Dialog Systems

(Key Element of Human–computer Interaction and Robotics)

Micha Schwab

First Edition, 2012

ISBN 978-81-323-3535-1

© All rights reserved.

Published by:
University Publications
4735/22 Prakashdeep Bldg,
Ansari Road, Darya Ganj,
Delhi - 110002
Email: info@wtbooks.com

Table of Contents

Chapter 1 - Dialog System by Modality

Chapter 2 - Speech Synthesis

Chapter 3 - Graphical User Interface

Chapter 4 - Dialog System Applications

Chapter 5 - Dialog System Devices

WT

Chapter- 1

Dialog System by Modality

Modality

In human-computer interaction, a *modality* is the general class of:

- a sense through which the human can receive the output of the computer (for example, *vision modality*)
- a sensor or device through which the computer can receive the input from the human

In less formal terms, a modality is a path of communication between the human and the computer.

When multiple modalities are available for some tasks or parts of tasks, the system is said to have **overlapping modalities**. When multiple modalities are available for all tasks, the system is said to have **redundant modalities**.

Computer to human modalities

Any human sense can be translated to a modality. In practice, the modalities of seeing and hearing are the most commonly employed. The following are modalities through which the computer can send information to the human:

- Major modalities
  - Seeing or *vision modality*
  - Hearing or *audition modality*
- *Haptic modalities*
  - Touch, tactile or *tactition modality* — the sense of pressure
  - *Proprioception modality* — the perception of body awareness
- Other modalities
  - Taste or *gustation modality*

- Smell or *olfaction modality*
- *Thermoception modality* — the sense of heat and the cold
- *Nociception modality* — the perception of pain
- *Equilibrioception modality* — the perception of balance

Human to computer modalities

The computer can be equipped with various types of input devices and sensors to allow it to receive information from the human.

Systems:-

Text-based (computing)

Usually used in reference to a computer application, a **text-based application** is one whose primary input and output are based on text rather than graphics. This does not mean that text-based applications do not have graphics, just that the graphics are secondary to the text.

History

Before the 1980s, most computers were text-based. The operator used the keyboard as the main input device to type in necessary commands into the terminal that could only display text on a low-resolution monochrome video monitor. The majority of end-user software was also written in text-based mode during this time. During this era, operating a computer was considered to be a challenging task because of the complexity of the text-based environment.

However, with the development of the graphical user interface and the improvement in hardware, many software engineers started adding graphics for their applications. As a result, the pointing device that controls the coordination of the cursor on the screen became a primary input source (such as a mouse), and the graphics displayed with some text on the screen became a primary output source.

There is a lot of text-based software in modern operating systems, particularly in Unix and Unix-like, which can usually be accessed through the shell running in a system (or virtual) console or a terminal emulator. In these operating systems text-based programs continue to be the primary software for system administration, programming and scripting. On the contrary, Microsoft Windows contains far less text-based software, which is essentially the remnants due to the MS-DOS ancestry, even though there are still several programs for system administration and critical maintenance.

When the method used to access the operating system itself is text-based, the interface is usually referred to as a Command Line Interface (CLI). This function is carried out by various shells in Unix and Unix-like operating systems, and CMD and PowerShell in Microsoft Windows.

Benefits of text-based software

Text-based applications typically run faster than software involving graphics does. Text-based applications run faster because the machine does not expend resources on processing the graphics, which generally requires more system resources than text does. For the same reason, text-based applications use memory more efficiently.

Command line interfaces often provide the user more control on the software than a graphical user interface, by taking all the details of a command as parameters and/or by redirecting the outputs between commands. Since the available parameters are not explicitly enumerated, the application can accept many more options than an equivalent GUI-based software: a high number of options in a GUI would make it too complex and impractical, but that doesn't happen in a CLI. Thus the text-based input can provide more flexibility at the cost of learnability and a burden on user's memory.

As a result, text-based software can offer more powerful features than graphic-based software, such as combining the command using a pipeline that allows output of the first command to be used as the input of the next command. Using this, a complex operation can be accomplished in a single command line using a sequence of commands connected together in a pipeline.

In certain circumstances, text-based application offers faster user interaction than the graphic-based software does. Assuming that the user is fluent with typing, the user can enter commands faster than when using the graphical interface, because the users do not have to move their hands from the keyboard to enter different commands.

Limitations of text-based software

Many users may not find an application with a text-based interface very user-friendly. This is especially true for beginner computer users. While the user may learn how to operate the software by simply playing around or navigating through given options, a text-based system usually requires users to have a more detailed understanding of the commands. Many text-based applications have a menu or help system that shows the user some (or all) of the available options of the software.

Text-based software also has limitations on its output as well. Text-based output may not be considered very attractive to many users, and may be difficult to understand (compared to an output involving graphics). Therefore, certain operations cannot be implemented using text-based software, due to its limitation of range in its input and output.

Spoken dialog system

A **Spoken dialog system** is a dialog system delivered through voice. It has two essential components that do not exist in a text dialog system: a speech recognizer and a text-to-speech module.

Components

Speech recognition

The display of the Speech Recognition screensaver on a laptop, in which the character responds to questions, e.g. "Where are you?" or statements, e.g. "Hello."

Speech recognition (also known as **automatic speech recognition** or **computer speech recognition**) converts spoken words to text. The term "voice recognition" is sometimes used to refer to recognition systems that must be trained to a particular speaker—as is the case for most desktop recognition software. Recognizing the speaker can simplify the task of translating speech.

Speech recognition is a broader solution which refers to technology that can recognize speech without being targeted at single speaker—such as a call center system that can recognize arbitrary voices.

Speech recognition applications include voice user interfaces such as voice dialing (e.g., "Call home"), call routing (e.g., "I would like to make a collect call"), domotic appliance control, search (e.g., find a podcast where particular words were spoken), simple data entry (e.g., entering a credit card number), preparation of structured documents (e.g., a

radiology report), speech-to-text processing (e.g., word processors or emails), and aircraft (usually termed Direct Voice Input).

History

The first speech recognizer appeared in 1952 and consisted of a device for the recognition of single spoken digits Another early device was the IBM Shoebox, exhibited at the 1964 New York World's Fair. Lately there have been numerous improvements like a high speed mass transcription capabilitiy on a single system like Sonic Extractor

One of the most notable domains for the commercial application of speech recognition in the United States has been health care and in particular the work of the medical transcriptionist (MT). According to industry experts, at its inception, speech recognition (SR) was sold as a way to completely eliminate transcription rather than make the transcription process more efficient, hence it was not accepted. It was also the case that SR at that time was often technically deficient. Additionally, to be used effectively, it required changes to the ways physicians worked and documented clinical encounters, which many if not all were reluctant to do. The biggest limitation to speech recognition automating transcription, however, is seen as the software. The nature of narrative dictation is highly interpretive and often requires judgment that may be provided by a real human but not yet by an automated system. Another limitation has been the extensive amount of time required by the user and/or system provider to train the software.

A distinction in ASR is often made between "artificial syntax systems" which are usually domain-specific and "natural language processing" which is usually language-specific. Each of these types of application presents its own particular goals and challenges.

Applications

Health care

In the health care domain, even in the wake of improving speech recognition technologies, medical transcriptionists (MTs) have not yet become obsolete. The services provided may be redistributed rather than replaced.

Speech recognition can be implemented in front-end or back-end of the medical documentation process.

Front-End SR is where the provider dictates into a speech-recognition engine, the recognized words are displayed right after they are spoken, and the dictator is responsible for editing and signing off on the document. It never goes through an MT/editor.

Back-End SR or Deferred SR is where the provider dictates into a digital dictation system, and the voice is routed through a speech-recognition machine and the recognized

draft document is routed along with the original voice file to the MT/editor, who edits the draft and finalizes the report. Deferred SR is being widely used in the industry currently.

Many Electronic Medical Records (EMR) applications can be more effective and may be performed more easily when deployed in conjunction with a speech-recognition engine. Searches, queries, and form filling may all be faster to perform by voice than by using a keyboard.

Military

High-performance fighter aircraft

Substantial efforts have been devoted in the last decade to the test and evaluation of speech recognition in fighter aircraft. Of particular note are the U.S. program in speech recognition for the Advanced Fighter Technology Integration (AFTI)/F-16 aircraft (F-16 VISTA), the program in France on installing speech recognition systems on Mirage aircraft, and programs in the UK dealing with a variety of aircraft platforms. In these programs, speech recognizers have been operated successfully in fighter aircraft with applications including: setting radio frequencies, commanding an autopilot system, setting steer-point coordinates and weapons release parameters, and controlling flight displays.

Working with Swedish pilots flying in the JAS-39 Gripen cockpit, Englund (2004) found recognition deteriorated with increasing G-loads. It was also concluded that adaptation greatly improved the results in all cases and introducing models for breathing was shown to improve recognition scores significantly. Contrary to what might be expected, no effects of the broken English of the speakers were found. It was evident that spontaneous speech caused problems for the recognizer, as could be expected. A restricted vocabulary, and above all, a proper syntax, could thus be expected to improve recognition accuracy substantially.

The Eurofighter Typhoon currently in service with the UK RAF employs a speaker-dependent system, i.e. it requires each pilot to create a template. The system is not used for any safety critical or weapon critical tasks, such as weapon release or lowering of the undercarriage, but is used for a wide range of other cockpit functions. Voice commands are confirmed by visual and/or aural feedback. The system is seen as a major design feature in the reduction of pilot workload, and even allows the pilot to assign targets to himself with two simple voice commands or to any of his wingmen with only five commands.

Speaker independent systems are also being developed and are in testing for The F35 Lightning II (JSF) and the Aermacchi M346 lead in fighter trainer. These systems have produced word accuracies in excess of 98%.

Helicopters

The problems of achieving high recognition accuracy under stress and noise pertain strongly to the helicopter environment as well as to the fighter environment. The acoustic noise problem is actually more severe in the helicopter environment, not only because of the high noise levels but also because the helicopter pilot generally does not wear a facemask, which would reduce acoustic noise in the microphone. Substantial test and evaluation programs have been carried out in the past decade in speech recognition systems applications in helicopters, notably by the U.S. Army Avionics Research and Development Activity (AVRADA) and by the Royal Aerospace Establishment (RAE) in the UK. Work in France has included speech recognition in the Puma helicopter. There has also been much useful work in Canada. Results have been encouraging, and voice applications have included: control of communication radios; setting of navigation systems; and control of an automated target handover system.

As in fighter applications, the overriding issue for voice in helicopters is the impact on pilot effectiveness. Encouraging results are reported for the AVRADA tests, although these represent only a feasibility demonstration in a test environment. Much remains to be done both in speech recognition and in overall speech recognition technology, in order to consistently achieve performance improvements in operational settings.

Battle management

Battle Management command centres generally require rapid access to and control of large, rapidly changing information databases. Commanders and system operators need to query these databases as conveniently as possible, in an eyes-busy environment where much of the information is presented in a display format. Human-machine interaction by voice has the potential to be very useful in these environments. A number of efforts have been undertaken to interface commercially available isolated-word recognizers into battle management environments. In one feasibility study speech recognition equipment was tested in conjunction with an integrated information display for naval battle management applications. Users were very optimistic about the potential of the system, although capabilities were limited.

Speech understanding programs sponsored by the Defense Advanced Research Projects Agency (DARPA) in the U.S. has focused on this problem of natural speech interface. Speech recognition efforts have focused on a database of continuous speech recognition (CSR), large-vocabulary speech which is designed to be representative of the naval resource management task. Significant advances in the state-of-the-art in CSR have been achieved, and current efforts are focused on integrating speech recognition and natural language processing to allow spoken language interaction with a naval resource management system.

Training air traffic controllers

Training for air traffic controllers (ATC) represents an excellent application for speech recognition systems. Many ATC training systems currently require a person to act as a "pseudo-pilot", engaging in a voice dialog with the trainee controller, which simulates the dialog which the controller would have to conduct with pilots in a real ATC situation. Speech recognition and synthesis techniques offer the potential to eliminate the need for a person to act as pseudo-pilot, thus reducing training and support personnel. In theory, Air controller tasks are also characterized by highly structured speech as the primary output of the controller, hence reducing the difficulty of the speech recognition task should be possible. In practice this is rarely the case. The FAA document 7110.65 details the phrases that should be used by air traffic controllers. While this document gives less than 150 examples of such phrases, the number of phrases supported by one of the simulation vendors speech recognition systems is in excess of 500,000.

The USAF, USMC, US Army,US Navy and FAA as well as a number of international ATC training organizations such as the Royal Australian Air Force and Civil Aviation Authorities in Italy, Brazil, Canada are currently using ATC simulators with speech recognition from a number of different vendors.

Telephony and other domains

ASR in the field of telephony is now commonplace and in the field of computer gaming and simulation is becoming more widespread. Despite the high level of integration with word processing in general personal computing, however, ASR in the field of document production has not seen the expected increases in use.

The improvement of mobile processor speeds made feasible the speech-enabled Symbian and Windows Mobile Smartphones. Speech is used mostly as a part of User Interface, for creating pre-defined or custom speech commands. Leading software vendors in this field are: Microsoft Corporation (Microsoft Voice Command), Nuance Communications (Nuance Voice Control), Vito Technology (VITO Voice2Go), Speereo Software (Speereo Voice Translator), Digital Syphon (Sonic Messager appliance) and SVOX.

Further applications

- Automatic translation;
- Automotive speech recognition (e.g., Ford Sync);
- Telematics (e.g. vehicle Navigation Systems);
- Court reporting (Realtime Voice Writing);
- Hands-free computing: voice command recognition computer user interface;
- Home automation;
- Interactive voice response;
- Mobile telephony, including mobile email;
- Multimodal interaction;
- Pronunciation evaluation in computer-aided language learning applications;

- Robotics;
- Video games, with Tom Clancy's EndWar and Lifeline as working examples;
- Transcription (digital speech-to-text);
- Speech-to-text (transcription of speech into mobile text messages);
- Air Traffic Control Speech Recognition.

Performance

The performance of speech recognition systems is usually specified in terms of accuracy and speed. Accuracy is usually rated with word error rate (WER), whereas speed is measured with the real time factor. Other measures of accuracy include Single Word Error Rate (SWER) and Command Success Rate (CSR).

In 1982, Kurzweil Applied Intelligence and Dragon Systems released speech recognition products. By 1985, Kurzweil's software had a vocabulary of 1,000 words—if uttered one word at a time. Two years later, in 1987, its lexicon reached 20,000 words, entering the realm of human vocabularies, which range from 10,000 to 150,000 words. But recognition accuracy was only 10% in 1993. Two years later, the error rate crossed below 50%. Dragon Systems released "Naturally Speaking" in 1997 which recognized normal human speech. Progress mainly came from improved computer performance and larger source text databases. The Brown Corpus was the first major database available, containing several million words. In 2001, recognition accuracy reached its current plateau of 80%, no longer growing with data or computing power. In 2006, Google published a trillion-word corpus, while Carnegie Mellon University researchers found no significant increase in recognition accuracy.

Algorithms

Both acoustic modeling and language modeling are important parts of modern statistically-based speech recognition algorithms. Hidden Markov models (HMMs) are widely used in many systems. Language modeling has many other applications such as smart keyboard and document classification.

Hidden Markov models

Modern general-purpose speech recognition systems are based on Hidden Markov Models. These are statistical models which output a sequence of symbols or quantities. HMMs are used in speech recognition because a speech signal can be viewed as a piecewise stationary signal or a short-time stationary signal. In a short-time (e.g., 10 milliseconds)), speech can be approximated as a stationary process. Speech can be thought of as a Markov model for many stochastic purposes.

Another reason why HMMs are popular is because they can be trained automatically and are simple and computationally feasible to use. In speech recognition, the hidden Markov model would output a sequence of n-dimensional real-valued vectors (with n being a

small integer, such as 10), outputting one of these every 10 milliseconds. The vectors would consist of cepstral coefficients, which are obtained by taking a Fourier transform of a short time window of speech and decorrelating the spectrum using a cosine transform, then taking the first (most significant) coefficients. The hidden Markov model will tend to have in each state a statistical distribution that is a mixture of diagonal covariance Gaussians which will give a likelihood for each observed vector. Each word, or (for more general speech recognition systems), each phoneme, will have a different output distribution; a hidden Markov model for a sequence of words or phonemes is made by concatenating the individual trained hidden Markov models for the separate words and phonemes.

Described above are the core elements of the most common, HMM-based approach to speech recognition. Modern speech recognition systems use various combinations of a number of standard techniques in order to improve results over the basic approach described above. A typical large-vocabulary system would need context dependency for the phonemes (so phonemes with different left and right context have different realizations as HMM states); it would use cepstral normalization to normalize for different speaker and recording conditions; for further speaker normalization it might use vocal tract length normalization (VTLN) for male-female normalization and maximum likelihood linear regression (MLLR) for more general speaker adaptation. The features would have so-called delta and delta-delta coefficients to capture speech dynamics and in addition might use heteroscedastic linear discriminant analysis (HLDA); or might skip the delta and delta-delta coefficients and use splicing and an LDA-based projection followed perhaps by heteroscedastic linear discriminant analysis or a global semitied covariance transform (also known as maximum likelihood linear transform, or MLLT). Many systems use so-called discriminative training techniques which dispense with a purely statistical approach to HMM parameter estimation and instead optimize some classification-related measure of the training data. Examples are maximum mutual information (MMI), minimum classification error (MCE) and minimum phone error (MPE).

Decoding of the speech (the term for what happens when the system is presented with a new utterance and must compute the most likely source sentence) would probably use the Viterbi algorithm to find the best path, and here there is a choice between dynamically creating a combination hidden Markov model which includes both the acoustic and language model information, or combining it statically beforehand (the finite state transducer, or FST, approach).

Dynamic time warping (DTW)-based speech recognition

Dynamic time warping is an approach that was historically used for speech recognition but has now largely been displaced by the more successful HMM-based approach. Dynamic time warping is an algorithm for measuring similarity between two sequences which may vary in time or speed. For instance, similarities in walking patterns would be detected, even if in one video the person was walking slowly and if in another they were walking more quickly, or even if there were accelerations and decelerations during the

course of one observation. DTW has been applied to video, audio, and graphics – indeed, any data which can be turned into a linear representation can be analyzed with DTW.

A well known application has been automatic speech recognition, to cope with different speaking speeds. In general, it is a method that allows a computer to find an optimal match between two given sequences (e.g. time series) with certain restrictions, i.e. the sequences are "warped" non-linearly to match each other. This sequence alignment method is often used in the context of hidden Markov models.

Further information

Popular speech recognition conferences held each year or two include SpeechTEK and SpeechTEK Europe, ICASSP, Eurospeech/ICSLP (now named Interspeech) and the IEEE ASRU. Conferences in the field of Natural language processing, such as ACL, NAACL, EMNLP, and HLT, are beginning to include papers on speech processing. Important journals include the IEEE Transactions on Speech and Audio Processing (now named IEEE Transactions on Audio, Speech and Language Processing), Computer Speech and Language, and Speech Communication. Books like "Fundamentals of Speech Recognition" by Lawrence Rabiner can be useful to acquire basic knowledge but may not be fully up to date (1993). Another good source can be "Statistical Methods for Speech Recognition" by Frederick Jelinek and "Spoken Language Processing (2001)" by Xuedong Huang etc. More up to date is "Computer Speech", by Manfred R. Schroeder, second edition published in 2004. The recently updated textbook of "Speech and Language Processing (2008)" by Jurafsky and Martin presents the basics and the state of the art for ASR. A good insight into the techniques used in the best modern systems can be gained by paying attention to government sponsored evaluations such as those organised by DARPA (the largest speech recognition-related project ongoing as of 2007 is the GALE project, which involves both speech recognition and translation components).

In terms of freely available resources, Carnegie Mellon University's SPHINX toolkit is one place to start to both learn about speech recognition and to start experimenting. Another resource (free as in free beer, not free software) is the HTK book (and the accompanying HTK toolkit). The AT&T libraries GRM library, and DCD library are also general software libraries for large-vocabulary speech recognition.

A useful review of the area of robustness in ASR is provided by Junqua and Haton (1995).

People with disabilities

People with disabilities can benefit from speech recognition programs. Speech recognition is especially useful for people who have difficulty using their hands, ranging from mild repetitive stress injuries to involved disabilities that preclude using conventional computer input devices. In fact, people who used the keyboard a lot and developed RSI became an urgent early market for speech recognition. Speech recognition

is used in deaf telephony, such as voicemail to text, relay services, and captioned telephone. Individuals with learning disabilities who have problems with thought-to-paper communication (essentially they think of an idea but it is processed incorrectly causing it to end up differently on paper) can benefit from the software.

Current research and funding

Measuring progress in speech recognition performance is difficult and controversial. Some speech recognition tasks are much more difficult than others. Word error rates on some tasks are less than 1%. On others they can be as high as 50%. Sometimes it even appears that performance is going backwards as researchers undertake harder tasks that have higher error rates.

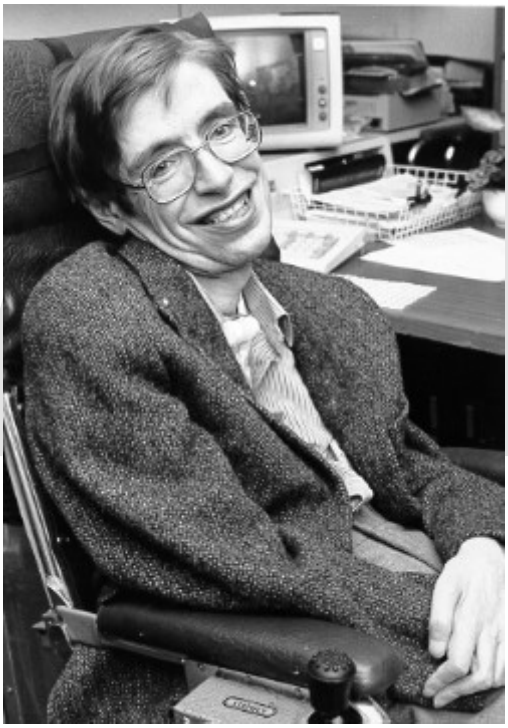
Because progress is slow and is difficult to measure, there is some perception that performance has plateaued and that funding has dried up or shifted priorities. Such perceptions are not new. In 1969, John Pierce wrote an open letter that did cause much funding to dry up for several years. In 1993 there was a strong feeling that performance had plateaued and there were workshops dedicated to the issue. However, in the 1990s funding continued more or less uninterrupted and performance continued slowly but steadily to improve.

For the past thirty years, speech recognition research has been characterized by the steady accumulation of small incremental improvements. There has also been a trend continually to change focus to more difficult tasks due both to progress in speech recognition performance and to the availability of faster computers. In particular, this shifting to more difficult tasks has characterized DARPA funding of speech recognition since the 1980s. In the last decade it has continued with the EARS project, which undertook recognition of Mandarin and Arabic in addition to English, and the GALE project, which focused solely on Mandarin and Arabic and required translation simultaneously with speech recognition.

