Zusammenfassung People-Oriented Computing

Inhalt

[1. Paradigms of Interaction 3](#_Toc24545727)

[1.1 What is People-Oriented Computing? 3](#_Toc24545728)

[1.2 Interaction and Interaction Paradigms 3](#_Toc24545729)

[1.2.1 Communication as Interaction 3](#_Toc24545730)

[1.3 History of Human & Computing 3](#_Toc24545731)

[1.3.1 Time sharing (1950-1960) 3](#_Toc24545732)

[1.3.2 Video display units 4](#_Toc24545733)

[1.3.3 Programming Toolkits (1960s) 4](#_Toc24545734)

[1.3.4 Personal Computing (1970s-1980s) 5](#_Toc24545735)

[1.3.5 Windows and WIMP (1980s) 5](#_Toc24545736)

[1.3.6 Interface Metaphors 6](#_Toc24545737)

[1.3.7 Direct manipulation (1980s) 6](#_Toc24545738)

[1.3.8 Hypertext (1940s-1960s) 7](#_Toc24545739)

[1.3.9 The World Wide Web (1990s) 7](#_Toc24545740)

[1.3.10 Agent-Based Interfaces 8](#_Toc24545741)

[1.3.11 Multi-Modality 8](#_Toc24545742)

[1.3.12 Ubiquitous Computing (1990s-2000s) 8](#_Toc24545743)

[1.3.13 Sensor-based and Context-aware Interaction 8](#_Toc24545744)

[1.3.14 Augmented/ Virtual Reality 8](#_Toc24545745)

[2. Humans and interactive Systems 10](#_Toc24545746)

[2.1 An Interaction Framework 10](#_Toc24545747)

[2.1.1 The Interface 10](#_Toc24545748)

[2.2 Input-Output channels 11](#_Toc24545749)

[2.2.1 Human-centric Models of interaction 11](#_Toc24545750)

[2.2.2 Human Input and Output 11](#_Toc24545751)

[2.2.3 Vision as Input 11](#_Toc24545752)

[3. Information Visualization Basics 12](#_Toc24545753)

[3.1 What is Information Visualization and why is it important? 12](#_Toc24545754)

[3.1.1 Information vs. Scientific Visualization 12](#_Toc24545755)

[3.2 Classic Visualization Examples: Successes and failures 12](#_Toc24545756)

[3.2.1 Broad Street Cholera Outbreak 12](#_Toc24545757)

[3.2.2 Nightingale’s Rose 12](#_Toc24545758)

[3.2.3 Napoleon’s March 13](#_Toc24545759)

[3.2.4 Challenger Explosion 13](#_Toc24545760)

[3.3 Visual thinking and perception 13](#_Toc24545761)

[3.3.1 Gestalt Laws 14](#_Toc24545762)

[3.3.2 Tufte’s Principles 15](#_Toc24545763)

[4. Principles for Design 16](#_Toc24545764)

[4.1 Design in interaction 16](#_Toc24545765)

[4.2 Design Principles and concepts 16](#_Toc24545766)

[4.2.1 Affordances 16](#_Toc24545767)

[4.2.2 Signifiers 16](#_Toc24545768)

[4.2.3 Constraints 17](#_Toc24545769)

[5. Modeling Interaction and Cognition 18](#_Toc24545770)

[5.1 Fitts’s law 18](#_Toc24545771)

[5.2 Keyboard-level model (KLM) 18](#_Toc24545772)

[5.3 GOMS Model 19](#_Toc24545773)

[5.4 Seven-stage model of interaction 19](#_Toc24545774)

[5.4.1 Bridging the gulfs 20](#_Toc24545775)

[5.5 Model Human Processor 20](#_Toc24545776)

# 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

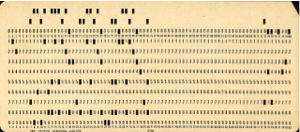
Communication includes means, abilities and channels of expressing intent, state or information and receiving input. It is necessary that there is enough shared understanding to interpret what is being communicated and to respond appropriately.

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)

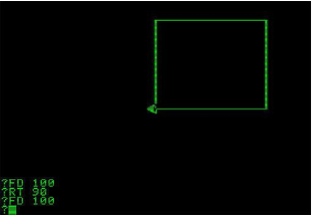
Up until this time Computers were considered something only experts and specialists could use. Douglas Engelbart’s vision was to enable humans to use computers to learn. He created programming tools that allow people to create complex programs more easily.

