SWE1018
Human Computer Interaction

Text Book

Gerard Jounghyun Kim, Human Computer Interaction – Fundamentals and Practice, – CRC press, 2015.

Reference Books

- Julie A. Jacko, The Human–Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, 3rd Edition, CRC Press (Taylor & Francis Group) 2012.
- Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Designing the User Interface: Strategies for Effective Human Computer Interaction, 5th Edition, Pearson, 2009.
- Alan Dix, Janet E. Finlay, Gregory D. Abowd, Russell Beale, Human - Computer Interaction, 3rd Edition, Pearson, 2003.

Module 1

Human Computer Interaction and its frameworks, Principles of HCI, Types of Interaction styles

- **Human Computer Interaction**
  - USA: Computer-Human Interaction (CHI)
  - USA: Man-Machine Interface (MMI)
- **Human Computer Interface**
- is the study of the interaction between people, computers and tasks involves the development and application of principals, guidelines and methods to support the design and evaluation of interactive systems

- **HCI** (**human-computer interaction**) is the study of how people interact with computers and to what extent computers are or are not developed for successful interaction with human beings. A significant number of major corporations and academic institutions now study **HCI**.

- Human-Computer Interaction (HCI) is a multidisciplinary field focusing on the design, evaluation, and implementation of interactive computing systems for human use and the study of major phenomena surrounding them. As technology continues to evolve, so does the landscape of HCI, shaping how we interact with machines in increasingly sophisticated ways.

The Journey of HCI: From Command Lines to Natural Interaction

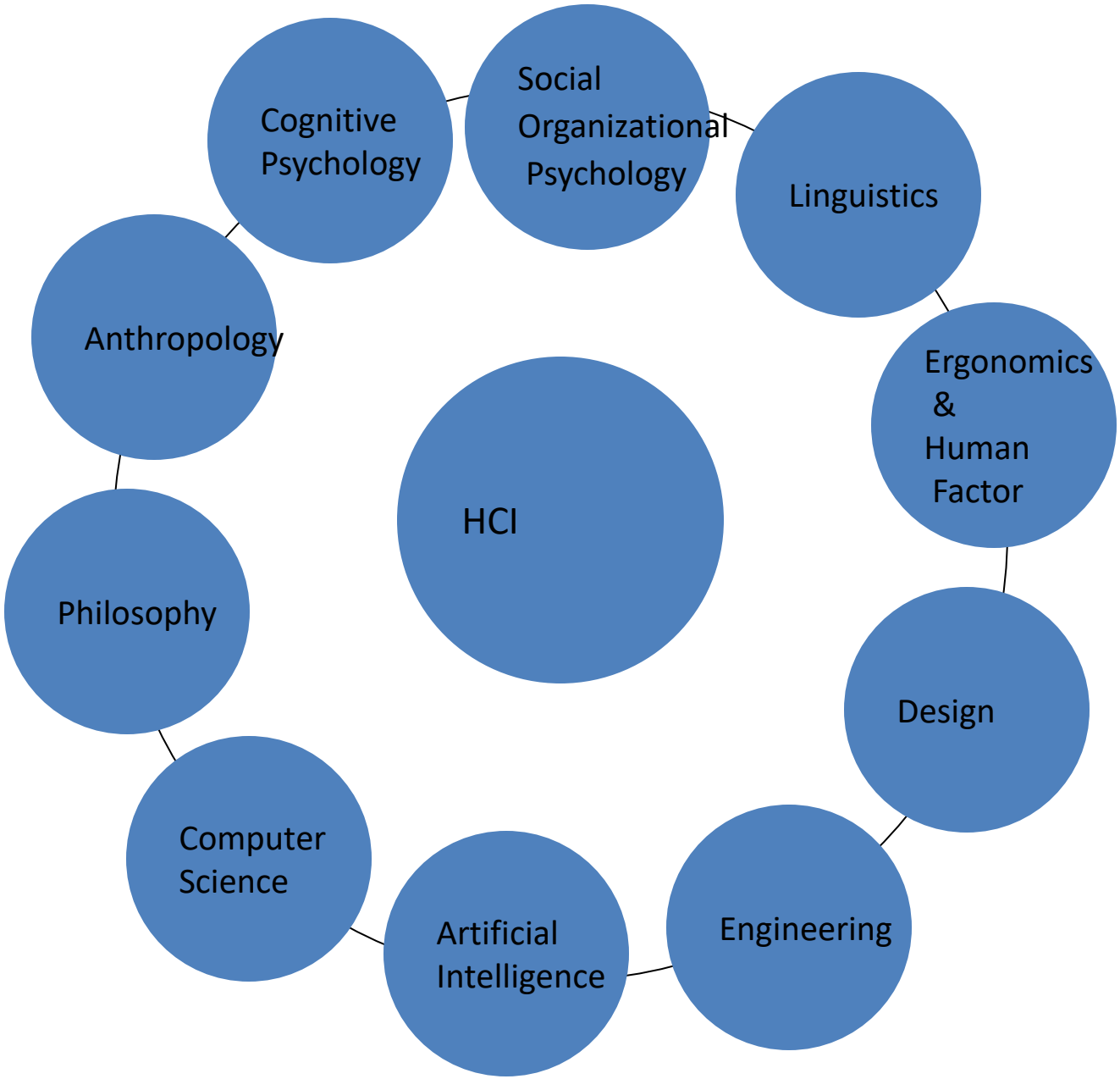
- The field of HCI has come a long way since its inception. Early computers required users to input commands through text-based interfaces. This mode of interaction was not user-friendly, limiting access to technology to those with specialized knowledge. The advent of graphical user interfaces (GUIs) in the 1980s revolutionized this dynamic, making computers more accessible to the general public. Icons, windows, and menus replaced text commands, paving the way for the mass adoption of personal computers.

- Fast forward to the 21st century, and we see a paradigm shift with the rise of touchscreens, voice assistants, and gesture-based controls. The introduction of smartphones and tablets brought about an era of intuitive touch interfaces, allowing users to interact with devices through simple taps and swipes. Voice assistants like Siri, Alexa, and Google Assistant further bridged the gap between humans and machines, enabling natural language interactions.

Key Innovations in HCI

- Voice Interaction: Voice assistants have become an integral part of our daily lives, offering hands-free control over devices and access to information. Advancements in natural language processing (NLP) and machine learning have made these interactions more seamless and context-aware.
- Augmented Reality (AR) and Virtual Reality (VR): AR and VR technologies are transforming the way we perceive and interact with digital content. From immersive gaming experiences to practical applications in training and remote collaboration, AR and VR are expanding the boundaries of interaction.

- Wearable Technology: Wearables like smartwatches and fitness trackers have integrated into our lives, providing real-time data and insights about our health and activities. These devices exemplify how HCI can enhance user experience through convenience and personalization.
- Brain-Computer Interfaces (BCIs): BCIs represent the frontier of HCI, allowing direct communication between the brain and computers. This technology holds immense potential for applications in accessibility, gaming, and beyond, offering new ways for individuals to interact with machines


HCI
Cognitive Psychology
Social Organizational Psychology
Linguistics
Ergonomics & Human Factor
Design
Engineering
Artificial Intelligence
Computer Science
Philosophy
Anthropology

