UNIT 5:

Software tools – Specification methods, interface – Building Tools.
Interaction Devices – Keyboard and function keys – pointing devices
Speech recognition digitization and generation – image and video displays– drivers.

Ben Shneidermann, Designing the user interface, 3rd Edition, PearsonEducation Asia.

Software Tools

Specification Methods

- Grammars
- Menu-selection and dialog-box trees
- Transition diagrams
- Statecharts

Interface-Building Tools

- Interface mockup tools
- Software-engineering tools

Evaluation and Critiquing Tools

A software tool is programmatic software used to create, maintain, or otherwise support other programs and applications.

UIs should be: reliable, standard, safe, inexpensive, effective widely acceptable and be build on a predictable schedule
Software tools help to achieve this!

UI Architect needs:

- Simple and quick methods of sketching UIs
- Precise methods for working out details with the client, for coordinating with team-members and for telling the developers whatto do

3 sets of tools are available:

- UI specification methods
- Software tools to support design and software engineering
- Evaluation and critiquing tools

Trends:

- Desktop ↓
- Web, Mobile, Universal Usability and Customization ↑
  - UI building tools need to be adapted for these changes

Specification Methods

The methods used to specify the GUI.

Natural-language specifications

- The default language for specifications in any field is natural language, e.g., English
- Communication medium, e.g., sketchpad, or blackboard

These specifications tend to be:

- lengthy
- vague
- ambiguous

Therefore often are difficult to prove:

- correct
- consistent
- complete

Formal and Semiformal languages

- Have a specified grammar
- Have specified and effective procedures to determine whether a string adheres to Grammar

Grammars for command languages are effective
But For GUI they are too short and are used to describe sequences of actions

Grammars

Useful to specify textual commands or expressions that a program should understand

Eg : **Backus-Naur Form – BNF**

Used to describe programming languages

High-level components are described by **nonterminals**
Specific strings are described as **terminals**

Grammars Example : Telephone book entry

```
<Telephone book entry> ::= <Name><Telephone number>
<Name> ::= <Last name>, <First name>
<Last name> ::= <string>
<First name> ::= <string>
<string> ::= <character>|<character><string>
<character> ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<Telephone number> ::= (<area code>) <exchange>-<local
number>
<area code> ::= <digit><digit><digit>
<exchange> ::= <digit><digit><digit>
<local number> ::= <digit><digit><digit><digit>
<digit> ::=   0|1|2|3|4|5|6|7|8|9
```

LHS are nonterminals given in < > and are defined by RHS

Examples of acceptable entries

- WASHINGTON, GEORGE (301) 555-1234
- BEEF, STU (726) 768-7878
- A, Z (999) 111-1111

Incomplete

- Good to specify the format, but fails to specify content, such as valid exchanges or area codes.
- A second level of software is used to validate content

Complex

- As BNF Grammar grows, it becomes very complex and difficult to follow or change

Multiparty grammars

- Used for showing interactions between more than one party (Shinderman, 1982)
- Non-terminals labeled by the party which generates (i.e. U for user and C and for Computer)
- Effective for text-oriented command sequences which have repeated exchanges

Eg : opening steps in a login process

```
<Session> ::=< U: Opening><C: Responding>
        <U: Opening>:= LOGIN <U: Name>
        <U: Name> ::=< U: string>
        <C: Responding>:= HELLO [<U: Name>]
```

Pros

- *Verification of completeness and correctness*

Cons:

- *Two-dimensional styles, such as form fillin or direct manipulation and graphical layouts, are more difficult to describe with multiparty grammars.*

Menu-selection and dialog-box trees

- A menu-selection tree has a simple structure that guides designers and users alike, making it an excellent selection style for many applications.
- Specification methods include online tools to help in the construction of menu trees and simple drawing tools that enable designers and users to see the entire tree at one time
- Due to the intention of avoiding clutter such trees are incomplete, are restricted to a “static view” and do not showthe entire structure of possible user actions

Problem:

- In some aspects of UI design a precise specification of every possible action is required
- Incases of non-menu systems there might be a need to express the set of possible states and allowed transitions

