structured and related to one another. It is usually abstracted from a theory coming from a contributing discipline, like psychology. For example, Don Norman (1988) developed a number of models of user interaction based on theories of cognitive processing, arising out of cognitive science, which were intended to explain the way users interacted with interactive technologies. These include the seven stages of action model that describes how users move from their plans to executing physical actions that they need to perform to achieve them to evaluating the outcome of their actions with respect to their goals. More recent models developed in interaction design are user models, which predict what information users want in their interactions and models that characterize core components of the user experience, such as Marc Hassenzahl's (2010) model of experience design.

- ***Frameworks***

Numerous frameworks have been introduced in interaction design to help designers con- strain and scope the user experience for which they are designing. In contrast to a model, a framework offers advice to designers as to what to design or look for. This can come in a variety of forms, including steps, questions, concepts, challenges, principles, tactics, and dimensions. Frameworks, like models, have traditionally been based on theories of human behavior, but they are increasingly being developed from the experiences of actual design practice and the findings arising from user studies.

Many frameworks have been published in the HCI/interaction design literature, cover- ing different aspects of the user experience and a diversity of application areas. For example, there are frameworks for helping designers think about how to conceptualize learning, working, socializing, fun, emotion, and so on, and others that focus on how to design particular kinds of technologies to evoke certain responses, for example, persuasive technologies.

A classic example of a conceptual framework that has been highly influential in HCI is

Don Norman's (1988) explanation of the relationship between the design of a conceptual model and a user's understanding of it. The framework comprises three interacting components: the designer, the user, and the system. Behind each of these are the following:

Designer's Model *The model the designer has of how the system should work*

System Image *How the system actually works, which is portrayed to the user through the interface, manuals, help facilities, and so on*

User's Model *How the user understands how the system works*

The framework makes explicit the relationship between how a system should function, how it is presented to users, and how it is understood by them. In an ideal world, users should be able to carry out activities in the way intended by the designer by interacting with the system image that makes it obvious what to do. If the system image does not make the designer's model clear to the users, it is likely that they will end up with an incorrect understanding of the system, which in turn will increase the likelihood of their using the system ineffectively and making errors. This has been found to happen often in the real world. By drawing attention to this potential discrepancy, designers can be made aware of the importance of trying to bridge the gap more effectively.

To recap, paradigms, visions, theories, models, and frameworks are not mutually exclusive, but rather they overlap in their way of conceptualizing the problem and design space, varying in their level of rigor, abstraction, and purpose. *Paradigms* are overarching approaches that comprise a set of accepted practices and framing of questions and phenomena to observe; *visions* are scenarios of the future that set up challenges, inspirations, and questions for interaction design research and technology development; *theories* tend to be comprehensive, explaining human-computer interactions; *models* are an abstraction that simplify some aspect of human-computer interaction, providing a basis for designing and evaluating systems; and *frameworks* provide a set of core concepts, questions, or principles to consider when designing for a user experience or analyzing data from a user study.

Assessment Task

ACTIVITY 3.1 – Formulating plan into reality:

- **Create a System for the plan given in Module 2 on a digital Timepiece for Basketball Game –**
- **This can be created in any computer language you want.**
- **This is to be presented before the class.**

These days, timepieces (such as clocks, wristwatches, and so on) have a variety of functions. Not only do they tell the time and date, but they can speak to you, remind you when it's time to do something, and record your exercise habits among other things. The interface for these devices, however, shows the time in one of two basic ways: as a digital number such as 11:40 or through an analog display with two or three hands—one to represent the hour, one for the minutes, and one for the seconds.

This in-depth activity is to design an innovative timepiece. This could be in the form of a wristwatch, a mantelpiece clock, a sculpture for a garden or balcony, or any other kind of timepiece you prefer. The goal is to be inventive and exploratory by following these steps:

(e) *Think about the interactive product that you are designing: What do you want it to do? Find three to five potential users, and ask them what they would want. Write a list of requirements for the clock, together with some usability criteria and user experience criteria based on the definitions in Module1.*

(f) *Look around for similar devices and seek out other sources of inspiration that you might find helpful. Make a note of any findings that are interesting, useful, or insightful.*

(g) *Sketch some initial designs for the timepiece. Try to develop at least two distinct alternatives that meet your set of requirements.*

(h) *Evaluate the two designs by using your usability criteria and by role-playing an interaction with your sketches. Involve potential users in the evaluation, if possible. Does it do what you want? Is the time or other information being displayed always clear? Design is iterative, so you may want to return to earlier elements of the process before you choose one of your alternatives.*

Summary

This Module explained the importance of conceptualizing the problem and design spaces before trying to build anything. It stressed throughout the need to be explicit about the claims and assumptions behind design decisions that are suggested. It described an approach to formulating a conceptual model and explained the evolution of interface metaphors that have been designed as part of the conceptual model. Finally, it considered other ways of conceptualizing interaction in terms of interaction types, paradigms, visions, theories, models, and frameworks.

Key Points

- A fundamental aspect of interaction design is to develop a conceptual model.
- A conceptual model is a high-level description of a product in terms of what users can do with it and the concepts they need to understand how to interact with it.
- Conceptualizing the problem space in this way helps designers specify what it is they are doing, why, and how it will support users in the way intended.
- Decisions about conceptual design should be made before commencing physical design (such as choosing menus, icons, dialog boxes).
- Interface metaphors are commonly used as part of a conceptual model.
- Interaction types (for example, conversing or instructing) provide a way of thinking about how best to support the activities users will be doing when using a product or service.
- Paradigms, visions, theories, models, and frameworks provide different ways of framing and informing design and research.

References

A. Biele, Cezary, "*Human movements in human-computer interaction (HCI"*, Springer, 2022

B. Li, Yang and Hilliges, Otmar, *ed.by Artificial intelligence for human computer interaction: a modern approach*, Springer, 2021

C. Thakur, Nirmalya, and B.D., Parameshachari, ed.by" *Human-computer interaction and beyond : advances towards smart and interconnected environments (part I)",* 2021

D. Jennifer Preece, Helen Sharp, Yvonne Rogers, "*Interaction Design: Beyond Human-Computer Interaction", 5th Edition*, Wiley 2019

E. HCI (human-computer interaction) from https://www.cs.bham.ac.uk/~rxb/Teaching/HCI%20II/intro.html

F. HCI (human-computer interface) from www.Tutorialspoint.com

G. Vijay Kanade, *What Is HCI (Human-Computer Interaction)? Meaning, Importance, Examples, and Goals*: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-hci/

H. What is Human Computer Interaction? By Simplilearn Last updated on Nov 7, 2023: https://www.simplilearn.com/what-is-human-computer-interaction-article

- END OF MODULE FOR PRELIMINARY TERM PERIOD –

SUBMISSION OF ASSESSMENT TASKS AND EXAMINATION FOR PRELIM PERIOD WILL BE ANNOUNCED VIA iLearnU LMS or GC

MAKE SURE TO CHECK THE DATES AND DO NOT FORGET TO TAKE IT AS SCHEDULED.