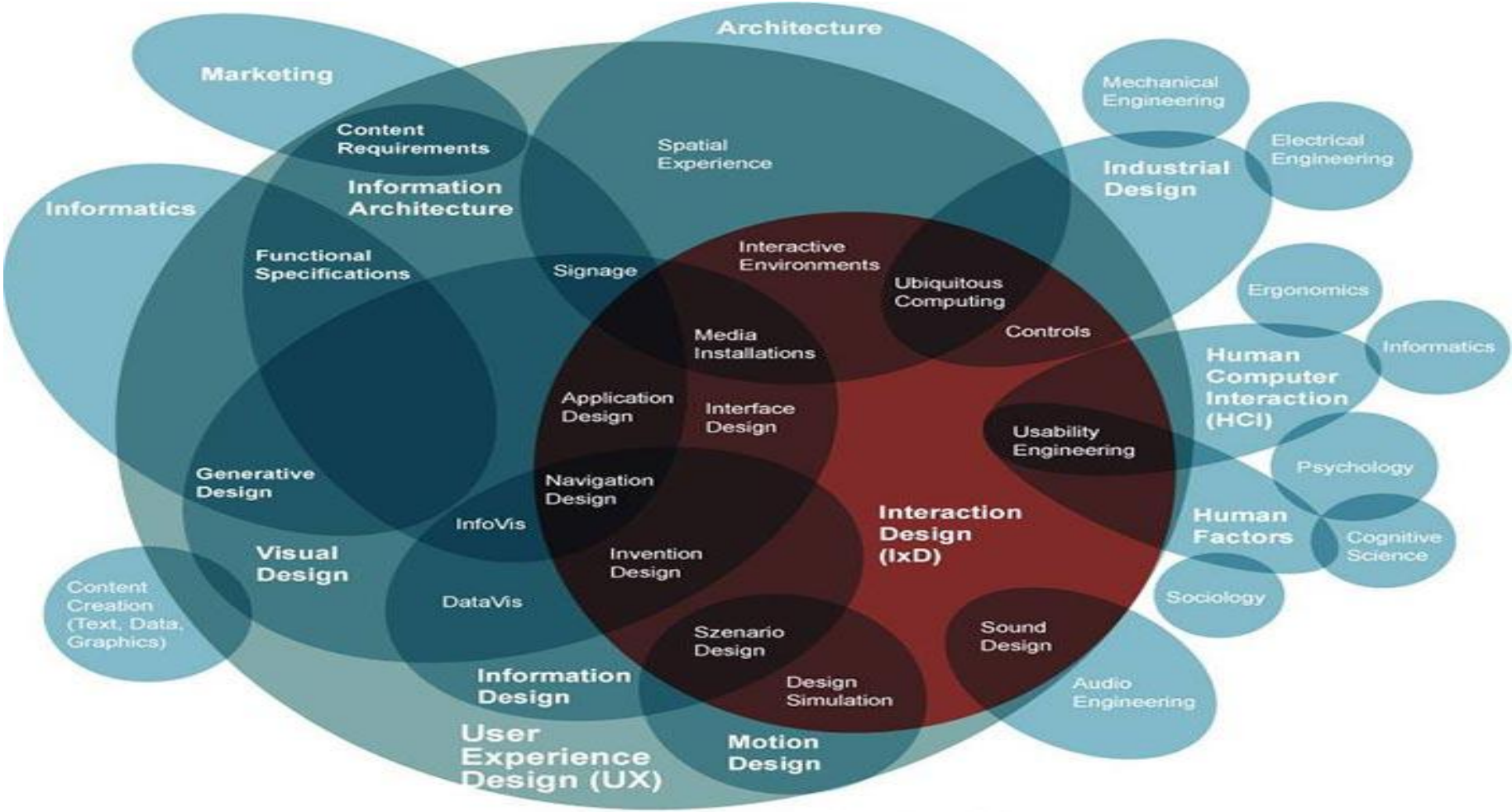
Disciplines

- Psychology
  - understanding the user
  - modelling the user
- Sociology
  - groupware
- Art
  - aesthetic appeal
- Design
  - user interface layout

- Engineering/Computer Science
  - faster machines
  - faster systems
  - means of building better interfaces
- Linguistics
  - language for commands
- Philosophy
  - creating consistency

- Physiology
  - physical capabilities
- Anthropology
  - user body shape
- Ergonomics
  - equipment design
- AI
  - help facilities
  - modelling the user


Computer Science
Technical Communication
Ergonomics
HCI
Content Strategy
Psychology
Human Factors
Cognitive Science
Branding
Experience Design
User Experience/ Usability
User Research
Anthropology
Analytics
Web Design
Market Research
Performance Support
Graphic Arts
Information Architecture
Interaction Design
Quality
Training
Library Science
Software Testing
Software Design



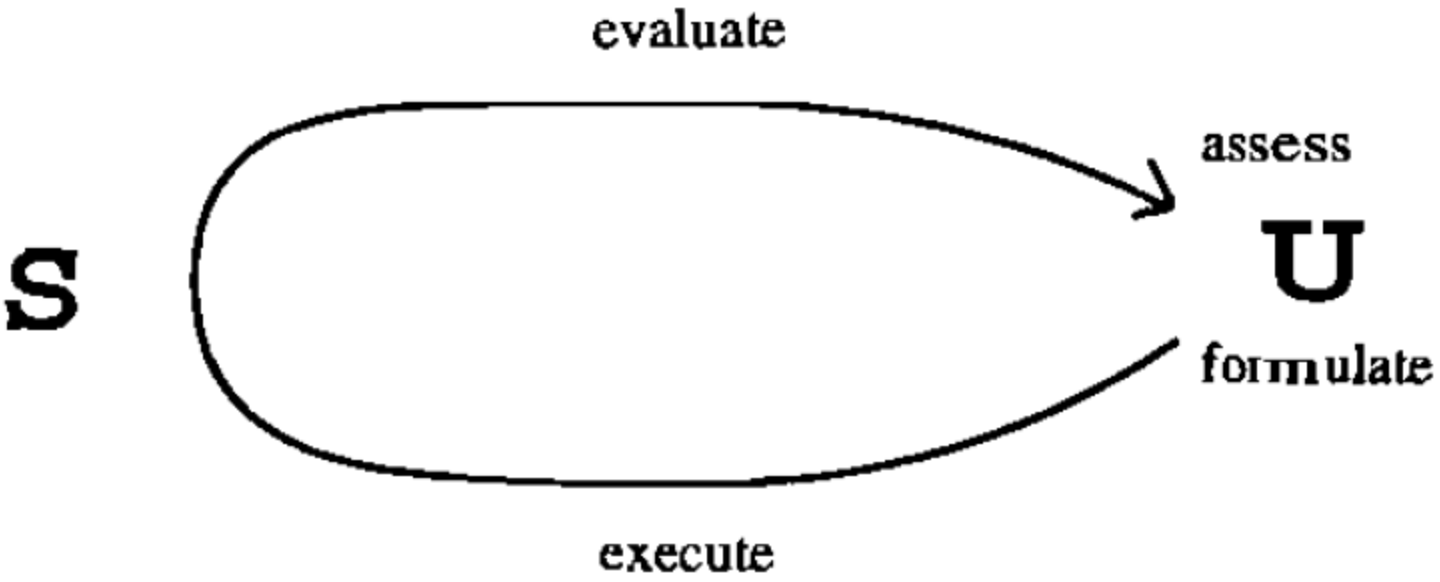
Main Factors

1.Organizational Factors
– training, job design, politics, roles, work organisation
2. Environmental Factors
– noise, heating, lighting, ventilation
3. Health and Safety Factors
– stress, headaches, musculo-skeletal disorders
4. The User
– motivation, enjoyment, satisfaction, experience level
5. Comfort Factors
– seating, equipment layout

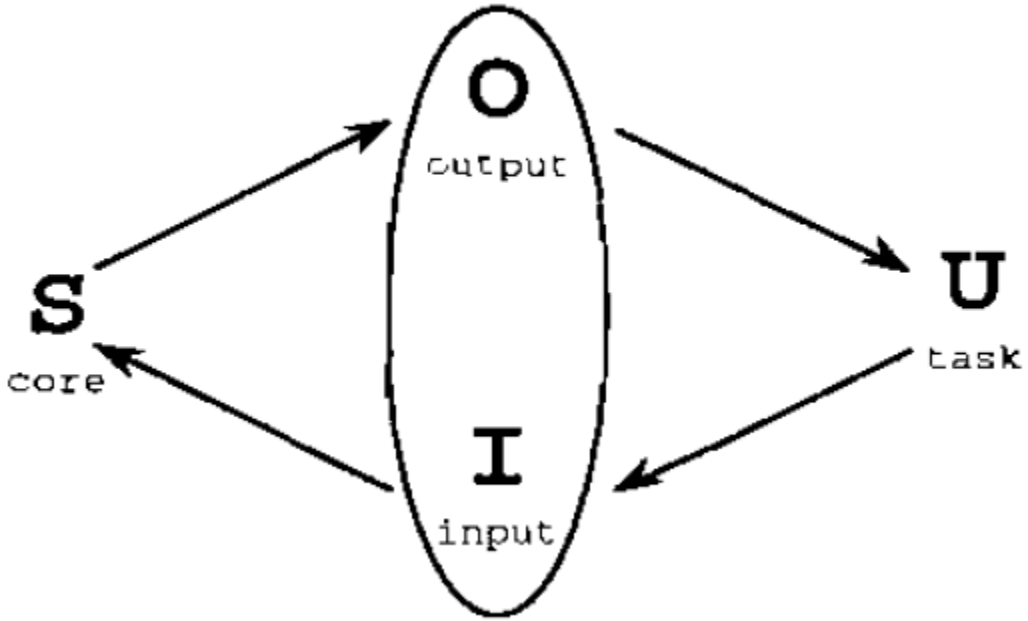
6. **User Interface**
– input devices, output devices, dialogue structures, use of colour, icons, commands, graphics, natural language, 3D, multimedia
7. **Task Factors**
– easy, complex, novel, task allocation, repetitive, monitoring, skills
8. **Constraints**
– costs, timescales, budgets, staff, equipment, building structure
9. **System Functionality**
– hardware, software, applications
10. **Productivity Factors**
– increase: output, quality
– decrease: costs, errors, labour requirements, production time