Solution: Use Transition Diagrams

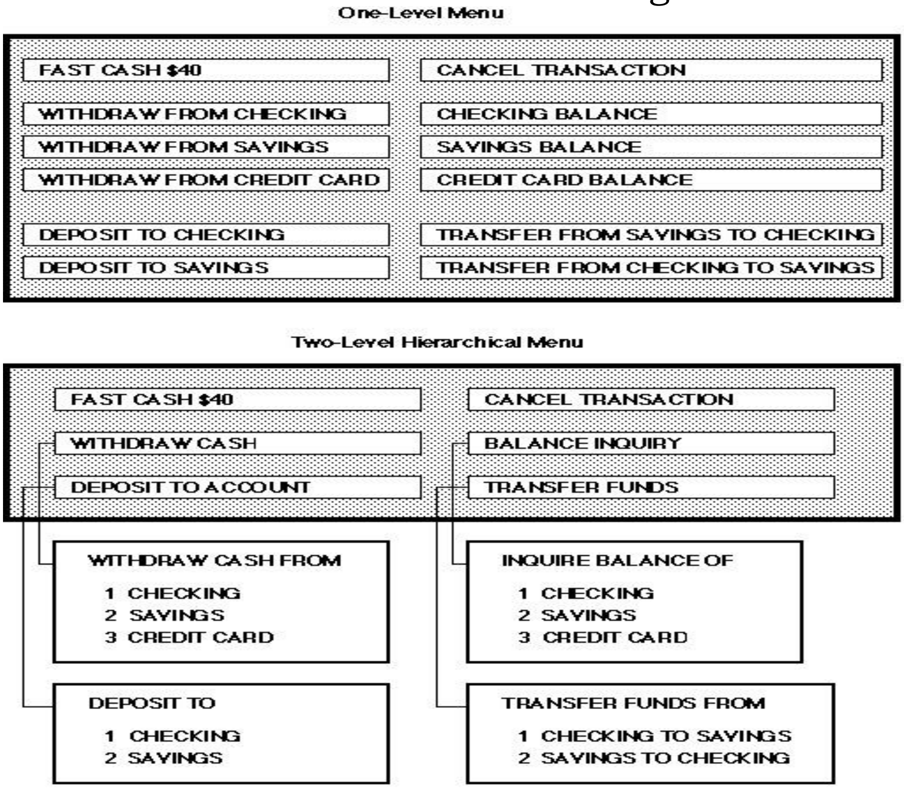


Figure 8.2. Two menu arrangements for automatic teller machines.

Transition Diagrams

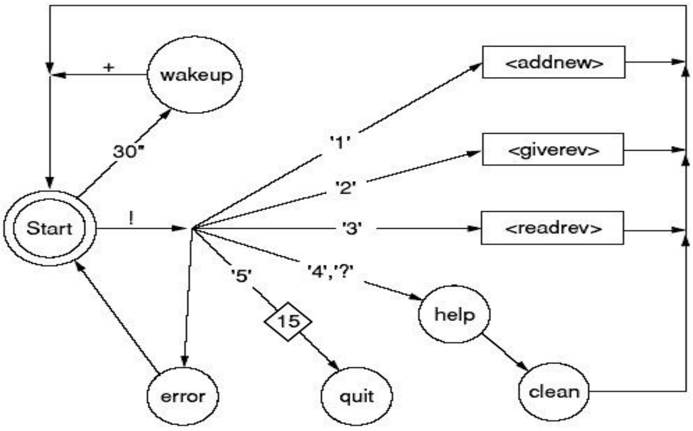
- Transition diagram has a set of nodes that represent system states and a set of links between the nodes that represent possible transitions.

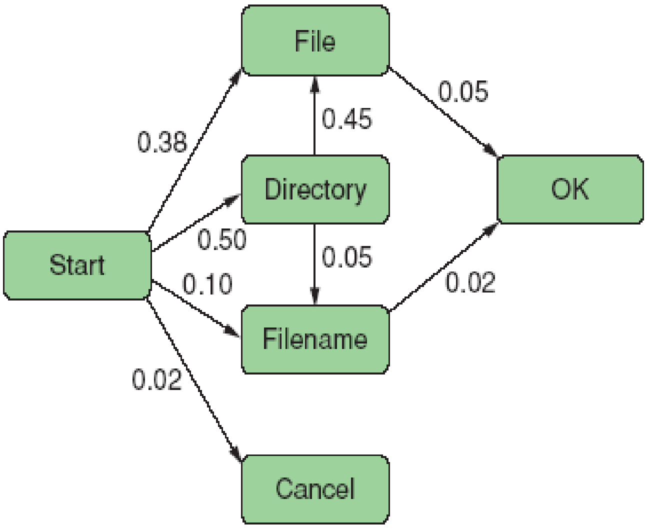
- Each link is labeled with the user action that selects that link and possible computer responses.

Pros: can be effective for keeping track of the current state and options of the UI/System, can also be verified

Cons: Too complex with growing system complexity (too many states and possible transitions), confusing when many nodes need a link to "help", "back", "quit", etc., in their basic form poor representation of concurrency and synchronization

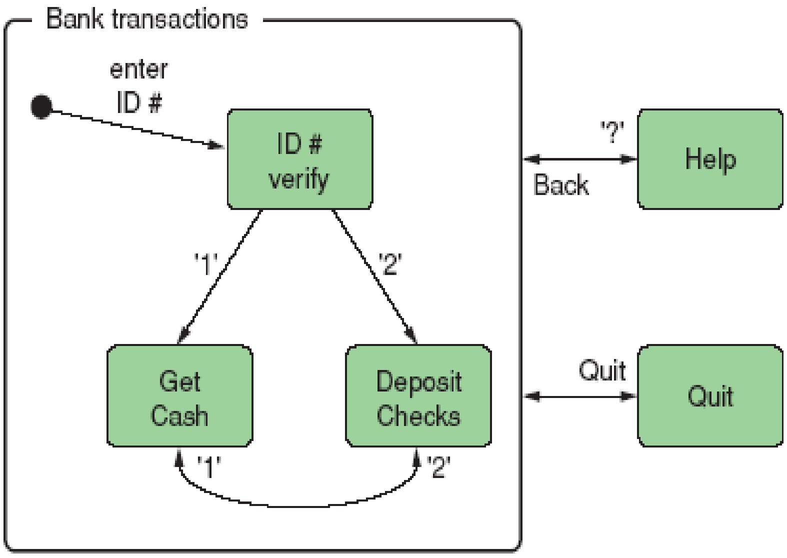
Alternative: State-Charts





Statecharts

- Statecharts have several virtues in specifying interfaces.
- Because a grouping feature is offered through nested roundtangles repeated transitions can be factored out to the surrounding roundtangle.
- Extensions to statecharts-such as concurrency, external interrupt events, and user actions-are represented in **I-Logix's**, which is a user-interface tool based on statecharts. The UML also uses statechart to specify the behavior of general program, thus they should be familiar to software engineers.