Commercial research and other academic research also continue to focus on increasingly difficult problems. One key area is to improve robustness of speech recognition performance, not just robustness against noise but robustness against any condition that causes a major degradation in performance. Another key area of research is focused on an opportunity rather than a problem. This research attempts to take advantage of the fact that in many applications there is a large quantity of speech data available, up to millions of hours. It is too expensive to have humans transcribe such large quantities of speech, so the research focus is on developing new methods of machine learning that can effectively utilize large quantities of unlabeled data. Another area of research is better understanding of human capabilities and to use this understanding to improve machine recognition performance.

Chapter- 2

Speech Synthesis



Stephen Hawking is one of the most famous people using speech synthesis to communicate

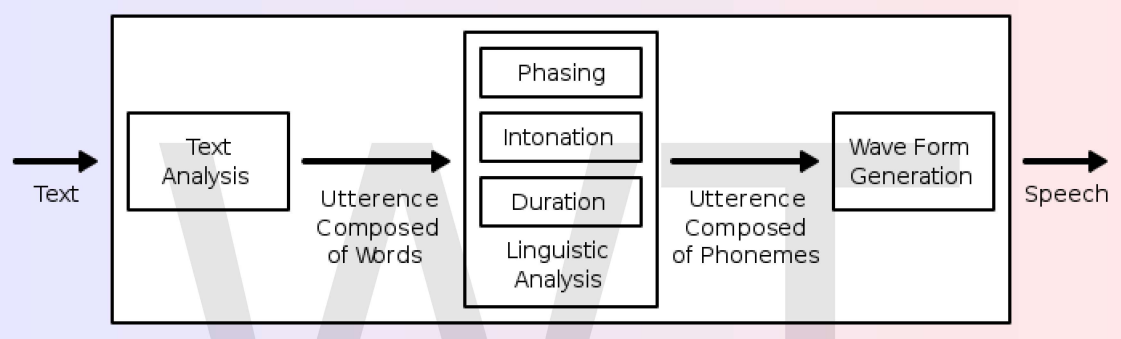
Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a **speech synthesizer**, and can be implemented in software or hardware. A **text-to-speech (TTS)** system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that

stores phones or diphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written works on a home computer. Many computer operating systems have included speech synthesizers since the early 1980s.

Overview of text processing



Overview of a typical TTS system

A text-to-speech system (or "engine") is composed of two parts: a front-end and a back-end. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called *text normalization*, *pre-processing*, or *tokenization*. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called *text-to-phoneme* or *grapheme-to-phoneme* conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end—often referred to as the *synthesizer*—then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the *target prosody* (pitch contour, phoneme durations), which is then imposed on the output speech.

History

Long before electronic signal processing was invented, there were those who tried to build machines to create human speech. Some early legends of the existence of "speaking heads" involved Gerbert of Aurillac (d. 1003 AD), Albertus Magnus (1198–1280), and Roger Bacon (1214–1294).

In 1779, the Danish scientist Christian Kratzenstein, working at the Russian Academy of Sciences, built models of the human vocal tract that could produce the five long vowel sounds. This was followed by the bellows-operated "acoustic-mechanical speech machine" by Wolfgang von Kempelen of Vienna, Austria, described in a 1791 paper. This machine added models of the tongue and lips, enabling it to produce consonants as well as vowels. In 1837, Charles Wheatstone produced a "speaking machine" based on von Kempelen's design, and in 1857, M. Faber built the "Euphonia". Wheatstone's design was resurrected in 1923 by Paget.

In the 1930s, Bell Labs developed the VOCODER, a keyboard-operated electronic speech analyzer and synthesizer that was said to be clearly intelligible. Homer Dudley refined this device into the VODER, which he exhibited at the 1939 New York World's Fair.

The Pattern playback was built by Dr. Franklin S. Cooper and his colleagues at Haskins Laboratories in the late 1940s and completed in 1950. There were several different versions of this hardware device but only one currently survives. The machine converts pictures of the acoustic patterns of speech in the form of a spectrogram back into sound. Using this device, Alvin Liberman and colleagues were able to discover acoustic cues for the perception of phonetic segments (consonants and vowels).

Dominant systems in the 1980s and 1990s were the MITalk system, based largely on the work of Dennis Klatt at MIT, and the Bell Labs system; the latter was one of the first multilingual language-independent systems, making extensive use of Natural Language Processing methods.

Early electronic speech synthesizers sounded robotic and were often barely intelligible. The quality of synthesized speech has steadily improved, but output from contemporary speech synthesis systems is still clearly distinguishable from actual human speech.

As the cost-performance ratio causes speech synthesizers to become cheaper and more accessible to the people, more people will benefit from the use of text-to-speech programs.

Electronic devices

The first computer-based speech synthesis systems were created in the late 1950s, and the first complete text-to-speech system was completed in 1968. In 1961, physicist John Larry Kelly, Jr and colleague Louis Gerstman used an IBM 704 computer to synthesize speech, an event among the most prominent in the history of Bell Labs. Kelly's voice recorder synthesizer (vocoder) recreated the song "Daisy Bell", with musical accompaniment from Max Mathews. Coincidentally, Arthur C. Clarke was visiting his friend and colleague John Pierce at the Bell Labs Murray Hill facility. Clarke was so impressed by the demonstration that he used it in the climactic scene of his screenplay for his novel *2001: A Space Odyssey*, where the HAL 9000 computer sings the same song as it is being put to sleep by astronaut Dave Bowman. Despite the success of purely

electronic speech synthesis, research is still being conducted into mechanical speech synthesizers.

Handheld electronics featuring speech synthesis began emerging in the 1970s. One of the first was the Telesensory Systems Inc. (TSI) *Speech+* portable calculator for the blind in 1976. Other devices were produced primarily for educational purposes, such as Speak & Spell, produced by Texas Instruments in 1978. The first multi-player game using voice synthesis was *Milton* from Milton Bradley Company, which produced the device in 1980.

Synthesizer technologies

The most important qualities of a speech synthesis system are *naturalness* and *intelligibility*. Naturalness describes how closely the output sounds like human speech, while intelligibility is the ease with which the output is understood. The ideal speech synthesizer is both natural and intelligible. Speech synthesis systems usually try to maximize both characteristics.

The two primary technologies for generating synthetic speech waveforms are *concatenative synthesis* and *formant synthesis*. Each technology has strengths and weaknesses, and the intended uses of a synthesis system will typically determine which approach is used.

Concatenative synthesis

Concatenative synthesis is based on the concatenation (or stringing together) of segments of recorded speech. Generally, concatenative synthesis produces the most natural-sounding synthesized speech. However, differences between natural variations in speech and the nature of the automated techniques for segmenting the waveforms sometimes result in audible glitches in the output. There are three main sub-types of concatenative synthesis.

Unit selection synthesis

Unit selection synthesis uses large databases of recorded speech. During database creation, each recorded utterance is segmented into some or all of the following: individual phones, diphones, half-phones, syllables, morphemes, words, phrases, and sentences. Typically, the division into segments is done using a specially modified speech recognizer set to a "forced alignment" mode with some manual correction afterward, using visual representations such as the waveform and spectrogram. An index of the units in the speech database is then created based on the segmentation and acoustic parameters like the fundamental frequency (pitch), duration, position in the syllable, and neighboring phones. At runtime, the desired target utterance is created by determining the best chain of candidate units from the database (unit selection). This process is typically achieved using a specially weighted decision tree.

Unit selection provides the greatest naturalness, because it applies only a small amount of digital signal processing (DSP) to the recorded speech. DSP often makes recorded speech sound less natural, although some systems use a small amount of signal processing at the point of concatenation to smooth the waveform. The output from the best unit-selection systems is often indistinguishable from real human voices, especially in contexts for which the TTS system has been tuned. However, maximum naturalness typically require unit-selection speech databases to be very large, in some systems ranging into the gigabytes of recorded data, representing dozens of hours of speech. Also, unit selection algorithms have been known to select segments from a place that results in less than ideal synthesis (e.g. minor words become unclear) even when a better choice exists in the database.

Diphone synthesis

Diphone synthesis uses a minimal speech database containing all the diphones (sound-to-sound transitions) occurring in a language. The number of diphones depends on the phonotactics of the language: for example, Spanish has about 800 diphones, and German about 2500. In diphone synthesis, only one example of each diphone is contained in the speech database. At runtime, the target prosody of a sentence is superimposed on these minimal units by means of digital signal processing techniques such as linear predictive coding, PSOLA or MBROLA. The quality of the resulting speech is generally worse than that of unit-selection systems, but more natural-sounding than the output of formant synthesizers. Diphone synthesis suffers from the sonic glitches of concatenative synthesis and the robotic-sounding nature of formant synthesis, and has few of the advantages of either approach other than small size. As such, its use in commercial applications is declining, although it continues to be used in research because there are a number of freely available software implementations.

Domain-specific synthesis

Domain-specific synthesis concatenates prerecorded words and phrases to create complete utterances. It is used in applications where the variety of texts the system will output is limited to a particular domain, like transit schedule announcements or weather reports. The technology is very simple to implement, and has been in commercial use for a long time, in devices like talking clocks and calculators. The level of naturalness of these systems can be very high because the variety of sentence types is limited, and they closely match the prosody and intonation of the original recordings.

Because these systems are limited by the words and phrases in their databases, they are not general-purpose and can only synthesize the combinations of words and phrases with which they have been preprogrammed. The blending of words within naturally spoken language however can still cause problems unless the many variations are taken into account. For example, in non-rhotic dialects of English the *"r"* in words like *"clear"* is usually only pronounced when the following word has a vowel as its first letter. Likewise in French, many final consonants become no longer silent if followed by a word that begins with a vowel, an effect called liaison. This alternation cannot be reproduced by a

simple word-concatenation system, which would require additional complexity to be context-sensitive.

Formant synthesis

Formant synthesis does not use human speech samples at runtime. Instead, the synthesized speech output is created using additive synthesis and an acoustic model (physical modelling synthesis). Parameters such as fundamental frequency, voicing, and noise levels are varied over time to create a waveform of artificial speech. This method is sometimes called *rules-based synthesis*; however, many concatenative systems also have rules-based components. Many systems based on formant synthesis technology generate artificial, robotic-sounding speech that would never be mistaken for human speech. However, maximum naturalness is not always the goal of a speech synthesis system, and formant synthesis systems have advantages over concatenative systems. Formant-synthesized speech can be reliably intelligible, even at very high speeds, avoiding the acoustic glitches that commonly plague concatenative systems. High-speed synthesized speech is used by the visually impaired to quickly navigate computers using a screen reader. Formant synthesizers are usually smaller programs than concatenative systems because they do not have a database of speech samples. They can therefore be used in embedded systems, where memory and microprocessor power are especially limited. Because formant-based systems have complete control of all aspects of the output speech, a wide variety of prosodies and intonations can be output, conveying not just questions and statements, but a variety of emotions and tones of voice.

Examples of non-real-time but highly accurate intonation control in formant synthesis include the work done in the late 1970s for the Texas Instruments toy Speak & Spell, and in the early 1980s Sega arcade machines. and in many Atari, Inc. arcade games using the TMS5220 LPC Chips. Creating proper intonation for these projects was painstaking, and the results have yet to be matched by real-time text-to-speech interfaces.

Articulatory synthesis

Articulatory synthesis refers to computational techniques for synthesizing speech based on models of the human vocal tract and the articulation processes occurring there. The first articulatory synthesizer regularly used for laboratory experiments was developed at Haskins Laboratories in the mid-1970s by Philip Rubin, Tom Baer, and Paul Mermelstein. This synthesizer, known as ASY, was based on vocal tract models developed at Bell Laboratories in the 1960s and 1970s by Paul Mermelstein, Cecil Coker, and colleagues.

Until recently, articulatory synthesis models have not been incorporated into commercial speech synthesis systems. A notable exception is the NeXT-based system originally developed and marketed by Trillium Sound Research, a spin-off company of the University of Calgary, where much of the original research was conducted. Following the demise of the various incarnations of NeXT (started by Steve Jobs in the late 1980s and merged with Apple Computer in 1997), the Trillium software was published under the

GNU General Public License, with work continuing as gnuspeech. The system, first marketed in 1994, provides full articulatory-based text-to-speech conversion using a waveguide or transmission-line analog of the human oral and nasal tracts controlled by Carré's "distinctive region model".

HMM-based synthesis

HMM-based synthesis is a synthesis method based on hidden Markov models, also called Statistical Parametric Synthesis. In this system, the frequency spectrum (vocal tract), fundamental frequency (vocal source), and duration (prosody) of speech are modeled simultaneously by HMMs. Speech waveforms are generated from HMMs themselves based on the maximum likelihood criterion.

Sinewave synthesis

Sinewave synthesis is a technique for synthesizing speech by replacing the formants (main bands of energy) with pure tone whistles.

Challenges

Text normalization challenges

The process of normalizing text is rarely straightforward. Texts are full of heteronyms, numbers, and abbreviations that all require expansion into a phonetic representation. There are many spellings in English which are pronounced differently based on context. For example, "My latest project is to learn how to better project my voice" contains two pronunciations of "project".

Most text-to-speech (TTS) systems do not generate semantic representations of their input texts, as processes for doing so are not reliable, well understood, or computationally effective. As a result, various heuristic techniques are used to guess the proper way to disambiguate homographs, like examining neighboring words and using statistics about frequency of occurrence.

Recently TTS systems have begun to use HMMs (discussed above) to generate "parts of speech" to aid in disambiguating homographs. This technique is quite successful for many cases such as whether "read" should be pronounced as "red" implying past tense, or as "reed" implying present tense. Typical error rates when using HMMs in this fashion are usually below five percent. These techniques also work well for most European languages, although access to required training corpora is frequently difficult in these languages.

Deciding how to convert numbers is another problem that TTS systems have to address. It is a simple programming challenge to convert a number into words (at least in English), like "1325" becoming "one thousand three hundred twenty-five." However, numbers occur in many different contexts; "1325" may also be read as "one three two five",

"thirteen twenty-five" or "thirteen hundred and twenty five". A TTS system can often infer how to expand a number based on surrounding words, numbers, and punctuation, and sometimes the system provides a way to specify the context if it is ambiguous. Roman numerals can also be read differently depending on context. For example "Henry VIII" reads as "Henry the Eighth", while "Chapter VIII" reads as "Chapter Eight".

Similarly, abbreviations can be ambiguous. For example, the abbreviation "in" for "inches" must be differentiated from the word "in", and the address "12 St John St." uses the same abbreviation for both "Saint" and "Street". TTS systems with intelligent front ends can make educated guesses about ambiguous abbreviations, while others provide the same result in all cases, resulting in nonsensical (and sometimes comical) outputs.

Text-to-phoneme challenges

Speech synthesis systems use two basic approaches to determine the pronunciation of a word based on its spelling, a process which is often called text-to-phoneme or grapheme-to-phoneme conversion (phoneme is the term used by linguists to describe distinctive sounds in a language). The simplest approach to text-to-phoneme conversion is the dictionary-based approach, where a large dictionary containing all the words of a language and their correct pronunciations is stored by the program. Determining the correct pronunciation of each word is a matter of looking up each word in the dictionary and replacing the spelling with the pronunciation specified in the dictionary. The other approach is rule-based, in which pronunciation rules are applied to words to determine their pronunciations based on their spellings. This is similar to the "sounding out", or synthetic phonics, approach to learning reading.

Each approach has advantages and drawbacks. The dictionary-based approach is quick and accurate, but completely fails if it is given a word which is not in its dictionary. As dictionary size grows, so too does the memory space requirements of the synthesis system. On the other hand, the rule-based approach works on any input, but the complexity of the rules grows substantially as the system takes into account irregular spellings or pronunciations. (Consider that the word "of" is very common in English, yet is the only word in which the letter "f" is pronounced [v].) As a result, nearly all speech synthesis systems use a combination of these approaches.

Languages with a phonemic orthography have a very regular writing system, and the prediction of the pronunciation of words based on their spellings is quite successful. Speech synthesis systems for such languages often use the rule-based method extensively, resorting to dictionaries only for those few words, like foreign names and borrowings, whose pronunciations are not obvious from their spellings. On the other hand, speech synthesis systems for languages like English, which have extremely irregular spelling systems, are more likely to rely on dictionaries, and to use rule-based methods only for unusual words, or words that aren't in their dictionaries.

Evaluation challenges

The consistent evaluation of speech synthesis systems may be difficult because of a lack of universally agreed objective evaluation criteria. Different organizations often use different speech data. The quality of speech synthesis systems also depends to a large degree on the quality of the production technique (which may involve analogue or digital recording) and on the facilities used to replay the speech. Evaluating speech synthesis systems has therefore often been compromised by differences between production techniques and replay facilities.

Recently, however, some researchers have started to evaluate speech synthesis systems using a common speech dataset.

Prosodics and emotional content

A recent study reported in the journal "**Speech Communication**" by Amy Drahota and colleagues at the University of Portsmouth, UK, reported that listeners to voice recordings could determine, at better than chance levels, whether or not the speaker was smiling. It was suggested that identification of the vocal features which signal emotional content may be used to help make synthesized speech sound more natural.

Dedicated hardware

- Votrax
  - SC-01A (analog formant)
  - SC-02 / SSI-263 / "Arctic 263"
- General Instruments SP0256-AL2 (CTS256A-AL2, MEA8000)
- Magnevation SpeakJet
- Savage Innovations SoundGin
- National Semiconductor DT1050 Digitalker (Mozer)
- Silicon Systems SSI 263 (analog formant)
- Texas Instruments LPC Speech Chips
  - TMS5110A
  - TMS5200
- Oki Semiconductor
  - ML22825 (ADPCM)
  - ML22573 (HQADPCM)
- Toshiba T6721A
- Philips PCF8200
- TextSpeak Embedded TTS Modules

Computer operating systems or outlets with speech synthesis

Atari

Arguably, the first speech system integrated into an operating system was the 1400XL/1450XL personal computers designed by Atari, Inc. using the Votrax SC01 chip in 1983. The 1400XL/1450XL computers used a Finite State Machine to enable World English Spelling text-to-speech synthesis. Unfortunately, the 1400XL/1450XL personal computers never shipped in quantity.

The Atari ST computers were sold with "stspeech.tos" on floppy disk.

Apple

The first speech system integrated into an operating system that shipped in quantity was Apple Computer's MacInTalk in 1984. Since the 1980s Macintosh Computers offered text to speech capabilities through The MacinTalk software. In the early 1990s Apple expanded its capabilities offering system wide text-to-speech support. With the introduction of faster PowerPC-based computers they included higher quality voice sampling. Apple also introduced speech recognition into its systems which provided a fluid command set. More recently, Apple has added sample-based voices. Starting as a curiosity, the speech system of Apple Macintosh has evolved into a fully-supported program, PlainTalk, for people with vision problems. VoiceOver was for the first time featured in Mac OS X Tiger (10.4). During 10.4 (Tiger) & first releases of 10.5 (Leopard) there was only one standard voice shipping with Mac OS X. Starting with 10.6 (Snow Leopard), the user can choose out of a wide range list of multiple voices. VoiceOver voices feature the taking of realistic-sounding breaths between sentences, as well as improved clarity at high read rates over PlainTalk. Mac OS X also includes say, a command-line based application that converts text to audible speech. The AppleScript Standard Additions includes a say verb that allows a script to use any of the installed voices and to control the pitch, speaking rate and modulation of the spoken text.

AmigaOS

The second operating system with advanced speech synthesis capabilities was AmigaOS, introduced in 1985. The voice synthesis was licensed by Commodore International from a third-party software house (Don't Ask Software, now Softvoice, Inc.) and it featured a complete system of voice emulation, with both male and female voices and "stress" indicator markers, made possible by advanced features of the Amiga hardware audio chipset. It was divided into a narrator device and a translator library. Amiga Speak Handler featured a text-to-speech translator. AmigaOS considered speech synthesis a virtual hardware device, so the user could even redirect console output to it. Some Amiga programs, such as word processors, made extensive use of the speech system.

Microsoft Windows

Modern Windows systems use SAPI4- and SAPI5-based speech systems that include a speech recognition engine (SRE). SAPI 4.0 was available on Microsoft-based operating systems as a third-party add-on for systems like Windows 95 and Windows 98. Windows 2000 added a speech synthesis program called Narrator, directly available to users. All Windows-compatible programs could make use of speech synthesis features, available through menus once installed on the system. Microsoft Speech Server is a complete package for voice synthesis and recognition, for commercial applications such as call centers.

Text-to-Speech (TTS) capabilities for a computer refers to the ability to play back text in a spoken voice. **TTS** is the ability of the operating system to play back printed text as spoken words.

An internal (installed with the operating system) driver (called a TTS engine): recognizes the text and using a synthesized voice (chosen from several pre-generated voices) speaks the written text. Additional engines (often use a certain jargon or vocabulary) are also available through third-party manufacturers.

Android

Version 1.6 of Android added support for speech synthesis (TTS).

Internet

The most recent TTS development in the web browser, is the JavaScript Text to Speech work of Yury Delendik, which ports the Flite C engine to pure JavaScript. This allows web pages to convert text to audio using HTML5 technology. The ability to use Yury's TTS port currently requires a custom browser build that uses Mozilla's Audio-Data-API. However, much work is being done in the context of the W3C to move this technology into the mainstream browser market through the W3C Audio Incubator Group with the involvement of The BBC and Google Inc.

Currently, there are a number of applications, plugins and gadgets that can read messages directly from an e-mail client and web pages from a web browser or Google Toolbar such as Text-to-voice which is an add-on to Firefox . Some specialized software can narrate RSS-feeds. On one hand, online RSS-narrators simplify information delivery by allowing users to listen to their favourite news sources and to convert them to podcasts. On the other hand, on-line RSS-readers are available on almost any PC connected to the Internet. Users can download generated audio files to portable devices, e.g. with a help of podcast receiver, and listen to them while walking, jogging or commuting to work.

A growing field in internet based TTS is web-based assistive technology, e.g. 'Browsealoud' from a UK company and Readspeaker. It can deliver TTS functionality to anyone (for reasons of accessibility, convenience, entertainment or information) with

access to a web browser. Additionally SPEAK.TO.ME from Oxford Information Laboratories is capable of delivering text to speech through any browser without the need to download any special applications, and includes smart delivery technology to ensure only what is seen is spoken and the content is logically pathed.

Others

- Some models of Texas Instruments home computers produced in 1979 and 1981 (Texas Instruments TI-99/4 and TI-99/4A) were capable of text-to-phoneme synthesis or reciting complete words and phrases (text-to-dictionary), using a very popular Speech Synthesizer peripheral. TI used a proprietary codec to embed complete spoken phrases into applications, primarily video games.
- IBM's OS/2 Warp 4 included VoiceType, a precursor to IBM ViaVoice.
- Systems that operate on free and open source software systems including Linux are various, and include open-source programs such as the Festival Speech Synthesis System which uses diphone-based synthesis (and can use a limited number of MBROLA voices), and gnuspeech which uses articulatory synthesis from the Free Software Foundation.
- Companies which developed speech synthesis systems but which are no longer in this business include BeST Speech (bought by L&H), Eloquent Technology (bought by SpeechWorks), Lernout & Hauspie (bought by Nuance), SpeechWorks (bought by Nuance), Rhetorical Systems (bought by Nuance).

