COURSE INFORMATION

Text:

Baecker, Grudin, Buxton & Greenberg
Human-Computer Interaction:
Toward the Year 2000, Second Edition
Morgan Kauffman: 1995

The textbook is a collection of reference articles. It is not intended to be read linearly. The chapter introductions (a book within the text) provide an excellent summary of the field of HCI. Below, I've divided the book's articles into groups depending on how general, interesting, and important they are. Each article is referenced by an *author and starting page number*.

Definitely should read:

	(starting page numbers)
Introductions to all Parts and Chapters:	
Introduction to human-computer interaction	1 23 35 49
Process of developing interactive systems	71 73 187 273 313
Interacting with computers	399 411 469 525
Psychology and human factors	571 573 667
Research frontiers in HCI	739 741 783 833 867 897

norman 5	johnson 53	kling 254	kim 304
myers 323	marcus 425	denning 684	sproull 755
fischer 822	glushko 849	berners-lee 907	weiser 933

Try to read, or at least skim:

mcgrath 152	bannon 205	suchman 233	myers 357
shneiderman 401	baecker 444	marcus 457	bier 517
olsen 603	landauer 659	lewis 686	henderson 793
maes 811	ellis 913		

Read only if you are particularily interested:

salomon 25	gould 93	lewis 122	mountford 128
vertelney 142	erickson 147	mack 170	kennedy 182
orlikowski 197	bodker 215	good 225	holtzblatt 241
boehm 281	grudin 293	pausch 344	wiecha 373
roseman 390	murch 442	ahlberg 450	mackensie 483
buxton 494	goldberg 500	pedersen 509	o'malley 539
peacocke 546	white 554	gaver 564	card 587
john 626	gray 634	barnard 640	norman 681
carroll 698	sellen 718	lazzaro 724	sauter 728
grudin 762	baecker 775	cypher 804	egan 843
davis 854	levine 871	perkins 881	francik 886

Evaluation

Available grades:

non-completion: Incomplete, Withdraw, etc.
completion: A A- B+ B B- C

A:	reserved for superior performance
A- or B+:	expected grade for conscientious performance
B:	adequate work
B-:	barely adequate
C:	equivalent to failing

Grading Options:

1. Grading Contract: specify a set of behaviors and an associated grade.
2. Performance Quality: work on assigned exercises (one per week)
3. Negotiation: student and instructor decide upon a personalized grade.

Discussion:
If you already understand the field, if you plan to excel, or if you need clear performance goals for motivation, then **Option 1** is a good idea. If you prefer a clearly defined agenda, if you do well with concrete task assignments, or if you need a schedule of activities for motivation, then **Option 2** is a good idea. If you are not concerned about grades, if you intend to do what you choose anyway, or if you are self-motivated, then **Option 3** is a good idea.

I will notify any student who is not on a trajectory for personal success.

Educational Philosophy

Ideally, a teacher facilitates the construction of an environment conducive to personal learning. This may include presenting facts, telling stories, creating opportunities for experience, pointing to relevant information and resources, sharing tools for thought, building mutually agreed upon territories, and creating quandaries and paradoxes. It does not include doing the student's thinking. One-liners:

* Computer Science is generative: program first, theorize later.
* Knowledge is actively constructed and is particular to the context and situation.
* Facilitating learning means not distinguishing between process and goal.
* Content is best conveyed by structuring the environment.
  Rather than putting information inside students, put students inside information.
* Learning exhibits itself through confusion and uncertainty.
* How you teach is as important as what you teach.
* One of the most difficult skills for a teacher to learn is keeping quiet.
* Never ask a question that you know the answer to.
* Real teaching is one-to-one.
  The larger the learning group, the more education looks like entertainment.

References

Conferences:

* CHI: ACM Special Interest Group on Computer and Human Interaction (annual)
* SIGGRAPH: ACM SIG on Computer Graphics and Interactive Techniques (annual)
* UIST: ACM Symposium on User Interface Software and Technology (annual)
* CSCW: ACM Conference on Computer-Supported Cooperative Work (biennial)

These are the large US conferences. CHI gets about 2500 attendants, quality of papers in proceedings is spotty. SIGGRAPH gets about 25000 attendants, and is the convergence of CS and the entertainment industry. Papers are superb but very technical. UIST is by invitation, papers are excellent and very relevant to HCI. CSCW is specialized, with strong industry support. Human factors, hypertext, VR, agent theory, interactivity design, and European interests also have specialized conferences.