Interface-Building Tools

1. Specification methods are important for the design of components of a system, such as command languages, data-entry sequences, and widgets.
2. Screen-transition diagrams drawn or printed on paper are an excellent means of providing an overview of the system.
3. They allow user-interface architects, designers, managers, users, and software engineers to sit around a table, discuss the design, and prepare for the big job that lies ahead.
3. Paper-based designs are a great way to start, but the detailed specification of complete user interfaces requires software tools such as :

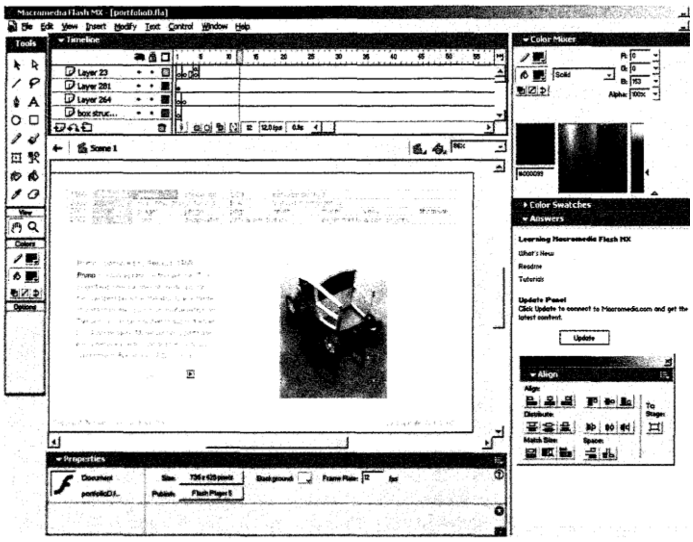
Interface Mockup Tools

i. User interface architects recognize that creating quick sketches is important during the early stages of design to explore multiple alternatives, to allow communication within the design team, and to convey to clients what the product will look like.

ii. User interface mockups can be created with paper and pencil, word processors, or slide show presentation software (such as Microsoft PowerPoint or Apple Keynote).

iii. Resourceful designers have also built user-interface prototypes with multimedia construction tools, such as Dreamweaver.

iv. These programs can quickly generate animated or even interactive programs and be distributed via the Web.



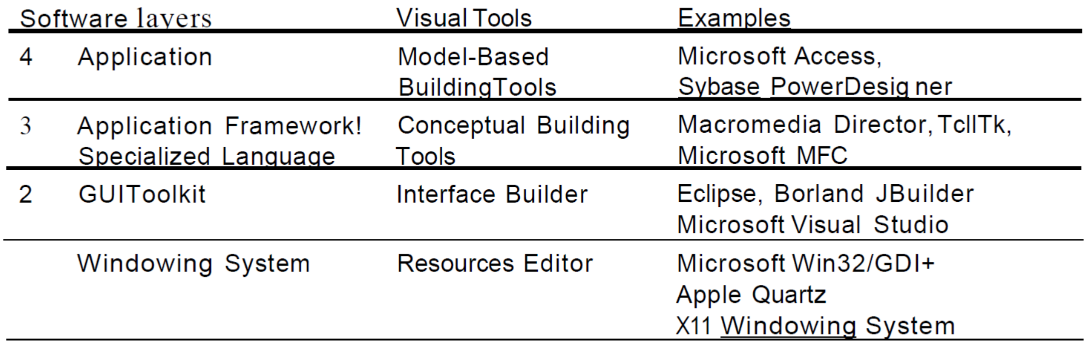
Click Update to connect to Macromedia.com and get the latest content.

This Flash MX design shows a multimedia presentation with a time line on top, a color palette on the top right, alignment options on the bottom right, and standard graphic tools on the left. Once created, a Flash file is saved that can be made available through the Internet and run on most web browsers using a free plug-in.

Software-engineering tools

- Main Purpose: Build the final product
- In earlier days, general purpose languages (e.g., Java,C++) were used to build UIs from scratch
- Today, this is supported through software tools, APIsand Frameworks, etc.
- A classification of UI building tools:
  - Layer 1: Windowing Toolkit
  - Layer 2: GUI Toolkit
  - Layer 3: Application Framework/Specialized Languages
  - Layer 4: Application Level

The following table lists the four software layers that can be used to build a user interface and their associated interactive tools



Software layers		Visual Tools	Examples
4	Application	Model-Based BuildingTools	Microsoft Access, Sybase PowerDesig ner
3	Application Framework! Specialized Language	Conceptual Building Tools	Macromedia Director, TclITk, Microsoft MFC
2	GUIToolkit	Interface Builder	Eclipse, Borland JBuilder Microsoft Visual Studio
	Windowing System	Resources Editor	Microsoft Win32/GDI+ Apple Quartz X11 Windowing System

Windowing-system layer

Sometimes when with new platforms, working at a low-level is required.