Speech synthesis markup languages

A number of markup languages have been established for the rendition of text as speech in an XML-compliant format. The most recent is Speech Synthesis Markup Language (SSML), which became a W3C recommendation in 2004. Older speech synthesis markup languages include Java Speech Markup Language (JSML) and SABLE. Although each of these was proposed as a standard, none of them has been widely adopted.

Speech synthesis markup languages are distinguished from dialogue markup languages. VoiceXML, for example, includes tags related to speech recognition, dialogue management and touchtone dialing, in addition to text-to-speech markup.

Applications

Speech synthesis has long been a vital assistive technology tool and its application in this area is significant and widespread. It allows environmental barriers to be removed for people with a wide range of disabilities. The longest application has been in the use of screen readers for people with visual impairment, but text-to-speech systems are now commonly used by people with dyslexia and other reading difficulties as well as by pre-literate children. They are also frequently employed to aid those with severe speech impairment usually through a dedicated voice output communication aid.

Sites such as Ananova and YAKiToMe! have used speech synthesis to convert written news to audio content, which can be used for mobile applications.

Speech synthesis techniques are used as well in the entertainment productions such as games, animation and similar. In 2007, Animo Limited announced the development of a software application package based on its speech synthesis software FineSpeech, explicitly geared towards customers in the entertainment industries, able to generate narration and lines of dialogue according to user specifications. The application reached maturity in 2008, when NEC Biglobe announced a web service that allows users to create phrases from the voices of Code Geass: Lelouch of the Rebellion R2 characters.

TTS applications such as YAKiToMe! and Speakonia are often used to add synthetic voices to YouTube videos for comedic effect, as in Barney Bunch videos. YAKiToMe! is also used to convert entire books for personal podcasting purposes, RSS feeds and web pages for news stories, and educational texts for enhanced learning.

Software such as Vocaloid can generate singing voices via lyrics and melody. This is also the aim of the Singing Computer project (which uses GNU LilyPond and Festival) to help blind people check their lyric input.

Next to these applications is the use of text to speech software also popular in Interactive Voice Response systems, often in combination with speech recognition.

Dialog manager

A **dialog manager** is the core component of a dialog system. It maintains the history of the dialog, adopts certain dialog strategy (see below), retrieve the content (stored in files or databases), and decides on the best response to the user. The dialog manager maintains the dialog flow.

The design of the dialog manager evolves over time.

- finite-state machine
- frame-based: The system has several slots to be filled. The slots can be filled in any order. This supports mixed-initiative dialog strategy.
- information-state based

Dialog Strategy

The dialog flow can have the following strategies:

- **System-initiative dialog**: The system is in control to guide the dialog at each step.

- **Mixed-initiative dialog**: Users can barge in and change the dialog direction. The system follows the user request, but tries to direct the user back the original course. This is the most commonly used dialog strategy in today's dialog systems.
- **User-initiative dialog**: The user takes lead, and the system respond to whatever the user directs.

Knowledge base

A **knowledge base** (abbreviated **KB**, **kb** or Δ) is a special kind of database for knowledge management, providing the means for the computerized collection, organization, and retrieval of knowledge. Also a collection of data representing related experiences, their results are related to their problems and solutions.

Types

The knowledge based systems are artificial intelligent tools working in a narrow domain to provide intelligent decisions with justification. Knowledge is acquired and represented using various knowledge representation techniques rules, frames and scripts. The basic advantages offered by such system are documentation of knowledge, intelligent decision support, self learning, reasoning and explanation.

Knowledge bases are categorized into two major types:

- **Machine-readable knowledge bases** store knowledge in a computer-readable form, usually for the purpose of having automated deductive reasoning applied to them. They contain a set of data, often in the form of rules that describe the knowledge in a logically consistent manner. An ontology can define the structure of stored data - what types of entities are recorded and what their relationships are. Logical operators, such as *And* (conjunction), *Or* (disjunction), *material implication* and *negation* may be used to build it up from simpler pieces of information. Consequently, classical deduction can be used to reason about the knowledge in the knowledge base. Some machine-readable knowledge bases are used with artificial intelligence, for example as part of an expert system that focuses on a domain like prescription drugs or customs law. Such knowledge bases are also used by the semantic web.

- **Human-readable knowledge bases** are designed to allow people to retrieve and use the knowledge they contain. They are commonly used to complement a help desk or for sharing information among employees within an organization. They might store troubleshooting information, articles, white papers, user manuals, knowledge tags, or answers to frequently asked questions. Typically, a search engine is used to locate information in the system, or users may browse through a classification scheme.

A text based system that can include groups of documents including hyperlinks between them is known as **Hypertext Systems**. Hypertext systems support the decision process by relieving the user of the significant effort it takes to relate and remember things." Knowledge bases can exist on both computers and mobile phones in a hypertext format.

- **Knowledge base analysis and design** (also known as KBAD) is an approach that allows people to conduct analysis and design in a way that results in a knowledge base, which can later be used to make informative decisions. This approach was first implemented by Dr. Steven H. Dam.

Types of Systems

Spoken dialog systems vary in their complexity. In general, the informational dialog system is the most simple one, and it gradually goes up to the most complex type, the entertainment dialog system.

- Informational: Examples include stock quotes, credit card and bank information. This also include Google 411 and Tellme directory assistance.
- Transactional: Examples include magazine subscription, and bank transation.
- Diagnostic: Technical support. Examples include Speech Cycle's LevelOne Broadband Agent, which diagnoses Internet access problem.
- Educational/Tutoring: Tutoring system for physics and language learning.
- Entertainment and chatting

WT

Chapter- 3

Graphical User Interface

A screenshot of a modern GUI (KDE Plasma Desktop).

The Macintosh 128K was the first commercially successful personal computer to use a graphical user interface

A **graphical user interface** (**GUI**), often pronounced *gooey*, is a type of user interface that allows users to interact with programs in more ways than typing such as computers; hand-held devices such as MP3 players, portable media players or gaming devices; household appliances and office equipment with images rather than text commands. A *GUI* offers graphical icons, and visual indicators, as opposed to text-based interfaces, typed command labels or text navigation to fully represent the information and actions available to a user. The actions are usually performed through direct manipulation of the graphical elements.

The term *GUI* is historically restricted to the scope of two-dimensional display screens with display resolutions capable of describing generic information, in the tradition of the computer science research at the Palo Alto Research Center (PARC). The term *GUI* earlier might have been applicable to other high-resolution types of interfaces that are non-generic, such as videogames, or not restricted to flat screens, like volumetric displays.

History

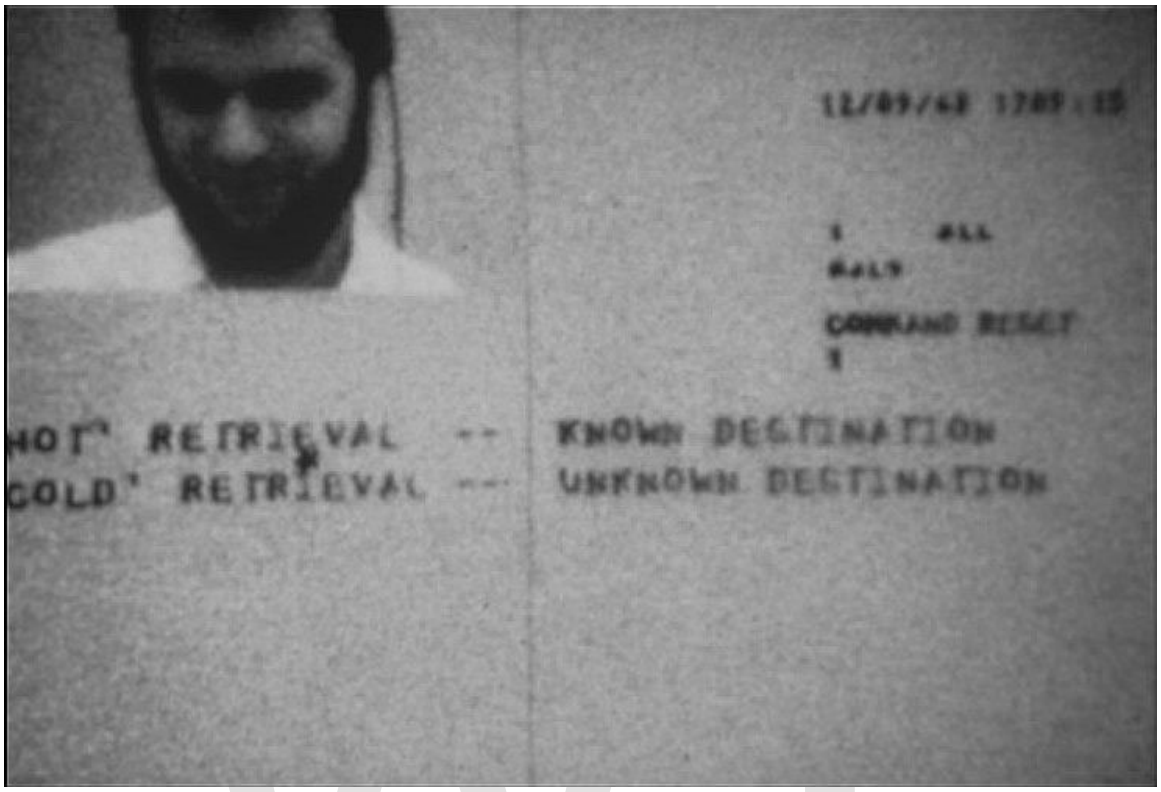
The **graphical user interface**, understood as the use of graphic icons and a pointing device to control a computer, has a four decade history of incremental refinements built on some constant core principles. Several vendors have created their own windowing systems based on independent code, but with basic elements in common that define the WIMP paradigm. There have been important technological achievements, and enhancements to the general interaction were given in small steps over previous systems. There have been a few significant breakthroughs in terms of use, but the same organizational metaphors and interaction idioms are still in use. Although many GUI Operating Systems are operated by using a mouse, the keyboard can also be used by using keyboard shortcuts or arrow keys.

Initial developments

Early dynamic information devices such as radar displays, where input devices were used for direct control of computer-created data, set the basis for later improvements of graphical interfaces.

The concept of a windowing system was introduced by the first real-time graphic display systems for computers: the SAGE Project and Ivan Sutherland's Sketchpad.

Augmentation of Human Intellect (NLS)



Videoconferencing on NLS (1968)

Doug Engelbart's Augmentation of Human Intellect project at SRI in the 1960s developed the On-Line System (NLS), which incorporated a mouse-driven cursor and multiple windows used to work on hypertext. Engelbart had been inspired, in part, by the memex desk-based information machine suggested by Vannevar Bush in 1945. Much of the early research was based on how children learn.

Xerox PARC

The Xerox Star Workstation introduced the first GUI Operating systems.

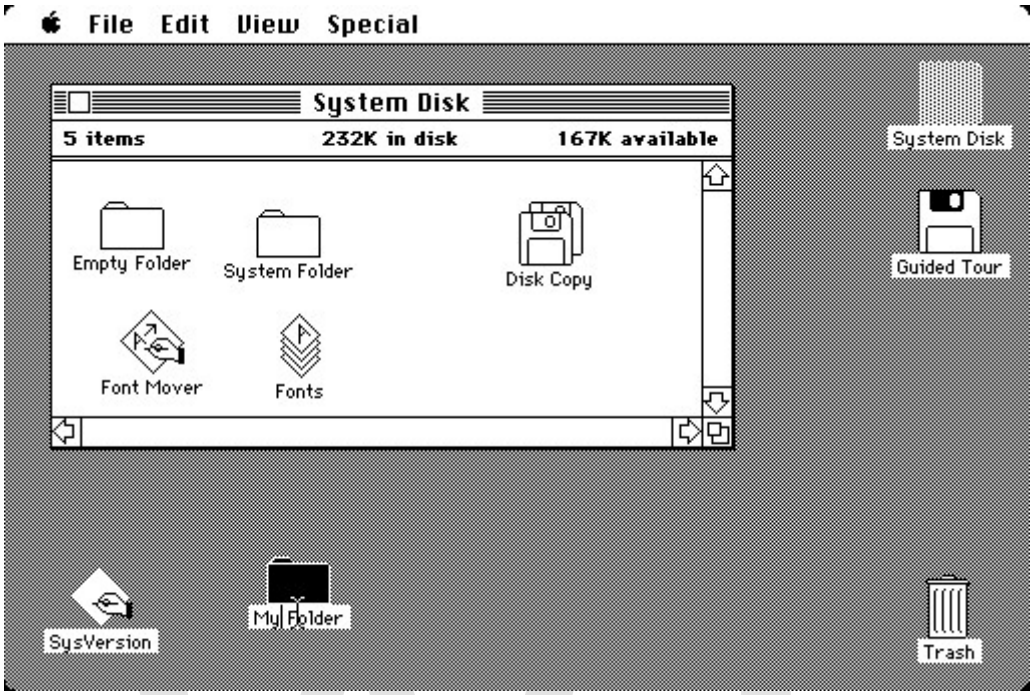
Engelbart's work directly led to the advances at Xerox PARC. Several people went from SRI to Xerox PARC in the early 1970s. In 1973 Xerox PARC developed the Alto personal computer. It had a bitmapped screen, and was the first computer to demonstrate the desktop metaphor and graphical user interface (GUI). It was not a commercial product, but several thousand units were built and were heavily used at PARC, as well as other XEROX offices, and at several universities for many years. The Alto greatly influenced the design of personal computers during the late 1970s and early 1980s, notably the Three Rivers PERQ, the Apple Lisa and Macintosh, and the first Sun workstations.

The GUI was first developed at Xerox PARC by Alan Kay, Larry Tesler, Dan Ingalls and a number of other researchers. It used windows, icons, and menus to support commands such as opening files, deleting files, moving files, etc. In 1981 Xerox introduced a pioneering product, Star, incorporating many of PARC's innovations. Although not commercially successful, Star greatly influenced future developments, for example at Apple, Microsoft and Sun Microsystems.

In 1974, work began at PARC on Gypsy, the first bitmap What-You-See-Is-What-You-Get (WYSIWYG) cut & paste editor. In 1975, Xerox engineers demonstrated a Graphical User Interface "including icons and the first use of pop-up menus".

The 80s: Early commercial developments

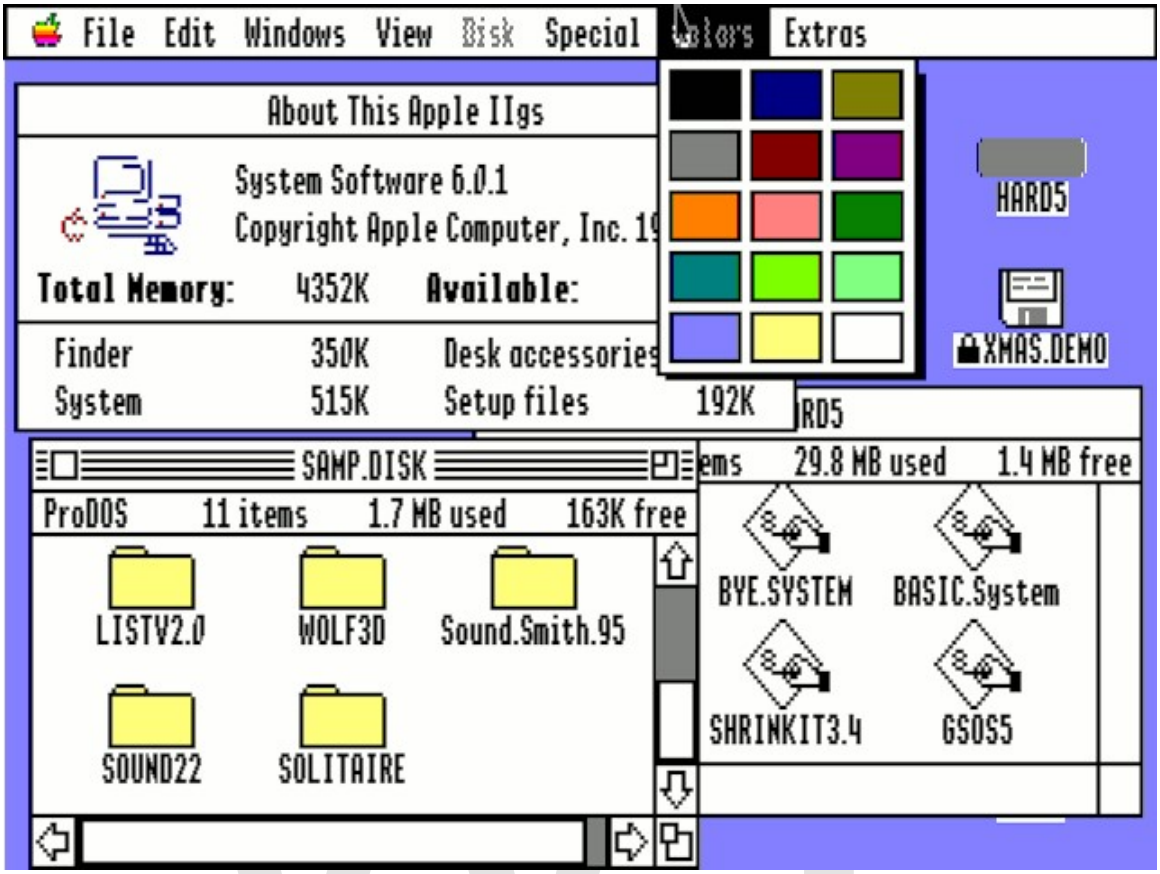
Apple Lisa and Macintosh (and later, the Apple IIgs)



Macintosh Desktop (1984)

Beginning in 1979, started by Steve Jobs and led by Jef Raskin, the Lisa and Macintosh teams at Apple Computer (which included former members of the Xerox PARC group) continued to develop such ideas. The Macintosh, released in 1984, was the first commercially successful product to use a GUI. A desktop metaphor was used, in which files looked like pieces of paper; directories looked like file folders; there were a set of desk accessories like a calculator, notepad, and alarm clock that the user could place around the screen as desired; and the user could delete files and folders by dragging them to a trash can on the screen. Drop down menus were also introduced.

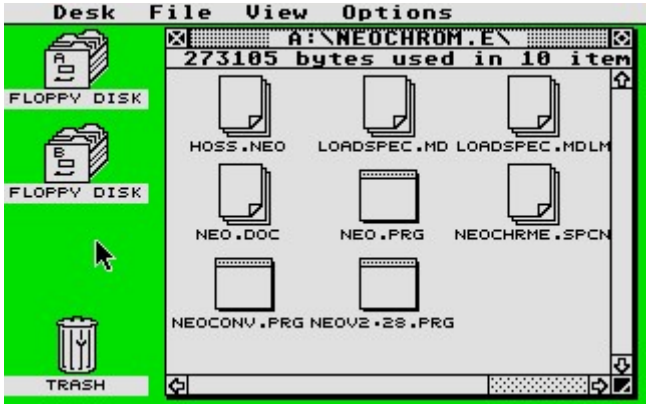
There is still some controversy over the amount of influence that Xerox's PARC work, as opposed to previous academic research, had on the GUIs of Apple's Lisa and Macintosh, but it is clear that the influence was extensive, because first versions of Lisa GUIs even lacked icons. These prototype GUIs are at least mouse driven, but completely ignored the WIMP concept. Rare screenshots of first GUIs of Apple Lisa prototypes are shown here and here. Note also that Apple was invited by PARC to view their research, and a number of PARC employees subsequently moved to Apple to work on the Lisa and Macintosh GUI. However, the Apple work extended PARC's considerably, adding manipulatable icons, a fixed drop-down menu bar and drag&drop manipulation of objects in the file system for example. A list of the improvements made by Apple to the PARC interface can be read here (folklore.org) It's hard to say which particular features were originated in which project, though. Jef Raskin warns that many of the reported facts in the history of the PARC and Macintosh development are inaccurate, distorted or even fabricated, due to the lack of usage by historians of direct primary sources.



The Apple GS/OS desktop (1986).

In 1986 the Apple IIgs was launched, a very advanced model of the Apple II successful series, based on 16-bit technology (in fact, virtually two machines into one). It came with a new operating system, the Apple GS/OS, which features a Finder-like GUI, very similar to that of the Macintosh series, able to deal with the advanced graphic abilities of its Video Graphics Chip (VGC).

Graphical Environment Manager (GEM)



GEM on the Atari ST (1985)

Digital Research (DRI) created the Graphical Environment Manager as an add-on program for personal computers. GEM was developed to work with existing CP/M and MS-DOS operating systems on business computers such as IBM-compatibles. It was developed from DRI software, known as GSX, designed by a former PARC employee. The similarity to the Macintosh desktop led to a copyright lawsuit from Apple Computer, and a settlement which involved some changes to GEM. This was to be the first of a series of 'look and feel' lawsuits related to GUI design in the 1980s.

GEM received widespread use in the consumer market from 1985, when it was made the default user interface built in to the TOS operating system of the Atari ST line of personal computers. It was also bundled by other computer manufacturers and distributors, such as Amstrad. Later, it was distributed with the best-sold Digital Research version of DOS for IBM PC compatibles, the DR-DOS 6.0. The GEM desktop faded from the market with the withdrawal of the Atari ST line in 1992 and with the popularity of the Microsoft Windows 3.0 in the PC front by the same years.

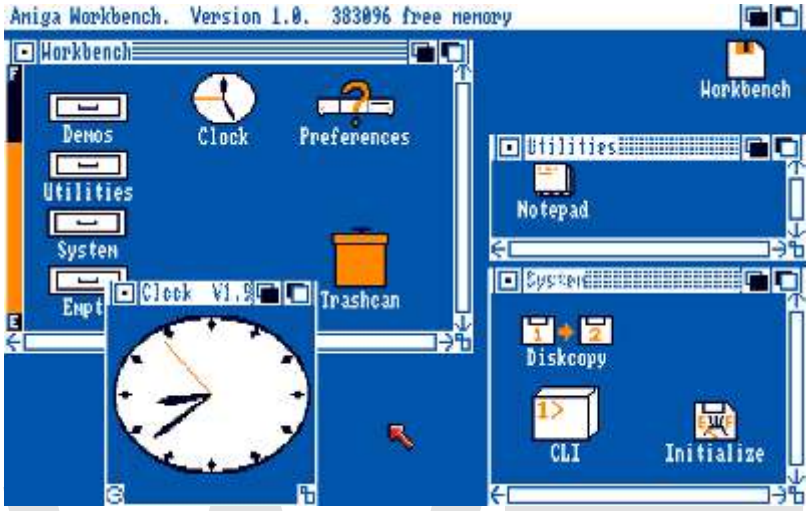
DeskMate

DeskMate 3.02 running in VGA mode

Tandy's DeskMate appeared in the early 1980s on its TRS-80 machines and was ported to its Tandy 1000 range in 1984. Like most PC GUIs of the time it depended on MS-DOS. The application was popular at the time and included a number of programs like Draw,

Text and Calendar as well as attracting outside investment such as Lotus 1-2-3 for DeskMate.