Journals:

* SIGCHI Bulletin (quarterly). For professionals in the HCI field
* Interactions (quarterly). Slick, excellent articles, for professionals.
* Presence (quarterly). Premier technical journal for virtual environments.

Internet:

http://www.cis.ohio-state.edu/~perlman/resources.html
six articles from ACM Interactions Magazine HCI Resources by Perlman
http://www.cs.bgsu.edu/HCI/
HCI resources collected by Instone
http://info.sigchi.acm.org/sigchi/
CHI homepage
http://www.Sun.COM/sun-on-net/www.sun.com/uidesign/
story of the SUN homepage design
http://www.yahoo.com/Science/Computer_Science/Human_Computer_Interaction/
resource list
http://www.cs.cmu.edu/afs/cs/project/amulet/www/amulet-home.html#overview
access to a research prototype UI toolkit
http://www.cis.ohio-state.edu/~perlman/hcibib.html
big bibliography
http://www.ida.liu.se/labs/aslab/groups/um/hci/
more references
http://www.cs.cmu.edu/afs/cs.cmu.edu/user/bam/www/toolnames.html
list of UI toolkits

Definitions

Definitions of HCI

* Encyclopedia of Computer Science and Engineering: no entry for HCI

* HCI is the study of the interaction between humans and computers. [Booth, 1989]
  - Interactional hardware and software
  - Matching models (understanding, meeting needs, usability)
  - Design and development of interactional systems
  - Organizational impact

* HCI is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them. [ACM-HCI, 1993]

machines:	computer workstations, aircraft cockpits, microwave ovens
humans:	groups, organizations, human work
interaction:	programming, TV remote control, VR games

* HCI is the main gating function to the successful use of technology. [Strong, 1995]

* The tension between the human use of computation and the computational use of humans. [Bricken, 1991]

Subject matter of HCI

Cross-disciplinary:

Computer Science	application design, interface engineering
Psychology	cognitive processes, user behavior
Sociology&Anthro	technology, work, organization
Industrial Design	product interactivity

ACM-HCI curriculum:

- nature of HCI
  - models and meta-models
- use and context of computers
  - social organization and work
  - application areas
  - human-machine fit and adaptation
- human characteristics
  - human information processing
  - language, communication and interaction
  - ergonomics
- computer system and interface architecture
  - input devices

- recognition
- output devices
- rendering and computer graphics
- dialogue and system architecture
- dialogue techniques
- dialogue genre

development process

- design approaches
- implementation techniques
- evaluation techniques
- example systems and case studies

Realities of HCI

Origins:

computer graphics	CRT and pen devices
man-machine symbiosis	WIMP
operating systems	i/o interface, response time
human factors	war equipment, sensory-motor
ergonomics	work efficiency, sensory-motor
industrial engineering	productivity, fatigue
cognitive psychology	human information processing
computer systems	sales and usability

Technical concerns:

- joint performance human/machine
- structure of communication
- human capabilities, learning
- programming
- engineering interfaces
- specification and design

Major trends:

ubiquitous communication	(the net)
high functionality systems	(configurable computing)
mass computer graphics	(killer video games)
mixed and multi media	
high bandwidth	
large, thin displays	
embedded computation	
group interfaces	
user tailorability	
information utilities	
virtual environments	
internet, internet, internet	

HCI career paths:

industry research
research practice and implementation
systems and requirements analysis
ergonomics and human factors engineering
software programming converts
personnel and support converts
graphical design

What HCI professionals say they need:

most:	user interface technology
	interactive systems design
less:	nature of HCI
	research methods
	programming the interface
least:	user modeling
	application areas

Most research interest:

UI design
CSCW
multimedia
software engineering
UIMS toolkits
information presentation/visualization
cognitive modeling
UI development
theories of HCI

Changing focus:

faster cheaper systems
portability
new display and packaging
network communication
multimodal i/o