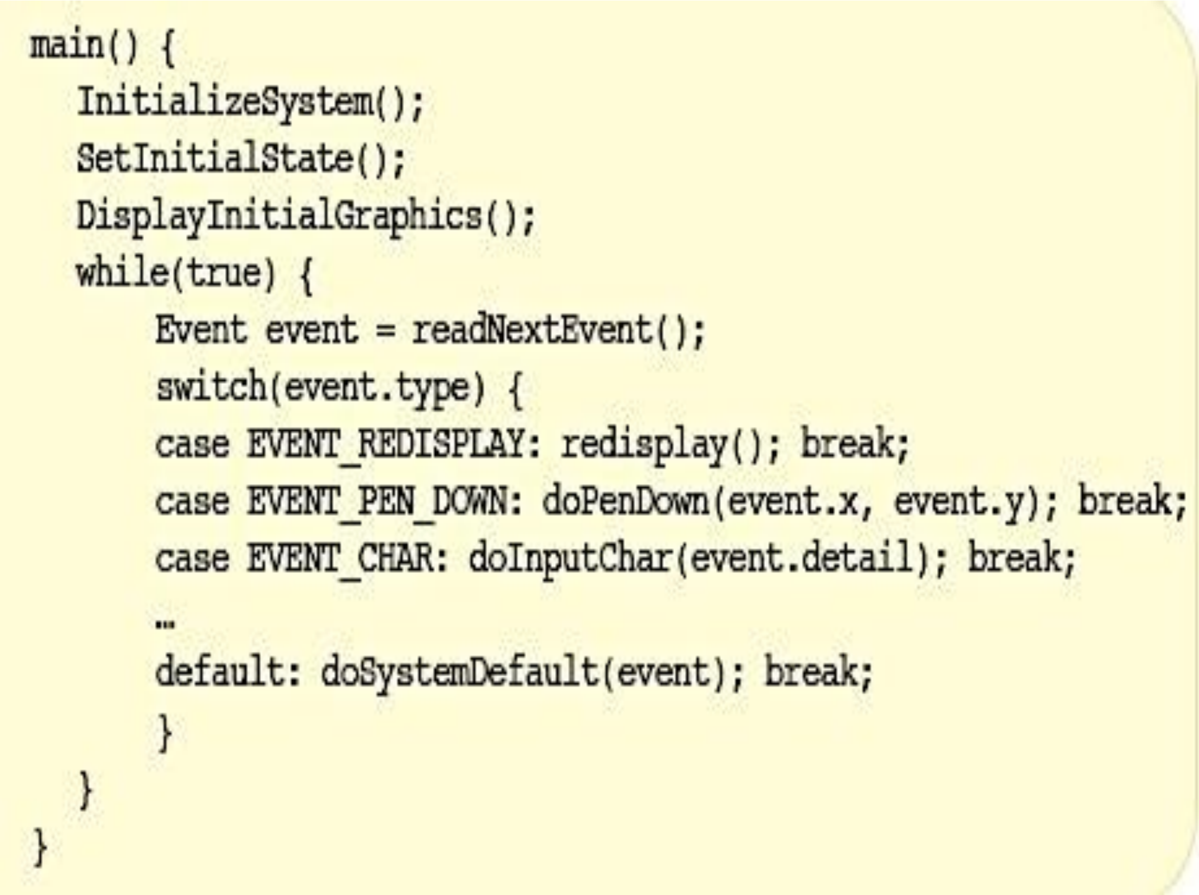
Involves Extensive programming

Interfaces for some mobile devices or cell phones should currently be done at the windowing-system level

E.g., for a new platform, all programs have the following form

- The while(true) main loop exits in every interactive program
- This loop manages the state transitions, triggered by events arriving from the windowing system or the operating system

Programming at this level is sometimes appropriate when no higher-level tool exists or when memory or performance is critical



```
main() {
  InitializeSystem();
  SetInitialState();
  DisplayInitialGraphics();
  while(true) {
      Event event = readNextEvent();
      switch(event.type) {
      case EVENT_REDISPLAY: redisplay(); break;
      case EVENT_PEN_DOWN: doPenDown(event.x, event.y); break;
      case EVENT_CHAR: doInputChar(event.detail); break;
      …
      default: doSystemDefault(event); break;
      }
  }
}
```

GUI-toolkit layer

- Provides software libraries and widgets (e.g., frames,dialog boxes, scrollbars) including event encapsulation as building blocks
- **Pros**: Abstraction, shorter development times, great flexibility, often based on general-purpose languages (makes it easier to combine UI with application logic)
- **Cons**: Possibly high learning times, building and maintenance time is still high, no support for consistency, experienced programmer needed
- Example for Toolkits: ILOG Views, Gtk, Qt, Java (AWT, Swing, SWT), .Net GUI toolkit)

Application framework and specialized language layer

- Application Framework
  - Software architecture specifically designed for building GUIs
  - **Idea**: Many UI-based programs have a similar structure, capture structure, translate it to classes, objects and methods, which can thenbe extended/reused
  - Commonly uses widgets from the GUI Toolkit layer
  - Based on convenient visual programming, simple

    - scripting languages,or general purpose languages
    - Less support for non-graphical part of the application
    - Example: Cocoa, MFC, Macromedia Director
- Specialized Languages
    - Languages specifically designed for building UIs
    - Can also be used for rapid prototyping
    - Example: Tcl, java script in combination with HTML

Application layer

- Interface Generators, also called: Model-BasedSystems or UI Management Systems
- Visual tools, that allow for most parts of an applicationto be built with one tool and without coding
- Only available for a small class of applications (e.g., DB front-ends) or research prototypes

Selecting the right tool

- Obviously higher-level tools are more efficient,however, they are the most restricting as well
- Finding the right tool is a tradeoff between 6 criteria:
    - Which part of the application do you need to build using thetool?
    - How long can learning time be / or are experts available?
    - How long can the building time be?
    - Do you accept a methodology being imposed or ratheradvised?
    - Is communication with other subsystems required?
    - Do you need to consider extensibility and modularity?