A Framework for Discussing Interaction

Four phases of interaction between *User (U) and System*

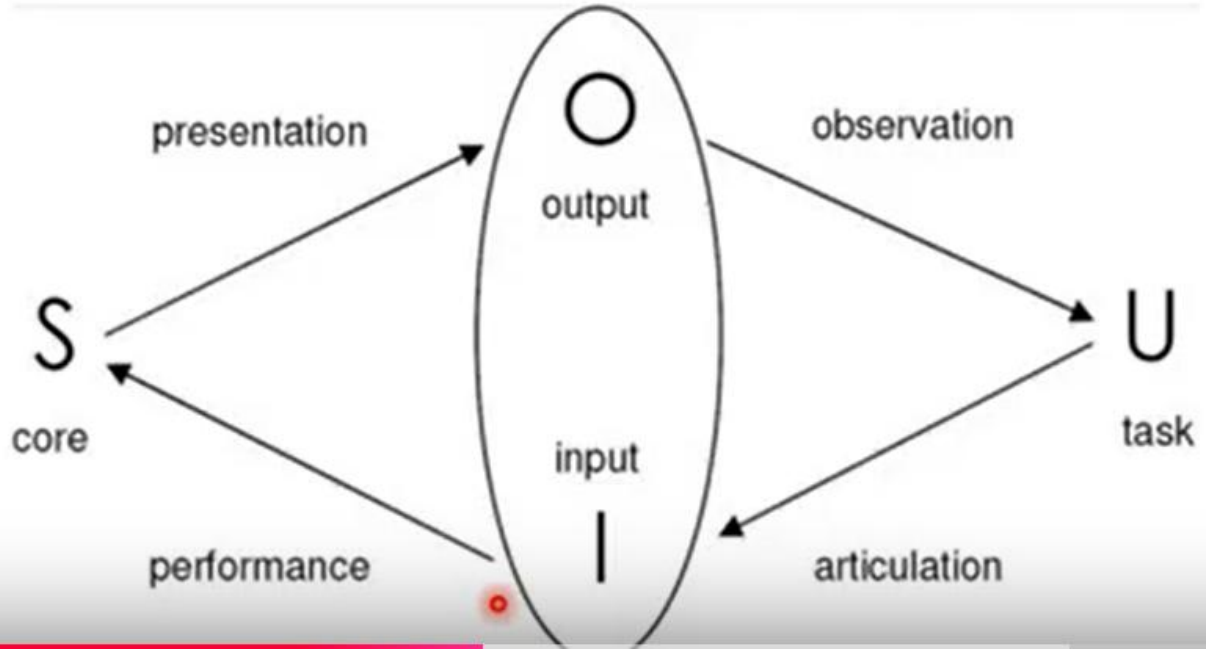


The general interaction framework



Interaction Framework

Abowd & Beale's model



Detail about Interaction Framework

1. **Core-**

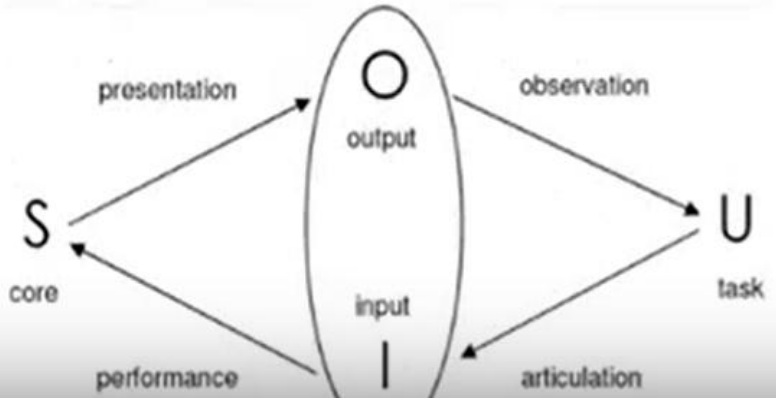
   The system language is known as core language which understand by system or its machine language.

2. **Task-**

   The users language referred as user task language understand by the user.

3. **Articulation-**

   Users produced some input to the system in various forms.



4. **Performance-**

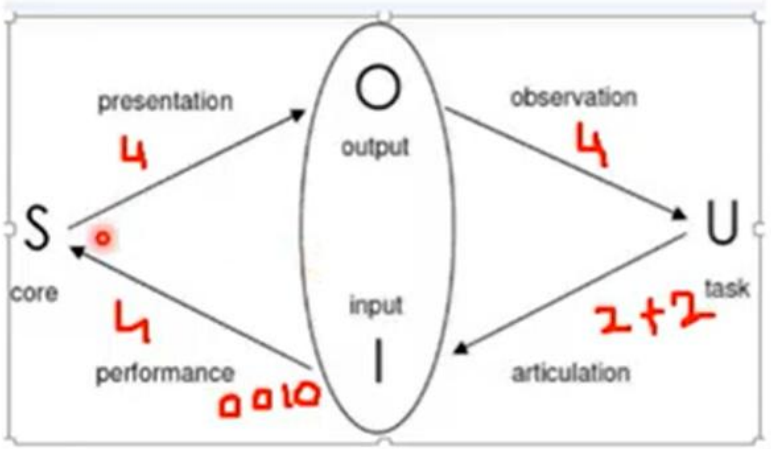
   System performs actions on those input.

5. **Presentation-**

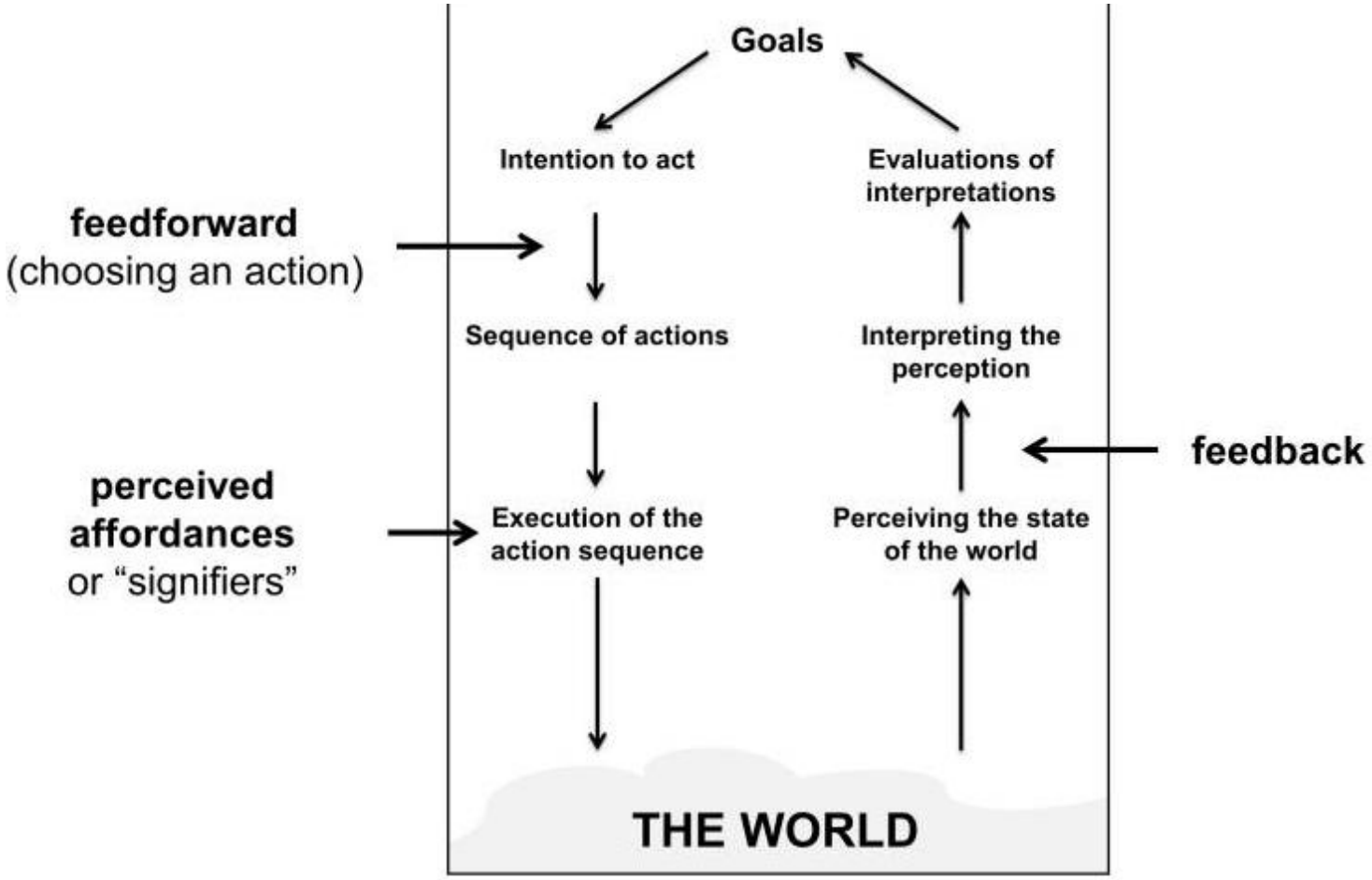
   System present the output to the user.

