People and Computers: Paradigms of Interaction

People-Oriented Computing

16.9.2019

image source: allottspta.org

Teaching Team

Lecturer:

- Prof. Dr. Elaine M. Huang

Teaching Support Leads:

- Alexander Eiselmayer
- Jan Gugler

Teaching Assistants and Tutors:

- Clara-Maria Barth, Sebastian Frison, Leah Melles, Natalia Obukhova, Sascha Sauer, Alexander Theus, Natalia Werner, Lynn Zumtaugwald

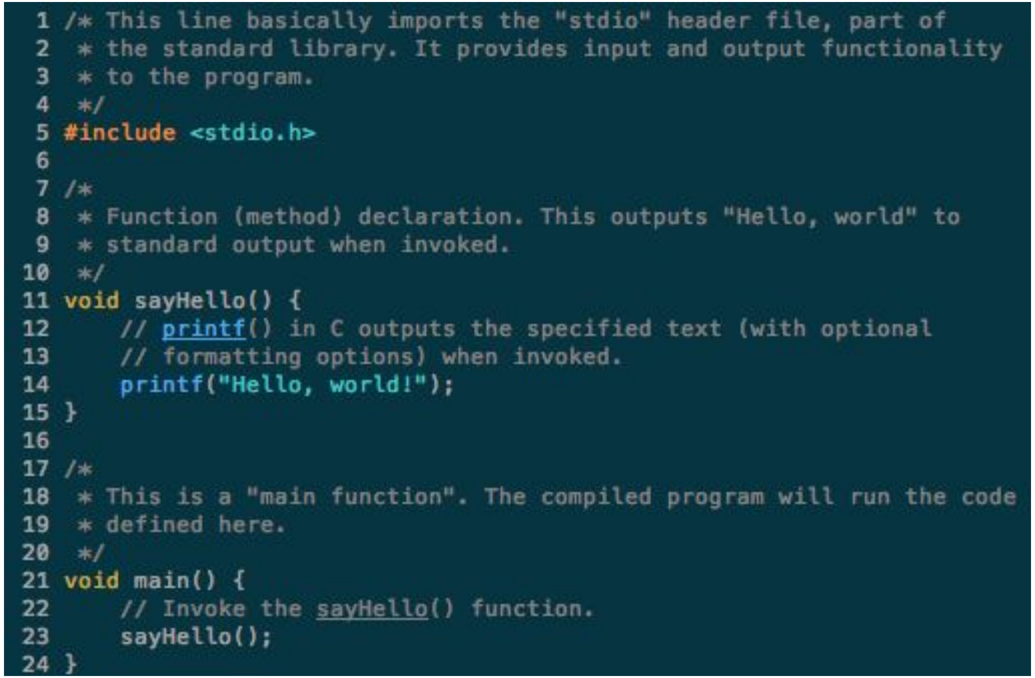
Agenda

- About this course
- Interaction paradigms – a brief history
- Course policies and syllabus

Learning Goals

After this lecture, you should:

- Understand the concept of interaction as communication
- Have familiarity with historical paradigm shifts in interaction with technology
- Be familiar with key systems and figures associated with technology paradigm shifts
- (Be familiar with class policies and expectations)



```c
/* This line basically imports the "stdio" header file, part of
 * the standard library. It provides input and output functionality
 * to the program.
 */
#include <stdio.h>

/*
 * Function (method) declaration. This outputs "Hello, world" to
 * standard output when invoked.
 */
void sayHello() {
    // printf() in C outputs the specified text (with optional
    // formatting options) when invoked.
    printf("Hello, world!");
}

/*
 * This is a "main function". The compiled program will run the code
 * defined here.
 */
void main() {
    // Invoke the sayHello() function.
    sayHello();
}
```

https://commons.wikimedia.org/w/index.php?curid=42810677

People-Oriented Computing

- Computer science is a broad field that has ever increasing impact on our lives and our world
- Humans shape the direction of computing as individuals, as groups, and as a collective society
- Computing in turn shapes what we do, how we work, how we engage with friends and family, how societies function

People-Oriented Computing

- This course provides a sampling of the many areas in which computing affects people, on
    - An individual level
    - A group or organizational level
    - A societal level
- The course touches upon many subject areas that you will have the opportunity to examine in depth later throughout your studies

INTERACTION AND INTERACTION PARADIGMS

source: maybusch.com

source: wsj.com

Communication as Interaction

- How do we communicate and why?
- What is language?
- What else do we use to communicate?
- What do we need in order to communicate?

Communication as Interaction

- Means, ability, and channels of expressing intent, state, or information
- Means, ability, and channels of receiving input
- Enough shared understanding to interpret what is being communicated and respond appropriately

Communication as Interaction

With humans, abilities develop naturally and structures have evolved over a long time

source: cbm.org

Interaction with Computers

Da, wenn man sie braucht:
die Verkaufsstelle fürs Monats-Abo.
STEIG EIN. KOMM WEITER.
ZVV