Features of User-interface building tools

1. User interface independence
   a. Separate interface design from internals
   b. enable multipleuser interface strategies and multiple platform support
   c. establish role of UI architect
   d. enforce standards
2. Methodology and Notation
   a. Develop design procedures
   b. find ways to talk aboutdesign
   c. create project management
3. Rapid Prototyping
   a. Try out new ideas very early
   b. test, revise, test, revise,...,
   c. engage end-users, managers and customers
4. Software Support
   a. Increase productivity
   b. offer constraint and consistencychecks
   c. facilitate team approach
   d. ease maintenance

Evaluation and CritiquingTools

- Build-Time Tools
  - Example: From simple spell checks to more sophisticated tools that provide metrics reporting number of displays, widgets etc. or even more advanced: check depth of menu-trees, redundancies, inconsistencies in labels etc.
- Runtime Logging Software
  - Capture user pattern of activity
  - Example: reports on error-frequencies, menu selections, dialogbox appearances, help invocations, or capture performance data

Interaction Devices

Several interactive devices are used for the human computer interaction. Some of them are known tools and some are recently developed or are a concept to be developed in the future.

Keyboard Layouts

QWERTY layout

- 1870 Christopher Latham Sholes
- good mechanical design and a clever placement of the letters that slowed down the users enough that key jamming was infrequent
- put frequently used letter pairs far apart, thereby increasing finger travel distances

Dvorak layout

- 1920
- reduces finger travel distances by at least one order of magnitude
- Acceptance has been slow despite the dedicated efforts of some devotees
- it takes about 1 week of regular typing to make the switch, but most users have been unwilling to invest the effort

ABCDE style

- 26 letters of the alphabet laid out in alphabetical order no typists will find it easier to locate the keys

Additional keyboard issues

- IBM PC keyboard was widely criticized because of the placement of a few keys
  - backslash key where most typists expect SHIFT key
  - placement of several special characters near the ENTER key
- Number pad layout
- wrist and hand placement

Keys

- 1/2 inch square keys
- 1/4 inch spacing between keys
- slight concave surface
- matte finish to reduce glare finger slippage
- 40- to 125-gram force to activate
- 3 to 5 millimeters displacement
- tactile and audible feedback important
    - certain keys should be larger (e.g. ENTER, SHIFT,CTRL)
    - some keys require state indicator, such as lowered position or light indicator (e.g. CAPS LOCK)
    - key labels should be large, meaningful, permanent
    - some "home" keys may have additional features, such as deeper cavity or small raised dot, to help user locate their fingers properly (caution - no standard for this)

Function keys

- users must either remember each key's function, identify them from the screen's display, or use a template over the keys in order to identify them properly
- can reduce number of keystrokes and errors
- meaning of each key can change with each application placement on keyboard can affect efficient use
- special purpose displays often embed function keys in monitor bezel
- lights next to keys used to indicate availability of the function, or on/off status
- Typically, simply labeled F1, F2, etc, though some may also have meaningful labels, such as CUT, COPY, etc.
- frequent movement between keyboard home position and mouse or function keys can be disruptive to use
- alternative is to use closer keys (e.g. ALT or

CTRL) and one letter to indicate special function

Cursor movement keys

- up, down, left, right
- some keyboards also provide diagonals
- best layout is natural positions
- inverted-T positioning allows users to place their middle three fingers in a way that reduces hand and finger movement
- cross arrangement better for novices than linear or box
- typically include typeamatic (auto-repeat)feature
- important for form-fill-in and direct manipulation
- Other movements may be performed with other keys, such as TAB, ENTER, HOME, etc.

Keyboard and keypads for small devices

- Wireless or foldable keyboards
- Virtual keyboards
- Cloth keyboards
- Soft keys
- Pens and touch screens

Pointing Devices

Pointing devices are applicable in six types of interaction tasks:

1. Select:
   - User chooses from a set of items.
   - Used for traditional menu selection, identification of a file in a directory, or marking of a part in an automobile design.
2. Position:
   - User chooses a point in a one-, two-, three-, or higher-dimensional space
   - Used to create a drawing, to place a new window, or to drag a block of text in a figure.
3. Orient:
   - User chooses a direction in a two-, three-, or higher-dimensional space.
   - Direction may simply rotate a symbol on the screen, indicate a direction of motion for a spaceship, or control the operation of a robot arm.
4. Path:
   - User rapidly performs a series of position and orient operations.
   - May be realized as a curving line in a drawing program, the instructions for a cloth cutting machine, or the route on a map.
5. Quantify:
   - User specifies a numeric value.
   - Usually a one-dimensional selection of integer or real values to set parameters, such as the page number in a document, the velocity of a ship, or the amplitude of a sound.
6. Text:
   - User enters, moves, and edits text in a two-dimensional space. The
   - Pointing device indicates the location of an insertion, deletion, or change.
   - More elaborate tasks, such as centering; margin setting; font sizes; highlighting, such as boldface or underscore; and page layout.