Theoretical issues:

utility of IPS
context and situation
human variability
human artifacts
social vs individual impact/design
role of theory in design

Freedom and privacy:

property vs freedom
constitution in cyberspace
falsifying electronic evidence
liability and sysadmin
cryptography
crime and law in cyberspace
privacy and freedom of speech
e-money
mass interactive communication
censorship

Selected Issues

* Social and psychological impact of computers
* Impact of the Web, networked users
* Cultural differences and human variability
* Speed of evolution of computers and design/learning strategies
* Interface within symbolic systems, programmer interfaces vs user interfaces
* Do you need to know programming or systems architecture in order to design interfaces?
* How much real-time is needed for interactivity?
* Formal or informal approach: clean/scruffy, artist/engineer
* Closed HCI society vs superstars vs poor academic acceptance
* Empiricism: task analysis, protocol analysis, the role of research in design
* User flexibility/choice vs designed constraints
* Program or interact or broadcast: where does the user begin and the system end
* Is the car a user interface for a fuel-injection computer?
* Is game and film design the dominant use of HCI?

Curriculum Exercise

"HCI Overview"

Make a chart/list of the major areas of HCI. Include what you think the subject is about, and the areas you have had experience in.

"Curriculum Engineering"

Draft your ideas of the curriculum for this course. What topics will we study, what activities will we do, how shall we determine success? Include preferred and requested topics, and what to avoid. I will build a group map focusing on general content and specific interests.

Emphasis?	machines	computer science
	humans	psychology, physiology
	design	art
	human groups	sociology

Final Curriculum Plan

Class Meeting	Topic	Text Sections
1)	introduction to the field	
2)	physiology	(Chs 2)
3)	simulation	
4)	simulation	
5)	evolution of interface	(Ch 1)
6)	psychology history	(Chs 7,9)
7)	interface theory	
8)	measurement, modeling	
9)	cognitive maps	
10)	current toolkits	(Ch 5) *research report due
11)	interface design	(Ch 6)
12)	dialog processes	(Ch 10)
13)	i/o devices, sound	(Ch 8)
14)	graphics, virtual environments	(Ch 12,13)
15)	information visualization	
16)	internet	(Ch 14)
17)	social issues	(Ch 3,11) *design critique due
18)	summary	

Students will be responsible for two projects during the quarter.

Interface Design Simulation

Objectives:

Experience HCI design using a detailed task specification. Integrate suggestions for design in the text into a task-oriented design context. Provides a context to discuss design methodology and choices.

Task:

A programming team in your organization has developed a new deductive engine which allows application programmers to manipulate data for expert systems. Your job is to design a prototype interface for this engine.

The engine provides functions for a knowledge engineer to restructure, optimize, verify, and in general manipulate the components of a knowledge base of logical and arithmetic constraints. What is neat about this engine is that it maintains a graphic, network description of the logical transformation processes, and like a circuit, distributes logic over many network nodes.

However, different departments in your organization have different formats for their knowledge-bases, want to do different things to their data, and require different outputs and views of their data. Furthermore, some users want automated functionality and some want fine-grain interactivity with transformations.

Due to organizational preferences, the interface is to be constructed by three separate teams, one team for each of the following aspects:

Aspect 1:	function calls to the interface	(programming, API)
Aspect 2:	screen layout and interactivity	(interface, dialog)
Aspect 3:	hardware i/o devices and functionalities	(architecture)

Fortunately, some members of each team can cross development boundaries and work with the other teams as advocates of their design process.

You are to add appropriate interface controls for things like opening and closing the system, trapping and notifying about input errors, and improper control configurations. Also, you should select appropriate names, labels, and displays for both naive and sophisticated users. You are not responsible for

explaining how the engine works,
the help system, or
the editors which allow databases to be constructed,

although you should include interface hooks to all three subsystems.

You are free to modify and enhance interface specifications to make the engine easy to use, so long as the requested functionality is available.

Human-Computer Interaction

Below is a (loosely) structured listing of some of the requirements for your interface.