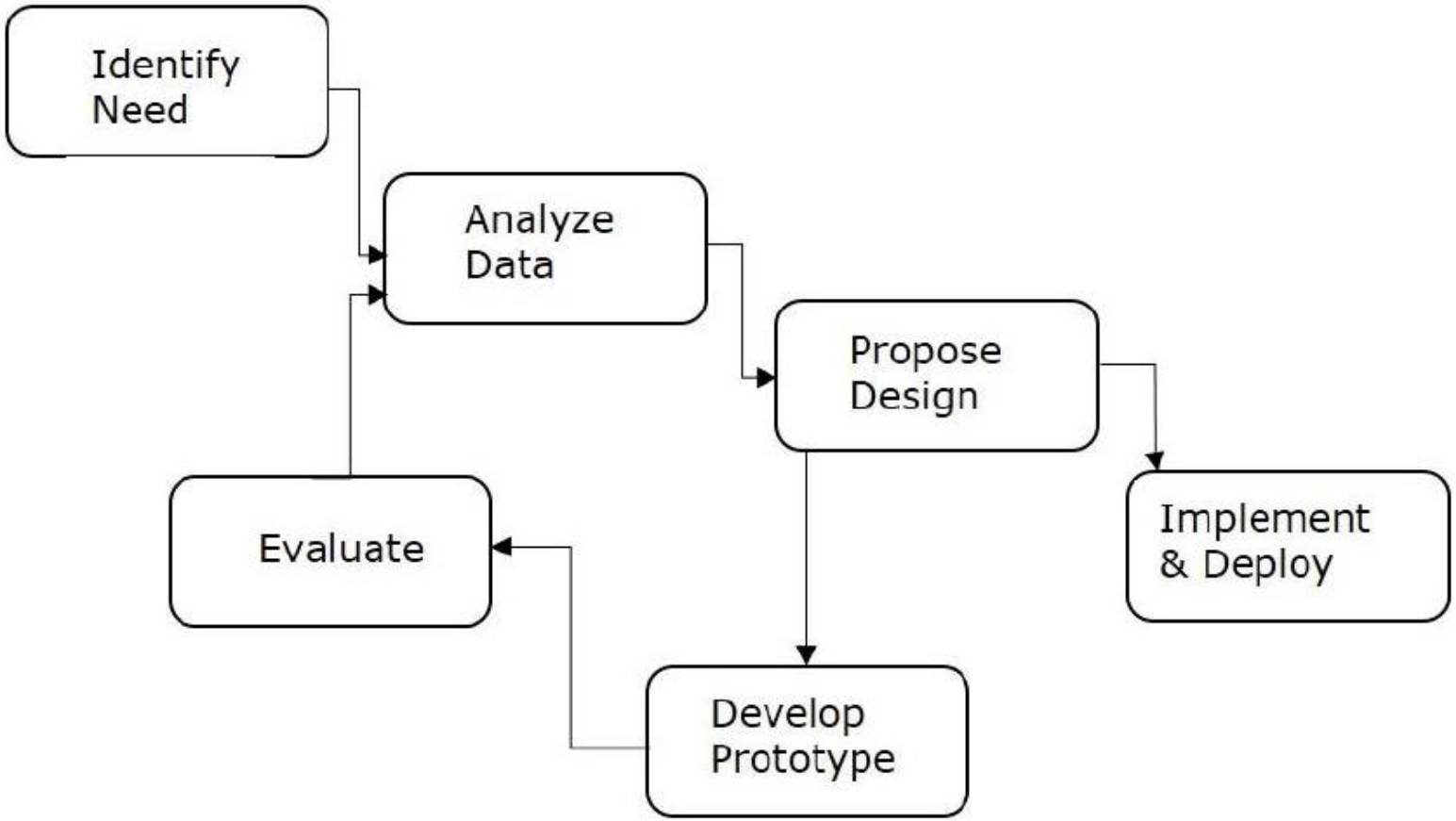
6. **Observation-**

   User observe those output and perform action on them.



Frame work of HCI




Identify Need
Analyze Data
Propose Design
Implement & Deploy
Evaluate
Develop Prototype

Donald Norman's Interaction framework

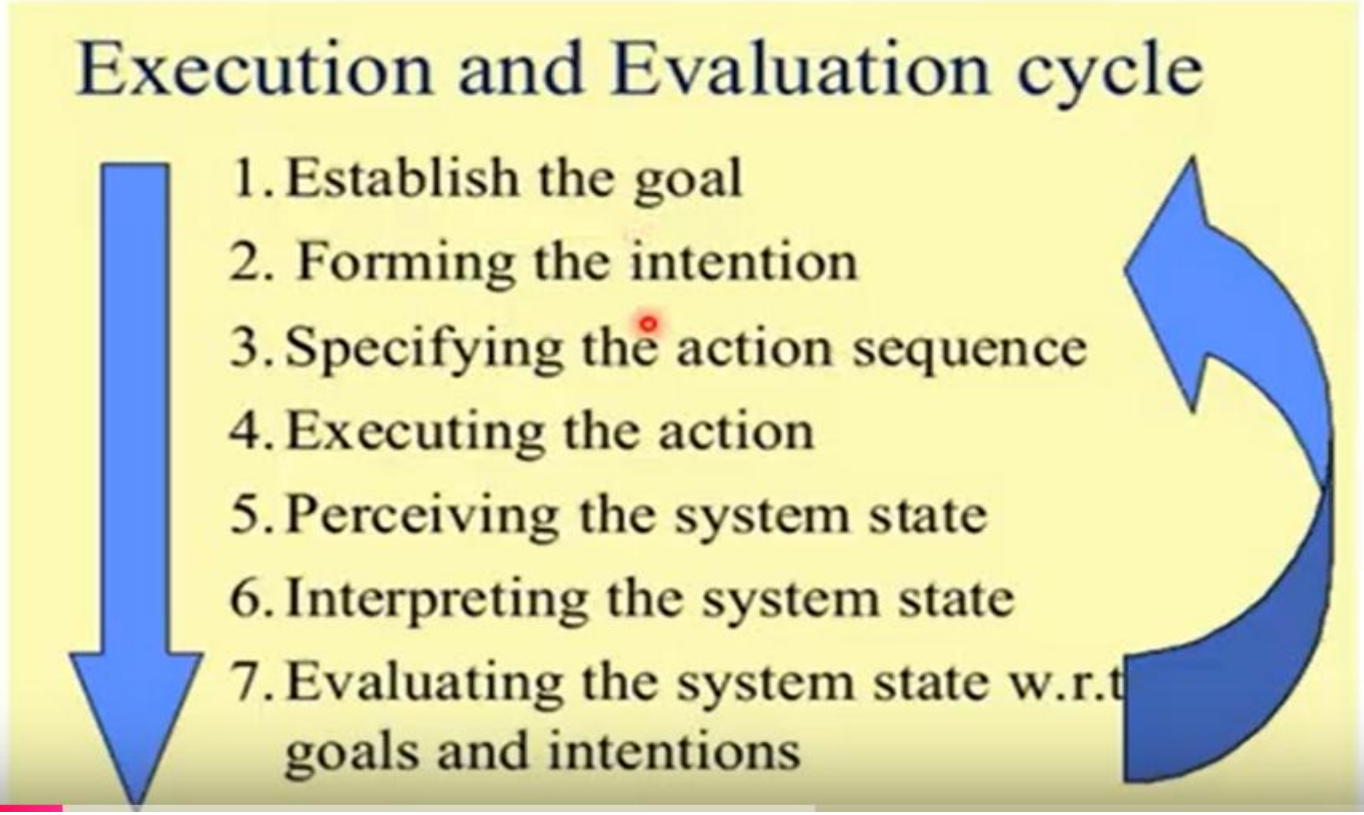
- In order to make good decisions, it is important to know how human beings are making decisions when they pursue an objective.
- Usually, when somebody is planning to interact with an object does several steps.

This steps could be divided into two categories called Gulfs.