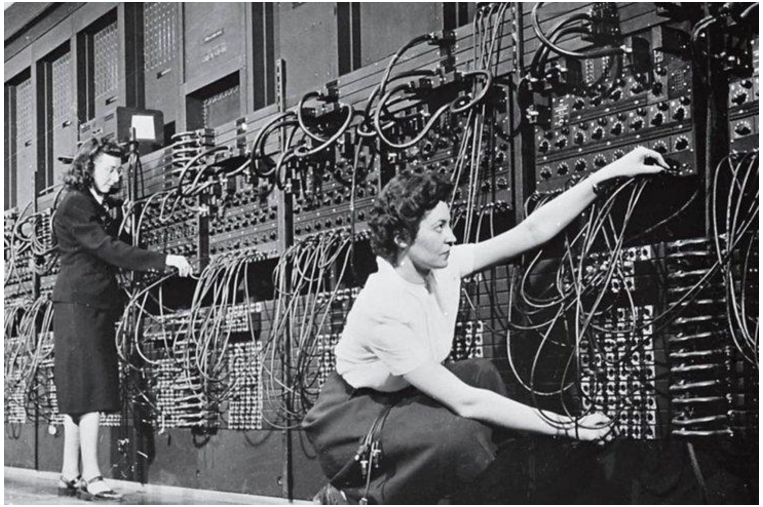
Interaction with Computers

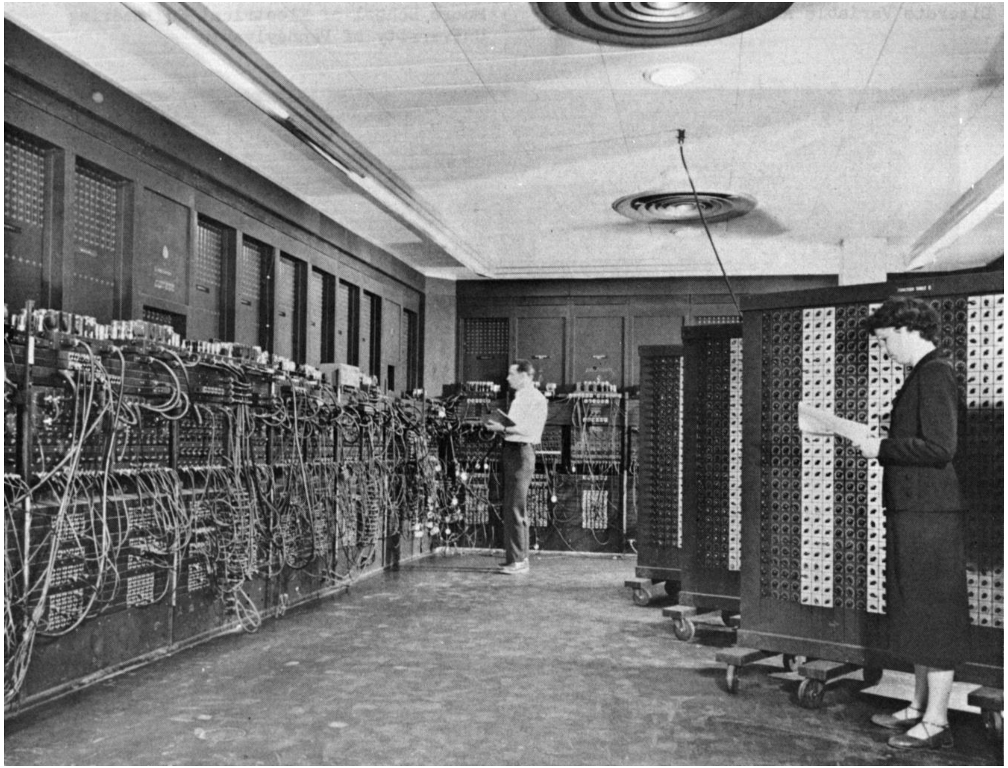
source: zvv.ch

Interaction with Computers

Da, wenn man sie braucht:
die Verkaufsstelle fürs Monats-Abo.
STEIG EIN. KOMM WEITER.
ZVV

Interacting with Computers





source: wikipedia.org

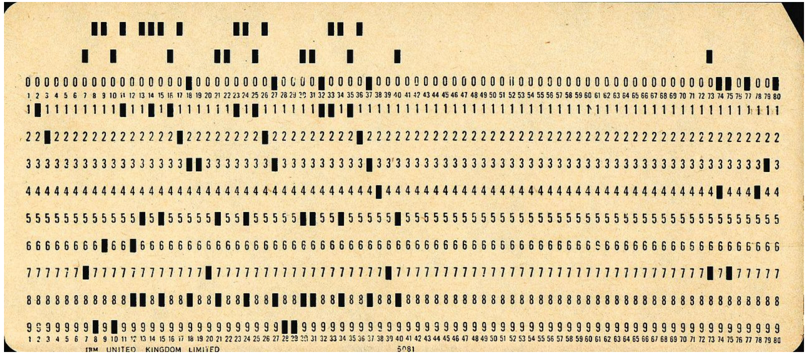
Interaction Paradigms

Successful approaches to interactive systems that have helped make it easier to use technology

Time Sharing (1950s-1960s)

Batch session

- Previous approach
- Individual programmers submitted complete jobs on punched cards or paper tape to an operator
- Operator ran individual jobs on a computer



Time Sharing

- Hardware advances in the 1940s and 1950s led to a massive increase in computing power
  - Mechanical relays -> vacuum electron tubes -> transistors -> integrated chips
- These hardware improvements necessitated parallel advancements in how to harness this power in use
- J.C.R. Licklider at ARPA (Advanced Research Projects Agency) financed research on how to apply computing technology

Time Sharing

Time sharing

- A single computer could support multiple users at once
- Programming became an interactive activity
- Gave rise to the “hacker” who could create increasingly complex programs
- Shift from programming as a pre-planned set of instructions for a computer to an exchange between programmer and computer
- Collaboration between computer and human

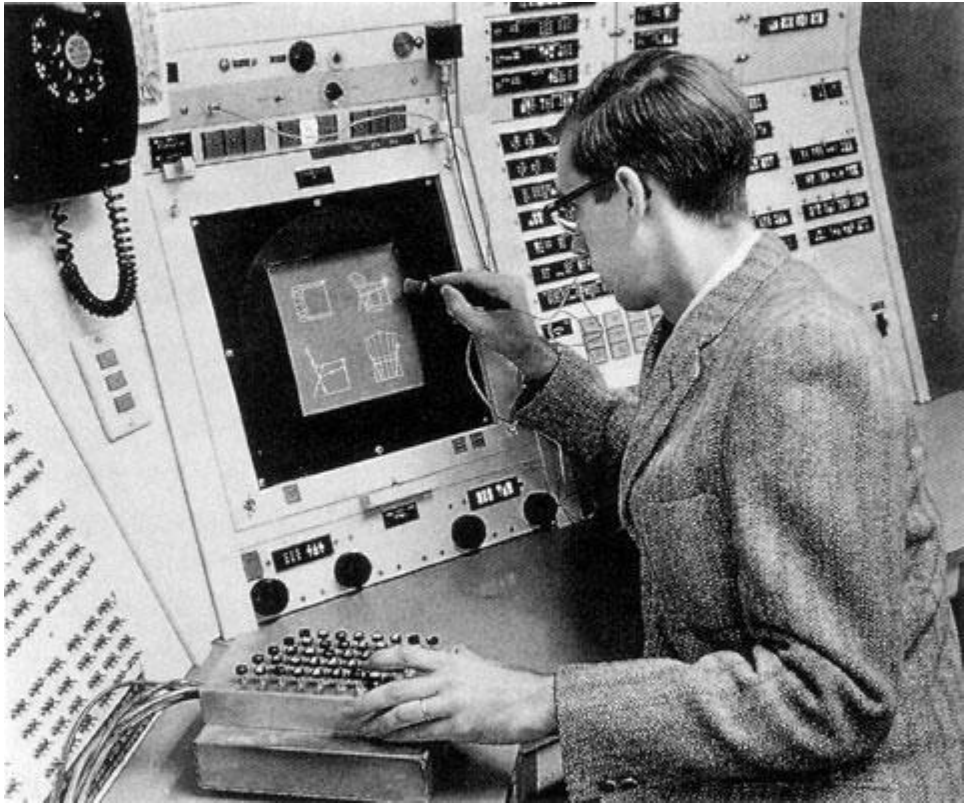
Video Display Units (1950s-1960s)

- Early research in video display units in the 1950s for displaying images for military purposes

Video Display Units

- Breakthrough in 1962 with Ivan Sutherland's *Sketchpad* program, developed at MIT (Massachusetts Institute of Technology)
    - Allowed data to be represented visually, abstracted, manipulated, and changed
    - Enabled truly visual interaction
    - A more human way of interacting with data
    - Computer adapting to human's way of thinking rather than vice versa

Video Display Units



source: history-computer.com

Programming Toolkits (1960s)

- Previous thinking:
    - Computers are complex technology that can only be used by a few experts and specialists