* Backend processor assignment: single, distributed

* Input language: logic, Prolog, Lisp

* Input form: files, keyboard

* Display:
  - linear (textual) view in any input syntax,
  - graphic (network) view,
  - internal computational view if requested

* Simple transformations: absorb, clarify, extract, coalesce

* Compound transformations: subsume, cancel, collect

* High-level transformations:
  - optimize relative to specified time and complexity parameters
  - identify contradictions, verify consistency
  - delete irrelevant facts
  - cluster facts in groupings relative to a particular set of variables

* Network display controls:
  - select an active subnetwork to perform transformation on
  - rearrange network by hand
  - rearrange network using energy minimization algorithm
    - parameterized by spring coefficient, spring divisor,
    - repulsion coefficient and relaxation stepsize
  - labels on or off

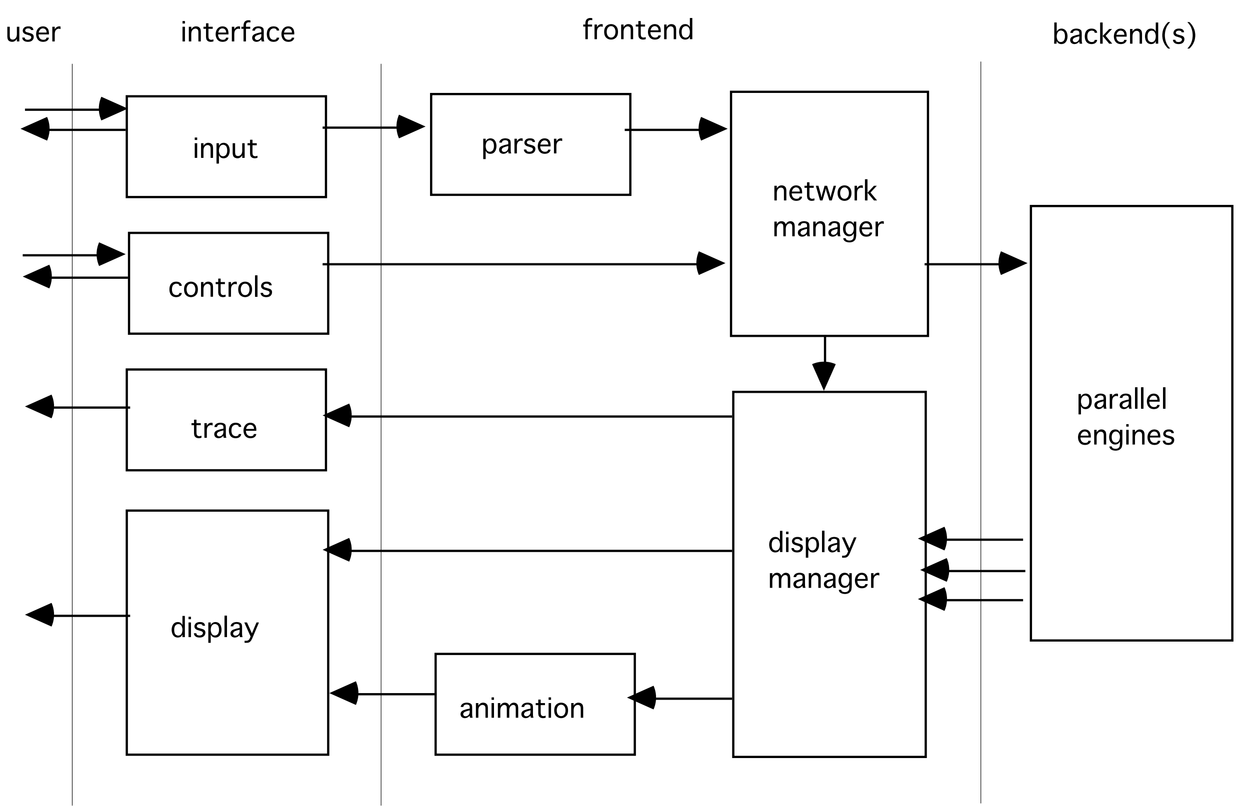
* Network animation:
  - show animation in forward or reverse order
  - stop and start animation freely
  - show active network components and their activity
  - specify rate of animation by
    - frames per second
    - transformations per second
    - specific transformations per second
    - activity indications per second

* Trace:
  - show engine activity by transformations performed and
  - by animation instructions performed

* Users also want to be able to :
  - focus on any display with full screen, especially the network display
  - reset display at any point
  - refresh display
  - load and display new logical databases
  - select textual parts of a database for analysis

The engine transformations can be applied individually or in any grouping. The backend engine(s) are much faster than the display, so the display manager collects engine activity and structures a display which makes sense to a person. It is important that the users of the engine understand the logic of the transformations. The engine transformations use an internal coding that is not easy to understand.