Types of Pointing Devices

Pointing devices can be grouped into :

- Direct control on the screen surface, such as the touchscreen or stylus, and
- Indirect control away from the screen surface, such as the mouse, trackball, joystick, graphics tablet, or touchpad.

Direct control pointing devices :

Lightpen : The lightpen is a device that enabled users to point to a spot on a screen and then press a button to perform a select, position, or other task.

Touchscreen : Touchscreen is robust and does not require picking up an external device; instead, it allows users to make direct-control touches on the screen with a finger.

Stylus : The stylus is an attractive device because it is familiar and comfortable for users, and users can guide the stylus tip to the desired location while keeping the whole context in view.

Indirect control pointing device :

- **Mouse**
  - the hand rests in a comfortable position, buttons on the mouse are easily pressed, even long motions can be rapid, and positioning can be precise
- **Trackball**
  - usually implemented as a rotating ball 1 to 6 inches in diameter that moves cursor
- **Joystick**
  - are appealing for tracking purposes
- **Graphics tablet**
  - A touch-sensitive surface separate from the screen
- **Touchpad**
  - built-in near the keyboard offers the convenience and precision of a touch screen while keeping the user's hand off the display surface

Direct control devices (easy to learn and use, but hand may obscure display)

- Lightpen
- Touchscreen
- Stylus

Indirect control devices (takes time to learn)

- Mouse
- Trackball
- Joystick
- Trackpoint
- Touchpad
- Graphics tablet

Novel devices and strategies (special purposes)

- Foot controls
- Eye tracking
- 3D trackers
- DataGloves
- Boom Chameleon
- Haptic feedback
- Bimanual input
- Tangible user interfaces
- Digital paper

Criteria for success

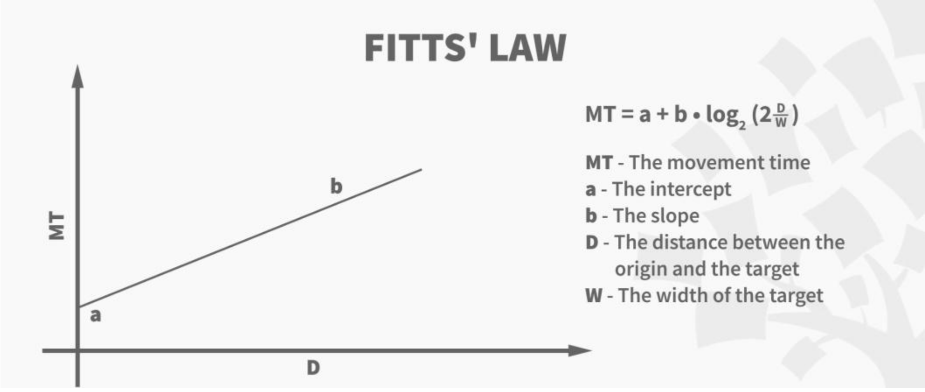
- Speed and accuracy
- Efficacy for task
- Learning time
- Cost and reliability
- Size and weight

Comparison of pointing devices

- Some results
  - direct pointing devices faster, but less accurate
  - graphics tablets are appealing when user can remain with device for long periods without switching to keyboard
  - mouse is faster than isometric joystick
  - for tasks that mix typing and pointing, cursor keys a faster and are preferred by users to a mouse
  - muscular strain is low for cursor keys

- **Fit’s Law**

Fitts's Law accurately predict the amount of time taken to move to and select a target. Fitts's Law is typically applied to movement through the graphical user interface using a cursor or other type of pointer.


FITTS' LAW
MT
b
a
D
$MT = a + b \cdot \log_2 (2\frac{D}{W})$
MT - The movement time
a - The intercept
b - The slope
D - The distance between the origin and the target
W - The width of the target

Speech Recognition Digitization and Generation

Speech Recognition

Speech recognition is the technology that enable machines, such as a computer or smartphone, to understand and interpret human speech. It is also known as Automatic Speech Recognition(ASR) or speech-to-text technology.

Speech recognition systems use algorithms and machine learning model to convert spoken language into the text. The process involves breaking down speech into the individual sounds of phonemes, and then using statistical model to match those sounds or word in a language model. Once the system has recognized the words, it can be then translate them into the text that can be processed and analyzed by the computer.

Speech recognition has numerous applications, including in virtual assistants, dictation software and voice activated devices. It can also be used in Healthcare, Education, Customer service to improve communication between the human and the Machine.

While speech recognition technology has come a long way, it is still face challenge in accurately recognizing speech in different languages, accents and the contexts. Ongoing research is focused on improving the accuracy of speech recognition system and making them more robust and adaptable to the different environments.

Speech Digitization

Speech digitization is the process to convert analog audio signals, which represent human speech, into digital data that can be processed and stored by a computer. The process involves two main stage: **Sampling and Quantization**.

In the Sampling stage, the continuous analog audio signal is converted into discrete signal by taking samples of the audio signals at the regular intervals. These samples are the **quantized**, which means that the analog signal is represented by a set of discrete value that can be stored as a digital data.

Speech digitization is important because it enable speech to be easily transmitted, stored and processed by the digital device any memory device such as computer or the smart phones. Digitized speech can be compressed to reduce its size for efficient storage and transmission, and it can also be easily manipulated and analyzed using Digital Signal Processing techniques.