Programming Toolkits

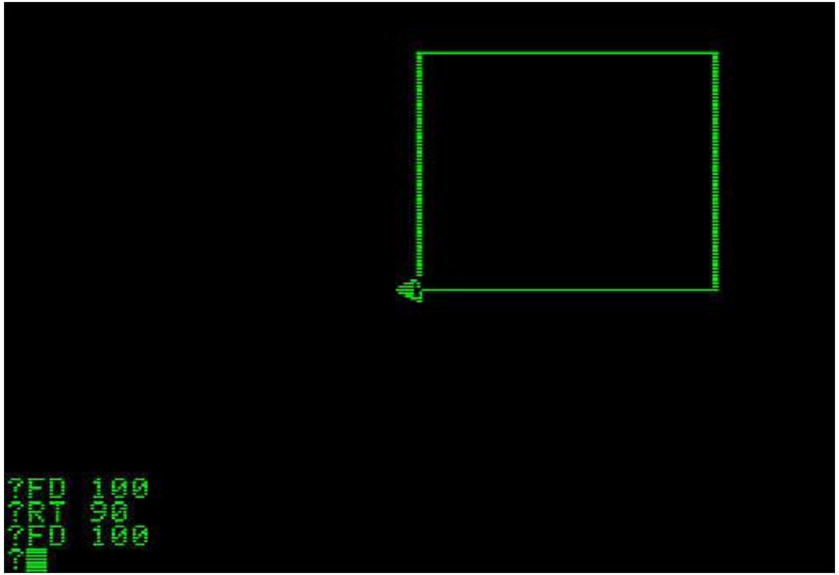
- Douglas Engelbart's (Stanford Research Institute) vision – to enable humans to use computers to learn
- Created programming tools that allow people to create complex programs more easily
- Small programming components can be combined to create larger ones
- Known as "bootstrapping"

Personal Computing (1970s-1980s)

- The notion of computing for the masses
- No need for substantial computing skills in order to benefit from computers

Personal Computing

- Seymour Papert created a programming language for children called LOGO



```
?FD 100
?RT 90
?FD 100
?
```

Personal Computing

- Seymour Papert created a programming language for children called LOGO
  - Demonstrated that powerful tools for hackers could be used by novices
  - Made use of a graphical “turtle” that could be commanded to draw shapes through simple English-based phrases, e.g. “turn left,”
  - Illustrated that ease of use makes a system more powerful

Personal Computing

- Alan Kay believed the future of computing was small, powerful machines dedicated to single users – *personal computers*
- Shift away from mainframe computing and time sharing
- With other PARC (Palo Alto Research Center) researchers, created Smalltalk, a simple but powerful, visually based programming environment especially for personal computing.
- Kay also conceived of the Dynabook in the 1970s, a handheld personal computer for children

Personal Computing

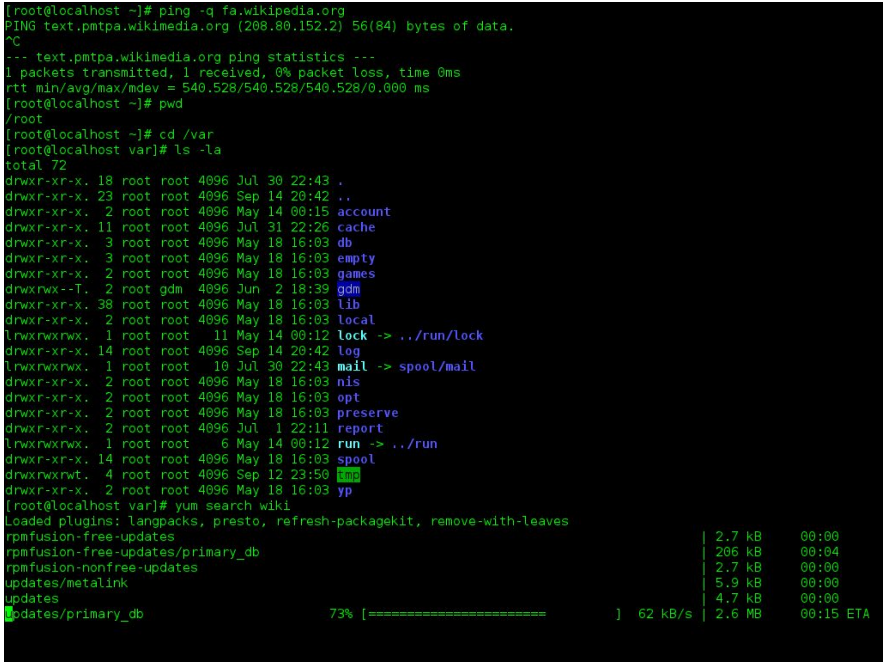
source: iconeye.com

Windows and WIMP (1980s)

- Advent of personal computing led to a focus on increased usability of single-user interaction with computers