Here is the functional architecture of the system:



Backend(s):
Computational machinery on which transformations are done; can be a parallel, distributed array of processors

Frontend:
Manages the interface and coordinates assignment of and communication with backend. Backend coordination can be organized by synchronous or asynchronous message-passing or by shared memory.

Interface:
The information and controls seen by the user.

HCI Assignments

In addition to readings in the text, (option 2) students are expected to complete two assignments, described below. We will work out the exact content for each student in class.

MID-TERM ASSIGNMENT — IN-DEPTH RESEARCH

Select a small area of HCI that you have interest in and explore it. Prepare a report back to the class about what you learned. Summarize the area in a discussion or oral report of 5-15 minutes.

You can use the web addresses provided earlier to locate writings on your topic of interest.

Content should be some small topic which you can find three or four articles/books on. (I will make suggestions to everyone who asks.) Best would be content that you encountered at work or in school. Another good idea is to select an issue which bothers you about some software of hardware system that you use regularly.

FINAL ASSIGNMENT — DESIGN CRITIQUE

Select one application or system interface that you are quite familiar with. Using the design principles discussed in class, in the lecture notes, and in the textbook readings, analyze and critique this interface. Consider:

- the cognitive model
- the visual layout
- the dialogue management
- the interactivity model
- the information structure
- the integration with i/o devices
- the task appropriateness

Build a model of the information structure, following the content being presented and its organizational structure.

Build a model of the interactivity, following the flow of control and communication throughout a dialog transaction.

Then redesign the interface in any areas that you think could be improved. Be specific about what you think is wrong and why your redesign is an improvement.

Justify your critique and your redesign by citing references from the HCI literature.

Interesting Books on Human Psychology

This selection is not intended to be balanced or objective, rather it represents both interesting and credible ideas about our minds and how we can be expected to respond to computers. The historical selections suggest ways in which people are similar, but the original authors make difficult reading. The current leaders selections represent several aspects of highly reputable modern ideas. The scholarly extensions contain advanced concepts which have grown out of the ideas in the previous selections.

Historical

Sigmund Freud, *Psychopathology of Everyday Life*
The Interpretation of Dreams
Three Contributions to the Theory of Sex
Civilization and its Discontents

Less than 100 years ago, Freud introduced/invented the concepts of psychoanalysis, ego defense (repression, projection, intellectualization, regression, denial, sublimation), transference, Freudian slips, unconscious, consciousness as an organ of perception, and

id	unconscious, primitive self, pleasure principle
ego	partly conscious, ethical self, reality principle
superego	inner conscious, morality
libido	sexual energy

Carl G. Jung, *The Archetypes and the Collective Unconscious*

Jung introduced the ideas of psychological unity across humans, differentiated mental functions (thinking, feeling, sensation, intuition) and attitudes (extroversion, introversion) and subconscious archetypes (mother/father, anima/animus, persona/shadow, child, maiden, wise old man, self).

Jean Piaget, *Genetic Epistemology*
The Child's Conception of the World
The Moral Judgment of the Child
The Construction of Reality in the Child

Piaget introduced the idea that children (and adults) construct reality through assimilation (matching perception of experience to thought) and accommodation (matching perception of thought to experience). Knowledge is identical to action. Intellect develops through stages (sensory-motor, 0-2 years; concrete thinking, 2-11 years; abstract thinking, 11-15 years).

Current Leaders in Psychology, Philosophy, and Computation

Jerome Bruner, *Actual Minds, Possible Worlds*

Understanding is always in context, meaning is always ambiguous. There is no reality independent of mental activity and symbolic language. We know the world and construct meaning through multiple perspectives, including emotion, culture, language, and stories.