Speech digitization has numerous applications, including in speech recognition system, Voice over Internet Protocol (VOIP) systems, and digital voice recording devices. It has also enabled the development of speech-based interface for various applications, such as virtual assistant and voice activated devices.

Overall, speech digitization has played a significant role in enabling human speech to be easily processed and transmitted into the digital age

Speech and Auditory Interfaces

Obstacle to Speech Recognition

- Increased cognitive load compared to pointing
- Interference from the noisy environment
- Unstable recognition across changing users environment and time

Obstacle to the Speech Output

- The slow pace of speech output compared to the visual displays
- Short term nature of the speech
- Difficulty is scanning/searching

Opportunities

- When users have vision impairments
- When the speakers hands are busy
- When Mobility is required
- When the speaker's eyes are occupied
- When harsh or cracked conditions preclude use of the keyboard

Technologies

- Speech store and forward
- Discrete-word recognition
- Continuous-speech recognition
- Voice information systems
- Speech generation

Speech Store and Forward

Speech Store and Forward refers to the process of capturing spoken language, storing it, and forwarding it to another location for later processing or analysis.

In the context of telecommunication, speech store and forward is often used to describe the process of capturing voicemail messages. When a caller leaves a voicemail message, their spoken language is captured and stored into the digital format. The voicemail system then forwards the message to the intended recipient, who can later listen to the message and respond accordingly.

Speech store and forward can also be used in other applications such as transcription services. In this case spoken language is captured and stored, and then forward to the human or machine transcriptionist who convert the speech into the written text.

The advantages of speech store and forward include the ability to capture and process spoken language at a later time, which can be useful for recording important information or messages. Additionally, speech store and forward can be used to enable Communication in situation when real-time communication is not possible or desirable, such as when communicating across different time zone and when dealing with sensitive or confidential information.

Overall, speech Stone forward is an important technology that enable the capture, storage and processing of a spoken language in variety of applications.

Discrete-word recognition

Discrete-word recognition, also known as isolated-word recognition, is a type of Automatic Speech Recognition (ASR) that focus on recognizing individual words rather than continuous speech. In discrete-word recognition, each spoken word is treated as a separate unit and is recognized independently.

The process of discrete-word recognition involves several steps. First, the spoken word is captured by a microphone and pre-processed to remove noise and the other distortion

Next, the audio signal is analyzed and segmented into individual word using techniques such as Dynamic Time Warping (DTW) or Hidden Markov models(HMMs).

Once the audio signal is segmented, each individual word is matched against a set of pre-defined word models, which are based on acoustic and language models. The acoustic model is used to match the sound of the spoken word with the corresponding phonemes, while the language model is used to determine the probability of each possible word based on the context of the speech.

Discrete-word recognition is commonly used in applications such as voice-activated assistants, voice dialing, and voice authentication. It is well-suited to situation where the user is expected to speak a limited set of predefined words, such as in phone menu system or in voice-controlled device with a limited vocabulary.

Discrete-word Recognition

- Recognizes individual words spoken by a specific person; works 90 -98% reliability

- Speaker-dependent training - repeat full vocabulary once or twice

- Speaker-independent training - not as accurate but eliminates training to expand the scope of commercial applications.

- Quite environments, head-mounted microphones, and careful word choices improve recognition rates to both.

Continuous-Speech Recognition

- Speech dictation products, IBM via voice: to help dictate letters and compose reports verbally for automatic transcription

- Can be used to scan and retrieve radio or television programs, court preceding, lectures or telephone calls for specific words or topics.

- Summarize can be generated from audio conversations.

- Close Caption generation
- Used for security systems and voice graph in court

Voice Information Systems

- Interactive voice Response systems: to check on airline flight departure/arrival times
- Voicemail Systems: Telephone based speech systems enabling users to store or forward messages.
- Apple iPod: allows to store large audio databases and retrieve selected segments

Speech Generation

Speech generation, is also known as text-to-speech (TTS) synthesis, is the process of converting written text into spoken words by a machine. TTS systems use algorithms and synthetic voices to read text aloud in a natural sounding way, giving the impression of a human speaker.

The process of speech generation involves several Stages. First, the text is analyzed and processed by Natural Language Processing (NLP) algorithm, which breaks it down into individual words and phrases. Next, the TTS system applies rules of pronunciation and intonation to generate a phonetic representation of the text.

This phonetic representation is then used for the synthesize the speech, either by concatenating pre-recorded speech fragments or by using a form of statistical parametric synthesis, which involves modelling the relationship between written text and corresponding acoustic features. Finally, the synthesized speech is outputted as an audio file or through a speaker.

Speech generation has numerous applications, including in assistive technologies for people with visual impairments or reading difficulties, as well as in virtual assistant and automated customer service systems. It can also be used in language learning and the pronunciation practice tools.

The development of speech generation technology has greatly improved the accessibility and usability of text-based content for individuals who may have difficulty reading or accessing written text. As the technology continues to advance, it is likely that speed generation will become more natural-sounding and capable of expressing a wide range of emotion and nuances in language.

Uses of speech Generation

- Widespread in consumer applications and the telephone applications.
- Used in automobile navigation systems, Internet services, utility-control rooms, and the children's games.

- Applications are good for the blind for text-to-speech utilities like Microsoft Windows Narrator
- Speech enabled readers: for documents, newspapers, statistical data and maps
- Web-based voice applications and telephone-based voice information systems.