Windows and WIMP

Previous interfaces were command-line based



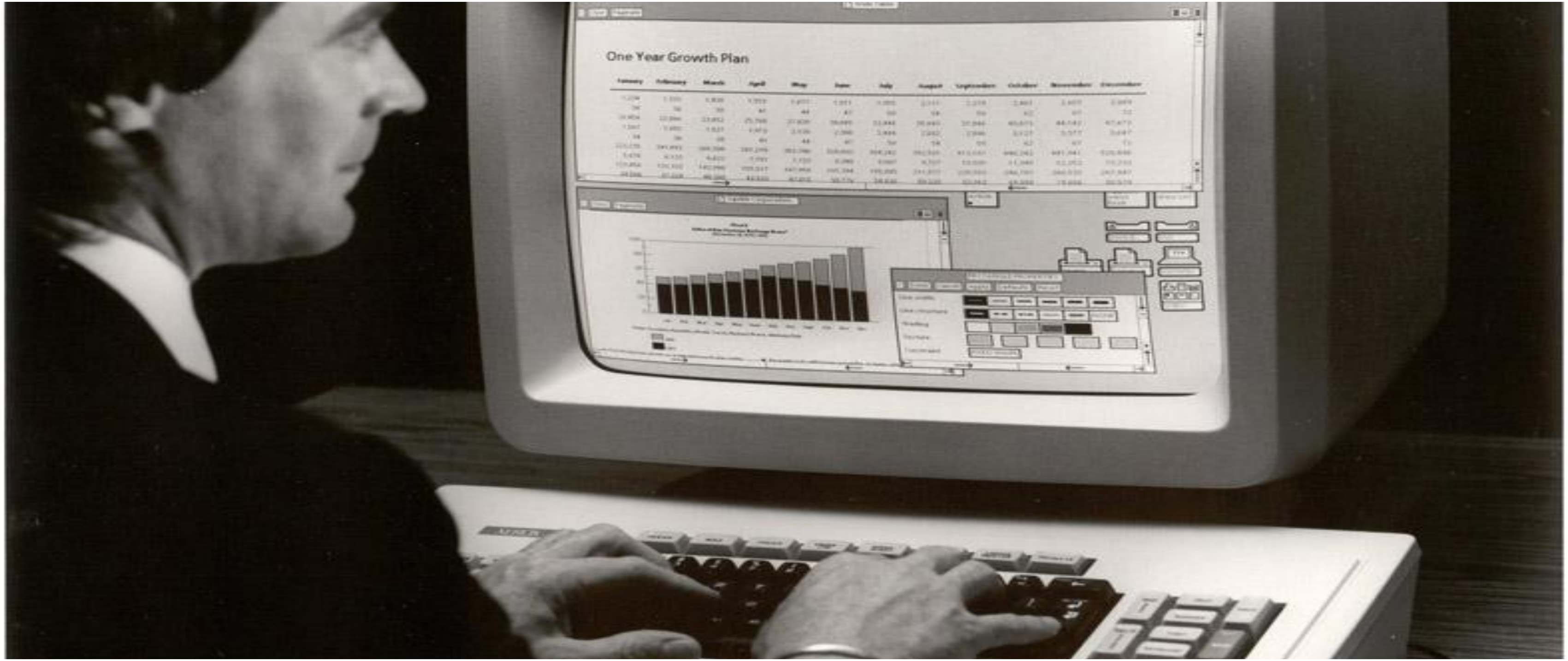
```
[root@localhost ~]# ping -q fa.wikipedia.org
PING text.pmtpa.wikimedia.org (208.80.152.2) 56(84) bytes of data.
^C
--- text.pmtpa.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 540.528/540.528/540.528/0.000 ms
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd /var
[root@localhost var]# ls -la
total 72
drwxr-xr-x. 18 root root 4096 Jul 30 22:43 .
drwxr-xr-x. 23 root root 4096 Sep 14 20:42 ..
drwxr-xr-x.  2 root root 4096 May 14 00:15 account
drwxr-xr-x. 11 root root 4096 Jul 31 22:26 cache
drwxr-xr-x.  3 root root 4096 May 18 16:03 db
drwxr-xr-x.  3 root root 4096 May 18 16:03 empty
drwxr-xr-x.  2 root root 4096 May 18 16:03 games
drwxrwx--T.  2 root gdm  4096 Jun  2 18:39 gdm
drwxr-xr-x. 38 root root 4096 May 18 16:03 lib
drwxr-xr-x.  2 root root 4096 May 18 16:03 local
lrwxrwxrwx.  1 root root   11 May 14 00:12 lock -> ../run/lock
drwxr-xr-x. 14 root root 4096 Sep 14 20:42 log
lrwxrwxrwx.  1 root root   10 Jul 30 22:43 mail -> spool/mail
drwxr-xr-x.  2 root root 4096 May 18 16:03 nis
drwxr-xr-x.  2 root root 4096 May 18 16:03 opt
drwxr-xr-x.  2 root root 4096 May 18 16:03 preserve
drwxr-xr-x.  2 root root 4096 Jul  1 22:11 report
lrwxrwxrwx.  1 root root    6 May 14 00:12 run -> ../run
drwxr-xr-x. 14 root root 4096 May 18 16:03 spool
drwxrwxrwt.  4 root root 4096 Sep 12 23:50 tmp
drwxr-xr-x.  2 root root 4096 May 18 16:03 yp
[root@localhost var]# yum search wiki
Loaded plugins: langpacks, presto, refresh-packagekit, remove-with-leaves
rpmfusion-free-updates                                                                    | 2.7 kB     00:00
rpmfusion-free-updates/primary_db                                                         | 206 kB     00:04
rpmfusion-nonfree-updates                                                                 | 2.7 kB     00:00
updates/metalink                                                                          | 5.9 kB     00:00
updates                                                                                   | 4.7 kB     00:00
updates/primary_db                        73% [=======================         ]  62 kB/s | 2.6 MB     00:15 ETA
```

source: wikipedia.org

Windows and WIMP

- Increased support for engaging in multiple tasks at once, with human in control
- Supporting multiple threads of interaction in conventional command line interfaces became complicated and difficult to manage
- Window-based systems supported physical and logical separation of tasks

Windows and WIMP



source: plyojump.com

Windows and WIMP

- The Xerox Star (Xerox Parc, 1981) computer introduced the first commercial WIMP interface
- Interface based on Windows, Icons, Menus, and Pointers

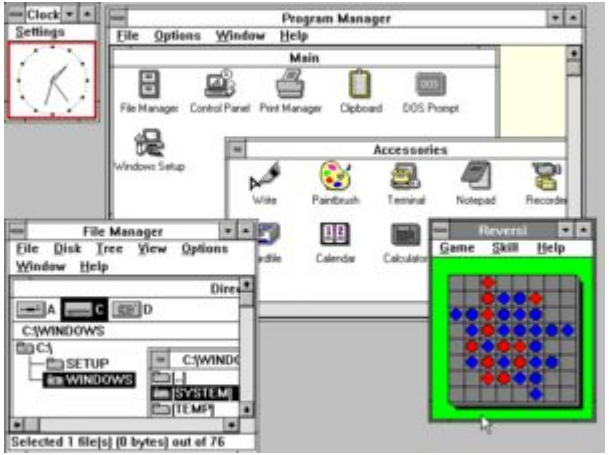
Interface Metaphors

- Metaphors help people learn new concepts by putting them in terms of known concepts
    - E.g. LOGO’s metaphor of a turtle dragging its tail in the dirt
- Metaphors applied to computer interactions make unfamiliar concepts familiar and reduce the perception of complexity or difficulty

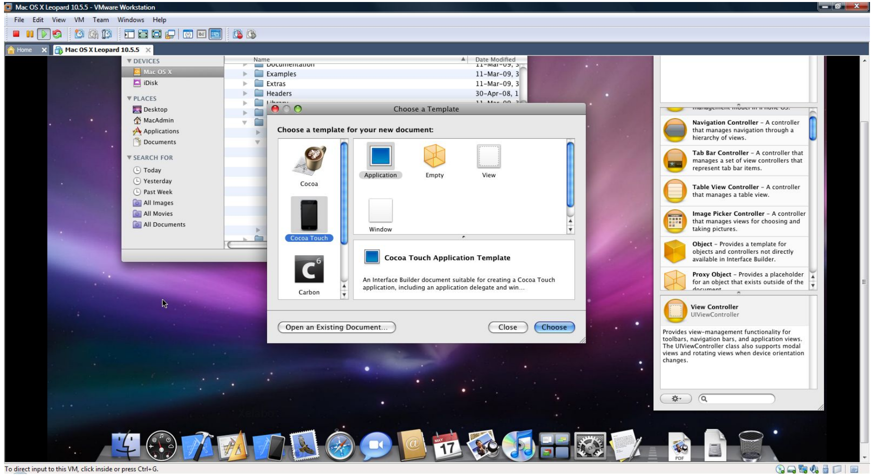
Interface Metaphors

- Window and WIMP interfaces make extensive use of real-world metaphors
  - Windows
  - Buttons
  - Menus
  - Palettes
- Xerox Star and successors made use of an office desktop metaphor
  - Desktop
  - Folders
  - Trash can
  - Etc.