Amiga Intuition and the Workbench



Amiga Workbench (1985)

The Amiga computer was launched by Commodore in 1985 with a GUI called Workbench. Workbench was based on an internal engine developed mostly by RJ Mical, called Intuition, which drove all the input events. The first versions used a blue/orange/white/black default palette, which was selected for high contrast on televisions and composite monitors. Workbench presented directories as drawers to fit in with the "workbench" theme. Intuition was the widget and graphics library that made the GUI work. It was driven by user events through the mouse, keyboard, and other input devices.

Due to a mistake made by the Commodore sales department, the first floppies of AmigaOS (released with the Amiga1000) named the whole OS "Workbench". Since then, users and CBM itself referred to "Workbench" as the nickname for the whole AmigaOS (including Amiga DOS, Extras, etc.). This common consent ended with release of version 2.0 of AmigaOS, which re-introduced proper names to the installation floppies of AmigaDOS, Workbench, Extras, etc.).

Early versions of AmigaOS did treat the Workbench as just another window on top of a blank screen, but this is due to the ability of AmigaOS to have invisible screens with a chromakey or a genlock – one of the most advanced features of Amiga platform – even without losing the visibility of Workbench itself. In later AmigaOS versions Workbench could be set as a borderless desktop.

Amiga users were able to boot their computer into a command line interface (aka. CLI/shell). This was a keyboard-based environment without the Workbench GUI. Later

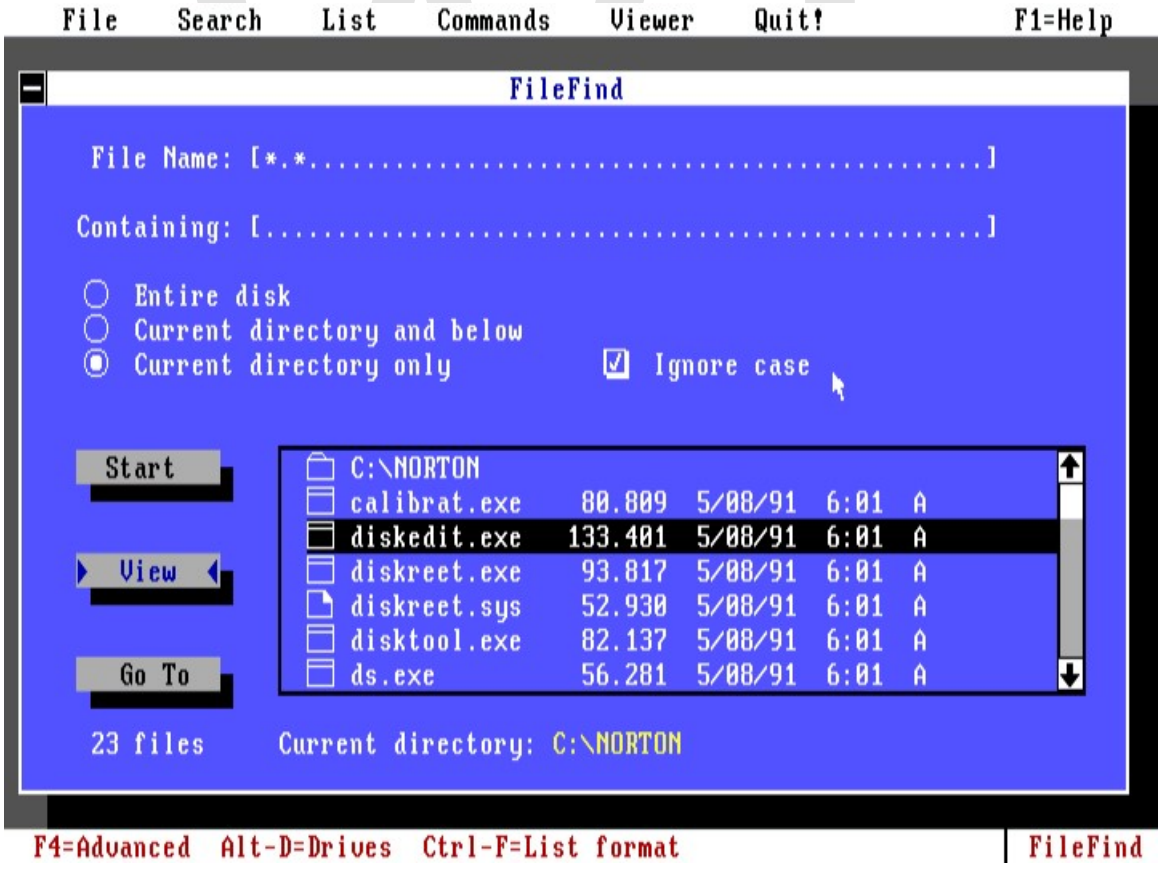
they could invoke it with the CLI/SHELL command **LoadWB** which loaded Workbench GUI.

One major difference between other OS's of the time and for some time after was the Amiga's fully Multi-Tasking Operating System, a powerful built-in animation system using a hardware blitter and copper and 4 channels of 26k 8 bit sampled sound. This made the Amiga the first Multi Media computer years before other OS's.

Like most GUIs of the day, Amiga's Intuition followed Xerox's, and sometimes Apple's, lead. But a CLI was included which dramatically extended the functionality of the platform. However, the CLI/Shell of Amiga is not just a simple text-based interface like in MS-DOS, but another graphic process driven by Intuition, and with the same gadgets included in Amiga's graphics.library. The CLI/Shell interface integrates itself with the Workbench, sharing privileges with the GUI.

The Amiga Workbench evolved over the 1990s, far beyond the official withdrawn from Commodore in 1994.

MS-DOS file managers and utility suites



Norton Utilities 6.01 (1991). Note the graphical widgets and the arrow pointer in Text mode.

Because most of the very early IBM PC and compatibles lacked any common true graphical capability (they used the 80-column basic text mode compatible with the original MDA display adapter), a series of file managers arose, including Microsoft's DOS Shell, which features typical GUI elements as menus, push buttons, lists with scrollbars and mouse pointer. The name Text user interface was later invented to name this kind of interface. Many MS-DOS text mode applications, like the default text editor for MS-DOS 5.0 (and related tools, like QBasic), also used the same philosophy. The IBM DOS Shell included with IBM DOS 5.0 (circa 1992) supported both text display modes and actual graphics display modes, making it both a TUI and a GUI, depending on the chosen mode.

Advanced file managers for MS-DOS were able to redefine character shapes with EGA and better display adapters, giving some basic low resolution icons and graphical interface elements, including an arrow (instead of a coloured cell block) for the mouse pointer. When the display adapter lacks the ability to change the character's shapes, they default to the CP437 character set found in the adapter's ROM. Some popular utility suites for MS-DOS, as Norton Utilities (pictured) and PC Tools used these techniques as well.

DESQview was a text mode multitasking program introduced in July 1985. Running on top of MS-DOS, it allowed users to run multiple DOS programs concurrently in windows. It was the first program to bring multitasking and windowing capabilities to a DOS environment in which existing DOS programs could be used. DESQview was not a true GUI but offered certain components of one, such as resizable, overlapping windows and mouse pointing.

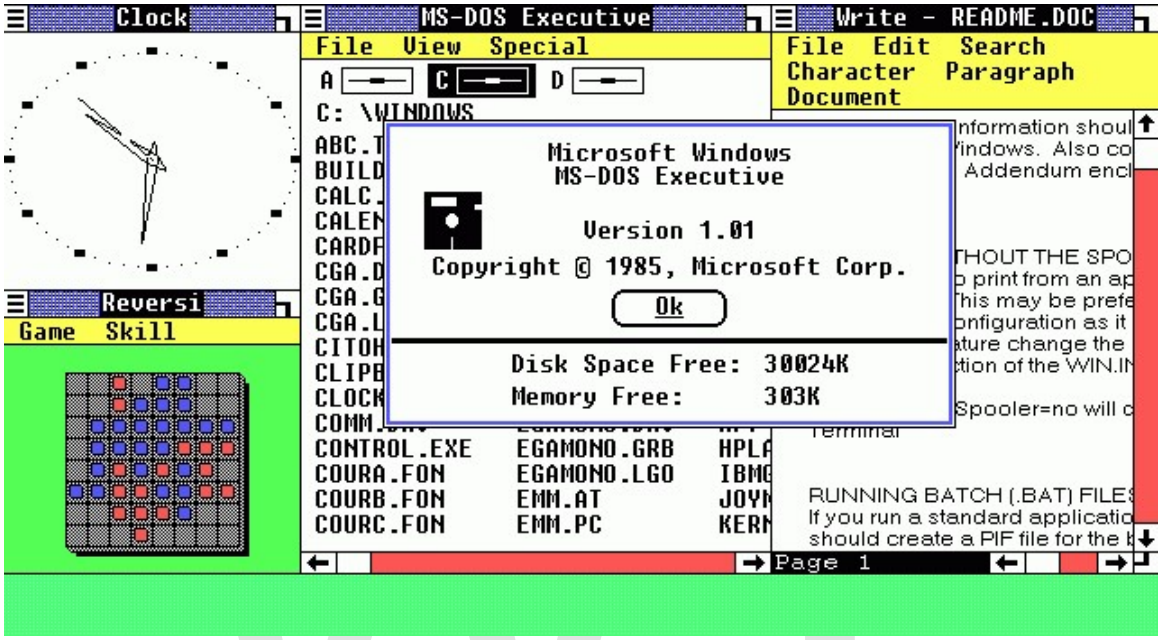
Applications under MS-DOS with proprietary true GUIs

DeluxePaint II for MS-DOS (1989)

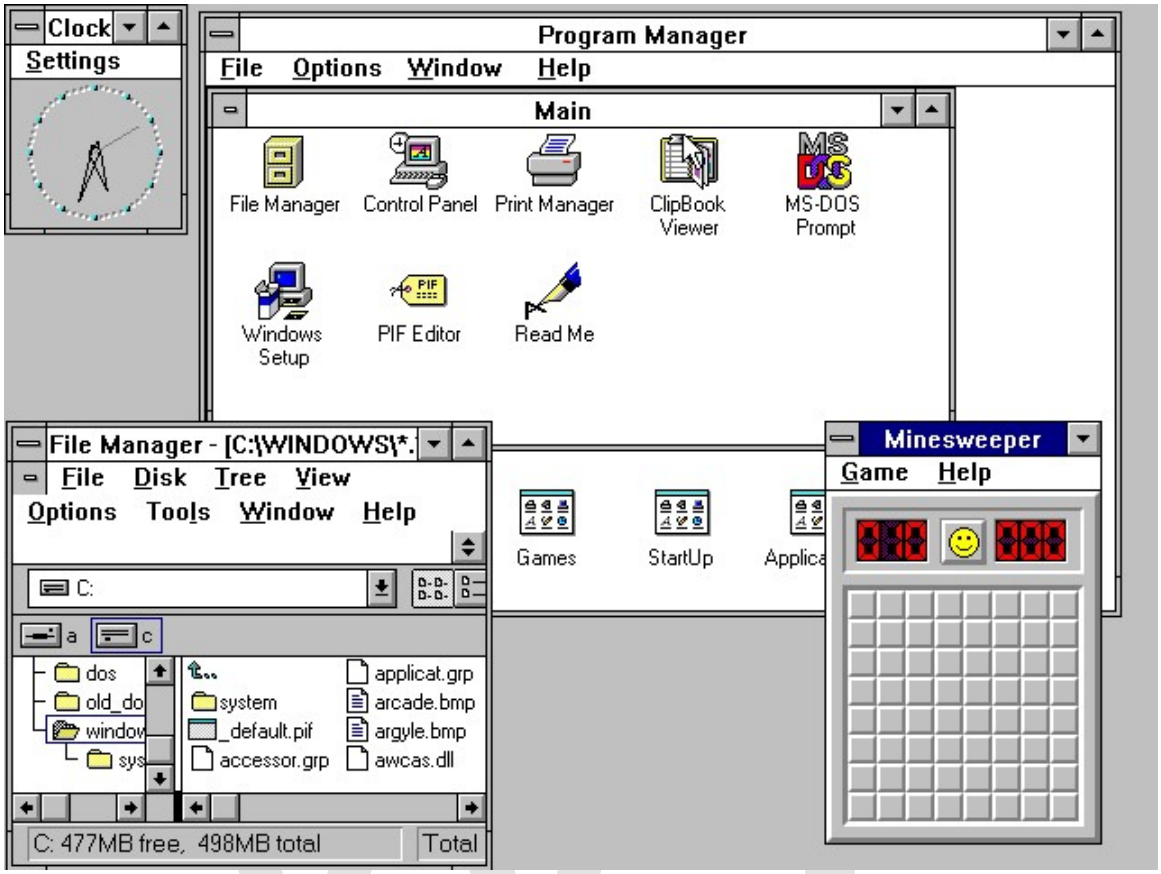
To take the maximum advantage possible in lack of a true common GUI under MS-DOS, the most of the graphical applications which worked with EGA, VGA and better graphic cards had proprietary built-in GUIs, before the MS-Windows age. One of the best known was Deluxe Paint, a popular painting software with a typical WIMP interface.

The original Adobe Acrobat Reader executable file for MS-DOS was able to run on both the standard Windows 3.x GUI and the standard DOS command prompt. When it was launched from the command prompt, it provides its own true GUI (on VGA), which provides the full of its functionality to read PDF files.

Microsoft Windows (16-bit versions)



Windows 1.01 (1985)



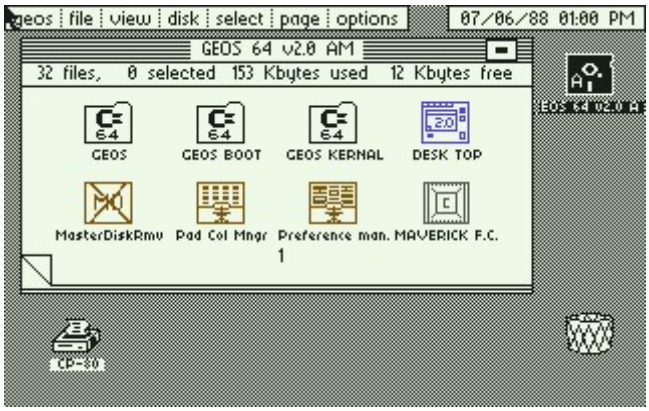
Windows 3.11 (1993)

Windows 1.0 was a GUI for the MS-DOS operating system that had been the OS of choice for IBM PC and compatible computers since 1981. Windows 2.0 followed, but it wasn't until the 1990 launch of Windows 3.0, based on Common User Access that its popularity truly exploded. The GUI has seen minor redesigns since, mainly the networking enabled Windows 3.11 and its Win32s 32-bit patch. The 16-bit line of MS Windows were discontinued with the introduction of Windows 95 and Windows NT 32-bit based architecture in the 1990s.

The main window of a given application can occupy the full screen in *maximized* status. The users must then to switch between maximized applications using the Alt+Tab keyboard shortcut; no alternative with the mouse except for de-maximize. When none of the running application windows is maximized, switching can be done by clicking on a partially visible window, as is the common way in other GUIs.

In 1988, Apple sued Microsoft for copyright infringement of the LISA and Apple Macintosh GUI. The court case lasted 4 years before almost all of Apple's claims were denied on a contractual technicality. Subsequent appeals by Apple were also denied. Microsoft and Apple apparently entered a final, private settlement of the matter in 1997.

GEOS

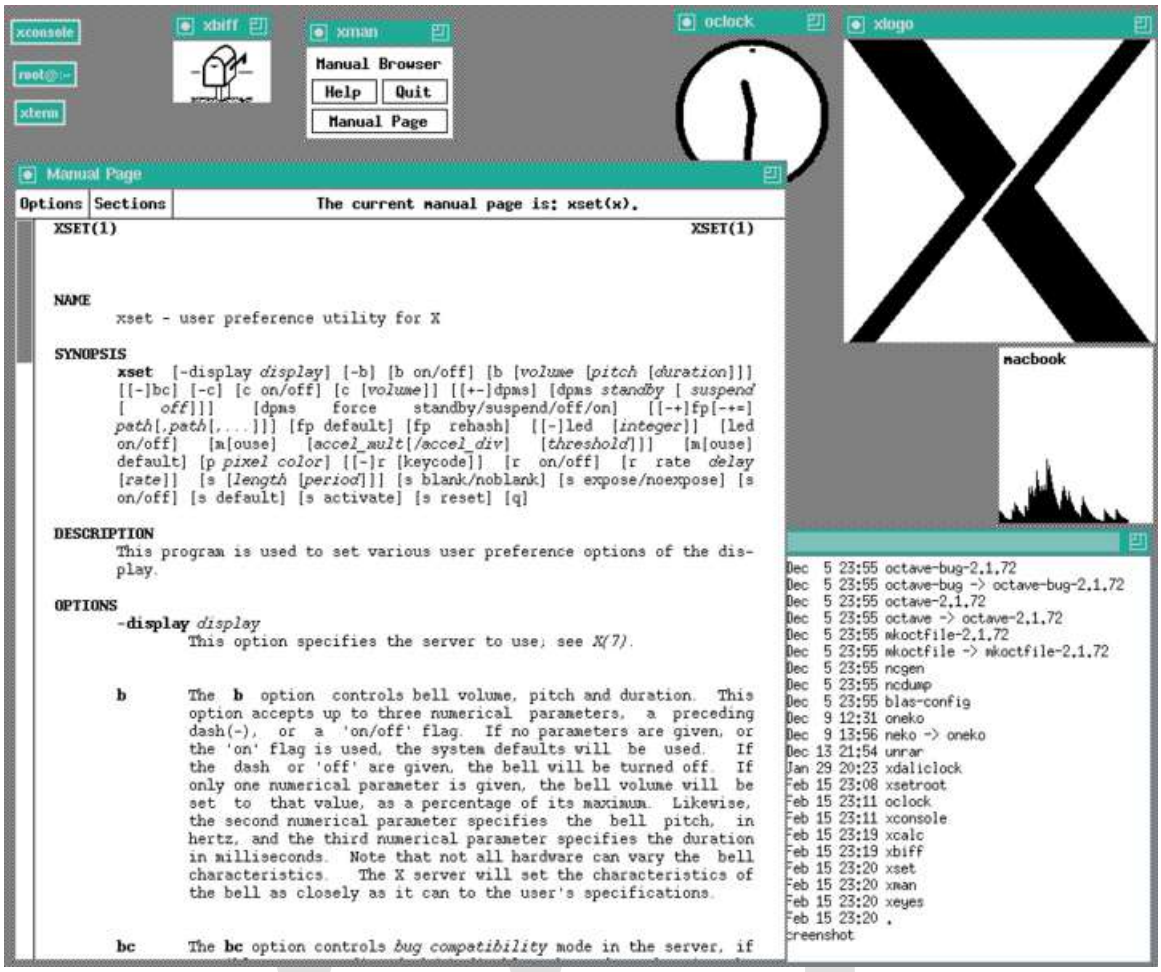


GEOS for the Commodore 64 (1986).

GEOS was launched in 1986. Originally written for the 8-bit home computer Commodore 64 and shortly after, the Apple II series it was later ported to IBM PC systems. It came with several application programs like a calendar and word processor, and a cut-down version served as the basis for America Online's DOS client. Compared to the competing Windows 3.0 GUI it could run reasonably well on simpler hardware. But it was targeted at 8-bit machines and the 16-bit computer age was dawning.

WT

The X Window System

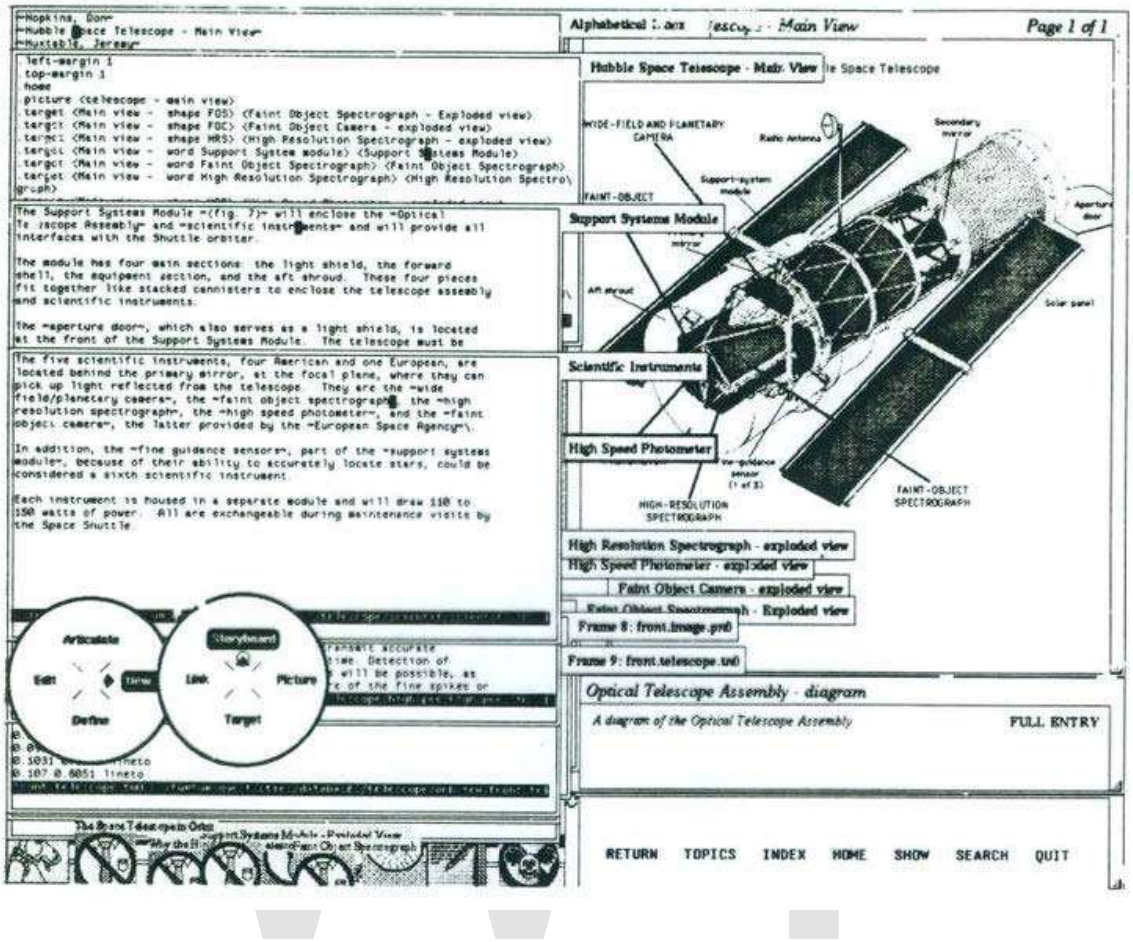


A Unix based X Window System desktop (circa 1990).

The standard windowing system in the Unix world is the X Window System (commonly X11 or X), first released in the mid-1980s. The W Window System (1983) was the precursor to X; X was developed at MIT as Project Athena. Its original purpose was to allow users of the newly emerging graphic terminals to access remote graphics workstations without regard to the workstation's operating system or the hardware. Due largely to the availability of the source code used to write X, it has become the standard layer for management of graphical and input/output devices and for the building of both local and remote graphical interfaces on virtually all Unix, Linux and other Unix-like operating systems, with the notable exception of Mac OS X.