Non-speech auditory interfaces

Non-speech auditory interfaces are interfaces that communicate with users through sound, but do not involve speech. These interfaces are often used in situation where visual interface may not be practical, such as in low light conditions, for people with visual impairments, or in situations where the user needs to be able to focus on task without looking at a screen.

Non-speech auditory interfaces can be particularly useful in situation where visual information is limited or unreliable such as dark or noisy environments. They can also be used to provide an additional layer of information to user who may be visually impaired or who prefer to interact with the technology in a non-visual way.

Some examples of non-speech auditory interface include:

- Tactile feedback devices that use sound to convey information through vibrations, such as wearable devices that notify the user of incoming calls or messages

- Sonification, which is the use of sound to represent data, such as in Scientific or Medical Visualizations.

- Auditory icons, which are sound that represent specific actions or events, such as the sound of camera shutter when taking a photo.

- Earcons, which are shorter musical phrases used to report specific actions or events, such as startup sound of the computer.

- Environmental sounds, such as the sound of car engine or the hum of the refrigerator which can be provide information about the status of the device or system

Image and Video displays

Visual display unit (VDU) has become the primary source of feedback to the user from the computer. Important features, of VDU are :

- Rapid operation
- Reasonable size
- Reasonable resolution
- Quiet operation
- No paper waste
- Relatively low cost
- Reliability
- Highlighting
- Graphics and animation

For certain applications, monochrome displays are adequate and are attractive because of their low cost.

Color displays can make video games educational simulations, CAD and many other applications more attractive and effective for users.

Display technologies include :

CRT :

- CRT is an evacuated glass tube equipped with various components.
- A beam of electrons (cathode rays), emitted by an electron gun, passes through focusing and deflection systems hits on the phosphor coated screen to generate the desired picture.
- The high speed electrons hit the phosphor coated screen to produce a spot of light controlled by a video controller.

Plasma panel :

- Plasma panels, also called gas discharge displays, are constructed by filling the region between two glass plates with a mixture of gases that usually includes neon.
- A series of vertical conducting ribbons is placed on one glass panel and a set of horizontal ribbons is built into the other glass panel.

Liquid Crystal Displays (LCDs) :

- LCDs are non-emissive devices which produce a picture by passing polarized light from the surroundings of an internal light source through a liquid crystal material.
- Liquid crystals are almost transparent substances, exhibiting the properties of both solid and liquid matter.

Flat panel display :

- Flat panel display is a display method that is designed to reduce the depth of the CRT display caused by the length of the tube.
- The screens of these flat panel displays are made up of pairs of electrodes.
- Each pair of electrodes is used to generate one picture element.

Characteristics of video display devices

Important characteristics of video display devices :

Persistence :

- Persistence is defined as the time it takes the emitted light from the screen to decay to one-tenth of its original intensity.
- Lower persistence phosphors require higher refreshing rates to maintain a picture on the screen without flicker. However it is useful for displaying animations.
- On the other hand higher persistence phosphors are useful for displaying static and highly complex pictures.

Resolution :

- Resolution indicates the maximum number of points that can be displayed without overlap on the CRT.
- It is defined as the number of points per centimeter that can be plotted horizontally and vertically.
- Resolution depends on the type of phosphor, the intensity to be displayed and the focusing and deflection systems used in the CRT.

Aspect ratio :

- It is the ratio of vertical points to horizontal points to produce equal length lines in both directions on the screen.
- An aspect ratio of 4/5 means that a vertical line plotted with four points has the same length as a horizontal line plotted with five points.

Printers

Important criteria for printers:

- Speed
- Print quality
- Cost
- Compactness
- Quiet operation
- Use of ordinary paper (fan folded or single sheet)
- Character set
- Variety of typefaces, fonts, and sizes
- Highlighting techniques (boldface, underscore, and soon)
- Support for special forms (printed forms, different lengths, and soon)
- Reliability

Dot-matrix printers

- print more than 200 characters per second, have multiple fonts, can print boldface, use variable width and size, and have graphics capabilities

Inkjet printers

- offer quiet operation and high-quality output

Thermal printers or fax machines

- offer quiet, compact, and inexpensive output on specially coated papers

Laser printers

- operate at 30,000 lines per minute

Color printers

- allow users to produce hardcopy output of color graphics, usually by an inkjet approach with three colored and black inks

Photographic printers

- allow the creation of 35-millimeter or larger slides (transparencies) and photographic prints

Drivers

- A driver is a program that interacts with a particular device or special (frequently optional) kind of software.

- The driver contains the special knowledge of the device or special software interface that programs using the driver do not. In personal computers, a driver is often packaged as a Dynamic Link Library (DLL) file.

- A driver in software provides a programming interface to control and manage specific lower level interface that is often linked to a specific type of hardware, or other low-level service.

- In the case of hardware, the specific subclass of drivers controlling physical or virtual hardware devices is known as device drivers.

- The main purpose of device drivers is to provide abstraction by acting as a translator between a hardware device and the applications or operating systems that use it.

- Programmers can write higher-level application code independently of whatever specific hardware the end-user is using.

  For example, a high-level application for interacting with a serial port may simply have two functions for “send data” and “receive data”. At a lower level, a device driver implementing these functions would communicate to the particular serial port controller installed on a user’s computer.