Interface Metaphors



source: wikipedia.org



Choose a template for your new document:

Cocoa Touch Application Template

An Interface Builder document suitable for creating a Cocoa Touch application, including an application delegate and win...

Navigation Controller – A controller that manages navigation through a hierarchy of views.

Tab Bar Controller – A controller that manages a set of view controllers that represent tab bar items.

Table View Controller – A controller that manages a table view.

Image Picker Controller – A controller that manages views for choosing and taking pictures.

Object – Provides a template for objects and controllers not directly available in Interface Builder.

Proxy Object – Provides a placeholder for an object that exists outside of the document.

Provides view-management functionality for toolbars, navigation bars, and application views. The UIViewController class also supports modal views and rotating views when device orientation changes.

To direct input to this VM, click inside or press Ctrl+G.

source: jdeveloper.eu

Interface Metaphors

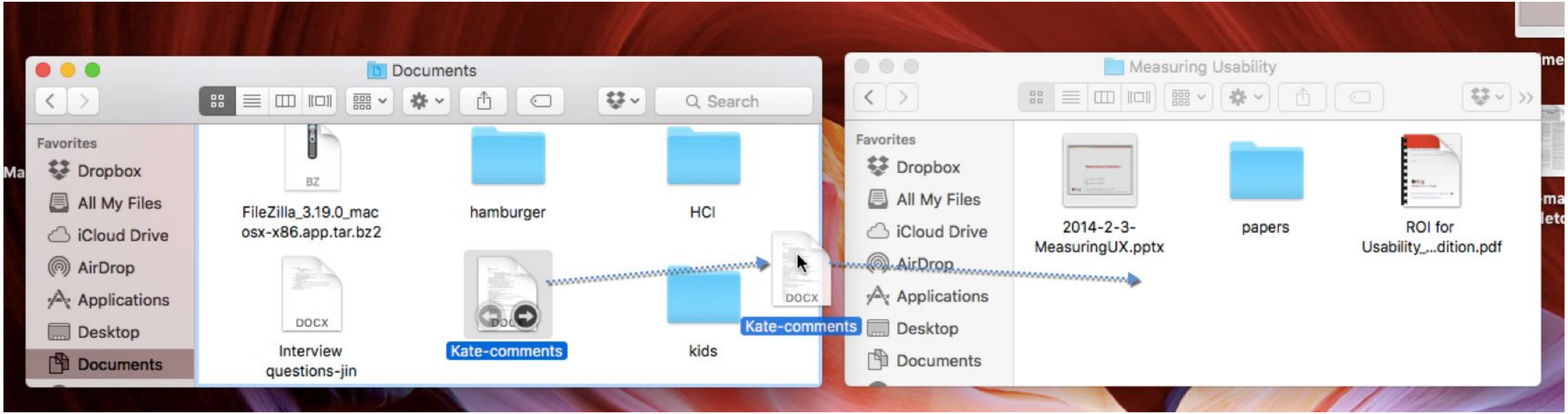
- Metaphors are naturally limited as it is not possible to completely map one set of concepts onto another
- Mismatches and false expectations can occur
    - Folders within folders
    - Dragging media into trash to eject

Direct Manipulation (1980s)

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

Direct Manipulation

- Rapid feedback facilitated an interaction technique called *direct manipulation* (Ben Shneiderman)
- Creates the illusion of operating directly on data and objects, rather than giving commands to a computer



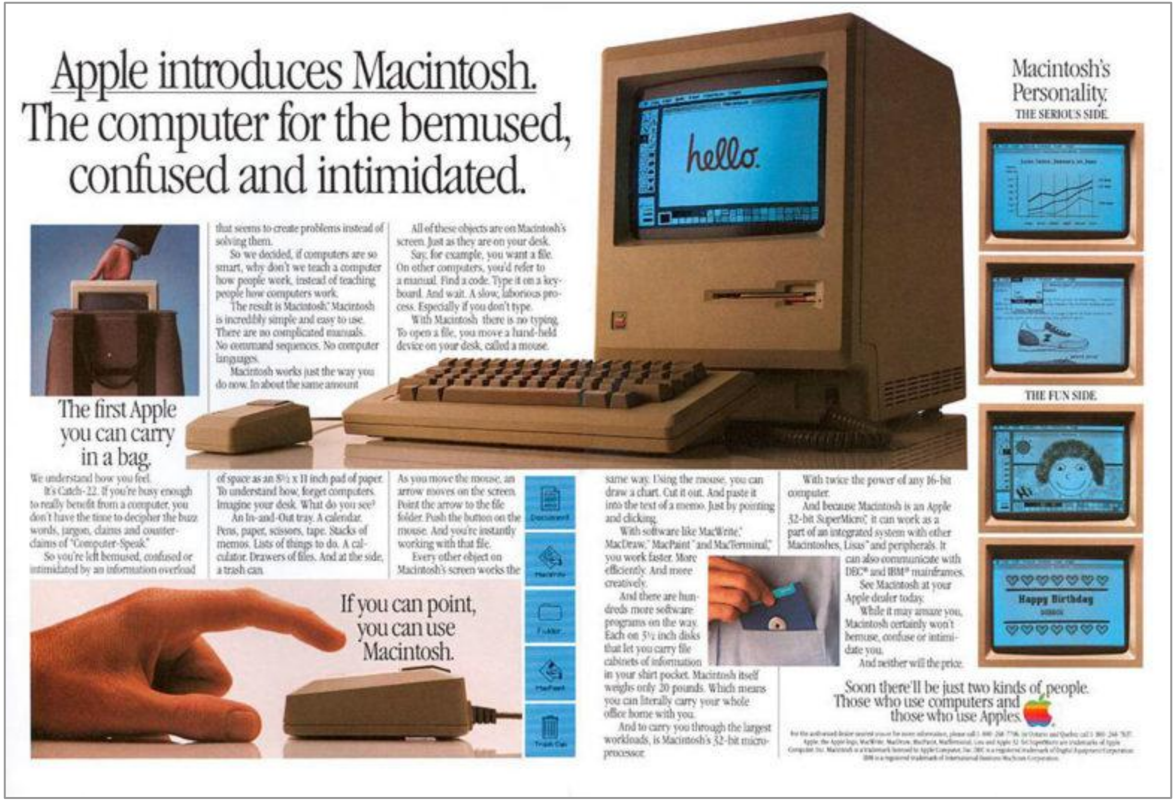
Direct Manipulation

- Features of direct manipulation
    - Visibility of all objects of interest
    - Incremental action at the interface with rapid feedback on all actions
    - Reversibility of all actions so that users can explore without severe penalties
    - Syntactic correctness of all actions so that possible action is a legal operation
    - Replacement of complex command languages with actions to manipulate visible objects directly

Direct Manipulation

- First commercial success of a direct manipulation interface was the Apple Macintosh computer (1984)
    - Made files and directory structure visible to the user
    - Operations such as moving files between directories were mirrored in an action on a visible document that could be picked up and dragged
    - Impossible to formulate a syntactically incorrect command
    - Continual visual feedback is provided while the operation is being carried out
    - Created illusion that the user is acting upon the objects represented in the interface, rather than giving commands

Direct Manipulation



Apple introduces Macintosh. The computer for the bemused, confused and intimidated.