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

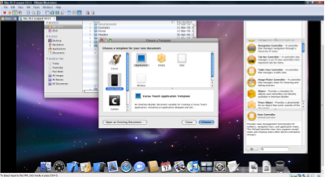
Metaphors helped people learn new concepts by putting them in terms of known concepts. Metaphors applied to computer interactions make unfamiliar concepts familiar and reduce the perception of complexity or difficulty.

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:

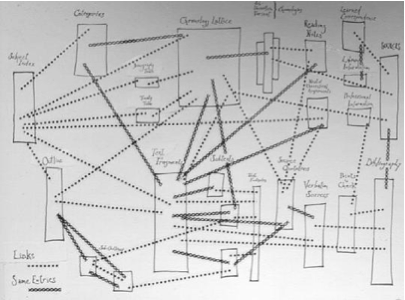
* Visibility of all objects of interest
* Incremental action at the interface with rapid feedback on all actions
* Reversibility of all actions so that users can explore without severe penalties
* Syntactic correctness of all actions so that possible action is a legal operation
* Replacement of complex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

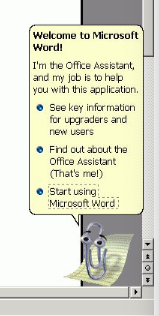
The **WWW** was a revolutionary paradigm which lowered the barrier for access to the internet, lowered the barriers to creating and publishing information and increased the purchases of computers and their use. It led to the rapid growth and increased value of internet content including leisure and commerce.

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

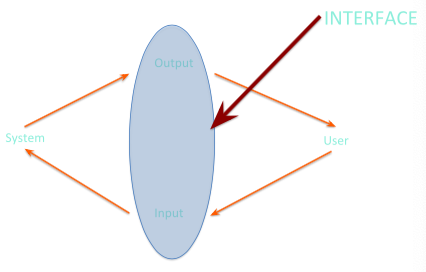
The purpose of an interactive system is traditionally to aid a user in accomplishing a goal within an application domain.

## 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

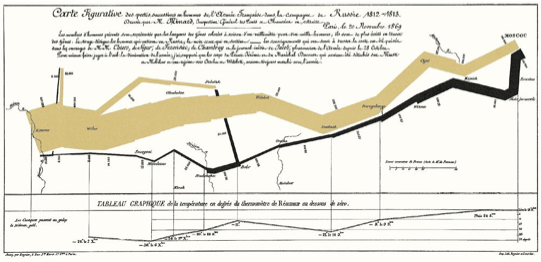
### 3.2.1 Broad Street Cholera Outbreak

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

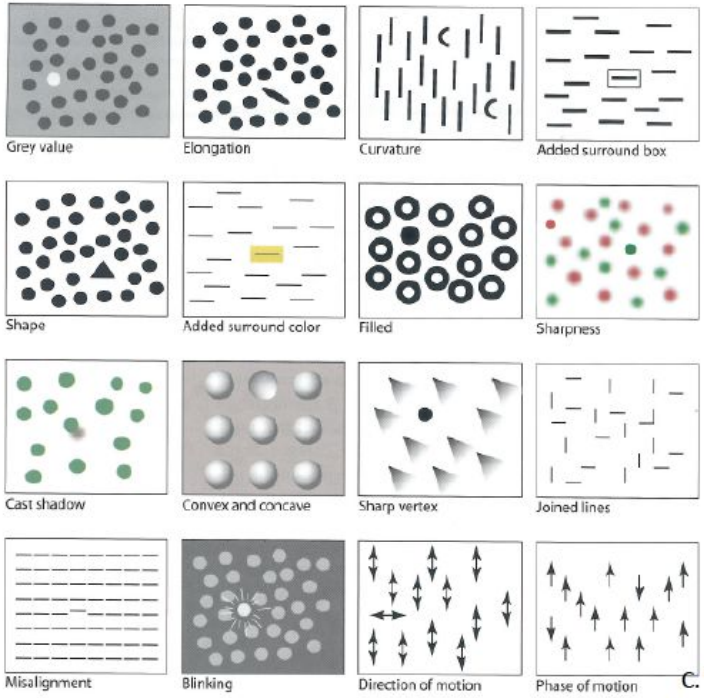


### 3.2.4 Challenger Explosion

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

An object can be made easier to find by making it different to it’s surroundings in:



### 3.3.1 Gestalt Laws

How to organize information in visual groups

1. **Proximity**: items located near each other are grouped together in the mind of the user and are considered to belong together.
2. **Similarity**: items with similar color, shape, size or orientation are grouped together and are considered to belong together. If the items are similar in more than one attribute the grouping process is done faster.
3. **Common** **fate**: If items have a similar motion they are considered to belong together. Motion is a very strong cue and is usually stronger than similarity!
   1. **Synchrony**: generalized common fate. If items move in synchrony to each other they are grouped together, even if their directions are different.
4. **Continuity**: items with a continuous contour are perceived as a group.