X allows a graphical terminal user to make use of remote resources on the network as if they were all located locally to the user by running a single module of software called the X server. The software running on the remote machine is called the client application. X's network transparency protocols allow the display and input portions of any application to be separated from the remainder of the application and 'served up' to any of a large number of remote users. X is available today as free software.

NeWS



HyperTIES authoring tool under NeWS window system.

The PostScript-based NeWS (Network extensible Window System) was developed by Sun Microsystems in the mid 1980's. For several years SunOS included a window system combining NeWS and the X Window System. Although NeWS was considered technically elegant by some commentators, Sun eventually dropped the product. Unlike X, NeWS was always proprietary software.

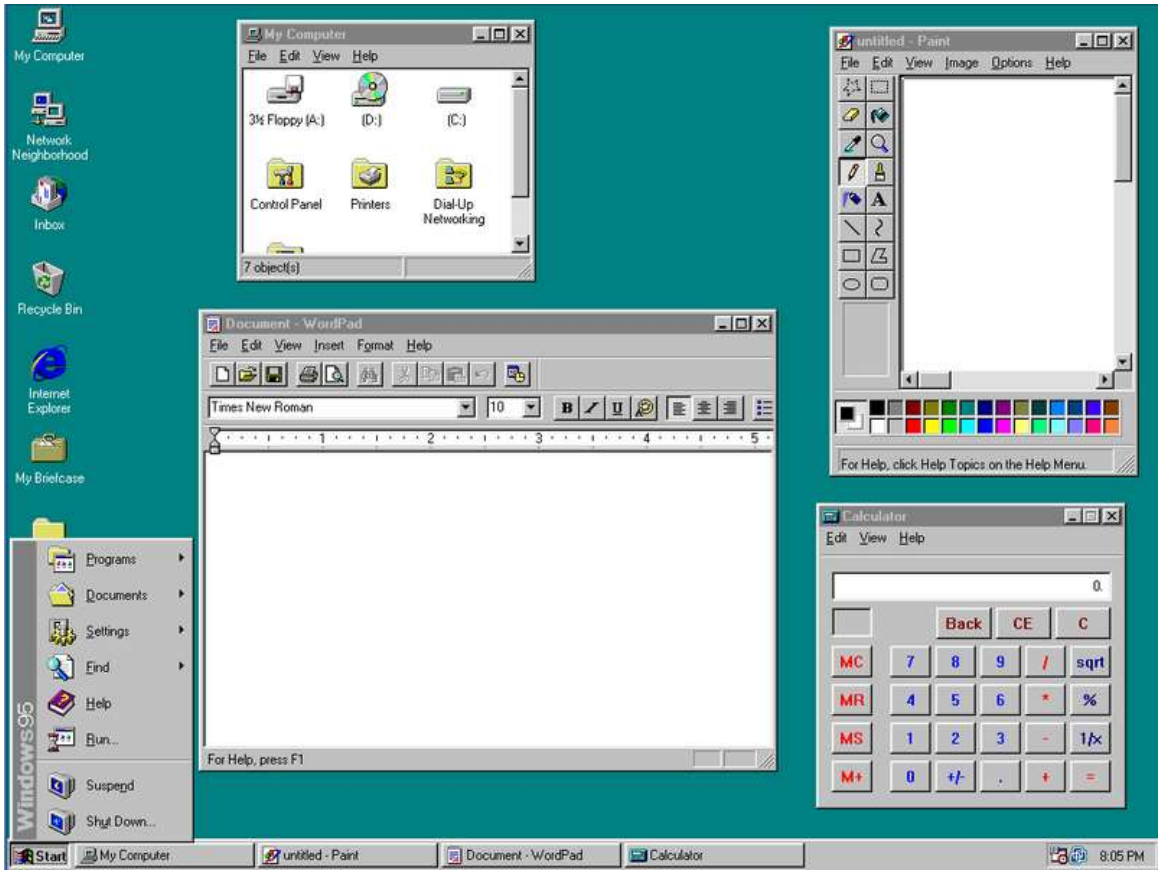
The 1990s: Mainstream usage of the desktop

The widespread adoption of the PC platform at homes and small business popularized computers among people with no formal training. This created a fast growing market, opening an opportunity for commercial exploitation and of easy-to-use interfaces and making economically viable the incremental refinement of the existing GUIs for home systems.

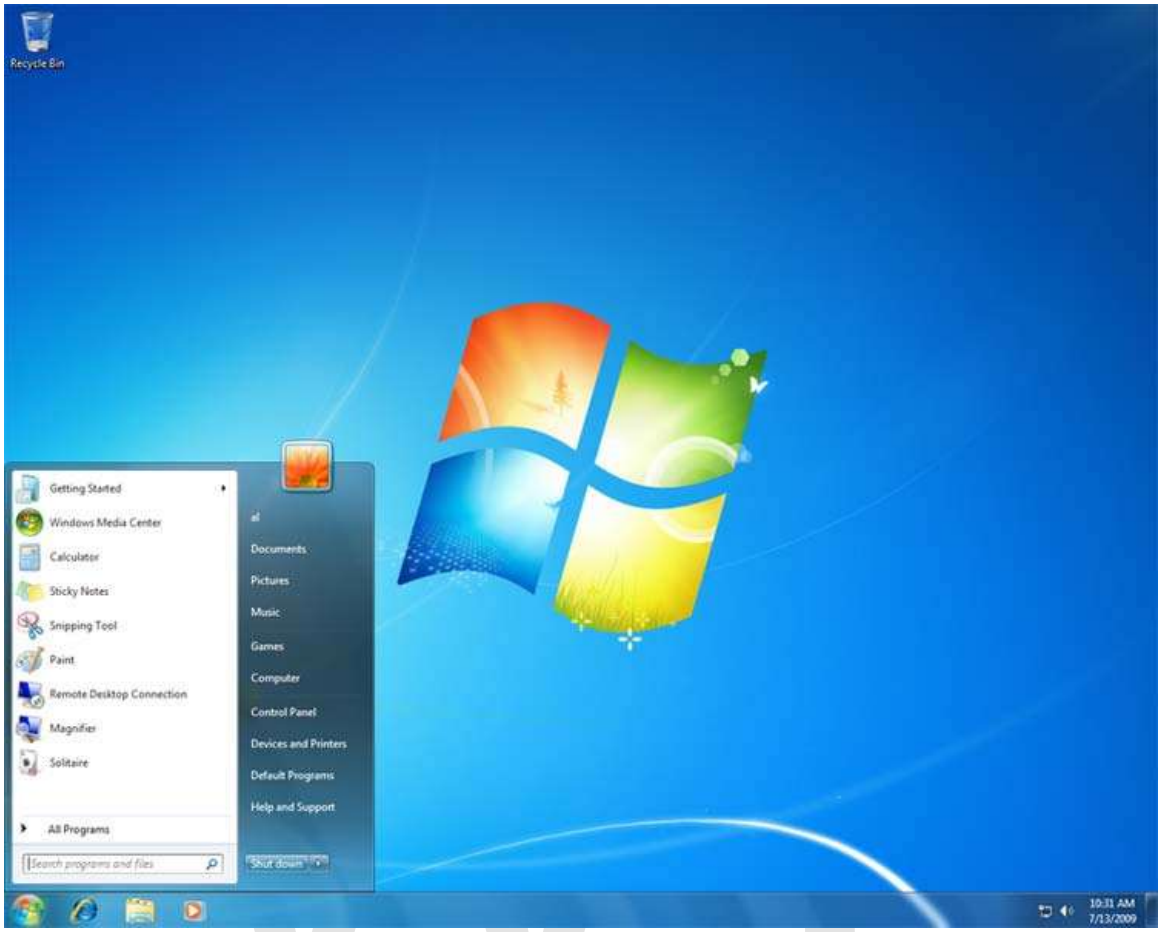
Also, the spreading of Highcolor and Truecolor capabilities of display adapters providing thousands and millions of colors, along with faster CPUs and accelerated graphic cards,

cheaper RAM, storage devices up to an order of magnitude larger (from megabytes to gigabytes) and larger bandwidth for telecom networking at lower cost helped to create an environment in which the common user was able to run complicated GUIs which began to favor aesthetics.

Windows 95 and "a computer in every home" (the 32-bit versions)



Windows 95 desktop (1995).



Windows 7 desktop (2009).

After Windows 3.11, Microsoft began to develop a new consumer-oriented version of the operating system. Windows 95 was intended to integrate Microsoft's formerly separate MS-DOS and Windows products and includes an enhanced version of DOS, often referred to as MS-DOS 7.0, although many people believe that Windows 95's MS-DOS was not really DOS at all. It also featured a significant redesign of the GUI, dubbed "Cairo. While Cairo never really materialized, parts of Cairo found their way into subsequent versions of the operating system starting with Windows 95. Both Win95 and WinNT could run 32-bit applications, and could exploit the abilities of the Intel 80386 CPU, as the preemptive multitasking and up to 4GiB of linear address memory space. Windows 95 was touted as a 32-bit based operating system but it was actually based on a hybrid kernel (VWIN32.VXD) with the 16-bit user interface (USER.EXE) and graphic device interface (GDI.EXE) of Windows for Workgroups (3.11), which had 16-bit kernel components with a 32-bit subsystem (USER32.DLL and GDI32.DLL) that allowed it to run native 16 bit applications as well as 32-bit applications. In the marketplace, Windows 95 was an unqualified success, promoting a general upgrade to 32-bit technology, and within a year or two of its release had become the most successful operating system ever produced.

Windows 95 saw the beginning of the Browser wars when the World Wide Web began receiving a great deal of attention in the popular culture and mass media. Microsoft at first did not see potential in the Web and Windows 95 was shipped with Microsoft's own online service called The Microsoft Network, which was dial-up only and was used primarily for its own content, not internet access. As versions of Netscape Navigator and Internet Explorer were released at a rapid pace over the following few years, Microsoft used its desktop dominance to push its browser and shape the ecology of the web mainly as a monoculture.

Windows 95 evolved through the years into Windows 98 and Windows ME. Windows ME was the last in the line of the Windows 3.x based operating systems from Microsoft. Windows underwent a parallel 32-bit evolutionary path, where Windows NT 3.1 was released in 1993. Windows NT (for New Technology) was a native 32-bit operating system with a new driver model, was unicode-based, and provided for true separation between applications. Windows NT also supported 16-bit applications in an NTVDM, but it did not support VXD based drivers. Windows 95 was supposed to be released before 1993 as the predecessor to Windows NT. The idea was to promote the development of 32-bit applications with backward compatibility - leading the way for more successful NT release. After multiple delays, Windows 95 was released without unicode and used the VXD driver model. Windows NT 3.1 evolved to Windows NT 4, then Windows 2000, Windows XP in 2003, Windows Vista, then Windows 7. Windows XP and higher were also made available in 64-bit modes. Windows server products branched off with the introduction of Windows 2003 (available in 64-bit IA64 or x64 (based on AMD's core) with or without R2), then Windows 2008. Windows 2000 and XP shared the same basic GUI themes (in XP, the user can even switch to the *classical Windows* 95/NT look). With Windows 98, the Active Desktop theme was introduced, allowing a HTML approach for the desktop, but this feature was coldly received by customers, who frequently disabled it. At the end, Windows Vista definitively discontinued it, but has a new SideBar on the desktop.

Mac OS

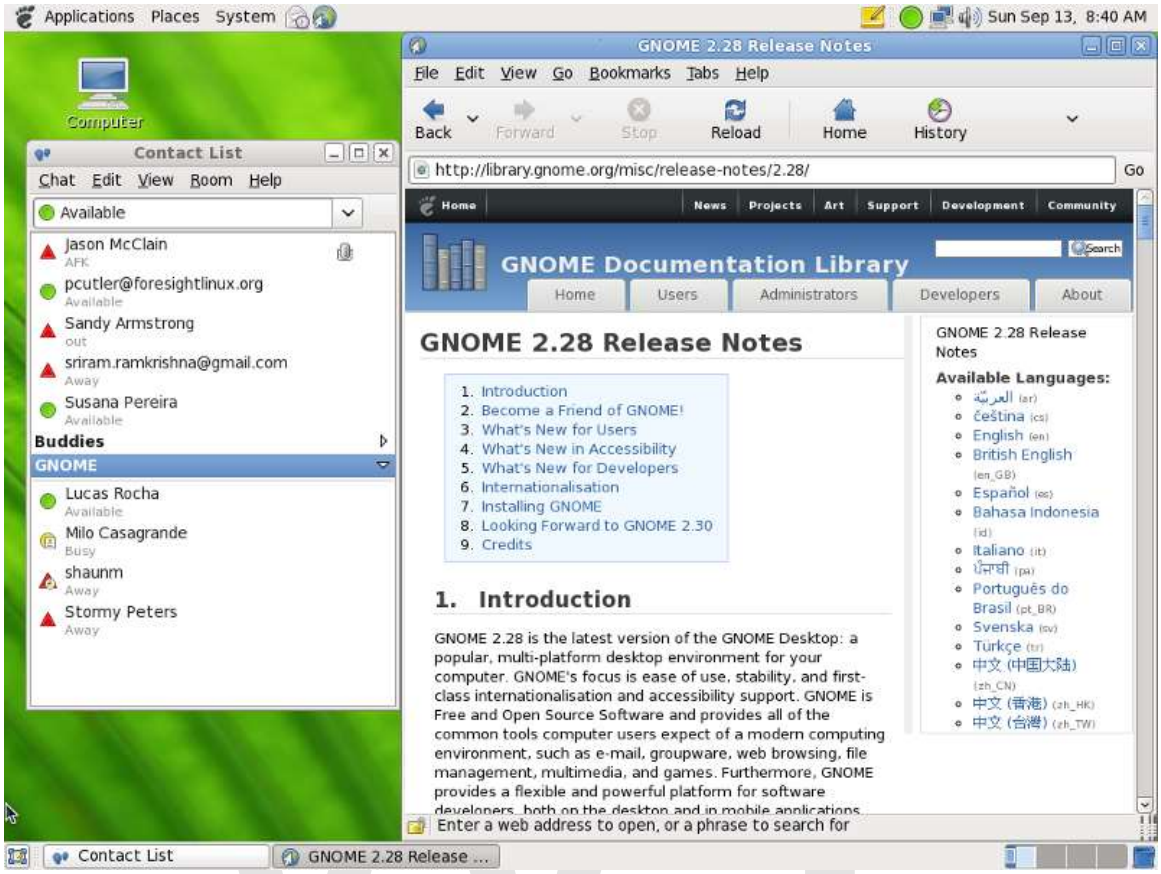
The Macintosh's GUI has been infrequently revised since 1984, with major updates including System 7, and underwent its largest revision with the introduction of the "Aqua" interface in 2001's Mac OS X. It was a new operating system built primarily on technology from NeXTStep with UI elements of the original Mac OS grafted on. Mac OS X uses a technology called Quartz for graphics rendering and drawing on-screen. Some interface features of Mac OS X are inherited from NeXTStep (such as the Dock, the automatic wait cursor, or double-buffered windows giving a solid appearance and flicker-free window redraws), while others are inherited from the old Mac OS operating system (the single system-wide menu-bar). Mac OS X v10.3 introduced features to improve usability including Exposé which is designed to make finding open windows easier.

With Mac OS X v10.4, new features were added, including Dashboard (a virtual alternate desktop for mini specific-purpose applications) and a search tool called Spotlight, which

provides users with an option for searching through files instead of browsing through folders.

GUIs built on the X Window System

KDE Plasma 4.5 desktop (2010)



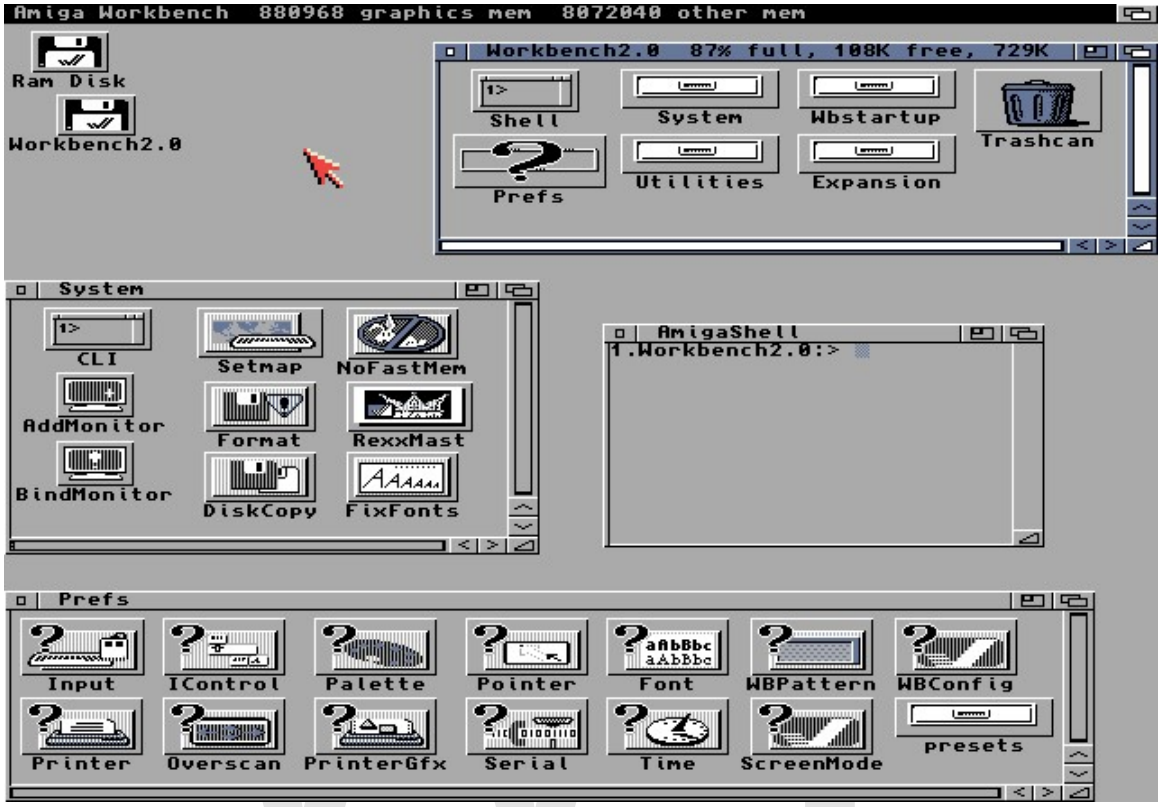
A GNOME 2.28 desktop (2010)

In the early days of X Window development, Sun Microsystems and AT&T attempted to push for a GUI standard called OPEN LOOK in competition with Motif. OPEN LOOK was a well-designed standard developed from scratch in conjunction with Xerox, while Motif was a collective effort that fell into place, with a look and feel patterned after Windows 3.11. Many who worked on OPEN LOOK at the time appreciated its design coherence. Motif prevailed in the UNIX GUI battles and became the basis for the Common Desktop Environment (CDE). CDE was based on VUE (Visual User Environment), a proprietary desktop from Hewlett-Packard that in turn was based on the Motif look and feel.

In the late 1990s, there was significant growth in the Unix world, especially among the free software community. New graphical desktop movements grew up around Linux and similar operating systems, based on the X Window System. A new emphasis on providing an integrated and uniform interface to the user brought about new desktop environments, such as KDE Plasma Desktop, GNOME and XFCE which are supplanting CDE in popularity on both Unix and Unix-like operating systems. The XFCE, KDE and GNOME look and feel each tend to undergo more rapid change and less codification than the earlier OPEN LOOK and Motif environments.

In the latter part of the first decade of the 21st century X Window GUIs such as Compiz Fusion and KWin began to incorporate translucency and drop shadow effects.

AmigaOS



Amiga Workbench 2.0 (1990)

Amiga Workbench 4.1 (2009)

Later releases added improvements over the original Workbench, like support for high-color Workbench screens, context menus, and embossed 2D icons with pseudo-3D aspect. But often Amiga users preferred alternative interfaces to standard Workbench, such as Directory Opus or ScalOS interface. An interesting article about these replacements is available here (in French language).

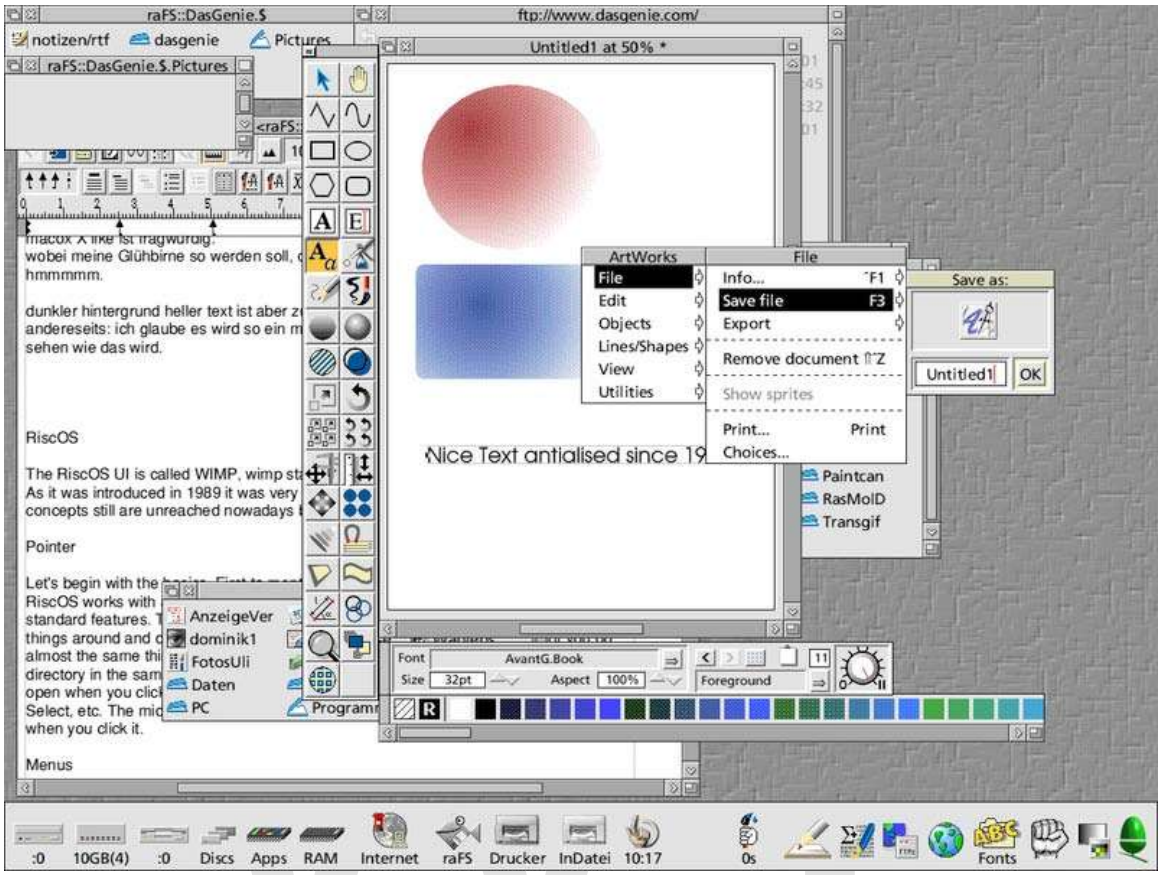
AmigaOS 4 Workbench with custom GUI (2007)

The use of improved, third-party GUI engines became common amongst users who preferred more attractive interfaces – such as Magic User Interface (MUI), and ReAction. These object-oriented graphic engines driven by "classes" of graphic objects and functions were then standardized into the Amiga environment and changed Amiga Workbench to a complete and modern guided interface, with new standard gadgets, animated buttons, true 24-bit-color icons, increased use of wallpapers for screens and windows, alpha channel, transparencies and shadows as any modern GUI requires.