The first Apple you can carry in a bag.

We understand how you feel.

It's Catch-22. If you're busy enough to really benefit from a computer, you don't have the time to decipher the buzz words, jargon, claims and counter-claims of "Computer-Speak."

So you're left bemused, confused or intimidated by an information overload that seems to create problems instead of solving them.

So we decided, if computers are so smart, why don't we teach a computer how people work, instead of teaching people how computers work.

The result is Macintosh.™ Macintosh is incredibly simple and easy to use. There are no complicated manuals. No command sequences. No computer languages.

Macintosh works just the way you do now. In about the same amount of space as an 8½ x 11 inch pad of paper.

To understand how, forget computers. Imagine your desk. What do you see?

An In-and-Out tray. A calendar. Pens, paper, scissors, tape. Stacks of memos. Lists of things to do. A calculator. Drawers of files. And at the side, a trash can.

All of these objects are on Macintosh's screen. Just as they are on your desk.

Say, for example, you want a file. On other computers, you'd refer to a manual. Find a code. Type it on a keyboard. And wait. A slow, laborious process. Especially if you don't type.

With Macintosh there is no typing. To open a file, you move a hand-held device on your desk, called a mouse.

If you can point, you can use Macintosh.

As you move the mouse, an arrow moves on the screen. Point the arrow to the file folder. Push the button on the mouse. And you're instantly working with that file.

Every other object on Macintosh's screen works the same way. Using the mouse, you can draw a chart. Cut it out. And paste it into the text of a memo. Just by pointing and clicking.

With software like MacWrite,™ MacDraw,™ MacPaint™ and MacTerminal,™ you work faster. More efficiently. And more creatively.

And there are hundreds more software programs on the way. Each on 3½ inch disks that let you carry file cabinets of information in your shirt pocket. Macintosh itself weighs only 20 pounds. Which means you can literally carry your whole office home with you.

And to carry you through the largest workloads, is Macintosh's 32-bit microprocessor. With twice the power of any 16-bit computer.

And because Macintosh is an Apple 32-bit SuperMicro,™ it can work as a part of an integrated system with other Macintoshes, Lisas™ and peripherals. It can also communicate with DEC® and IBM® mainframes.

See Macintosh at your Apple dealer today.

While it may amaze you, Macintosh certainly won't bemuse, confuse or intimidate you.

And neither will the price.

Macintosh's Personality.

THE SERIOUS SIDE.

THE FUN SIDE.

Soon there'll be just two kinds of people.
Those who use computers and
those who use Apples.

For the authorized dealer nearest you or for more information, please call 1-800-268-7796. In Ontario and Quebec call 1-800-268-7637. Apple, the Apple logo, MacWrite, MacDraw, MacPaint, MacTerminal, Lisa and Apple 32 SuperMicro are trademarks of Apple Computer Inc. Macintosh is a trademark licensed to Apple Computer, Inc. DEC is a registered trademark of Digital Equipment Corporation. IBM is a registered trademark of International Business Machines Corporation.

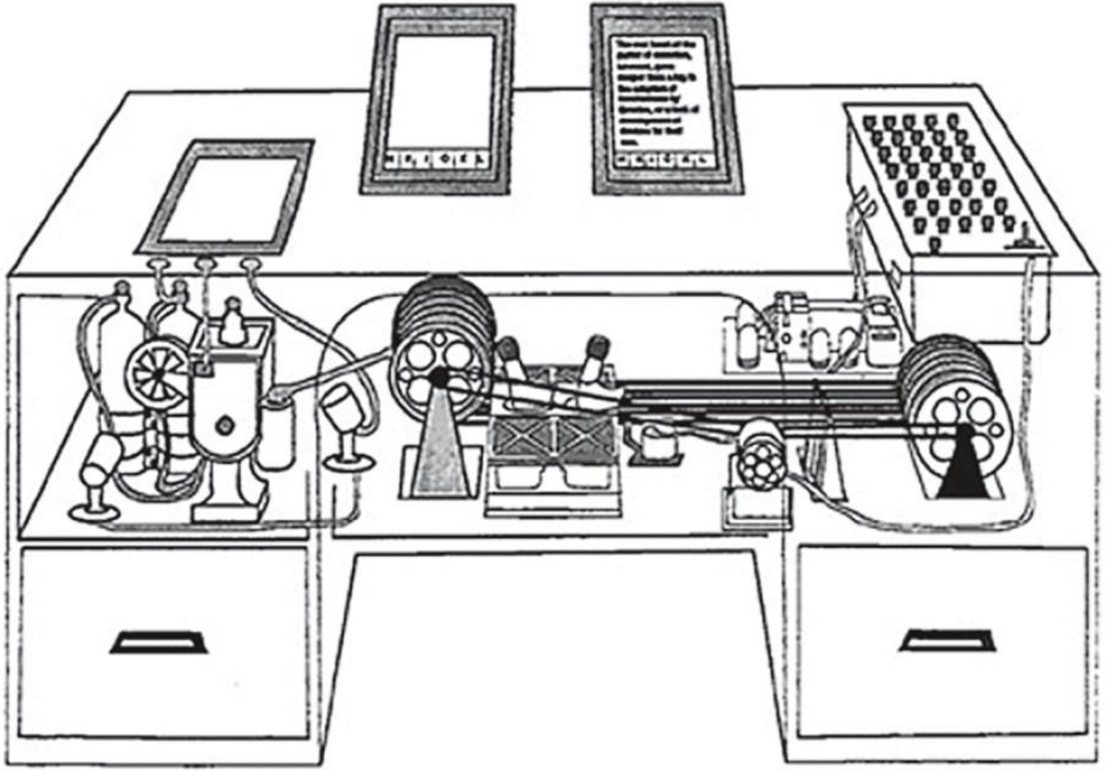
Hypertext (1940s-1960s)

- In 1945, Vannevar Bush published “As We May Think” in *The Atlantic Monthly*
- A response to the proliferation of scientific knowledge produced during WWII and the challenges of keeping track of the growing body of scientific literature
- Bush believed that keeping abreast of the increasing flow of information was crucial for progress

Hypertext

- “As We May Think” proposed an innovative future for information storage and retrieval to improve human capacity of knowledge access
- The proposed “memex” apparatus was a desk with the ability to produce and store massive amounts of photographic copies of documents
- Memex could keep track of links between parts of different documents, created an interconnected mesh of data, similar to information storage in the human brain
- Revolutionary idea to store not only information between associations among information