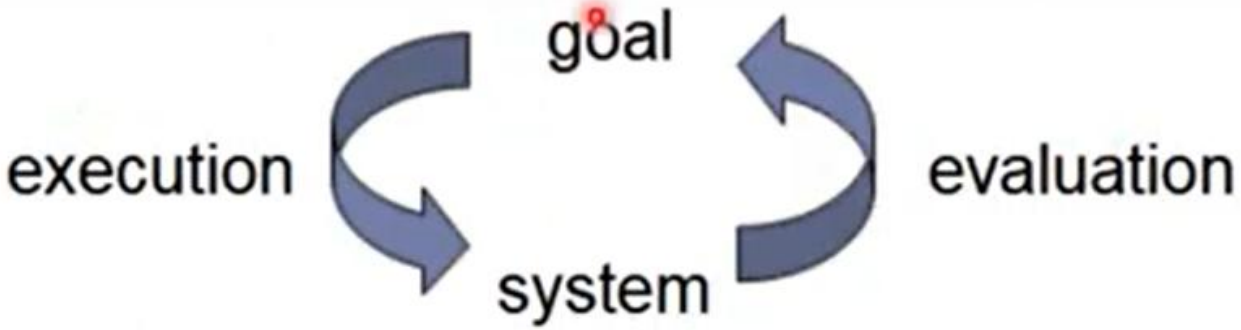
1. **Gulf of Execution** where we tend to discover the object's properties
2. **Gulf of Evaluation** where we get feedback related to our interaction.

- Norman's model of interaction (1988)- Most influential in HCI.

Seven Stages of Donald Norman's Model



Execution and Evaluation loop



- user establishes the goal
- formulates intention
- specifies actions at interface
- executes action
- perceives system state
- interprets system state
- evaluates system state with respect to goal

Example

- Lets you are imagine you are alone at home & bored, So you go & watch a movie.

Though this looks simple, our human brain executed it in the following way:

1. I want to kill my boredom **(Goal)**
2. Movie seems to be a good idea **(Plan)**
3. You check for the nearest theatre and show time **(Specify)**
4. Purchase ticket and sit in the movie hall **(Perform)**
5. Watch audio, Video affects of the movie **(Perceive)**
6. Interpret effects to your understanding **(Reflect)**

Nielsen's 10 Usability Heuristics

1. *Visibility of system status*
2. *Match between system and the real world*
3. *User control and freedom*
4. *Consistency and standards*
5. *Error prevention*
6. *Recognition rather than recall*
7. *Flexibility and efficiency of use*
8. *Aesthetic and minimalist design*
9. *Help users recognize, diagnose, and recover from errors*
10. *Help and documentation*

Shneiderman's 8 Golden Rules

1. *Strive for consistency*
2. *Enable frequent users to use shortcuts*
3. *Offer informative feedback*
4. *Design dialogs to yield closure*
5. *Offer error prevention and simple error handling*
6. *Permit easy reversal of actions*
7. *Support internal locus of control*
8. *Reduce short-term memory load*

Norman's 7 Principles

1. *Use both knowledge in the world and knowledge in t*
2. *Simplify the structure of tasks.*
3. *Make things visible: bridge the gulfs of Execution and Evaluation.*
4. *Get the mappings right.*
5. *Exploit the power of constraints, both natural and artificial.*
6. *Design for error.*
7. *When all else fails, standardize.*

Principles of HCI

- Despite its importance, good HCI design is generally difficult, mainly because it is a multiobjective task that involves simultaneous consideration of many things, such as the types of users, characteristics of the tasks, capabilities and cost of the devices, lack of objective or exact quantitative evaluation measures, and changing technologies, to name just a few.
- A considerable knowledge in many different fields is required. Over the relatively young history of HCI, researchers and developers in the field have accumulated and established basic principles for good HCI design in hopes of achieving some of the main objectives (as a whole) that were laid out in the previous section. These HCI principles are general, fundamental, and commonsensical, applicable to almost any HCI design situation. Here, we provide a short review of the main HCI principles.

Principles of HCI

- *Know Thy User”*
- *Understand the Task*
- *Reduce Memory Load*
- *Strive for Consistency*
- *Remind Users and Refresh Their Memory*
- *Prevent Errors/Reversal of Action*
- *Naturalness*

Know Thy User

- The foremost creed in HCI is to devise interaction and interfaces around the target users.
- “Know thy user,” so-called user-centered design approach has become a buzzword only in recent years.
- This principle simply states that the interaction and interface should cater to the needs and capabilities of the target user of the system in design.

- Consider a situation where a developer is working to change an inter-face, supposedly to achieve higher usability. However, we might need to remember that while young adults are extremely adept at and open to adopting new interfaces, older generations are much less so
- If a direct field study is not feasible, an experienced and humble HCI designer will at least try to leverage the vast knowledge available from cognitive psychology, ergonomics, and anthropomorphic data to assess the capa-bilities and characteristics of the target user group.

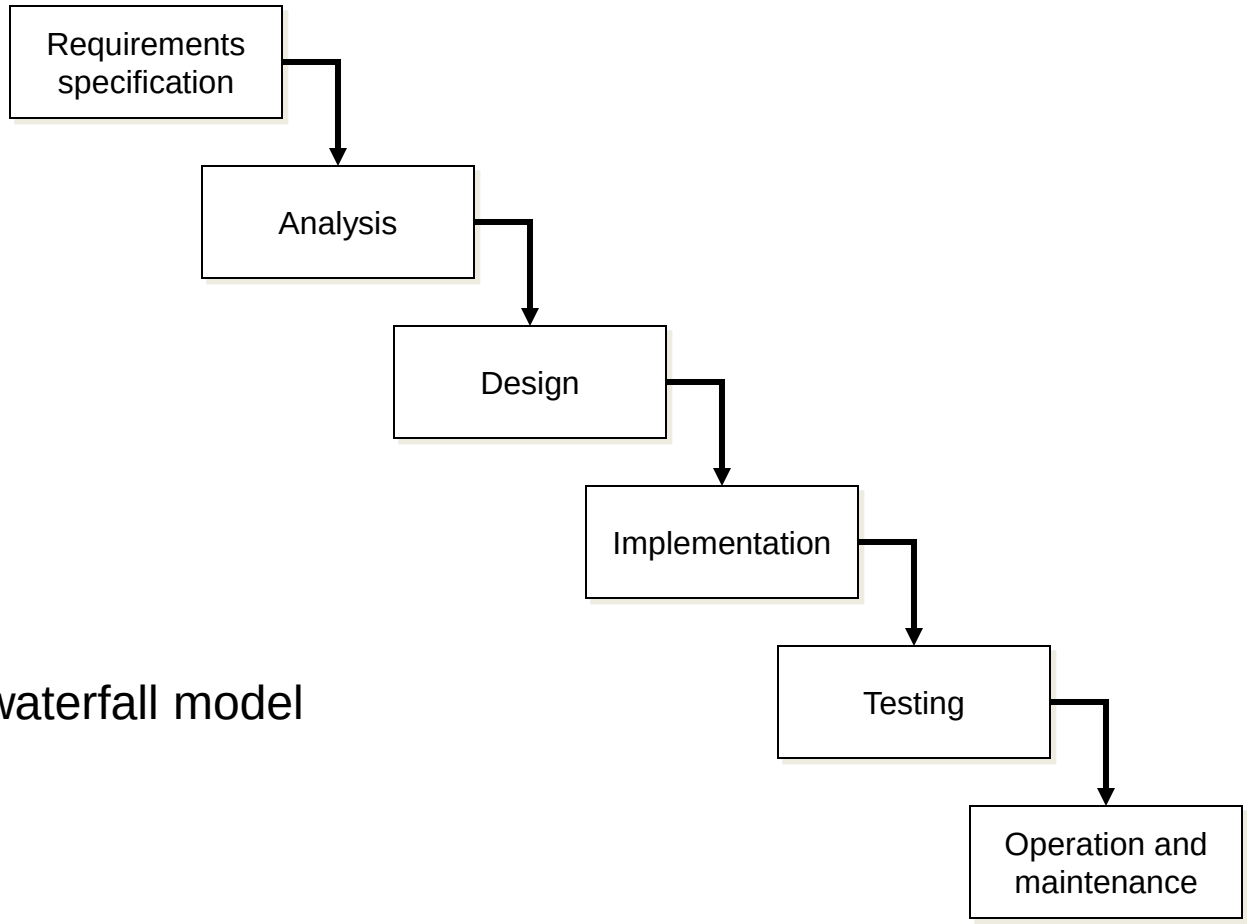
Examples of user-centered designs of web pages for (a) kids (courtesy of Junior Naver and (b) the elderly (courtesy of SilverNet News,

Traditional approaches to system development

- Is concerned with producing software, software specification, maintainability, and testing

- Generally considers the interface to be just another software component.