Modern derivatives of Workbench are Ambient for MorphOS, ScalOS, Workbench for AmigaOS 4 and Wanderer/Zune for AROS. There is a brief article on Ambient and descriptions of MUI icons, menus and gadget here (aps.fr) and images of Zune stay at main AROS site.

Use of object oriented graphic engines (ReAction) dramatically changes the look and feel of a GUI to match actual styleguides.

RISC OS



A typical *RISC OS 3.7* session

Early versions of what came to be called RISC OS were known as Arthur, which was released in 1987 by Acorn Computers. RISC OS was a colour GUI operating system which used three-button mice, a taskbar (called the icon bar), and a file navigator similar to that of Mac OS. Acorn created RISC OS in the 1980s for their ARM-CPU based computers. The GUI of RISC OS has developed over versions of RISC OS from 1987 to the present day with version 4.39 having a great ability to customise the interface.

OS/2

OS/2 Workplace Shell

Originally collaboratively developed by Microsoft and IBM to replace DOS, OS/2 version 1.0 (released in 1987) had no GUI at all. Version 1.1 (released 1988) included Presentation Manager (PM), which looked a lot like the later Windows 3.0 UI. After the split with Microsoft, IBM developed the Workplace Shell (WPS) for version 2.0 (released in 1992), a quite radical, object-oriented approach to GUIs. Microsoft later imitated much of this in Windows 95.

NeXTSTEP

NeXTSTEP Desktop

The NeXTSTEP user interface was used in the NeXT line of computers. NeXTSTEP's first major version was released in 1989. It used Display PostScript for its graphical underpinning. The NeXTSTEP interface's most significant feature was the Dock, carried with some modification into Mac OS X, and had other minor interface details that some found made it easier and more intuitive to use than previous GUIs. NeXTSTEP's GUI was the first to feature opaque dragging of windows in its user interface, on a comparatively weak machine by today's standards, ideally aided by high performance graphics hardware.

BeOS

BeOS Desktop

BeOS was developed on custom AT&T Hobbit-based computers before switching to PowerPC hardware by a team lead by former Apple executive Jean-Louis Gassée as an alternative to Mac OS. BeOS was later ported to Intel hardware. It used an object-oriented kernel written by Be, and did not use the X Window System, but a different GUI written from scratch. Much effort was spent by the developers to make it an efficient platform for multimedia applications. Be Inc. was acquired by PalmSource, Inc. (Palm Inc. at the time) in 2001. The BeOS GUI still lives in Haiku, an open source software reimplementation of the BeOS.

Current trends

3D User Interface

Compiz running on Fedora Core 6 with AIGLX.

As of 2009, a recent trend in desktop technology is the inclusion of 3D effects in window management. It's based in experimental research in User Interface Design trying to expand the expressive power of the existing toolkits in order to enhance the physical cues that allow for direct manipulation. New effects common to several projects are scale resizing and zooming, several windows transformations and animations (wobbling windows, smooth minimization to system tray...), composition of images (used for window drop shadows and transparency) and enhancing the global organization of open windows (zooming to virtual desktops, desktop cube, Exposé, etc.) The proof-of-concept BumpTop desktop combines a physical representation of documents with tools for document classification possible only in the simulated environment, like instant reordering and automated grouping of related documents.

These effects are popularized thanks to the widespread use of 3D video cards (mainly due to gaming) which allow for complex visual processing with low CPU use, using the 3D acceleration in most modern graphics cards to render the application clients in a 3D scene. The application window is drawn off-screen in a pixel buffer and the graphics card renders it into the 3D scene.

This can have the advantage of moving some of the window rendering to the GPU on the graphics card and thus reducing the load on the main CPU, but the facilities that allow this must be available on the graphics card to be able to take advantage of this.

Examples of 3D user interface software include XGL and Compiz from Novell, and AIGLX bundled with Red Hat Fedora. Quartz Extreme for Mac OS X and Windows 7's Aero interface use 3D rendering for shading and transparency effects as well as Expose and Windows Flip and Flip 3D, respectively. AmigaOS 4.1 uses Cairo 2D vector based interface integrated with 3D hardware accelerated Porter-Duff image composition engine, while its counterpart clone MorphOS 2.0 features Ambient, a complete 3D GUI based on a subset of OpenGL. Vista uses Direct3D to accomplish this, whereas the other interfaces use OpenGL.

Portable devices

Portable devices such as mp3 players and cell phones are a burgeoning area of deployment for GUIs in recent years. Since the mid-2000s, a vast majority of portable devices have advanced to having high screen resolutions and sizes (The iPhone's 640x960 display for example). Because of this, these devices have their own famed user interfaces and operating systems that have large homebrew communities dedicated to creating their own visual elements, such as Icons, Menus, Wallpapers, and more.

Precursors

A precursor to GUIs was invented by researchers at the Stanford Research Institute, led by Douglas Engelbart. They developed the use of text-based hyperlinks manipulated with a mouse for the On-Line System. The concept of hyperlinks was further refined and extended to graphics by researchers at Xerox PARC, who went beyond text-based hyperlinks and used a GUI as the primary interface for the Xerox Alto computer. Most modern general-purpose GUIs are derived from this system. As a result, some people call this class of interface a PARC User Interface (PUI) (note that PUI is also an acronym for perceptual user interface).

Ivan Sutherland developed a pointer-based system called the Sketchpad in 1963. It used a light-pen to guide the creation and manipulation of objects in engineering drawings.

PARC user interface

The PARC user interface consisted of graphical elements such as windows, menus, radio buttons, check boxes and icons. The PARC user interface employs a pointing device in addition to a keyboard. These aspects can be emphasized by using the alternative acronym WIMP, which stands for *windows*, *icons*, *menus* and *pointing device*.

Evolution

The Xerox Star Workstation introduced the first GUI operating systems as shown above.

Following PARC the first GUI-centric computer operating model was the Xerox 8010 Star Information System in 1981, followed by the Apple Lisa (which presented the concept of menu bar as well as window controls) in 1983, the Apple Macintosh 128K in 1984, and the Atari ST and Commodore Amiga in 1985.

The GUIs familiar to most people today are Mac OS X, Microsoft Windows, and X Window System interfaces. Apple, IBM and Microsoft used many of Xerox's ideas to develop products, and IBM's Common User Access specifications formed the basis of the user interface found in Microsoft Windows, IBM OS/2 Presentation Manager, and the Unix Motif toolkit and window manager. These ideas evolved to create the interface found in current versions of Microsoft Windows, as well as in Mac OS X and various desktop environments for Unix-like operating systems, such as Linux. Thus most current GUIs have largely common idioms.

Components

A GUI uses a combination of technologies and devices to provide a platform the user can interact with, for the tasks of gathering and producing information.

A series of elements conforming a visual language have evolved to represent information stored in computers. This makes it easier for people with few computer skills to work with and use computer software. The most common combination of such elements in GUIs is the WIMP ("window, icon, menu, pointing device") paradigm, especially in personal computers.

The WIMP style of interaction uses a physical input device to control the position of a cursor and presents information organized in windows and represented with icons.

Available commands are compiled together in menus, and actions are performed making gestures with the pointing device. A window manager facilitates the interactions between windows, applications, and the windowing system. The windowing system handles hardware devices such as pointing devices and graphics hardware, as well as the positioning of the cursor.

In personal computers all these elements are modeled through a desktop metaphor, to produce a simulation called a desktop environment in which the display represents a desktop, upon which documents and folders of documents can be placed. Window managers and other software combine to simulate the desktop environment with varying degrees of realism.

Post-WIMP interfaces

Smaller mobile devices such as PDAs and smartphones typically use the WIMP elements with different unifying metaphors, due to constraints in space and available input devices. Applications for which WIMP is not well suited may use newer interaction techniques, collectively named as post-WIMP user interfaces.

Some touch-screen-based operating systems such as Apple's iOS and Android OS currently use post-WIMP styles of interaction. The iPhone's use of more than one finger in contact with the screen allows actions such as pinching and rotating, which are not supported by a single pointer and mouse.

A class of GUIs sometimes referred to as post-WIMP include 3D compositing window manager such as Compiz, Desktop Window Manager, and LG3D. Some post-WIMP interfaces may be better suited for applications which model immersive 3D environments, such as Google Earth.

User interface and interaction design

Designing the visual composition and temporal behavior of GUI is an important part of software application programming. Its goal is to enhance the efficiency and ease of use for the underlying logical design of a stored program, a design discipline known as usability. Techniques of user-centered design are used to ensure that the visual language introduced in the design is well tailored to the tasks it must perform.

Typically, the user interacts with information by manipulating visual widgets that allow for interactions appropriate to the kind of data they hold. The widgets of a well-designed interface are selected to support the actions necessary to achieve the goals of the user. A Model-view-controller allows for a flexible structure in which the interface is independent from and indirectly linked to application functionality, so the GUI can be easily customized. This allows the user to select or design a different *skin* at will, and eases the designer's work to change the interface as the user needs evolve. Nevertheless, good user interface design relates to the user, not the system architecture.

The visible graphical interface features of an application are sometimes referred to as "chrome". Larger widgets, such as windows, usually provide a frame or container for the main presentation content such as a web page, email message or drawing. Smaller ones usually act as a user-input tool.

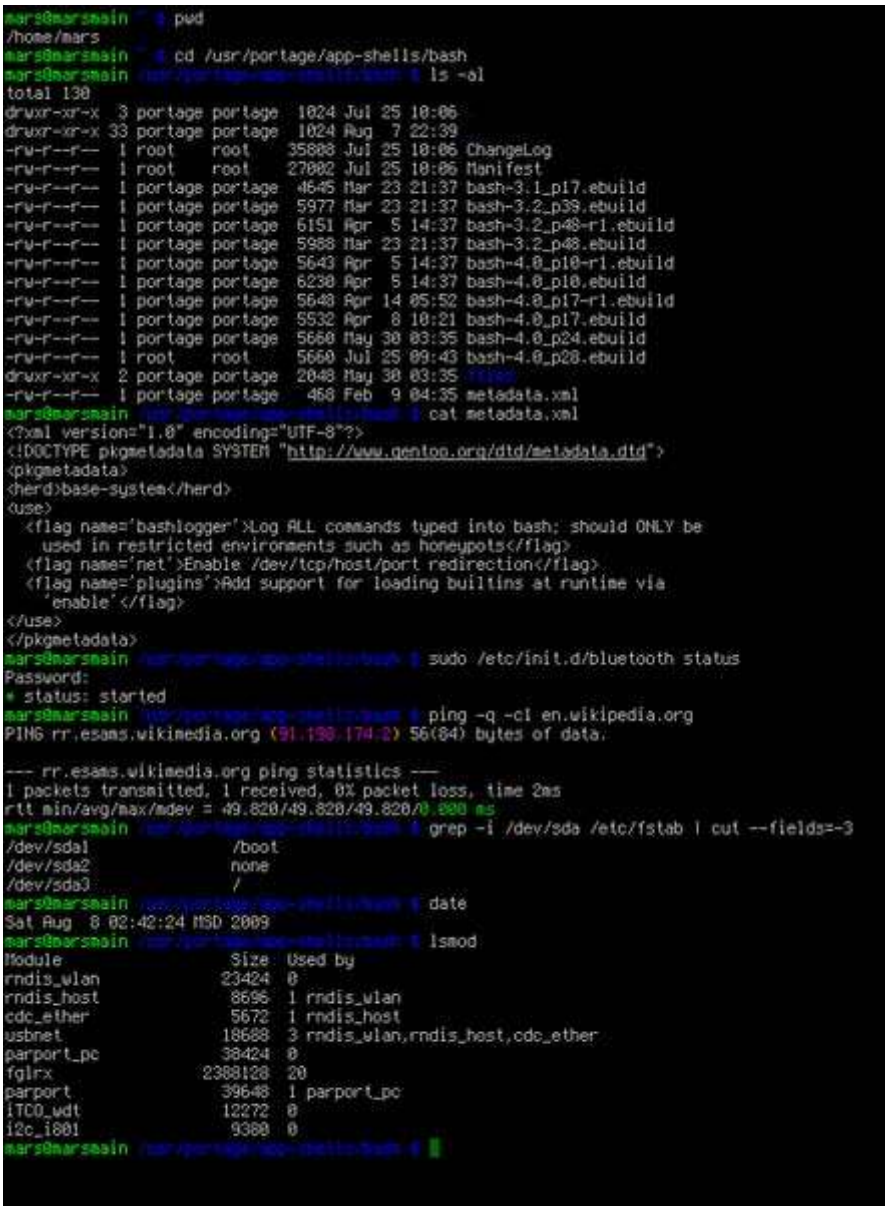
A GUI may be designed for the rigorous requirements of a vertical market. This is known as an "application specific graphical user interface." Among early application specific GUIs was Gene Mosher's 1986 Point of Sale touchscreen GUI. Other examples of an application specific GUIs are:

- Self-service checkouts used in a retail store
- Automated teller machines (ATM)
- Airline self-ticketing and check-in
- Information kiosks in a public space, like a train station or a museum
- Monitors or control screens in an embedded industrial application which employ a real time operating system (RTOS).

The latest cell phones and handheld game systems also employ application specific touchscreen GUIs. Newer automobiles use GUIs in their navigation systems and touch screen multimedia centers.

Comparison to other interfaces

Command-line interfaces



Modern CLI

GUIs were introduced in reaction to the steep learning curve of command-line interfaces (CLI), which require commands to be typed on the keyboard. Since the commands available in command line interfaces can be numerous, complicated operations can be completed using a short sequence of words and symbols. This allows for greater efficiency and productivity once many commands are learned, but reaching this level takes some time because the command words are not easily discoverable and not mnemonic. WIMPs ("window, icon, menu, pointing device"), on the other hand, present the user with numerous widgets that represent and can trigger some of the system's available commands.

WIMPs extensively use modes as the meaning of all keys and clicks on specific positions on the screen are redefined all the time. Command line interfaces use modes only in limited forms, such as the current directory and environment variables.

Most modern operating systems provide both a GUI and some level of a CLI, although the GUIs usually receive more attention. The GUI is usually WIMP-based, although occasionally other metaphors surface, such as those used in Microsoft Bob, 3dwm or File System Visualizer (FSV).

Applications may also provide both interfaces, and when they do the GUI is usually a WIMP wrapper around the command-line version. This is especially common with applications designed for Unix-like operating systems. The latter used to be implemented first because it allowed the developers to focus exclusively on their product's functionality without bothering about interface details such as designing icons and placing buttons. Designing programs this way also allows users to run the program non-interactively, such as in a shell script.

Three-dimensional user interfaces

For typical computer displays, *three-dimensional* is a misnomer—their displays are two-dimensional. Semantically, however, most graphical user interfaces use three dimensions - in addition to height and width, they offer a third dimension of layering or stacking screen elements over one another. This may be represented visually on screen through an illusionary transparent effect, which offers the advantage that information in background windows may still be read, if not interacted with. Or the environment may simply hide the background information, possibly making the distinction apparent by drawing a drop shadow effect over it.

Some environments use the techniques of 3D graphics to project virtual three dimensional user interface objects onto the screen. As the processing power of computer graphics hardware increases, this becomes less of an obstacle to a smooth user experience.

Motivation

Three-dimensional GUIs are quite common in science fiction literature and movies, such as in *Jurassic Park*, which features Silicon Graphics' three-dimensional file manager, "File system navigator", an actual file manager that never got much widespread use as the user interface for a Unix computer. In fiction, three-dimensional user interfaces are often immersible environments like William Gibson's Cyberspace or Neal Stephenson's Metaverse.

Three-dimensional graphics are currently mostly used in computer games, art and computer-aided design (CAD). There have been several attempts at making three-dimensional desktop environments like Sun's Project Looking Glass or SphereXP from Sphere Inc. A three-dimensional computing environment could possibly be used for

collaborative work. For example, scientists could study three-dimensional models of molecules in a virtual reality environment, or engineers could work on assembling a three-dimensional model of an airplane. This is a goal of the Croquet project and Project Looking Glass.

Technologies

The use of three-dimensional graphics has become increasingly common in mainstream operating systems, from creating attractive interfaces—eye candy— to functional purposes only possible using three dimensions. For example, user switching is represented by rotating a cube whose faces are each user's workspace, and window management is represented via a Rolodex-style flipping mechanism in Windows Vista. In both cases, the operating system transforms windows on-the-fly while continuing to update the content of those windows.

Interfaces for the X Window System have also implemented advanced three-dimensional user interfaces through compositing window managers such as Beryl, Compiz and KWin using the AIGLX or XGL architectures, allowing for the usage of OpenGL to animate the user's interactions with the desktop.

Another branch in the three-dimensional desktop environment is the three-dimensional GUIs that take the desktop metaphor a step further, like the BumpTop, where a user can manipulate documents and windows as if they were "real world" documents, with realistic movement and physics.

The Zooming User Interface (ZUI) is a related technology that promises to deliver the representation benefits of 3D environments without their usability drawbacks of orientation problems and hidden objects. It is a logical advancement on the GUI, blending some three-dimensional movement with two-dimensional or "2.5D" vector objects.

Multimodal interaction

Multimodal interaction provides the user with multiple modes of interfacing with a system.

Multimodal input

Two major groups of multimodal interfaces have merged. The first group of interfaces combined various user input modes beyond the traditional keyboard and mouse input/output, such as speech, pen, touch, manual gestures, gaze and head and body movements. The most common such interface combines a visual modality (e.g. a display, keyboard, and mouse) with a voice modality (speech recognition for input, speech synthesis and recorded audio for output). However other modalities, such as pen-based

input or haptic input/output may be used. Multimodal user interfaces are a research area in human-computer interaction (HCI).

The advantage of multiple input modalities is increased usability: the weaknesses of one modality are offset by the strengths of another. On a mobile device with a small visual interface and keypad, a word may be quite difficult to type but very easy to say (e.g. Poughkeepsie). Consider how you would access and search through digital media catalogs from these same devices or set top boxes. And in one real-world example, patient information in an operating room environment is accessed verbally by members of the surgical team to maintain an antiseptic environment, and presented in near realtime aurally and visually to maximize comprehension.

Multimodal input user interfaces have implications for accessibility. A well-designed multimodal application can be used by people with a wide variety of impairments. Visually impaired users rely on the voice modality with some keypad input. Hearing-impaired users rely on the visual modality with some speech input. Other users will be "situationally impaired" (e.g. wearing gloves in a very noisy environment, driving, or needing to enter a credit card number in a public place) and will simply use the appropriate modalities as desired. On the other hand, a multimodal application that requires users to be able to operate all modalities is very poorly designed.

The most common form of input multimodality in the market makes use of the XHTML+Voice (aka X+V) Web markup language, an open specification developed by IBM, Motorola, and Opera Software. X+V is currently under consideration by the W3C and combines several W3C Recommendations including XHTML for visual markup, VoiceXML for voice markup, and XML Events, a standard for integrating XML languages. Multimodal browsers supporting X+V include IBM WebSphere Everyplace Multimodal Environment, Opera for Embedded Linux and Windows, and ACCESS Systems NetFront for Windows Mobile. To develop multimodal applications, software developers may use a software development kit, such as IBM WebSphere Multimodal Toolkit, based on the open source Eclipse framework, which includes an X+V debugger, editor, and simulator.

Multimodal input and output

The second group of multimodal systems presents users with multimedia displays and multimodal output, primarily in the form of visual and auditory cues. Interface designers have also started to make use of other modalities, such as touch and olfaction. Proposed benefits of multimodal output system include synergy and redundancy. The information that is presented via several modalities is merged and refers to various aspects of the same process. The use of several modalities for processing exactly the same information provides an increased bandwidth of information transfer. Currently, multimodal output is used mainly for improving the mapping between communication medium and content and to support attention management in data-rich environment where operators face considerable visual attention demands.

An important step in multimodal interface design is the creation of natural mappings between modalities and the information and tasks. The auditory channel differs from vision in several aspects. It is omnidirection, transient and is always reserved . Speech output, one form of auditory information, received considerable attention. Several guidelines have been developed for the use of speech. Michaelis and Wiggins (1982) suggested that speech output should be used for simple short messages that will not be referred to later. It was also recommended that speech should be generated in time and require an immediate response.

The sense of touch was first utilized as a medium for communication in the late 1950s . It is not only a promising but also a unique communication channel. In contrast to vision and hearing, the two traditional senses employed in HCI, the sense of touch is proximal: it senses objects that are in contact with the body, and it is bidirectonal in that it supports both perception and acting on the environment.

Examples of auditory feedback include auditory icons in computer operating systems indicating users' actions (e.g. deleting a file, open a folder, error), speech output for presenting navigational guidance in vehicles, and speech output for warning pilots on modern airplane cockpits. Examples of tactile signals include vibrations of the turn-signal lever to warn drivers of a car in their blind spot, the vibration of auto seat as a warning to drivers, and the stick shaker on modern aircraft alerting pilots to an impending stall.

Invisible interface spaces became available using sensor technology. Infrared, ultrasound and cameras are all now commonly used (e.g., http://doi.acm.org/10.1145/1282040.1282059). Transparency of interfacing with content is enhanced providing an immediate and direct link via meaningful mapping is in place, thus the user has direct and immediate feedback to input and content response becomes interface affordance (Gibson 1979).

Chapter- 4

Dialog System Applications

Edutainment

Edutainment (also **Info-tainment** or **entertainment-education**) is a form of entertainment designed to educate as well as to amuse.

Overview

It can be argued that edutainment has existed for millennia in the form of parables and fables that promoted social change.

Modern forms include television productions, film, museum exhibits, and computer software which use entertainment to attract and maintain an audience, while incorporating deliberate educational content or messages.

Since the 1970s, various groups in the United States, the United Kingdom, and Latin America have used edutainment to address such health and social issues as substance abuse, immunization, teenage pregnancy, HIV/AIDS, and cancer. Initiatives in major universities, such as Johns Hopkins University and the University of Wisconsin-Madison, NGOs such as PCI-Media Impact, and government agencies such as the Center for Disease Control (CDC) have produced edutainment content.

One form of edutainment popular in Latin America is the educational telenovela. Miguel Sabido, a producer of telenovelas from the 1970s on, has combined communication theory with pro-health/education messages to educate audiences throughout Latin America about family planning, literacy, and other topics. He developed a model which incorporated the work of Albert Bandura and other theorists, as well as research to determine whether programs impacted audience behavior.