Hypertext

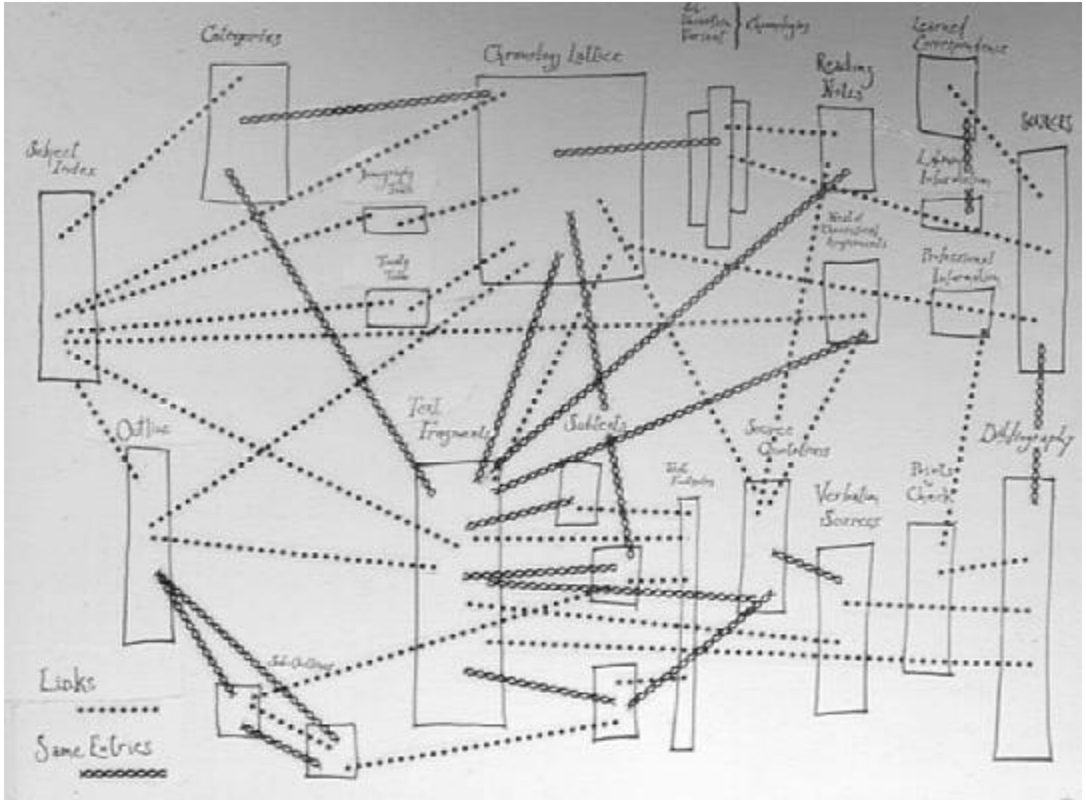


source: boingboing.net

Hypertext

- Ted Nelson (1960s) inspired by Bush's work to create a machine language equivalent of memex
- Nelson created Xanadu, a revolutionary information publishing an retrieval system based on interconnected, non-linear text and other media
- Supported new way of information browsing and non-linear consumption of information
- Coined the term "hypertext" to distinguish the non-linear structure of media storage and presentation
- This concept is the basis of underlying structure of the modern World Wide Web

Hypertext



source: xanadu.com.au

Multi-Modality

- Interaction advances allowed people to engage in multiple tasks simultaneously
- Multi-modality enables the use of multiple “channels” of human communication means to interact
  - Visual channel
  - Audio channel
  - Haptic (touch) channel

Multi-Modality

- Conventional computer systems with keyboard, pointing device, visual display and audio output are inherently multimodal
- However, genuinely multimodal systems rely more heavily on simultaneous or flexible use of multiple communication channels
  - E.g. combining gestures and voice commands

Computer-Supported Cooperative Work

- Advent of computer networks beginning in the 1960s enabled computers to communicate with each other
- Networks facilitated collaboration among individuals using computers resulting in CSCW (Computer-Supported Cooperative Work) systems

Computer-Supported Cooperative Work

- Transition from systems for individual use to systems for group and organizational use
- CSCW systems allowed for interaction between humans via computers and need to support the needs of multiple users
- Email is an early example of a CSCW system
- Subsequent computer-based communication tools and social media platforms arose from this paradigm

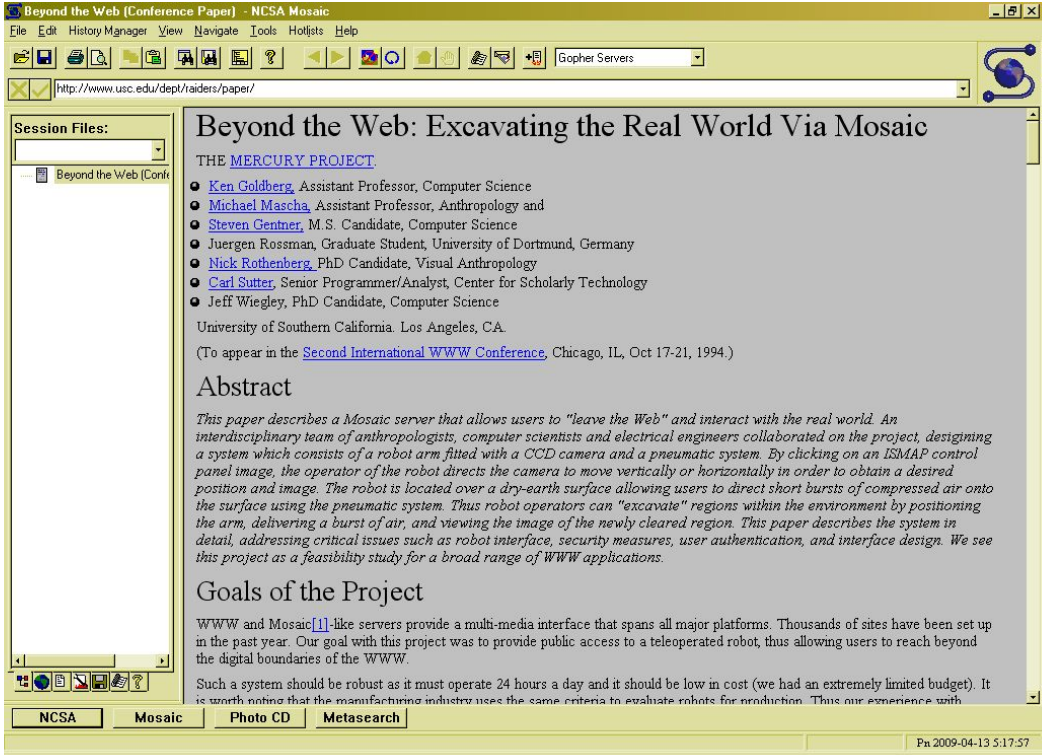
The World Wide Web (1990s)

- The internet refers to a collection of computers linked by data connections
- Although the internet has been in existence since the 1960s there were no interfaces to make it easily usable and accessible

The World Wide Web

- In 1989, Tim Berners-Lee (CERN Geneva) started the World Wide Web project as a way to facilitate distribution of scientific data over the internet
- The World Wide Web (WWW), built atop the internet provides access to information on the internet through standard protocols, notations, and addressing systems
- The WWW abstracts underlying complexities of transmission protocols and data access from users
- First WWW browser (1991) was text based
- By 1993 several graphical web browsers (e.g. Mosaic) were available

The World Wide Web



Beyond the Web (Conference Paper) - NCSA Mosaic
File Edit History Manager View Navigate Tools Hotlists Help
Gopher Servers
http://www.usc.edu/dept/raiders/paper/
Session Files:
Beyond the Web (Confe
NCSA
Mosaic
Photo CD
Metasearch
Pn 2009-04-13 5:17:57

Beyond the Web: Excavating the Real World Via Mosaic

THE MERCURY PROJECT.