1. **Closure**: items with a closed contour are grouped together.



* 1. **Contour closure**: Tendency to fill in the gaps in continuous contours.



* 1. **Enclosure**: Tendency to perceive closed shape as foreground.



1. **Common** **region**: Items inside the same region are grouped together.
2. **Element** **connectedness**: Connected elements are grouped together. Element connectedness is stronger than similarity, but weaker than common region.

### 3.3.2 Tufte’s Principles

#### Graphical excellence

An excellent graphic consists of complex ideas communicated with clarity, precision and efficiency. It gives the viewer the greatest number of ideas in the shortest time in the smallest space. It’s nearly always multivariate and tells the truth about the data.

#### Graphical integrity

The representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the numerical quantities represented.

The graphic is clear, detailed, and thorough labeling should be used to prevent distortion and ambiguity. It shows data variation, not design variation and shouldn’t quote data out of context.

#### Data graphics

* Words are spelled out, elaborate encoding avoided
* Words run from left to right
* Little messages to help explain data
* Labels are placed on the graphics, no legend required
* Graphics attract viewer, provoke curiosity
* Colors, if used, are chosen so that the color-deficient and color-blind can make sense of the graphic
* Type is clear, precise, modest
* Type is upper-and-lower case, with serifs

#### Errors that can be made

* Selecting the wrong data
* Selecting the wrong data structure
* Filtering out important data
* Failed understanding of the types of things that need to be shown
* Selecting the wrong representation
* Choosing the wrong presentation format

# 4. Principles for Design

## 4.1 Design in interaction

**Usability**: Is a system or object easy to use

**Usefulness**: Does a system or object serve a function that is valuable to me?

#### Three fields of design

1. **Industrial design**: Focuses on function and appearance of products and systems, often physical
2. **Interaction design**: Focuses on how people interact with technology, particularly understanding how to use it
3. **Experience design**: Focuses on quality and enjoyment of experience, particularly of services, environments, and events

**Human-Centered Design**: Not a field but an approach. In its center are human capabilities and behavior and designs must accommodate these. This approach focuses on communication between the person and the system.

## 4.2 Design Principles and concepts

**Conceptual Model**: usually simplified explanation of how something works. It’s not necessarily an accurate reflection of the actual workings of the system. It’s not necessarily complete and varies from person to person.

**Mental Model**: Conceptual models are the conceptual models of a system that people have in their minds. Mental models are often developed from experience.

**User’s Conceptual Model**: varies from person to person and in completeness and correctness. It’s based on experience and influenced by system image.

**System Image**: The total information that is available to the user. This involves Appearance of the system, instructions, articles about products, … The system image can be incomplete or contradictory.

Good design facilitates communication of the designer’s conceptual model via the system image to the user and enables user to develop a good conceptual model.

### 4.2.1 Affordances

**Affordance**: The relationship between an object and a person (or other entity). It determines how the object could possibly be used and depends on both the properties of the object and the capabilities of the person. Affordances need to be perceivable to be effective

### 4.2.2 Signifiers

**Signifiers**: any perceivable indicator that communicates an appropriate behavior to a person. They don’t communicate everything that can be done, only what should be done. They’re important in design for fostering discoverability, but can be misleading, poor, or superfluous. Perceived affordances often serve as signifiers. Simple objects or systems should be self-explanatory.

### 4.2.3 Constraints

**Constraints**: Clues that help you discover what to do by putting limits on the set of possible actions. Constraints especially useful for helping people determine proper course of action in new situations.

Constraints can be physical, cultural, semantic and/or logical.

# 5. Modeling Interaction and Cognition

## 5.1 Fitts’s law

Essentially a formulation of the idea that movement time is proportional to distance and target size.