Etymology

The term edutainment was used as early as 1948 by The Walt Disney Company to describe the *True Life Adventures* series..

The noun *edutainment* is a neologistic portmanteau used by Robert Heyman in 1973 while producing documentaries for the National Geographic Society. It was also used by Dr. Chris Daniels in 1975 to encapsulate the theme of his Millennium Project. This project later became known as The Elysian World Project.

Edutainment can be used to describe various learning modules.

Theories

Entertainment-Education uses a blend of core communication theories and fundamental entertainment pedagogy to guide the preparation of the programming. Additionally the CDC has a tip sheet available on its website that provides additional guidance for writers and producers .

The major communication theories that influence Entertainment-Education include:

- Persuasion Theory: (Aristotle, Petty, Cacioppo) Psychological characteristics affect the response of a person to messages. Also indicates the message and source factors that influence a person's response such as the credibility, attractiveness, and expertise of the source.
- Theory of Reasoned Action: (Ajzen, Fishbein) Social influences affect behavior, including beliefs and perceived social norms.
- Social Learning Theory: (Bandura) People learn by observing others and the consequences of their behavior. If the person so chooses, they then emulate the behavior by rehearsing the action, taking action, comparing their experiences to the experiences of others, and then adopting the new behavior.
- Diffusion Theory: (Rogers) Behavior spreads through a community or group over a period of time. Television may plant the idea, but social networks reinforce it and cause it to grow.

Pedagogy involved with Entertainment-Education include:

- Relevance: Learning is more likely when people can see the usefulness of the knowledge they are given.
- Incremental Learning: Learning is most effective when people can learn at their own pace.
- Distributed Learning: (Fossard) Different people learn in different ways over different periods of time. It is important to present information differently so that people can absorb it.

In media

Video games

In 1983, the term "edutainment" was used to describe a package of software games for the Oric 1 and Spectrum Microcomputers in the UK. Dubbed "arcade edutainment" an advertisement for the package can be found in various issues of "Your Computer" magazine from 1983. The software package was available from Telford ITEC a government sponsored training program. The originator of the name was Chris Harvey who worked at ITEC at the time. Since then, many other computer games such as Electronic Arts computer game *Seven Cities of Gold*, released 1984, have also used the term *edutainment* to describe their product. Most edutainment games seek to teach players using a game based learning approach. Criticism as to what video games can be considered "educational" has led to the creation of Serious games whose primary focus is to teach rather than entertain.

Simon Egenfeldt-Nielsen (PhD, Psychologist) has spent a great deal of time researching the educational use and potential of computer games and has written many articles on the subject. One paper dealing specifically with Edutainment breaks it down into 3 generational categories to separate the cognitive methods most predominantly used to teach. In his papers he is critical of the research that has been done in the areas of the educational use of computer games cited their biases and weaknesses in method causing them to lack scientific validity in their findings.

Film and television programming

Motion pictures with educational contents appeared as early as 1943, such as Private Snafu, and can still be seen in modern films such as An Inconvenient Truth.

After World War II, edutainment shifted towards television. The presence of edutainment is especially evident in children's television series, such as *Sesame Street*, *Dora the Explorer*, and *Teletubbies*. For older viewers, individual situation comedy episodes also occasionally serve as edutainment vehicles. These episodes are sometimes described in United States television commercial parlance as very special episodes. The American sitcom *Happy Days* produced an especially effective edutainment episode which was reported to have prompted a 600% increase in the U.S. demand for library cards. As early as the 1950s, children-aimed shows like "Watch Mr Wizard" were made which could be considered edutainment.

Discovery Channel is also known for its various shows that follow the theme, such as MythBusters.

There are many television programs that incorporate Entertainment-Education as well. The Sentinel Award, which is administered by the University of Southern California's Annenberg Center for Communication, the CDC and the National Cancer Institute (NCI),

is given each year to programs that address health and medical issues in their storylines. 2006's nominees/winners include:

- Numb3rs - for a storyline about the shortage of organ donations.
- Grey's Anatomy - for story lines about organ transplantation and cancer.
- As the World Turns - for a breast cancer storyline that involved a major character.
- The George Lopez Show - for a storyline about a kidney transplant.
- Don Pedro's Diabetes - a telenovela about a major character's struggle with diet, exercise, and medication to control diabetes.
- Ben & Izzy - for a storyline about two children, Ben from America and Izzy from Jordan, who form a close bond despite their different cultural backgrounds.

Radio

Radio can also serve as an effective vehicle for edutainment. The British radio soap opera *The Archers* has for decades been systematically educating its audience on agricultural matters; likewise, the Tanzanian radio soap opera *Twende na Wakati* ("Let's Go With the Times") was written primarily to promote family planning.

Other successful radio programs that have incorporated Entertainment-Education principles include:

- DJ Nihal's BBC Radio 1 radio show which centered around 'edutainment'. He mentions this term each time the show is broadcast.
- "The Lawsons/Blue Hills" - a radio program that was designed to help Australian farmers adjust to new farming methods.
- "Tinka Tinka Sukh" - a Hindi-language radio program that results in environmental and health improvements in India.
- Soul City - A successful South African radio serial drama that carried AIDS prevention messages.
- The Donut Shop - A successful internet radio show talk about educational games that they think could be used in today's schools.
- Radio Ado and its radiodrama "Pildoritas de la Vida Real" a Mexican radio soap opera designed to disseminate sexual education among teenagers. This radiodrama was produced by the University of Guadalajara and teenagers from Morelia, Michoacan, Mexico.

Cultural impact

Some college professors have adopted the practice of edutainment in order to keep the interest of adult students in long classroom lectures . Here the instructor entertains the students while meeting course objectives. An important teaching technique of education is to use variety, by utilizing various mediums such as video, in-class skits, demonstrations, and Power Point slides along with lectures. Within the lecture, the instructor can add comedy and discussions of personal experiences of the professor or students.

Corporate learning

The concept of Edutainment is also being used for building e learning programs for organizations also.High technology is used to make the programs entertaining and educational.One of the companies offering such services is Eduledge Solutions.

Criticism

Edutainment is also a growing paradigm within the science museum community in the United States. This approach emphasizes fun and enjoyment, often at the expense of educational content. The idea is that people are used to flashy, polished entertainment venues like movie theaters and theme parks that they demand similar experiences at science centers and museums. Thus, a museum is seen as just another business competing for entertainment dollars from the public, rather than as an institution that serves the public welfare through education or historical preservation.

WT

Chapter- 5

Dialog System Devices

Personal digital assistant

The Palm TX

EO Personal Communicator (440) from AT&T

A **personal digital assistant** (**PDA**), also known as a **palmtop computer**, is a mobile device that functions as a personal information manager. Current PDAs often have the ability to connect to the Internet. A PDA has an electronic visual display, enabling it to include a web browser, but some newer models also have audio capabilities, enabling them to be used as mobile phones or portable media players. Many PDAs can access the Internet, intranets or extranets via Wi-Fi or Wireless Wide Area Networks. Many PDAs employ touchscreen technology.

The term *PDA* was first used on January 7, 1992 by Apple Computer CEO John Sculley at the Consumer Electronics Show in Las Vegas, Nevada, referring to the Apple Newton. In 1996, Nokia introduced the first mobile phone with full PDA functionality, the 9000 Communicator, which grew to become the world's best-selling PDA. The Communicator spawned a new category of mobile phones: the smartphone. Today, the vast majority of all PDAs are smartphones. Over 150 million smartphones are sold each year, while "stand-alone" PDAs without phone functionality sell only about 3 million units per year. Popular smartphone brands include HTC, Apple, Palm, Nokia N-Series, and RIM BlackBerry.

Typical features

A typical PDA has a touchscreen for entering data, a memory card slot for data storage, and IrDA, Bluetooth and/or Wi-Fi. However, some PDAs may not have a touch screen, using softkeys, a directional pad, and a numeric keypad or a thumb keyboard for input; this is typically seen on telephones that are incidentally PDAs.

In order to have the functions expected of a PDA, a device's software typically includes an appointment calendar, a to-do list, an address book for contacts, and some sort of memo (or "note") program. PDAs with wireless data connections also typically include an email client and a Web browser.

Touch screen

Many of the original PDAs, such as the Apple Newton and Palm Pilot, featured a touchscreen for user interaction, having only a few buttons—usually reserved for shortcuts to often-used programs. Touchscreen PDAs, including Windows Mobile devices, may have a detachable stylus to facilitate making selections. The user interacts with the device by tapping the screen to select buttons or issue commands, or by dragging a finger or the stylus on the screen to make selections or scroll.

Typical methods of entering text on touchscreen PDAs include:

- A **virtual keyboard**, where a keyboard is shown on the touchscreen. Text is entered by tapping the on-screen keyboard with a finger or stylus.
- An **external keyboard** connected via USB, Infrared, or Bluetooth. Some users may choose a chorded keyboard for one-handed use.
- **Handwriting recognition**, where letters or words are written on the touchscreen, and the PDA converts the input to text. Recognition and computation of handwritten horizontal and vertical formulas, such as "1 + 2 =", may also be a feature.
- **Stroke recognition** allows the user to make a predefined set of strokes on the touchscreen, sometimes in a special input area, representing the various characters to be input. The strokes are often simplified character shapes, making them easier for the device to recognize. One widely-known stroke recognition system is Palm's Graffiti).

Despite rigorous research and development projects, end-users experience mixed results with handwriting recognition systems. Some find it frustrating and inaccurate, while others are satisfied with the quality of the recognition.

Touchscreen PDAs intended for business use, such as the BlackBerry and Palm Treo, usually also full keyboards and scroll wheels or thumbwheels to facilitate data entry and navigation.

Many touchscreen PDAs support some form of external keyboard as well. Specialized folding keyboards, which offer a full-sized keyboard but collapse into a compact size for transport, are available for many models. External keyboards may attach to the PDA directly, using a cable, or may use wireless technology such as infrared or Bluetooth to connect to the PDA.

Newer PDAs, such as the Apple iPhone, Apple iPod Touch, HTC HD2, and Palm Pre, include more advanced forms of touchscreen that can register multiple touches simultaneously. These "multi-touch" displays allow for more sophisticated interfaces using various gestures entered with one or more fingers.

Memory cards

Although many early PDAs did not have memory card slots, now most have either some form of Secure Digital (SD) slot or a CompactFlash slot. Although originally designed for memory, Secure Digital Input/Output (SDIO) and CompactFlash cards are available that provide accessories like Wi-Fi or digital cameras, if the device can support them. Some PDAs also have a USB port, mainly for USB flash drives. Some PDAs use microSD cards, which are electronically compatible with SD cards, but have a much smaller physical size.

Wired connectivity

While early PDAs connected to a user's personal computer via serial ports or another proprietary connection, many today connect via a USB cable. PDAs are not typically able to connect to each other via USB, as USB requires one machine to act as a "host," which isn't a typical PDA function.

Some early PDAs were able to connect to the Internet indirectly by means of an external modem connected via the PDA's serial port or "sync" connector, or directly by using an expansion card that provided an Ethernet port.

Wireless connectivity

Most modern PDAs have Bluetooth a popular wireless protocol for mobile devices. Bluetooth can be used to connect keyboards, headsets, GPS receivers, and other nearby accessories. It's also possible to transfer files between PDAs that have Bluetooth.

Many modern PDAs have Wi-Fi wireless network connectivity, and can connect to Wi-Fi hotspots.

All smartphones, and some other modern PDAs like the Apple iPod touch, can connect to Wireless Wide Area Networks, such as those provided by cellular telecommunications companies.

Older PDAs typically had an IrDA (infrared) port allowing short-range, line-of-sight wireless communication. Few current models use this technology, as it has been supplanted by Bluetooth and Wi-Fi. IrDA allows communication between two PDAs, or between a PDA and any device with an IrDA port or adapter. Some printers have IrDA receivers, allowing IrDA-equipped PDAs to print to them, if the PDA's operating system supports it. Most universal PDA keyboards use infrared technology because many older PDAs have it. Infrared technology is low-cost and has the advantage of being allowed aboard aircraft.

Synchronization

Most PDAs can synchronize their data with applications on a user's personal computer. This allows the user to update contact, schedule, or other information on their computer, using software such as Microsoft Outlook or ACT!, and have that same data transferred to PDA—or transfer updated information from the PDA back to the computer. This eliminates the need for the user to update their data in two places.

Synchronization also prevents the loss of information stored on the device if it is lost, stolen, or destroyed. When the PDA is repaired or replaced, it can be "re-synced" with the computer, restoring the user's data.

Some users find that data input is quicker on their computer than on their PDA, since text input via a touchscreen or small-scale keyboard is slower than a full-size keyboard. Transferring data to a PDA via the computer is therefore a lot quicker than having to manually input all data on the handheld device.

Most PDAs come with the ability to synchronize to a computer. This is done through *synchronization software* provided with the handheld, or sometime with the computer's operating system. Examples of synchronization software include:

- **HotSync Manager**, for Palm OS PDAs
- **Microsoft ActiveSync**, used by Windows XP and older Windows operating systems to synchronize with Windows Mobile, Pocket PC, and Windows CE PDAs, as well as PDAs running iOS, Palm OS, and Symbian
- **Microsoft Windows Mobile Device Center** for Windows Vista, which supports Microsoft Windows Mobile and Pocket PC devices.
- **Apple iTunes**, used on Mac OS X and Microsoft Windows to sync iOS devices (such as the iPhone and iPod touch)
- **iSync**, included with Mac OS X, can synchronize many SyncML-enabled PDAs
- **BlackBerry Desktop Software**, used to sync BlackBerry devices

These programs allow the PDA to be synchronized with a personal information manager, which may be part of the computer's operating system, provided with the PDA, or sold separately by a third party. For example, the RIM BlackBerry comes with RIM's *Desktop Manager* program, which can synchronize to both Microsoft Outlook and ACT!.

Other PDAs come only with their own proprietary software. For example, some early Palm OS PDAs came only with Palm Desktop, while later Palm PDAs—such as the Treo 650—have the ability to sync to Palm Desktop aor Microsoft Outlook. Microsoft's ActiveSync and Windows Mobile Device Center only synchronize with Microsoft Outlook or a Microsoft Exchange server.

Third-party synchronization software is also available for some PDAs from companies like CommonTime and CompanionLink. Third-party software can be used to synchronize

PDAs to other personal information managers that are not supported by the PDA manufacturers (for example, GoldMine and IBM Lotus Notes).

Wireless synchronization

Some PDAs can synchronize some or all of their data using their wireless networking capabilities, rather than having to be directly connected to a personal computer via a cable.

Apple iOS devices, like the iPhone, iPod Touch, and iPad, can use Apple's MobileMe subscription service to synchronize calendar, address book, mail account, Internet bookmark, and other data with one or more Macintosh or Windows computers using Wi-Fi or cellular data connections.

Palm's webOS smartphones primarily sync with the cloud. For example, if Gmail is used, information in contacts, email, and calendar can be synchronized between the phone and Google's servers.

RIM sells BlackBerry Enterprise Server to corporations so that corporate BlackBerry users can wirelessly synchronize their PDAs with the company's Microsoft Exchange Server, IBM Lotus Domino, or Novell GroupWise servers. Email, calendar entries, contacts, tasks, and memos kept on the company's server are automatically synchronized with the BlackBerry.

Automobile navigation

Some PDAs include Global Positioning System (GPS) receivers; this is particularly true of smartphones. Other PDAs are compatible with external GPS-receiver add-ons that use the PDA's processor and screen to display location information.

PDAs with GPS functionality can be used for automotive navigation. PDAs are increasingly being fitted as standard on new cars.

PDA-based GPS can also display traffic conditions, perform dynamic routing, and show known locations of roadside mobile radar guns. TomTom, Garmin, and iGO offer GPS navigation software for PDAs.

Ruggedized PDAs

Some businesses and government organizations rely upon rugged PDAs, sometimes known as enterprise digital assistants (EDAs), for mobile data applications. EDAs often have extra features for data capture, such as barcode readers, radio-frequency identification (RFID) readers, magnetic stripe card readers, or smart card readers.

Typical applications include:

- supply chain management in warehouses
- package delivery
- route accounting
- medical treatment and recordkeeping in hospitals
- facilities maintenance and management
- parking enforcement
- access control and security
- capital asset maintenance
- meter reading by utilities
- "wireless waitress" applications in restaurants and hospitality venues

Medical and scientific uses

Many companies have developed PDA products aimed at the medical professions' unique needs, such as drug databases, treatment information, and medical news. Services such as AvantGo translate medical journals into PDA-readable formats. WardWatch organizes medical records, providing reminders of information such as the treatment regimens of patients and programs to doctors making ward rounds. Pendragon and Syware provide tools for conducting research with PDAs, allowing the user to enter data into a centralized database using their PDA. Microsoft Visual Studio and Sun Java also provide programming tools for developing survey instruments on the handheld. These development tools allow for integration with SQL databases that are stored on the handheld and can be synchronized with a desktop- or server-based database.

PDAs have been shown to aid diagnosis and drug selection and some studies have concluded that when patients use PDAs to record their symptoms, they communicate more effectively with hospitals during follow-up visits.

The development of Sensor Web technology may lead to wearable bodily sensors to monitor ongoing conditions, like diabetes or epilepsy, which would alert patients and doctors when treatment is required using wireless communication and PDAs.

Educational uses

As mobile technology becomes more common, it is increasingly being used as a learning tool. Some educational institutions have embraced **M-Learning**, integrating PDAs into their teaching practices.

PDAs and handheld devices are allowed in many classrooms for digital note-taking. Students can spell-check, modify, and amend their class notes on the PDA. Some educators distribute course material through the Internet or infrared file-sharing functions of the PDA. Textbook publishers have begun to release e-books, or electronic textbooks, which can be uploaded directly to a PDA, reducing the number of textbooks students must carry.

Software companies have developed PDA programs to meet the instructional needs of educational institutions, such as dictionaries, thesauri, word processing software, encyclopedias, and digital lesson planners.

Recreational uses

PDAs may be used by music enthusiasts to play a variety of music file formats. Many PDAs include the functionality of an MP3 player.

Road rally enthusiasts can use PDAs to calculate distance, speed, and time. This information may be used for navigation, or the PDA's GPS functions can be used for navigation.

Underwater divers can use PDAs to plan breathing gas mixtures and decompression schedules using software such as "V-Planner."

PDAs for people with disabilities

PDAs offer varying degrees of accessibility for people with differing abilities, based on the particular device and service. People with vision, hearing, mobility, or speech impairments may be able to use PDAs on a limited basis. This use may be enhanced by accessibility software (e.g., speech recognition for verbal input instead of manual input). Universal design is relevant to PDAs as well as other technology, and a viable solution for many user-access issues, though it has yet to be consistently integrated into the design of popular consumer PDA devices.

PDAs are useful for people with traumatic brain injury or posttraumatic stress disorder, as seen in troops returning home from the Iraq War and Operation Enduring Freedom. PDAs help address memory problems, helping affected people with daily life organization and reminders. As of quite recently, the Department of Veterans' Affairs has issued thousands of PDAs to troops who need them. Occupational therapists have taken on a crucial role within this population helping these veterans

Virtual machine

A **virtual machine** (**VM**) is a software implementation of a machine (i.e. a computer) that executes instructions like a physical machine.

Definitions

When trying to explain the concept of virtual machines, it is first necessary to understand the meaning of the term virtual. Virtual is a term that originally came from optics, to understand objects in a mirror. Objects in a mirror are reflections of an actual physical object but mirrors are not actually that object. This means that the image looks exactly like the actual object and looks to be in the same location

A virtual machine (VM) is a software implementation of a machine (i.e. a computer) that executes programs like a physical machine. Virtual machines are classified in two major categories based on their use and degree of correspondence to any real machine

1. A system virtual machine (SVM) provides support for execution of complete operating system (OS).

2. A process virtual machine (PVM) that provides support for execution of a single process.

A virtual machine was originally defined by Popek and Goldberg as "an efficient, isolated duplicate of a real machine". Current use includes virtual machines which have no direct correspondence to any real hardware.

Virtual machines are separated into two major categories, based on their use and degree of correspondence to any real machine. A **system virtual machine** provides a complete system platform which supports the execution of a complete operating system (OS). In contrast, a **process virtual machine** is designed to run a single program, which means that it supports a single process. An essential characteristic of a virtual machine is that the software running inside is limited to the resources and abstractions provided by the virtual machine—it cannot break out of its virtual world.

System virtual machines

System virtual machines (sometimes called **hardware virtual machines**) allow the sharing of the underlying physical machine resources between different virtual machines, each running its own operating system. The software layer providing the virtualization is called a **virtual machine monitor** or hypervisor. A hypervisor can run on bare hardware (**Type 1** or **native** VM) or on top of an operating system (**Type 2** or **hosted** VM).

The main advantages of system VMs ares:

- multiple OS environments can co-exist on the same computer, in strong isolation from each other
- the virtual machine can provide an instruction set architecture (ISA) that is somewhat different from that of the real machine
- application provisioning, maintenance, high availability and disaster recovery

The main disadvantages of system VMs are:

- a virtual machine is less efficient than a real machine when it accesses the hardware indirectly
- when multiple VMs are concurrently running on the same physical host, each VM may exhibit a varying and unstable performance, which highly depends on the

workload imposed on the system by other VMs, unless proper techniques are used for temporal isolation among virtual machines.

Multiple VMs each running their own operating system (called **guest operating system**) are frequently used in **server consolidation**, where different services that used to run on individual machines in order to avoid interference are instead run in separate VMs on the same physical machine.

The desire to run multiple operating systems was the original motivation for virtual machines, as it allowed time-sharing a single computer between several single-tasking OSes. In some respects, a system virtual machine can be considered a generalization of the concept of virtual memory that historically preceded it. IBM's CP/CMS, the first systems to allow full virtualization, implemented time sharing by providing each user with a single-user operating system, the CMS. Unlike virtual memory, a system virtual machine allowed the user to use privileged instructions in their code. This approach had certain advantages, for instance it allowed users to add input/output devices not allowed by the standard system.

The guest OSes do not have to be all the same, making it possible to run different OSes on the same computer (e.g., Microsoft Windows and Linux, or older versions of an OS in order to support software that has not yet been ported to the latest version). The use of virtual machines to support different guest OSes is becoming popular in embedded systems; a typical use is to support a real-time operating system at the same time as a high-level OS such as Linux or Windows.

Another use is to sandbox an OS that is not trusted, possibly because it is a system under development. Virtual machines have other advantages for OS development, including better debugging access and faster reboots.

Process virtual machines

A process VM, sometimes called an *application virtual machine*, runs as a normal application inside an OS and supports a single process. It is created when that process is started and destroyed when it exits. Its purpose is to provide a platform-independent programming environment that abstracts away details of the underlying hardware or operating system, and allows a program to execute in the same way on any platform.

A process VM provides a high-level abstraction — that of a high-level programming language (compared to the low-level ISA abstraction of the system VM). Process VMs are implemented using an interpreter; performance comparable to compiled programming languages is achieved by the use of just-in-time compilation.

This type of VM has become popular with the Java programming language, which is implemented using the Java virtual machine. Other examples include the Parrot virtual machine, which serves as an abstraction layer for several interpreted languages, and the .NET Framework, which runs on a VM called the Common Language Runtime.