Example of traditional approach



The waterfall model

Introduction to User Centered approach

This approach normally involves a number of key activities throughout the development of the software including:

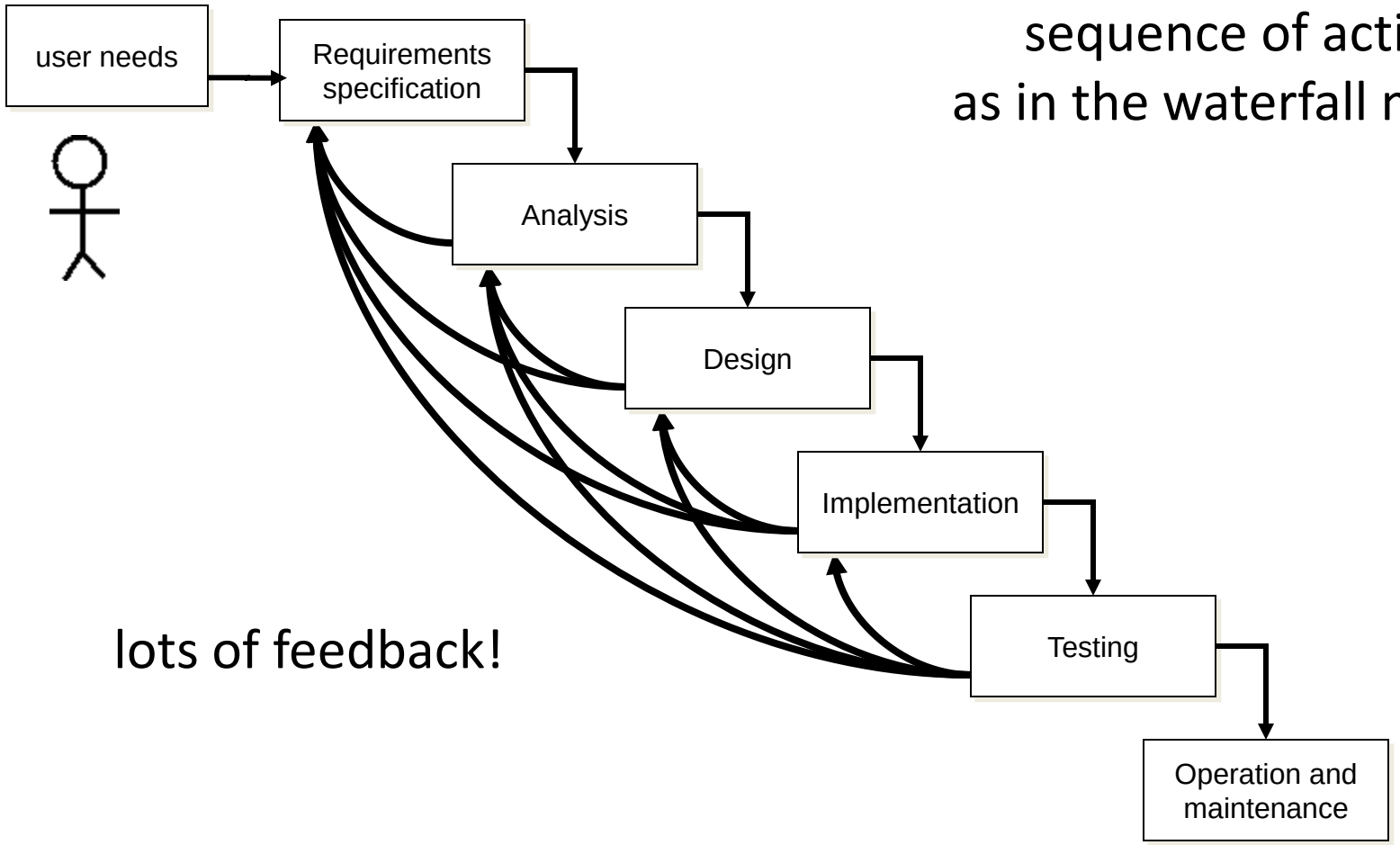
- Involving users
- Obtaining their feedback on the design
- Providing prototypes for system evaluation and re-design in light of user feedback and comments.

User Centered approach

- Real users involved at each step of the process
- Find out about the users before requirement specification
- Design and implementation
- Review (usability test) with the users

The life cycle for interactive systems

cannot assume a linear sequence of activities as in the waterfall model



lots of feedback!

User Centered Development

1. Data Collection
2. Data Analysis
3. Prototyping
4. Design
5. Evaluation

1. Data Collection

- Data recording
  - Using media
- Interviews
  - Stakeholder interviews
  - Subject Matter Expert interviews
  - User and customer interviews
- Questionnaires
  - Surveys, product reviews
- Literature review
  - Studying existing systems

2. Data Analysis

- Requirement analysis
  - Formal specifications of the system
- User analysis
  - Identifying and understanding the user
- Task analysis
  - Steps user take to accomplish this task
- Functional analysis
  - Functions that system perform to help the users carry out their task

3. Prototyping

Advantages of Prototyping:

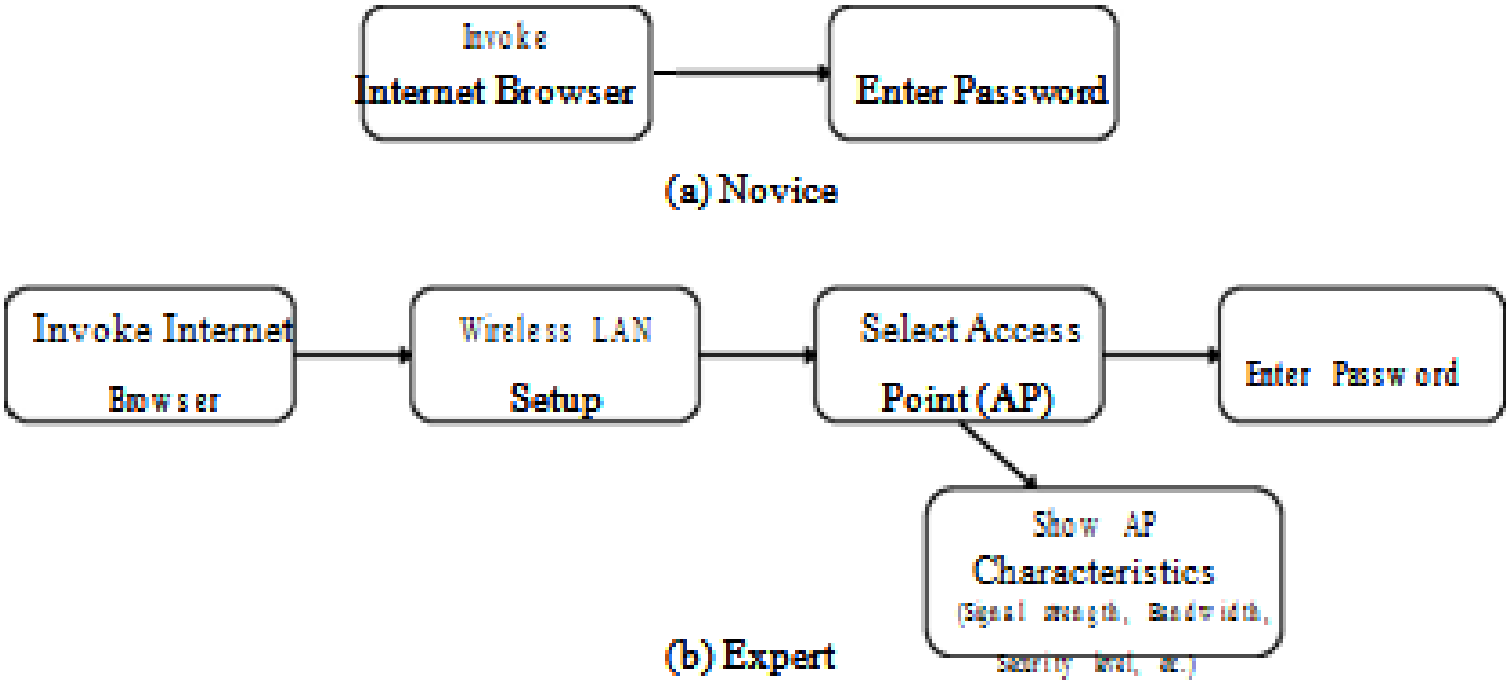
- Users are actively involved in the development
- It provides a better system to users
- The users get a better understanding of the system being developed.
- Errors can be detected much earlier
- Quicker user feedback is available leading to better solutions

4. Design

- Goals
  - Achieving goals
- Users and systems
  - Understanding the raw materials: computer and human
- Limitations
  - Accepting limitations of humans and of design

Understand the Task

- The term *task* refers to the job to be accomplished by the user through the use of the interactive system
- Different users will have different mental models of the task at hand, and this must be reflected in the structure of the interface to simplify implementation for all users
- Example: Take the subtask (for a larger application) for “changing the Wi-Fi connec-tion access point” for a smartphone
- task/interaction model may sometimes be developed based solely on the general human capacity.