Movement Time (MT) is proportional to the Index of Difficulty (ID) of a selection task. I.E., the harder the selection task is, the longer it will take. The movement time for a well-rehearsed selection task:

* Increases as the distance A to the target increases
* Decreases as the size of the target W increases

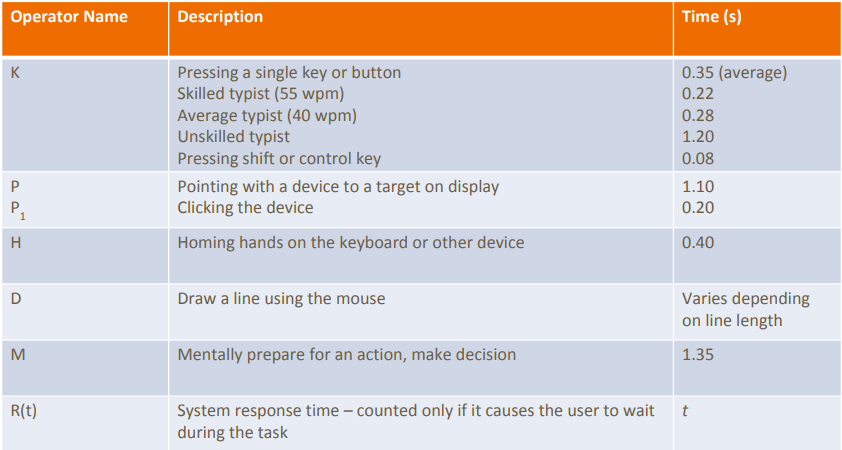
Difficulty of a selection task can be calculated as:



## 5.2 Keyboard-level model (KLM)

Not all human tasks involve no cognition. Even making simple decisions about how to accomplish a task or what the next step should be involves some mental processing. KLM is a simple model that begins to incorporate mental processes.

* **K** – striking keys
* **B** – pressing a mouse button
* **P** – pointing (dragging a pointer to a target)
* **H** – homing (switching the hand between the mouse and keyboard)
* **D** – drawing lines using the mouse
* **M** – mentally preparing for a physical action
* **R** – system response time



## 5.3 GOMS Model

Stands for **G**oals, **O**perators, **M**ethods, and **S**election rules. IT attempts to model the knowledge and cognitive processes involved when users interact with system

* **Goal**: a particular state the user wants to achieve
* **Operators**: the cognitive processes and physical actions that need to be performed in order to attain goals
* **Methods**: learned procedures for accomplishing goals. Consist of the exact sequence of steps required
* **Selection rules**: determine which method to select when there is more than one available for a given stage of a task

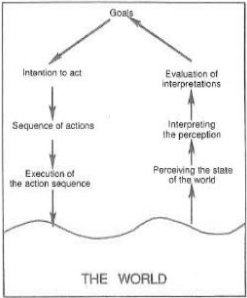
## 5.4 Seven-stage model of interaction

**Execution**

* **Forming the Intention**: What does the person want to do in this step?
* **Specifying an Action**: What are the exact steps the person decides to take to address the intention?
* **Executing the Action**: Actually doing the steps that have been chosen, thus acting upon the world

**Evaluation**

* **Perceiving the state of the world**: The person must physically perceive the current state of the world, whether changed or unchanged (i.e., see, hear, feel, etc.)
* **Interpreting the state of the world**: The person must figure out what the perceived changes mean, i.e., what just happened?
* **Evaluating the outcome**: The person must come to a conclusion about whether the original goal has been addressed



### 5.4.1 Bridging the gulfs

The **Gulf of Execution** is small when:

* The actions provided by the system match the intentions of the user
* The actions can be executed without extra effort

The **Gulf of Evaluation** is small when:

* The system provides information about its state that can be easily accessed and interpreted
* The system’s state matches the way the user thinks about the system

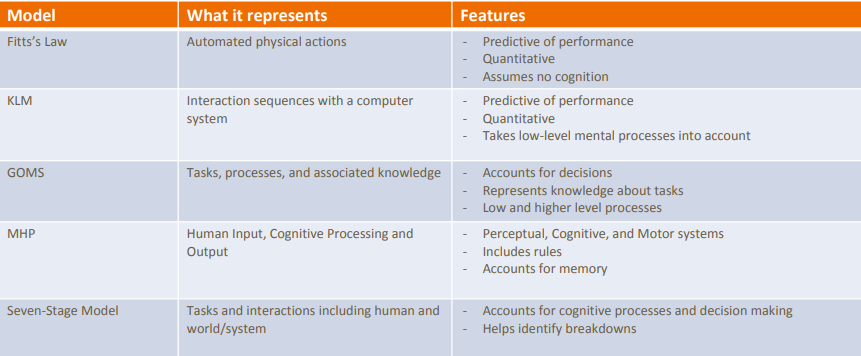
## 5.5 Model Human Processor

Three systems: perceptual, cognitive, and motor. Each system has processor and memory and each system has principles of operation.