Hilary Putnam, *Representation and Reality*

Mental states cannot be computational states. Meaning is always individually unique, interpretive, plastic, normative, social, interactive, and holistic. Symbols and languages are always ambiguous, vague, open, relative, and situated in a particular context. Truth is quasi-mythological and is not independent, bivalent or unique. Concepts depend on evolution.

Terry Winograd and Fernando Flores, *Understanding Computers and Cognition*
Design is the interaction between understanding and creation. Knowledge depends on individual interpretation and intuitive understanding rather than on logical deduction and conscious reflection. Language and interface are socially embedded, impossible to articulate/analyze, and conversational/interactive. Error is equivalent to non-obviousness, is an interpretation, and constructs positive information.

Imre Lakatos, *Proofs and Refutations: The Logic of Mathematical Discovery*
Even formal mathematics is situational, negotiated, informal, and completely non-rational. Mathematical knowledge grows through a cycle of conjecture, proof, identification of exceptions, redefinition of the meaning of proof, and redefinition of the conjecture. Facts are linguistic blindness. Proof, truth, consequence, counterexample, and criticism are inseparable.

Oliver Sacks, *The Man Who Mistook his Wife for a Hat*,
Clinical stories about how broken brains still construct a coherent reality.

Scholarly Extensions

Julian Jaynes, *The Origin of Consciousness in the Breakdown of the Bicameral Mind*
Human consciousness is a recently learned process. Before we associated thoughts with ego, we associated them with the voice of the gods. We construct thinking, locate it in our heads, fill in holes to create an illusion of continuity and wholeness, invent "I" and "me" so that the story has a main character, and blind ourselves to the inconsistencies of our own cognitive invention.

William Thompson et al, *Gaia, A Way of Knowing*
Nature is made of processes rather than objects. The unique, autonomous individual is the primary organizational unity in evolution. Natural processes are structural couplings between unities/individuals. Every thought is inseparable from every other thought; every action is inseparable from the entire environment; every individual is unique. Organization comes from disorganization induced by chaos followed by reorganization.

Francisco Varela, Evan Thompson and Eleanor Rosch, *The Embodied Mind*
Thought is embodied action. Each sense has a different consciousness. The omnipresent mental factors are contact, feeling, discernment, intention, and attention. The principal activity of the mind is changing itself.

Tarthang Tulku, *Time, Space, and Knowledge*
Objects (solid things) are formed by our choice of a point-of-view. Concepts and beliefs are formed by placing our viewpoint in an opaque (unclear) space. Time is the location which enables experience. Knowledge is the appreciation of space and time.

Notes on Psychology

Summary of design perspectives

Interface refers to the static look; interaction refers to the dynamic feel.

Machines are characterized by extreme similarity (replacability, predictability)
Humans are characterized by extreme uniqueness (individuality, unpredictability)
Therefore measure artifacts, but consult people.

Fundamental Principle of Design: Minimize cognitive load.
Fundamental Method of Design: Measure and modif

Generations of Interface

Year	*machine*	*substrate*	*access*	*interface*	*human activity*
1945-55	ENIAC	vacuum tube	knobs & dials	plugboard	plug wires, watch tubes
1955-65	PDP1	transistor	batch mode	countertop	punch cards, read print
1965-80	VAX	integrated circuit	timesharing	dumb terminal	type keys, read terminal
1980-90	68020	VLSI chip	menu	desktop	click mouse, watch monitor
1990-99	RISC	multimedia chip	multisensory	simulation	touch, talk, watch
2000!	parallel	array	whole body	movement	act naturally

A Simple Model of Human-Computer Interaction

```
        [eyes, human sensing]     <--meaning--      [display, machine acting]
                 /                                          \
                /                                            \
{cognition/concept}                                         {software/computation}
                \                                            /
                 \                                          /
        [hands, human acting]     --language-->     [keyboard, machine sensing]
```

Note: connections in human between sensing and acting are two-directional
connections in machine between sensing and acting are one-directional

Friendliness

People use conceptual models to guide their actions.
These models are not necessarily symbolic or encoded.
Friendliness of an interface: the match between conceptual model and input options
common language from idea to human action to machine sensing
Friendliness of a software tool: the match between conceptual model and display output
common meaning from machine acting to human sensing to idea

Formalism

Computers use formal systems to guide their actions.
These models are necessarily symbolic and encoded.