Invoke
Internet Browser
Enter Password
(a) Novice
Invoke Internet
Browser
Wireless LAN
Setup
Select Access
Point (AP)
Enter Password
Show AP
Characteristics
(Signal strength, Bandwidth,
Security level, etc.)
(b) Expert

Reduce Memory Load

- Designing interaction with as little memory load as possible is a principle that also has a theoretical basis
- The capacity of the human's short-term memory (STM) is about 5–9 chunks of information (or items meaning-ful with respect to the task), famously known as the "magic number"

Interfaces designed for minimal short-term memory

Strive for Consistency

- the user is likely to get confused and exhibit erroneous responses if the same subtask is involved, at different times, for different interaction steps or interface methods
- Aside from being able to remember what to do, consistency and familiarity also lead to higher acceptability and preference

Remind Users and Refresh Their Memory

- Any significant task will involve the use of memory, so another good strategy is to employ interfaces that give continuous reminders of important information and thereby refresh the user's memory
- Even a single task may proceed in different contextual spans. For instance, in an online shopping application, one might cycle through the entry of different types of information: item selection, delivery options, address, credit card number, number of items, etc.
- To maintain the user's awareness of the situation and further elicit correct responses, informative, momentary, or continuous feedback will refresh the user's memory and help the user complete the task easily.
- One particular type of informative feedback (aside from the cur-rent status) is the reaffirmation of the user action to signal the closure of a larger process

Prevent Errors/Reversal of Action

- error-free operation is equally important
- the interaction and interface should be designed to avoid confusion and mental overload
- Inactive menu items are good examples of such a technique. Also, having the system require the user to choose from possibilities (e.g., menu system)
- Despite employing some of the principles and techniques described here, there is always a chance that the user will make mistakes

Naturalness

- “natural” interaction and interfaces.
- *Naturalness* refers to a trait that is reflective of various operations in our everyday life. For instance, a perfect HCI may one day be realized when a natural language–based conversational interface is possible
- A natural or metaphoric interface (assuming that the metaphor is not contrived) will also have *affordance*, a property (or additional cues) that appeals to our innate perception and cognition, thus making it so intuitive that the interface would require almost no learning

Interaction Styles

- Command Line
- Menu-Based Interface
- Form Fill-In
- Question and Answer
- Direct Manipulation
- Metaphors
- Web Navigation
- Three-Dimensional Environments
- Zoomable Interface
- Natural Language

Interaction Styles - *Command Line*

- Command-line interfaces are fast and powerful.
  - Many commands are abbreviated
    - quick and efficient
  - Commands can be applied to many objects simultaneously
    - fast input
  - Some commands have multiple parameters that can be set and altered
    - precise and flexible

Interaction Styles - *Command Line*

- Command Line and the EECA
  - Intention formation, specification of the action, and the execution stages are complex
  - Requires a rather accurate mental model of the computer's internal processing
- Command Line and the Interaction Framework
  - Translating the user's task language into the input language requires knowledge of the core language
  - The output language can be confusing for inexperienced users - there is very little feedback

Interaction Styles - *Command Line*

- Command Line and Articulatory Distance

  - Articulatory distance is large because we are presented with only the command prompt - no indication of functionality

Interaction Styles - *Command Line*

- Advantages of command-line interfaces:
  - Suitable for repetitive tasks
  - Advantageous for expert users
  - Offer direct access to system functionality
  - Efficient and powerful
  - Not encumbered with graphic controls
    - Low visual load
    - Not taxing on system resources

Interaction Styles - *Command Line*

- Disadvantages of command-line interfaces:
  - Low command retention
  - Steep learning curve
  - High error rates
  - Heavy reliance on memory
  - Frustrating for novice users

Interaction Styles - *Menu-Based Interface*

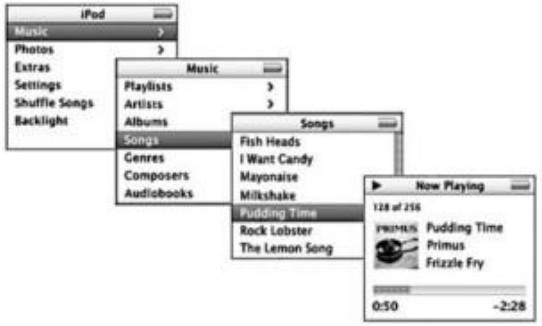
- Menu-driven interfaces present users with sequential hierarchal menus that offer lists of functions.
  - Textual: key-in number of option
  - Graphical: use arrow keys or pointing device

Interaction Styles - *Menu-Based Interface*

Maxim

Menus are based on recognition as opposed to recall

- No need to remember commands
- Users search from a list of possible choices
- List provides constraints
- Appropriate for small screens (iPod)



Interaction Styles - *Menu-Based Interface*

- Menu-based interfaces and the EEAC

  - Menu constraints can help the user to form the proper intentions and specify the proper action sequence

  - Provide a context to evaluate the output language

Interaction Styles - *Menu-Based Interface*

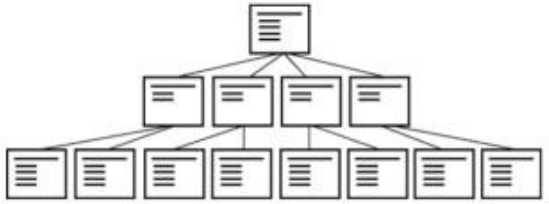
- Menu-based interfaces and :
  - **Articulatory Distance**
    - Menu options create small articulatory distance
  - **Mental Models**
    - Menu construction has a direct impact on user's mental model
  - **Affordances**
    - Menu elements present affordances

Interaction Styles - *Menu-Based Interface*

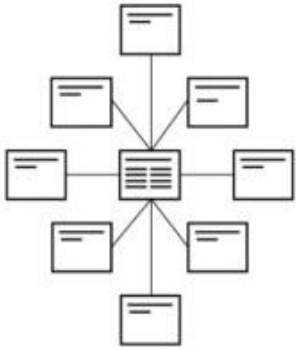
- Most menus are a variation on a few basic categories:

Single

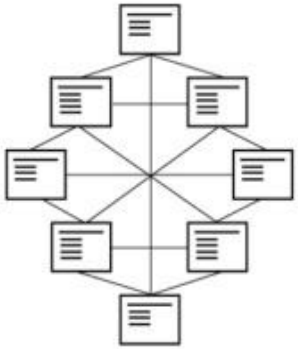
Sequential



Hierarchal



Star network



Web network

Interaction Styles - *Menu-Based Interface*

- Advantages of menu-based interfaces:
  - Low memory requirements
  - Self-explanatory
  - Easy to undo errors
  - Appropriate for beginners

- Disadvantages of menu-based interfaces:
  - Rigid and inflexible navigation
  - Inefficient for large menu navigation
  - Inefficient use of screen real estate
  - Slow for expert users

Interaction Styles - *Form Fill-In*

• Similar to menu interfaces - present screens of information

• Different than menu interfaces - used to capture information and proceed linearly not to navigate a hierarchical structure

Interaction Styles - *Form Fill-In*

MAXIM

Always inform the user about the length of paged forms and where they are within the structure

- Forms can be presented using
  - Single scrolling screens
  - Multiple linked pages
- Form elements must be grouped logically
- Include “You Are Here” indications

Interaction Styles - *Form Fill-In*

MAXIM

Form elements must be unambiguously labeled to increase data integrity

- Users must understand what data is required and what format should be used
  - Date information formats

1/29/2005, 29/1/2005, or January 29, 2005?

Interaction Styles - *Form Fill-In*

- Advantages of form fill-in interfaces:
  - Low memory requirements
  - Self-explanatory
  - Can gather a great deal of information in little space
  - Present a context for input information

- Disadvantages of form fill-in interfaces:
  - Require valid input in valid format
  - Require familiarity with interface controls
  - Can be tedious to correct mistakes

Interaction Styles - *Question and Answer*

- Question and answer interfaces are also called wizards.
- They are restricting for expert users
- They are easy for novice users
  - However, they may not know the required information

MAXIM

Users must be able to cancel a menu without affecting the state of the computer

Interaction Styles - *Question and Answer*

- Microsoft Add Network Place Wizard

(a) Add Network Place wizard. (b) Select a service provider. (c) Address of the network place.

Interaction Styles - *Question and Answer*

- Advantages of question and answer interfaces:
  - Low memory requirements
  - Self-explanatory
  - Simple linear presentation
  - Easy for beginners

- Disadvantages of question and answer interfaces:
  - Require valid input supplied by user
  - Require familiarity with interface controls
  - Can be tedious to correct mistakes

Interaction Styles - *Direct Manipulation*

- Ben Shneiderman (1982)

  - Continuous representations of the objects and actions of interest with meaningful visual metaphors.

  - Physical actions or presses of labeled buttons instead of complex syntax.

  - Rapid, incremental, reversible actions whose effects on the objects of interest are visible immediately.

