Zusammenfassung People-Oriented Computing

Inhalt

[1. Paradigms of Interaction 2](#_Toc20744409)

[1.1 What is People-Oriented Computing? 2](#_Toc20744410)

[1.2 Interaction and Interaction Paradigms 2](#_Toc20744411)

[1.2.1 Communication as Interaction 2](#_Toc20744412)

[1.3 History of Human & Computing 2](#_Toc20744413)

[1.3.1 Time sharing (1950-1960) 2](#_Toc20744414)

[1.3.2 Video display units 3](#_Toc20744415)

[1.3.3 Programming Toolkits (1960s) 3](#_Toc20744416)

[1.3.4 Personal Computing (1970s-1980s) 4](#_Toc20744417)

[1.3.5 Windows and WIMP (1980s) 4](#_Toc20744418)

[1.3.6 Interface Metaphors 5](#_Toc20744419)

[1.3.7 Direct manipulation (1980s) 5](#_Toc20744420)

[1.3.8 Hypertext (1940s-1960s) 6](#_Toc20744421)

[1.3.9 The World Wide Web (1990s) 6](#_Toc20744422)

[1.3.10 Agent-Based Interfaces 7](#_Toc20744423)

[1.3.11 Multi-Modality 7](#_Toc20744424)

[1.3.12 Ubiquitous Computing (1990s-2000s) 7](#_Toc20744425)

[1.3.13 Sensor-based and Context-aware Interaction 7](#_Toc20744426)

[1.3.14 Augmented/ Virtual Reality 7](#_Toc20744427)

[2. Humans and interactive Systems 9](#_Toc20744428)

[2.1 An Interaction Framework 9](#_Toc20744429)

[2.1.1 The Interface 9](#_Toc20744430)

[2.2 Input-Output channels 10](#_Toc20744431)

[2.2.1 Human-centric Models of interaction 10](#_Toc20744432)

[2.2.2 Human Input and Output 10](#_Toc20744433)

[2.2.3 Vision as Input 10](#_Toc20744434)

[3. 11](#_Toc20744435)

# 1. Paradigms of Interaction

## 1.1 What is People-Oriented Computing?

People-Oriented Computing provides a sampling of the many areas in which computing affects people, on

* An individual level
* A group or organizational level
* A societal level

## 1.2 Interaction and Interaction Paradigms

### 1.2.1 Communication as Interaction

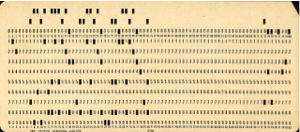
Communication includes means, abilities and channels of expressing intent, state or information and receiving input. It is necessary that there is enough shared understanding to interpret what is being communicated and to respond appropriately.

With humans, these abilities develop naturally, and certain structures have evolved over a long time. With computers these interactions are not natural. This means that it is necessary that computers can receive input so that it knows what it should do and must give the user feedback.

**Interaction Paradigms**: Successful approaches to interactive systems that have helped make it easier to use technology.

## 1.3 History of Human & Computing

### 1.3.1 Time sharing (1950-1960)



Previous approach **Batch sessions**: Individual programmers submitted jobs on punched cards or paper tape to an operator who then ran the individual jobs on a computer. Hardware advances in the 1940s and 1950s led to a massive increase in computing power through integrated chips. These hardware improvements necessitated parallel advancements in how to harness this power in use.

**Time sharing** was significant as a single computer could support multiple users at once and programming became an interactive activity. It gave rise to the “hacker” who could create increasingly complex programs. Time sharing shifted programming as a preplanned set of instructions for a computer to an exchange between programmer and computer.

### 1.3.2 Video display units

The first research in video display came in the 1950s for displaying images for military purposes.

In 1962 Ivan Sutherland landed a breakthrough with **Sketchpad**. It allowed data to be represented visually, abstracted, manipulated and changed. It enabled truly visual interaction and a more human way of interacting with data. It adapted the computer to the human’s way of thinking rather than the other way around.



### 1.3.3 Programming Toolkits (1960s)

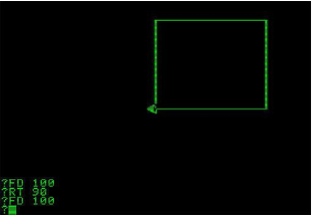
Up until this time Computers were considered something only experts and specialists could use. Douglas Engelbart’s vision was to enable humans to use computers to learn. He created programming tools that allow people to create complex programs more easily.

This allowed for bootstrapping to be possible.

**Bootstrapping**: Small programming components can be combined to create larger ones

### 1.3.4 Personal Computing (1970s-1980s)

The notion of computing for the masses without the need for substantial computing skills in order to benefit from computers became popular. Seymour Papert created a programming language for children called **LOGO** which demonstrated that powerful tools for hackers could be used by novices. It made use of a graphical “turtle” that could be commanded to draw shapes through simple English-based phrases (e.g. “turn left”). It illustrated that ease of use makes a system more powerful.