- Ken Goldberg, Assistant Professor, Computer Science
- Michael Mascha, Assistant Professor, Anthropology and
- Steven Gentner, M.S. Candidate, Computer Science
- Juergen Rossman, Graduate Student, University of Dortmund, Germany
- Nick Rothenberg, PhD Candidate, Visual Anthropology
- Carl Sutter, Senior Programmer/Analyst, Center for Scholarly Technology
- Jeff Wiegley, PhD Candidate, Computer Science

University of Southern California. Los Angeles, CA.

(To appear in the Second International WWW Conference, Chicago, IL, Oct 17-21, 1994.)

Abstract

This paper describes a Mosaic server that allows users to "leave the Web" and interact with the real world. An interdisciplinary team of anthropologists, computer scientists and electrical engineers collaborated on the project, desigining a system which consists of a robot arm fitted with a CCD camera and a pneumatic system. By clicking on an ISMAP control panel image, the operator of the robot directs the camera to move vertically or horizontally in order to obtain a desired position and image. The robot is located over a dry-earth surface allowing users to direct short bursts of compressed air onto the surface using the pneumatic system. Thus robot operators can "excavate" regions within the environment by positioning the arm, delivering a burst of air, and viewing the image of the newly cleared region. This paper describes the system in detail, addressing critical issues such as robot interface, security measures, user authentication, and interface design. We see this project as a feasibility study for a broad range of WWW applications.

Goals of the Project

WWW and Mosaic[1]-like servers provide a multi-media interface that spans all major platforms. Thousands of sites have been set up in the past year. Our goal with this project was to provide public access to a teleoperated robot, thus allowing users to reach beyond the digital boundaries of the WWW.

Such a system should be robust as it must operate 24 hours a day and it should be low in cost (we had an extremely limited budget). It is worth noting that the manufacturing industry uses the same criteria to evaluate robots for production. Thus our experience with

source: wikipedia.org

The World Wide Web

- The WWW was a revolutionary paradigm
- Lowered the barrier for access to the internet
- Lowered the barriers to creating and publishing information, thus leading to the rapid growth and increased value of internet content including leisure and commerce
- Increased access to internet in turn led to increase in computer purchases and use

Agent-Based Interfaces

- A departure from direct manipulation
- Creates the illusion of someone working on your behalf to perform tasks
- Can perform repetitive tasks, respond to events, and learn from user actions

Agent-Based Interfaces

- Agents can be simple actors that follow commands or intelligent and proactive
- Early example was Eager, a cat icon that would observe HyperCard programmers and suggest next actions
- Interaction language has always been a challenge for designing agents
- Recent innovations, e.g. Apple’s Siri have made great strides in agent interaction

Ubiquitous Computing (1990s-2000s)

- Researchers at Xerox PARC (1980s) attempted to move computing “off the desktop” and into everyday life
- Mark Weiser (1991) wrote “The Computer for the 21st Century” which introduced the vision of Ubiquitous Computing
- Making computing seamless with everyday activities
- Computing at different scales from handheld to wall-size displays

Ubiquitous Computing

- Ubiquitous Computing also refers to a shift in computer to human ratios
  - 1950s: many humans to one computer
  - 1970s – 1980s: one computer to one human
  - 1990s –present: many computers to one human

Ubiquitous Computing

source: adaptivepath.org

Sensor-based and Context-aware Interaction

- Context-aware computing extends the notion of ubiquitous computing
- More invisibility and seamlessness of computing with everyday life
- Information gathered from people’s activities through sensing and context
- Sensed information used by systems to provide functionality and support for human activities

Sensor-based and Context-aware Interaction

- Presently systems pose challenges relating to privacy of information collection and use
- Systems that act based on collected data using artificial intelligence may also make poor decisions or act in ways that are undesirable

COURSE POLICIES

Course Format

- Weekly lectures (Monday 14:00-15:45, 1.B.01)
- Lab (Wednesday 12:15-13:45, 0.K.02)
- Exercises (distributed after lecture)
- Two assignments (mid-October/mid-November)
- Final Exam (18.12.2019, 13:00 – 14:30, location TBA)

Course Language

- Lectures, lab, exercises and final exam are in English
- Assignments will be accepted only in English
- Assignments will not be graded for language quality, BUT
  - Should be comprehensible and readable such that they can be assessed and graded
  - Should correctly use terminology from the course
  - Should be run through an English language spell checker when appropriate

Course Communications

- All students should enroll on OLAT for this course
- All course materials, grades, announcements, schedule information, and other relevant information for the course will be posted on OLAT
- All questions pertaining to class should be posted on the OLAT forum

Course Expectations

- Attendance at lectures
- Completion of reading assignments
- Completion of exercises
- Completion of project assignments
- Final exam

Grading

- Graded components of the course:
  - Project assignments (20% + 20%)
  - Final Exam (60%)
- Lecture and lab attendance, exercise completion, reading assignment completion are not explicitly graded, but are necessary for successful completion of course

Exercises

- Completion of exercises and attendance at lab sessions is important and highly recommended
- Lecture attendance and completion of associated reading assignments are necessary for exercises
- Exercises will be posted on OLAT along with the corresponding lecture
- Understanding how to do the exercises is key to doing well on the final exam!

Lab Sessions

- Take place on 7 Wednesdays from 12:15-13:45
- Time to work through the exercises independently or in groups
- TAs and Tutors will walk through the exercise and present solutions
- TAs and Tutors will answer questions and provide feedback as necessary
- Lab sessions are a resource for you to do the exercises - it is strongly recommended that you take advantage of them

Course Content

- Course content for which students are responsible includes:
    - All lecture content
    - All lecture slide content
    - All (non-optional) reading assignments
    - All material from exercises
- Any material above may be tested in the exam
- Lecture slides are necessarily an abbreviation of lecture material
- Lectures may contain information not present in slides or readings, and vice versa

Late Assignments

- Late submission of assignments and substitute assignments for missed exercises are not permitted without a medical certificate
- Requests for deadline extensions out of medical necessity will be considered only if requests are made **in a timely fashion**

Plagiarism

Plagiarism will not be tolerated and will be handled in accordance with standard University procedures. Students should familiarize themselves with University plagiarism policies.

https://www.uzh.ch/cmsssl/de/studies/teaching/plagiate.html

Course Schedule

- Course schedule (tentative, subject to change) posted on OLAT
- NO EXERCISE 18.9.2019 or 25.9.2019
- First exercise will be distributed on 30.9.2019
- First lab will take place on 2.10.2019

HAVE A GREAT START TO THE SEMESTER!