A special case of process VMs are systems that abstract over the communication mechanisms of a (potentially heterogeneous) computer cluster. Such a VM does not consist of a single process, but one process per physical machine in the cluster. They are designed to ease the task of programming parallel applications by letting the programmer focus on algorithms rather than the communication mechanisms provided by the interconnect and the OS. They do not hide the fact that communication takes place, and as such do not attempt to present the cluster as a single parallel machine.

Unlike other process VMs, these systems do not provide a specific programming language, but are embedded in an existing language; typically such a system provides bindings for several languages (e.g., C and FORTRAN). Examples are PVM (Parallel Virtual Machine) and MPI (Message Passing Interface). They are not strictly virtual machines, as the applications running on top still have access to all OS services, and are therefore not confined to the system model provided by the "VM".

Techniques

Emulation of the underlying raw hardware (native execution)

This approach is described as full virtualization of the hardware, and can be implemented using a Type 1 or Type 2 hypervisor. (A Type 1 hypervisor runs directly on the hardware; a Type 2 hypervisor runs on another operating system, such as Linux). Each virtual machine can run any operating system supported by the underlying hardware. Users can thus run two or more different "guest" operating systems simultaneously, in separate "private" virtual computers.

The pioneer system using this concept was IBM's CP-40, the first (1967) version of IBM's CP/CMS (1967–1972) and the precursor to IBM's VM family (1972–present). With the VM architecture, most users run a relatively simple interactive computing single-user operating system, CMS, as a "guest" on top of the VM control program (VM-CP). This approach kept the CMS design simple, as if it were running alone; the control program quietly provides multitasking and resource management services "behind the scenes". In addition to CMS, VM users can run any of the other IBM operating systems, such as MVS or z/OS. z/VM is the current version of VM, and is used to support hundreds or thousands of virtual machines on a given mainframe. Some installations use Linux for zSeries to run Web servers, where Linux runs as the operating system within many virtual machines.

Full virtualization is particularly helpful in operating system development, when experimental new code can be run at the same time as older, more stable, versions, each in a separate virtual machine. The process can even be recursive: IBM debugged new versions of its virtual machine operating system, VM, in a virtual machine running under an older version of VM, and even used this technique to simulate new hardware.

The standard x86 processor architecture as used in modern PCs does not actually meet the Popek and Goldberg virtualization requirements. Notably, there is no execution mode

where all sensitive machine instructions always trap, which would allow per-instruction virtualization.

Despite these limitations, several software packages have managed to provide virtualization on the x86 architecture, even though dynamic recompilation of privileged code, as first implemented by VMware, incurs some performance overhead as compared to a VM running on a natively virtualizable architecture such as the IBM System/370 or Motorola MC68020. By now, several other software packages such as Virtual PC, VirtualBox, Parallels Workstation and Virtual Iron manage to implement virtualization on x86 hardware.

Intel and AMD have introduced features to their x86 processors to enable virtualization in hardware.

Emulation of a non-native system

Virtual machines can also perform the role of an emulator, allowing software applications and operating systems written for another computer processor architecture to be run.

Some virtual machines emulate hardware that only exists as a detailed specification. For example:

- One of the first was the p-code machine specification, which allowed programmers to write Pascal programs that would run on any computer running virtual machine software that correctly implemented the specification.
- The specification of the Java virtual machine.
- The Common Language Infrastructure virtual machine at the heart of the Microsoft .NET initiative.
- Open Firmware allows plug-in hardware to include boot-time diagnostics, configuration code, and device drivers that will run on any kind of CPU.

This technique allows diverse computers to run any software written to that specification; only the virtual machine software itself must be written separately for each type of computer on which it runs.

Operating system-level virtualization

Operating System-level Virtualization is a server virtualization technology which virtualizes servers on an operating system (kernel) layer. It can be thought of as partitioning: a single physical server is sliced into multiple small partitions (otherwise called virtual environments (VE), virtual private servers (VPS), guests, zones, etc.); each such partition looks and feels like a real server, from the point of view of its users.

For example, Solaris Zones supports multiple guest OSes running under the same OS (such as Solaris 10). All guest OSes have to use the same kernel level and cannot run as different OS versions. Solaris native Zones also requires that the host OS be a version of

Solaris; other OSes from other manufacturers are not supported.,however you need to use Solaris Branded zones to use another OSes as zones.

Another example is System Workload Partitions (WPARs), introduced in the IBM AIX 6.1 operating system. System WPARs are software partitions running under one instance of the global AIX OS environment.

The operating system level architecture has low overhead that helps to maximize efficient use of server resources. The virtualization introduces only a negligible overhead and allows running hundreds of virtual private servers on a single physical server. In contrast, approaches such as full virtualization (like VMware) and paravirtualization (like Xen or UML) cannot achieve such level of density, due to overhead of running multiple kernels. From the other side, operating system-level virtualization does not allow running different operating systems (i.e. different kernels), although different libraries, distributions etc. are possible.

List of hardware with virtual machine support

- Alcatel-Lucent 3B20D/3B21D emulated on commercial off-the-shelf computers with **3B2OE** or **3B21E** system
- AMD-V (formerly code-named Pacifica)
- ARM TrustZone
- Boston Circuits gCore (grid-on-chip) with 16 ARC 750D cores and Time-machine hardware virtualization module.
- Freescale PowerPC MPC8572 and MPC8641D
- IBM System/370, System/390, and zSeries mainframes
- Intel VT-x (formerly code-named Vanderpool)
- Sun Microsystems sun4v (UltraSPARC T1 and T2) – utilized by Logical Domains

- HP vPAR and cell based nPAR
- GE Project MAC then
- Honeywell Multics systems
- Honeywell 200/2000 systems Liberator replacing IBM 14xx systems, Level 62/64/66 GCOS
- IBM System/360 Model 145 Hardware emulator for Honeywell 200/2000 systems
- RCA Spectra/70 Series emulated IBM System/360
- NAS CPUs emulated IBM and Amdahl machines
- Honeywell Level 6 minicomputers emulated predecessor 316/516/716 minis
- Xerox Sigma 6 CPUs were modified to emulate GE/Honeywell 600/6000 systems

List of virtual machine software

Template:Col-1

Process (Application) virtual machine software

- Baan Bshell Virtual Machine – Baan 4GL
- Common Language Infrastructure – C#, Visual Basic .NET, J#, C++/CLI (formerly Managed C++)
- Dalvik virtual machine – part of the Android mobile phone platform
- Dis – Inferno operating system and its Limbo programming language
- Dosbox
- EiffelStudio for the Eiffel programming language
- Erlang programming language
- Forth virtual machine – Forth
- Glulx – Glulx, Z-code
- Hec – Hasm Assembler
- Java Virtual Machine – Java, Nice, NetREXX, Scala, Groovy, Clojure, JRuby
- Juke Virtual Machine – A public domain ECMA-335 compatible virtual machine hosted at Google code.
- Low Level Virtual Machine (LLVM) – currently C, C++, Stacker
- Lua
- Macromedia Flash Player – SWF
- Memory Array Redcode Simulator (MARS) – Virtual machine that executes Corewars programs.
- MMIX – MMIXAL
- Neko virtual machine – currently Neko and haXe
- O-code machine – BCPL
- p-code machine – Pascal
- Parrot – Perl 6
- Perl virtual machine – Perl
- CPython – Python
- YARV – Ruby MRI
- Rubinius – Ruby
- ScummVM – Scumm
- SECD machine – ISWIM, Lispkit Lisp
- Sed the stream-editor can also be seen as a VM with 2 storage spaces.
- Smalltalk virtual machine – Smalltalk
- SQLite virtual machine – SQLite opcodes
- Squeak virtual machine – Squeak
- SWEET16
- Tamarin (JavaScript engine) – ActionScript VM in Flash 9
- TrueType virtual machine – TrueType
- V8 (JavaScript engine) - Javascript VM
- Valgrind – checking of memory accesses and leaks in x86/x86-64 code under Linux

- Virtual Processor (VP) from Tao Groumadarchodp (UK).
- VX32 virtual machine – application-level virtualization for native code
- Waba – Virtual machine for small devices, similar to Java
- Warren Abstract Machine – Prolog, CSC GraphTalk
- Z-machine – Z-Code
- Zend Engine – PHP
- libJIT Just-In-Time compilation library – libJIT bytecode
- BytePusher - a ByteByteJump VM

Extended descriptions of selected virtualization software

The following software products are able to virtualize the hardware so that several operating systems can share it.

- Adeos is a Hardware Abstraction Layer that can be loaded as a Kernel Module in Linux. It allows the loading of a real-time kernel as a module, at the same time as Linux but with higher priority.
- Denali uses paravirtualisation to provide high-performance virtual machines on x86 computers. Denali's virtual machines support specialised minimal OSs for Internet services. The system can scale to thousands of virtual machines. Denali does not preserve the application binary interface (ABI), and so applications must be recompiled to run within a library operating system; in this sense it is similar to the Exokernel.
- OKL4 from Open Kernel Labs is designed for use in embedded systems and is primarily deployed in mobile phones. It is the only commercial hypervisor employing capability-based security.
- OpenVZ – Operating System-level server virtualization solution, built on Linux.
- Parallels provides virtualization of x86 for running unmodified PC operating systems. It also uses a lightweight hypervisor technology in order to improve security and to increase the efficiency. Parallels has become popular for its ability to run Windows as a guest under Mac OS X on the Apple–Intel architecture.
- QEMU is a simulator based on a virtual machine, which gives it the ability to emulate a variety of guest CPU architectures on many different host platforms.
- Returnil Virtual System allows you to virtualize your Windows based system and requires only a restart to rebuild the virtual machine.
- VirtualBox is an open source (GPL)/proprietary virtual machine developed by Sun Microsystems. It allows virtualization of x86 and supports various host operating systems including Windows, Linux, BSD and Solaris. It also supports VMware Workstation Virtual Machine Disk Format.
- Virtual Iron provides virtual machines for x86 that run unmodified operating systems, such as Windows, Red Hat and SUSE. Virtual Iron open source virtualization technology implements native virtualization, which delivers near-native performance for x86 operating systems.
- VMware provides virtual machines for x86 that can run unmodified PC operating systems. The technology involved in doing this is complex and also incurs (sometimes significant) performance overheads with hosted VMware products

(VM Server and Workstation). ESX server provides near-native performance and a fully virtualized option (along with para-virtualization of some hardware components). Xen trades running of existing operating systems for running modified (paravirtualized) operating systems with improved performance. Virtual Iron provides full OS compatibility for existing or new OSes with near-native performance without the performance trade-offs between paravirualization and binary translation.

- Xen Virtualization system whose motivation differs from that of Denali in that it is intended to run a moderate number of full-featured operating systems, rather than a large number of specialised, lightweight ones.
- KVM is a Linux kernel module that enables a modified QEMU program to use hardware virtualization.
- libJIT Just-In-Time Compilation library is a library for development of advanced Just-in-time compilation (JIT) in Virtual Machine implementations, Dynamic programming languages, and Scripting languages. Currently it is used for Common Intermediate Language, Ruby, Java, Domain-specific programming languages.

Virtual reality

U.S. Navy personnel using a VR parachute trainer

World Skin (1997), Maurice Benayoun's virtual reality interactive installation

Virtual reality (**VR**) is a term that applies to computer-simulated environments that can simulate physical presence in places in the real world, as well as in imaginary worlds. Most current virtual reality environments are primarily visual experiences, displayed either on a computer screen or through special stereoscopic displays, but some simulations include additional sensory information, such as sound through speakers or headphones. Some advanced, haptic systems now include tactile information, generally known as force feedback, in medical and gaming applications. Furthermore, virtual reality covers remote communication environments which provide virtual presence of users with the concepts of telepresence and telexistence.

Users can interact with a virtual environment or a virtual artifact (VA) either through the use of standard input devices such as a keyboard and mouse, or through multimodal devices such as a wired glove, the Polhemus, and omnidirectional treadmills. The simulated environment can be similar to the real world—for example, in simulations for pilot or combat training—or it can differ significantly from reality, such as in VR games. In practice, it is currently very difficult to create a high-fidelity virtual reality experience, due largely to technical limitations on processing power, image resolution, and communication bandwidth; however, the technology's proponents hope that such limitations will be overcome as processor, imaging, and data communication technologies become more powerful and cost-effective over time.

Virtual reality is often used to describe a wide variety of applications commonly associated with immersive, highly visual, 3D environments. The development of CAD software, graphics hardware acceleration, head mounted displays, database gloves, and miniaturization have helped popularize the notion. In the book *The Metaphysics of Virtual Reality* by Michael R. Heim, seven different concepts of virtual reality are identified: simulation, interaction, artificiality, immersion, telepresence, full-body immersion, and network communication. The definition still has a certain futuristic romanticism attached. People often identify VR with head mounted displays and data suits.

Background

Terminology and concepts

The term "artificial reality", coined by Myron Krueger, has been in use since the 1970s; however, the origin of the term "virtual reality" can be traced back to the French playwright, poet, actor, and director Antonin Artaud. In his seminal book *The Theatre and Its Double* (1938), Artaud described theatre as "*la réalite virtuelle*", a virtual reality "in which characters, objects, and images take on the phantasmagoric force of alchemy's visionary internal dramas". It has been used in *The Judas Mandala*, a 1982 science-fiction novel by Damien Broderick, where the context of use is somewhat different from that defined above. The concept of virtual reality was popularized in mass media by movies such as *Brainstorm* and *The Lawnmower Man*. The VR research boom of the 1990s was accompanied by the non-fiction book *Virtual Reality* (1991) by Howard Rheingold. The book served to demystify the subject, making it more accessible to less technical researchers and enthusiasts, with an impact similar to that which his book *The Virtual Community* had on virtual community research lines closely related to VR. *Multimedia: from Wagner to Virtual Reality*, edited by Randall Packer and Ken Jordan and first published in 2001, explores the term and its history from an avant-garde perspective. Philosophical implications of the concept of VR are systematically discussed in the book *Get Real: A Philosophical Adventure in Virtual Reality* (1998) by Philip Zhai, wherein the idea of VR is pushed to its logical extreme and ultimate possibility. According to Zhai, virtual reality could be made to have an ontological status equal to that of actual reality. *Digital Sensations: Space, Identity and Embodiment in Virtual Reality* (1999), written by Ken Hillis, offers a more critical and theoretical academic assessment of the complex set of cultural and political desires and practices culminating in the development of the technology.

Timeline

Virtual reality can trace its roots to the 1860s, when 360-degree art through panoramic murals began to appear. An example of this would be Baldassare Peruzzi's piece titled, *Sala delle Prospettive*. In the 1920s, vehicle simulators were introduced. Morton Heilig wrote in the 1950s of an "Experience Theatre" that could encompass all the senses in an effective manner, thus drawing the viewer into the onscreen activity. He built a prototype of his vision dubbed the Sensorama in 1962, along with five short films to be displayed in

it while engaging multiple senses (sight, sound, smell, and touch). Predating digital computing, the Sensorama was a mechanical device, which reportedly still functions today. Around this time, Douglas Englebart uses computer screens as both input and output devices. In 1966, Tom Furness introduces a visual flight stimulator for the Air Force. In 1968, Ivan Sutherland, with the help of his student Bob Sproull, created what is widely considered to be the first virtual reality and augmented reality (AR) head mounted display (HMD) system. It was primitive both in terms of user interface and realism, and the HMD to be worn by the user was so heavy it had to be suspended from the ceiling. The graphics comprising the virtual environment were simple wireframe model rooms. The formidable appearance of the device inspired its name, The Sword of Damocles. Also notable among the earlier hypermedia and virtual reality systems was the Aspen Movie Map, which was created at MIT in 1977. The program was a crude virtual simulation of Aspen, Colorado in which users could wander the streets in one of three modes: summer, winter, and polygons. The first two were based on photographs—the researchers actually photographed every possible movement through the city's street grid in both seasons—and the third was a basic 3-D model of the city. In the late 1980s, the term "virtual reality" was popularized by Jaron Lanier, one of the modern pioneers of the field. Lanier had founded the company VPL Research in 1985, which developed and built some of the seminal "goggles and gloves" systems of that decade. In 1991, Antonio Medina, a MIT graduate and NASA scientist, designed a virtual reality system to "drive" Mars rovers from Earth in apparent real time despite the substantial delay of Mars-Earth-Mars signals. The system, termed "Computer-Simulated Teleoperation" as published by Rand, is an extension of virtual reality.

Future

In the short run, the graphics displayed in the HMD will soon reach a point of near visual (but not behavioral) realism. The audio capabilities will move into a new realm of three dimensional sound, which refers to the addition of sound channels both above and below the individual or a Holophony approach. Within existing technological limits, sight and sound are the two senses which best lend themselves to high quality simulation. There are, however, attempts being made to simulate smell. The purpose of current research is linked to a project aimed at treating Post Traumatic Stress Disorder (PTSD) in veterans by exposing them to combat simulations, complete with smells. Although it is often seen in the context of entertainment by popular culture, this illustrates the point that the future of VR is very much tied into therapeutic, training, and engineering demands. Given that fact, a full sensory immersion beyond basic tactile feedback, sight, sound, and smell is unlikely to be a goal in the industry. It is worth mentioning that simulating smells, while realistically possible, requires costly research and development to make each odor, and the machine itself is expensive and specialized, using capsules tailor made for it. As a result, basic and very strong smells such as burning rubber, cordite, and gasoline fumes have been made. Japan's NTT Communications has recently finished testing an Internet-connected odor-delivery system to be used by retailers and restaurants to attract customers. But as new trials and applications are tried out and more data is gathered, Hamada says he is sure the technology "will take communications to a new level in

content richness, compared to today's communications, which only offers images and sounds".

In order to engage the other sense of taste, the brain must be manipulated directly. This would move virtual reality into the realm of simulated reality, similar to the brain interface ports used in *The Matrix*. Although no form of this has been seriously developed at this point, Sony has taken the first step. On April 7, 2005, Sony went public with information that they had filed for, and received a patent for the idea of non-invasive beaming of different frequencies and patterns of ultrasonic waves directly into the brain to recreate all five senses.

Virtual reality is a costly development in technology. Because of this, the future of VR is dependent on whether or not those costs can be reduced in some way. If VR technology becomes affordable, it could be very widespread, but for the moment, major industries are the sole buyers that have the opportunity to utilize this resource.

Impact

There has been an increase in interest in the potential social impact of new technologies, such as virtual reality. Mychilo S. Cline, in his book *Power, Madness, and Immortality: The Future of Virtual Reality*, argues that virtual reality will lead to a number of important changes in human life and activity. He argues that:

- Virtual reality will be integrated into daily life and activity, and will be used in various human ways.
- Techniques will be developed to influence human behavior, interpersonal communication, and cognition.
- As we spend more and more time in virtual space, there will be a gradual "migration to virtual space", resulting in important changes in economics, worldview, and culture.
- The design of virtual environments may be used to extend basic human rights into virtual space, to promote human freedom and well-being, and to promote social stability as we move from one stage in socio-political development to the next.
- Virtual reality can also be used to induce body transfer illusions.

Heritage and archaeology

The use of VR in heritage and archaeology has potential in museum and visitor centre applications, but its use has been tempered by the difficulty in presenting a "quick to learn" real time experience to numerous people at any given time. Many historic reconstructions tend to be in a pre-rendered format to a shared video display, thus allowing more than one person to view a computer generated world, but limiting the interaction that full-scale VR can provide. The first use of a VR presentation in a heritage application was in 1994, when a museum visitor interpretation provided an interactive "walk-through" of a 3D reconstruction of Dudley Castle in England as it was in 1550.

This consisted of a computer controlled laserdisc-based system designed by British based engineer Colin Johnson. It is a little known fact that one of the first users of virtual reality was Queen Elizabeth II, when she officially opened the visitor centre in June 1994. The system was featured in a conference held by the British Museum in November 1994, and in the subsequent technical paper, *Imaging the Past - Electronic Imaging and Computer Graphics in Museums and Archaeology*.

VR reconstruction

Virtual reality enables heritage sites to be recreated extremely accurately, so that the recreations can be published in various media. The original sites are often inaccessible to the public, or may even no longer exist. This technology can be used to develop virtual replicas of caves, natural environment, old towns, monuments, sculptures and archaeological elements.

Therapeutic uses

The primary use of VR in a therapeutic role is its application to various forms of exposure therapy, ranging from phobia treatments to newer approaches to treating PTSD. A very basic VR simulation with simple sight and sound models has been shown to be invaluable in phobia treatment, like zoophobia, and acrophobia, as a step between basic exposure therapy such as the use of simulacra and true exposure. A much more recent application is being piloted by the U.S. Navy to use a much more complex simulation to immerse veterans suffering from PTSD in simulations of urban combat settings. Much as in phobia treatment, exposure to the subject of the trauma or fear leads to desensitization, and a significant reduction in symptoms.

Other research fields in which the use of virtual reality is being explored are physical medicine, rehabilitation, physical therapy, and occupational therapy. In adult rehabilitation, a variety of virtual reality applications are currently being evaluated within upper and lower limb motor rehabilitation for individuals recovering from stroke or spinal cord injury. In pediatrics, the use of virtual reality is being evaluated to promote movement abilities, navigational abilities, or social skills in children with cerebral palsy, acquired brain injury, or other disabilities. Research evidence is emerging rapidly in the field of virtual reality for therapeutic uses. A number of recent reviews published in peer-reviewed journals have summarized the current evidence for the use of Virtual Reality within pediatric and adult rehabilitation. One such review concluded that the field is potentially promising.

Implementation

To develop a real time virtual environment, a computer graphics library can be used as embedded resource coupled with a common programming language, such as C++, Perl, Java, or Python. Some of the most popular computer graphic libraries are OpenGL, Direct3D, Java3D, and VRML, and their use are directly influenced by the system demands in terms of performance, program purpose, and hardware platform. The use of

multithreading can also accelerate 3D performance and enable cluster computing with multi-user interactivity.

Manufacturing

Virtual reality can serve to new product design, helping as an ancillary tool for engineering in manufacturing processes, new product prototypes, and simulation. Among other examples, Electronic Design Automation, CAD, Finite Element Analysis, and Computer Aided Manufacturing are widely utilized programs. The use of Stereolithography and 3D printing shows how computer graphic modeling can be applied to create physical parts of real objects used in naval, aerospace, and automotive industries, which can be seen, for example, in the VR laboratory of VW in Mladá Boleslav. Beyond modeling assembly parts, 3D computer graphics techniques are currently used in the research and development of medical devices for therapies, treatments, patient monitoring, and early diagnoses of complex diseases.

Urban design

3D virtual reality is becoming widely used for urban regeneration and planning and transport projects, especially in East Asia.

Challenges

Virtual reality has been heavily criticized for being an inefficient method for navigating non-geographical information. At present, the idea of ubiquitous computing is very popular in user interface design, and this may be seen as a reaction against VR and its problems. In reality, these two kinds of interfaces have different goals and are complementary. The goal of ubiquitous computing is to bring the computer into the user's world, rather than force the user to go inside the computer. The current trend in VR is actually to merge the two user interfaces to create a fully immersive and integrated experience.