Alan Kay believed the future of computing was small, powerful machines dedicated to single users: **personal computers**. He believed in a shift away from mainframe computing and timesharing. With his team he created **Smalltalk**, a simple, but powerful, visually based programming environment especially for personal computing.

Kay also conceived of the **Dynabook** in the 1970s, a handheld personal computer for children.

### 1.3.5 Windows and WIMP (1980s)

The beginning of personal computing led to a focus on increased usability of single-user interaction with computers. Previous interfaces were command-line based. There was increased support for engaging in multiple tasks at once, with humans in control. Supporting multiple threads of interaction in conventional command line interfaces became complicated and difficult to manage. **Window-based** systems supported physical and logical separations of tasks.

The Xerox Star (1981) computer introduced the first commercial WIMP interface.

**WIMP**: Interface based on Windows, Icons, Menus and Pointers.

### 1.3.6 Interface Metaphors

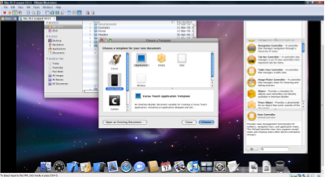
Metaphors helped people learn new concepts by putting them in terms of known concepts. Metaphors applied to computer interactions make unfamiliar concepts familiar and reduce the perception of complexity or difficulty.

Window and WIMP interfaces make extensive use of real-world metaphors:

* Windows
* Buttons
* Menus
* Palettes

Xerox Star and successors made use of an office desk metaphor:

* Desktop
* Folders
* Trash can
* …



Metaphors are naturally limited as it is not possible to completely map one set on concepts onto another. Because of this mismatches and false expectations can occur (i.e. Folders within folders, dragging media into trash eject).

### 1.3.7 Direct manipulation (1980s)

Traditional command line interfaces provided very limited feedback in interactions. Advancement in displays allowed for rapid audio and visual feedback with every interaction.

Rapid feedback facilitated an interaction technique called **direct manipulation**. It creates the illusion of operating directly on data and objects, rather than giving commands to a computer.

Features of direct manipulation are:

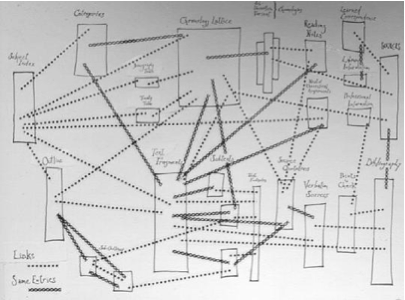
* Visibility of all objects of interest
* Incremental action at the interface with rapid feedback on all actions
* Reversibility of all actions so that users can explore without severe penalties
* Syntactic correctness of all actions so that possible action is a legal operation
* Replacement of complex command languages with actions

The first commercial success of a direct manipulation interface was the Apple Macintosh computer (1984).

It made files and directory structure visible to the user. Operations such as moving files between directories were mirrored in an action on a visible document that could be picked up and dragged. It was impossible to formulate a syntactically incorrect command and it gave continual visual feedback while the operation was being carried out.

### 1.3.8 Hypertext (1940s-1960s)

Vannevar Bush published “As We May Think”, which proposed an innovative future for information storage and retrieval to improve human capacity of knowledge access. He proposed a “memex” apparatus, a desk with the ability to produce and store massive amounts of photographic copies of documents.



### 1.3.9 The World Wide Web (1990s)

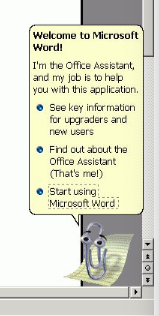
The **WWW** was a revolutionary paradigm which lowered the barrier for access to the internet, lowered the barriers to creating and publishing information and increased the purchases of computers and their use. It led to the rapid growth and increased value of internet content including leisure and commerce.

The beginning of computer networks can be traced back to the 1960s where computers started to communicate with each other. This enabled **CSCW** (Computer-Supported Cooperative Work).

CSCW enabled a transition from individual use to group and organizational use of computers. Email is an early example of a CSCW system. Subsequent computer-based communication tools and social media platforms arose from this paradigm.

### 1.3.10 Agent-Based Interfaces

Agent-Based Interfaces started a departure from direct manipulation. It created the illusion of someone working on your behalf to perform the tasks.

Agents can be simple actors that follow commands or intelligent and proactive. An early example was Eager, a cat icon that would observe HyperCard programmers and suggest next actions.

A lot of recent innovations like Siri have made great strides in agent interaction.

### 1.3.11 Multi-Modality

Multimodality allows people to engage in multiple tasks at once and to give input in different ways. Some of these communication “channels” can be:

* Visual channel (mostly output, but also gestures)
* Audio channel (Sound and voice input)
* Haptic (touch) channel (touchscreen)
* Keyboard & mouse

A normal computer with Keyboard, Mouse or pointing device, visual and audio output are inherently multimodal.