A formal system:

* a map between meaning and symbols that is invariant over symbol transformation
* what you do as machine input does not undermine your understanding of the output
* the software does not violate the user's model

Using a Formal System

```
    [situation]     --difficult  route--> [solution]
         |                                   /\
         |                                   |
  (meaning in)                          (meaning out)
         |                                   |
        \/               (blind rules)       |
    [representation]   --easy  route--> [symbolic  result]
```

Interaction Styles

```
command line                    Progression is from symbolic and abstract
programming language                           /\
semi-natural language                          |       formal
menu                                           |
forms
icons
windows                                        |
direct manipulation                            |       friendly
graphical interaction                          \/
inclusion                              to spatial and interactive
```

Text, Multimedia, and VR

Property	*text*	*multimedia*	*virtual environment*
access	sequential	parallel	experiential
space	1D	2D	nD
transfer	slow	fast	immediate
representation	abstract	pictorial	as-if-real
reference	indirect	graphic	interactive
display	static	dynamic	inclusive
metaphors	symbolic	iconic	natural

Schools of Psychology

```
1900             introspection (thought)
                 /                          \
1920   behavioral (action)              psychoanalytic (fantasy)
                 |                               |
1950   cognitive (information)          humanistic (potential)
                 |                               |
2000   constructive (organism)          organizational (task)
```

Cognitive Science

```
Cognition is information processing  (not sensing, intuition, emotion, action, faith)
    \/                   \/
thinking              psychology
remembering           linguistics
understanding         neuroscience
learning              computer science
languaging            philosophy
perceiving
```

Hilary Putnam

1960: the father of functionalism. The mind is functionally equivalent to a computer.
1990: Functionalism is completely false.
- Meaning is holistic
- Meaning is normative
- Concepts depend on evolution
- Mental states do not exist.

Information Processing Model of Human Cognition

```
     [effectors]  <--  [ response         ]       <-----------------  [              ]
     /                 [      generator   ]    <----                  [     long     ]
    /                                                   \             [              ]
{environment}                                            |            [     term     ]
    \                                                    |            [              ]
     \                                                   |            [    memory    ]
      [receptors] --> [sensory register] --> [working memory] <-->[              ]
```

The VR Paradigm Shift

from — We adapt to digital processes.
to — Digital processes adapt to us.
through — broader information interaction
- activity within an environment, multiple models of intelligence
- multiple sensory modalities, intelligence amplification (interactive problem solving)

Mechanical/information models

Old technology models embedded in new technology capabilities
mismatch of affordances

Cognitive/implementation models

Software design is toward cognitive not implementation model
so software engineers don't design
analogy to carpenters vs architects
- use hammer to buy a house

ask: what is goal of interaction
- how to make task easier
- how to hide implementation details
- usability is orthogonal to efficiency

Examples of
- calendar as single page of paper vs scroll
- tabs as mechanical rather than spatial
- trains and no moving camera in early film
- horseless carriage looks like carriage
- early tv as live radio

Shifts
- visual <- graphical
- user-centered <- machine-centered
- sensory <- silicon
- interactive <- symbolic

reduce cognitive complexity:
- vision is subconscious
- text is conscious

levels of human understanding
- perceive
- respond, recognize
- evaluate
- apply
- understand, analyze, synthesize

Models of computer

computer levels of architecture
- hierarchy of abstraction specification languages
  - machine language specification
    - vonNeumann tradeoff

circuit behavioral specification
hierarchy of realization specification languages

design model	abstract behavior
architecture model	abstract structure
performance model	abstract efficiency
correctness of behavior	functionality
efficiency of behavior	performance
actual behavior of physical circuit	reality

bit, word, instruction, program, message, application, user interface

Program levels, hierarchy

User interface: metaphoric system which makes design interface accessible to non-experts.

Design interface: hidden symbolic system which provide conceptual language for non-expert human to specify design abstractions.

Design abstraction: pure symbolic system which expresses a human objective

High-level programming language: symbolic system which closely models expert human models (math) and hides machine needs [Often math and algorithm are confused.]