Interaction Styles - *Direct Manipulation*

• Three phases in Direct Manipulation - Cooper, Reimann (2003)

- **Free Phase—**How the screen looks before any user actions
- **Captive Phase—**How the screen looks during a user action (click, click-drag, etc.)
- **Termination Phase—**How the screen looks after a user action

Interaction Styles - *Direct Manipulation*

- Direct Manipulation and the EEAC
  - The range of possible intentions is consistently wide
  - Users usually have multiple options for specifying action sequences
    - Can be overwhelming of novice users
  - Provide multiple ways of executing action sequences

Interaction Styles - *Direct Manipulation*

- Advantages of direct manipulation interfaces:
  - Easy to learn
  - Low memory requirements
  - Easy to undo
  - Immediate feedback to user actions
  - Enables user to use spatial cues
  - Easy for beginners

- Disadvantages of direct manipulation interfaces:
  - Not self-explanatory
  - Inefficient use of screen real estate
  - High graphical system requirements

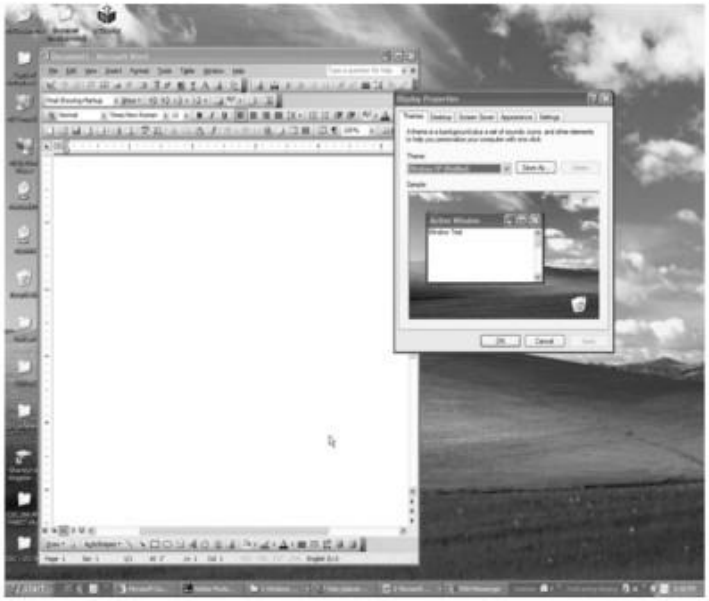
Interaction Styles - *Metaphors*

- GUIs use visual relationships to real-world objects (metaphors)
- Metaphors can help people relate to complex concepts and procedures by drawing on real-world knowledge
- Real-world affordances can be reflected

- What metaphors are used by contemporary GUIs?

Interaction Styles - *Metaphors*

Microsoft Windows XP



Apple OS X

Interaction Styles - *Metaphors*

MAXIM

A metaphor's function must be consistent with real-world expectations

- Metaphors that do not behave the way people expect will cause confusion and frustration
- Macintosh trashcan

Interaction Styles - *Metaphors*

MAXIM

Don’t force a metaphor

- Potential problems with metaphors
  - Run out of metaphors
    - Some virtual processes and objects have no real-world counter parts
  - Mixed metaphors
  - Carry connotations and association

Interaction Styles - *Web Navigation*

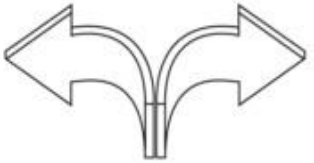
- Two basic interaction styles
  - Link-based navigation
    - Sensitive to articulatory distance
    - Ambiguous link labels increase the gulf of evaluation
  - Search
    - Sensitive to semantic distance
    - Inadequate search engine algorithms increase the gulf of execution
    - Slight advantage in development of mental models

Interaction Styles - *3D Environments*

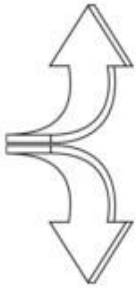
- 3D interaction is natural in the real-world
- 3D environments are common in digital games
- Rich graphical 3D environment are processor intensive

Interaction Styles - *3D Environments*

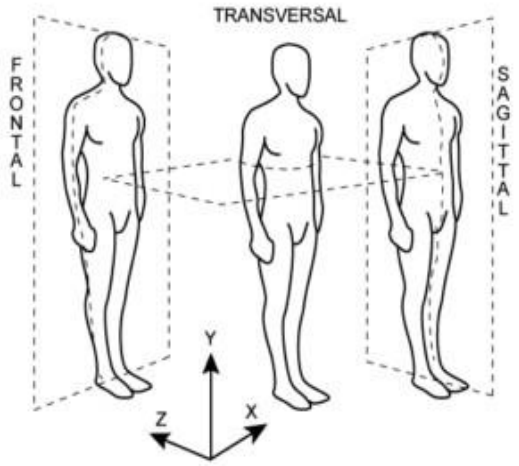
- 3D Navigation
  - Involves two types of movement
    - Translation - movement on a plane
    - Rotation - movement around an axis



Yaw



Pitch



Interaction Styles - *3D Environments*

- Web-based 3D
  - Use vector-based graphics to decrease file size
  - Virtual Reality Modeling Language (VRML)
    - Uses polygons with parameters
      - Transparency
      - Texture maps
      - shininess
  - X3-D is XML based - **Web3D.org**
    - Offers greater flexibility and control

Interaction Styles - *3D Environments*

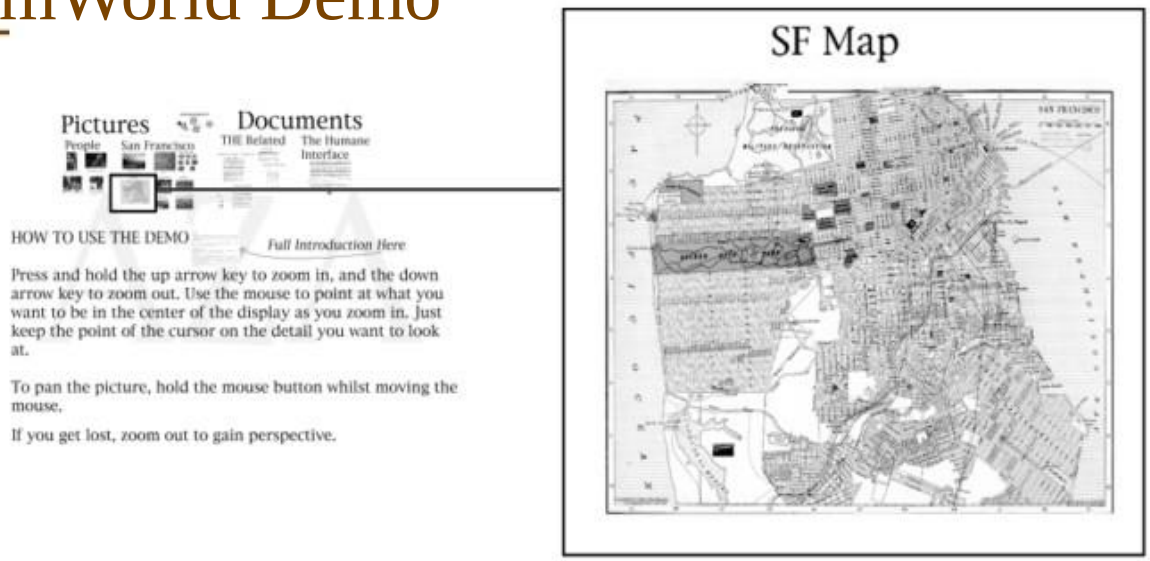
- Desktop 3D
  - Current GUIs are predominantly 2D
  - 3D environments presented on 2D screens are difficult to navigate

MAXIM

Three-dimensional navigation can quickly become difficult and confusing

Interaction Styles - *Zoomable Interface*

- ZoomWorld (Jeff Raskin) is based on the zooming interface paradigm (ZIP)
- ZoomWorld Demo



Interaction Styles - *Zoomable Interface*

MAXIM

Zoomable interfaces allow us to use our sense of relative positioning

- ZIP is based on landmarks and relative positioning (organizational cues)
  - Proportion
  - Color
  - Patterns
  - Proximity
- **Pad++:** Zoomable User Interface (ZUI)

Interaction Styles - *Natural Language*

- Natural Language Interaction (NLI) - Interacting with computers using everyday language
- Obstacles
  - Language is ambiguous
  - Meaning depends on context
    - "Search results"
    - "She said she did not know"
  - Dependant on visual cues

Interaction Styles - *Natural Language*

- Applications for NLI
  - Speech Input
    - Hands-free operation
    - Poor Lighting Situations
    - Mobile Applications
    - In the home
  - Speech Output
    - On-board navigational systems

Interaction Styles - *Natural Language*

- Two areas of development
  - Speech recognition
  - Semantics
    - Grammar issues
    - Vague meanings
    - Contradictory statements

MAXIM

NLIs may require constant clarification of linguistic ambiguities

Interaction Styles - *Natural Language*

- Advantages of NLI:
  - Ease of learning
  - Low memory requirements
  - Flexible interaction
  - Low screen requirements
  - Appropriate for beginners

- Disadvantages of NLI:
  - Requires knowledge of the task domain
  - May require tedious clarification dialogues
  - Complex system development