### 1.3.12 Ubiquitous Computing (1990s-2000s)

### 1.3.13 Sensor-based and Context-aware Interaction

### 1.3.14 Augmented/ Virtual Reality

**Augmented reality** combines physical world and digital content. It requires knowledge of the environment like QR codes or IR sensors.

**Virtual reality** replaces the physical world with a digital world. It’s a full immersion with 3D interaction. It uses gesture recognition, eye gaze and/or full body sensing.

Both have many applications: entertainment, medicine or training.

# 2. Humans and interactive Systems

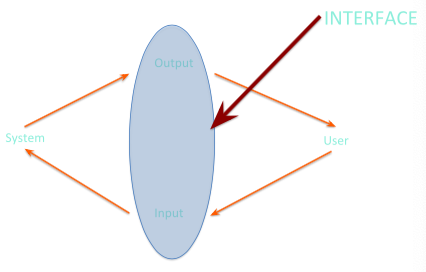
The purpose of an interactive system is traditionally to aid a user in accomplishing a goal within an application domain.

## 2.1 An Interaction Framework

An interaction Framework has four major components:

* The systems
* The user
* The input
* The output

One of the main issues in the interaction framework is that the system and the user have different languages.



The translation between these two languages happens in the interface. An interactive system’s interface can be thought of as the combination of the input provided by the user and the output provided by the system.

### 2.1.1 The Interface

An interface has the following requirements:

* The user needs to be able to articulate their goals and tasks in the input language specified by the interface
* The input needs to be translated into stimuli for the systems upon which the system can perform
* The new state of the system must be presented as output as specified by the interface
* The output must be observed and interpreted by the user

## 2.2 Input-Output channels

### 2.2.1 Human-centric Models of interaction

The first model is the **Model Human Processor**. It considers the human as an input-output machine that processes physical stimuli and produces a physical response.

The second model of interaction is the **7-Stage Model of interaction**. It conceives of interaction in two human-centric phases of execution and evaluation in the world.

### 2.2.2 Human Input and Output

Input to humans occurs primarily through the five senses. Human output occurs through motor control of effectors, i.e. physical action.

In an interactive systems human input is primarily through vision, hearing and touch. Other input is still very rare. Output is traditionally finger and hand movement based (keyboards and pointing devices, touch screens).

In conjunction with advances in sensor and recognition technologies there’s an increase in other output being used (e.g. head movement, voice commands, facial gestures, hand and body gestures).

### 2.2.3 Vision as Input

# 3. Information Visualization Basics

## 3.1 What is Information Visualization and why is it important?

In 2012 there were 2,8 zettabytes of data produced. It is estimated to be 40 zettabytes in 2020. 10^21= 1 Zettabyte

**Information Visualization**: The use of computer-supported, interactive visual representations of abstract data to amplify cognition.

### 3.1.1 Information vs. Scientific Visualization

**Information Visualization**: Abstract data with no physical correspondence. There’s free mapping of data to 2D or 3D space.

**Scientific Visualization**: Scientific data corresponding to physical phenomena with fixed positions in space for visualizations.

Some of the key challenges in Information Visualization:

* Creating meaningful and useful mappings of abstract data onto 2D or 3D space
* Representing extremely large sets of data in a finite amount of space
* Representing diverse types and forms of data within visualitazion

## 3.2 Classic Visualization Examples: Successes and failures

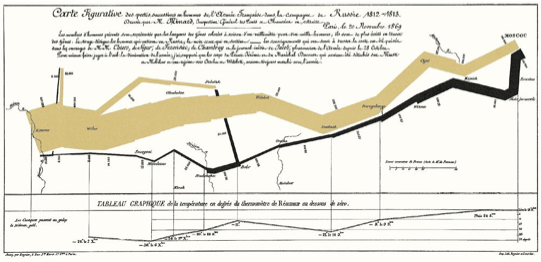
### 3.2.1 Broad Street Cholera Outbreak

In 1845 there was a severe outbreak of cholera in London. The cause was believed to be bad air. Dr. John Snow was skeptical and made a map of the cases. He concluded that the outbreak was caused by a well, as almost all cases got water from the same source. The combination of the data he collected and his visualizations explained the cases and exceptions.

### 3.2.2 Nightingale’s Rose

Florence Nightingale created a visualization to convince the British military that sanitation was a greater hazard to soldiers than battlefield combat.

### 3.2.3 Napoleon’s March



### 3.2.4 Challenger Explosion

In 1986 a NASA space shuttle exploded after damage to an O-Ring of one of the rockets. The problem was that the data showed to the NASA executives was not visualized enough to prove the engineer’s point.

## 3.3 Visual thinking and perception