This model is not meant to explain how the human brain works, but intended to help understand, predict, and calculate human performance in interaction.

## 5.6 Quick Synopsis of Models



# 6. Computer-Supported Cooperative Work

## 6.1 CSWC Background and history

Focus thus far has been on interaction between individual humans and computer systems. Much of current technology goes beyond a closed interaction between an individual and a technology – how does technology connect humans?

**Computer-Supported Cooperative Work (CSWC)**: A field of research concerned with understanding social interaction and technologies supporting social interaction in groups, organizations, and communities.

“In its most general form, CSCW examines the possibilities and effects of technological support for humans involved in collaborative group communication and work processes” -*Bowers and Benford*

### 6.1.1 Focus of CSWC

* Technologies to support group work and processes
  + How can computing technology be applied to support and enhance cooperative work and organizations?
  + How should such technologies be designed to support the diverse roles and needs in groups and organizations?
* Understanding socio-technical aspects of work in groups and organizations
  + What is the nature of cooperative work carried out in groups as opposed to by individuals?
  + What are the various roles and relationships in organizations, and how do these affect and influence the work?
  + What are the challenges to cooperative work?
* Present day computing technologies
  + Make ample use of social features
  + Are seldom purely for individual use
    - Spectrum of group/individual use, e.g. MS-Word change tracking vs. synchronous Google docs editing
  + Often blur the distinction between work and leisure

CSWC differs from Social Computing in term of the amount of people they are made for. CSWC is made for a smaller amount of people (1k max) while social computing is created for entire societies.

**Groupware**: Refers to the technical systems that arise from CSCW research and development.

“Computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment.” - *Ellis, 1991*

### 6.1.2 Early CSWC Efforts

* Placed heavy emphasis on facilitating synchronous, remote communication
* “Holy Grail” goal of CSCW was to simulate “being there” or face-to-face communication
* Focused extensively on improving the experience of videoconferencing to create the illusion of colocation

At the same time as research focused on videoconferencing, other tools were proving to be successful for supporting group work: Wiki, Instant messaging

As simulating face-to-face interaction turned out to be very difficult, the focus moved away and towards other improvements of face-to-face communication:

* Asynchronous communication
* Anonymous communication
* Automatic archiving of communication

## 6.2 CSCW Technologies

### 6.2.1 Classes of CSWC technologies

* **Computer-mediated communication**: systems that support the direct communication between participants
* **Meeting and decision support systems**: systems that capture common understanding
* **Shared applications and artifacts**: systems that support participant interaction with shared work objects – the artifacts of work
* **Awareness applications**: systems that promote awareness of individual and group status

### 6.2.2 Computer-Mediated Communication

CSCW systems that support direct communication. Usually support remote communication and Can be asynchronous (often text based) or synchronous

#### Asynchronous CMC

The oldest forms of CMC. Primary examples: Email, Bulletin boards/online forums

#### Synchronous CMC

Can support synchronous communication (though many can also be used asynchronously). Examples: Instant messaging, chat, video conferencing, virtual reality, virtual worlds

#### Conferencing Tools

Support for real time meetings via high bandwidth connections, sometimes allow for sharing of artifacts, e.g., shared documents, agendas, slides, electronic whiteboards, still difficulties with quality, interaction, turn-taking

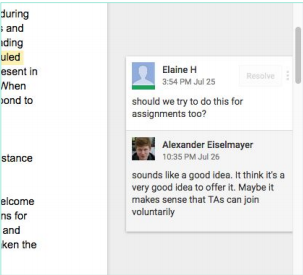
### 6.2.3 Meeting and Decision Support Systems

Participants in shared work must establish a common understanding about tasks to be performed. Often work involves generating and recording ideas. Systems can support idea generation and idea and decision recording.

#### Meeting Support

* May impose structure on the process of the meeting
* May support brainstorming and voting processes
* May provide support for group decision making, e.g., prioritization of projects, tracking of arguments
* May support shared editing of documents or artifacts
* May provide shared surfaces, e.g. electronic whiteboards, shared tabletops

#### Argumentation tools

Record arguments used to arrive at a decision and often support asynchronous co-located teams

#### Meeting Rooms

Support face-to-face groups (synchronous, co-located) in brainstorming and management meetings

#### Shared surfaces

Can be used for synchronous remote or co-located collaboration