Programming language: symbolic system which expresses assembly steps in human writable form. Does not cleanly differentiate between requirements of the human and those of the machine.

Assembly language: symbolic system which expresses machine language in process steps over specific logic function systems

Machine language: symbolic system which transfers low level machine instructions into processes within designed physical logic function systems

Logic function systems: physical system, integrated networks of gates

Gates: physical system represented by dnet graph format which has parens and pun components

Transistors: ignored physical system, assumed to be bundled in gates

Physiology

The *human sensorium*

cranial:	smell, see, taste, hear, motion, balance
integumentary:	touch, pressure, cold, heat, pain
muscular:	position, tension

Retinal variables

size, value, hue, orientation, texture, shape, position

Gestalts

proximity ... ...
similarity ..,,..,,..,,
continuity - - - | - - -
closure ( ) ( )
area (( ))
symmetry <<<::.::>>>

Varieties of Meaning

physical semantics:
map between digital representation and activity in the physical world
virtual semantics:
map between digital representation and perceived virtual world activity
natural semantics:
hiding the digital layer, map between physical activity and virtual consequence

Environments

- qualities
  - partially observable
  - strongly parallel
  - apparently infinite
  - turns into an object when viewed from outside
- coordination of interaction between contained entities
- hierarchy of relative containment systems
- holds global attributes
  - consistency (common time)
  - continuity (metric and origin of space)
  - linearity
  - flows and fluids
  - granularity
  - invariants (laws of the local universe)

User Interface System

An example of **a complete user interface system:** Mac Common Lisp

- **windowing** tools
  - menus
  - dialogs
  - views and windows

- **display** interface tools
  - graphics system
  - fonts and points
  - color

- **programming** interface tools
  - object-oriented class, instance, and message system <initialize-, make->
  - load, compile, link, and evaluate
  - text editor
  - interface construction toolkit
  - debugging and error handling
  - foreign function interface

- **interactivity** system tools
  - event handling and management
  - streams and buffers

- **operating system** tools
  - multitasking
  - memory management
  - file system interface
  - low level: internal data structures, pointers, memory blocks
  - high-level: traps and records

Generic object operators/functions:

constructors:	make-, initialize-, set-
assessors:	get-
queries:	? -
functions:	act-on-
relations:	constrain-

Turnkey dialog boxes

throw-cancel and catch-cancel <aborts>
message-dialog
yes-or-no-dialog
get-string-from-user-dialog
select-item-from-list-dialog

Windows

nested-views, size, position, scroller, click-handler
title, font, color, active?, layer, zoom, grow, drag

Menu Class structure

- menu-element
  - menubar (class, variable, function)
    - set-menubar
    - find-menu
    - <color-functions>
    - *default-menubar*
  - menu
    - initialize-, set-
      - menu-title, menu-items, menu-colors
      - update-function
      - help-spec (balloon-help system)
    - install, deinstall, installed?
    - enable, disable, enabled?
    - font-style
    - add-menu-items, remove-menu-items, get-menu-items, find-menu-item
    - <color-functions>
    - menu-item
      - initialize-, set-, get-, query?-
        - owner — title
        - command-key — action-function (call vs get)
        - disabled? — colors
        - checked — font-style
        - update-function — help-spec
    - window-menu-item
      - close, save, save-as, save-copy-as, revert, hardcopy
      - undo, undo-more
      - cut, copy, paste, clear, select-all
      - load/evaluate-selection, load/evaluate-whole-buffer

Dialog-items

- initialize-, set-, get-, make-
  - view-size, view-container, view-position, view-nickname, view-font
  - dialog-item-text, dialog-item-handle, dialog-item-enabled?
  - part-color-list, dialog-item-action, help-spec, window-pointer
- install, activate, activate-event-handler, default

- button-dialog-item
  - press-button, default-button-dialog-item (make-, get-, set-, ?-)
- static-text-dialog-item
- editable-text-dialog-item <key-stroke-handlers>
- check-box-dialog-item
  - check-box-check, -uncheck, -checked?
- radio-button-dialog-item
  - radio-button-cluster, -push, -unpush, -pushed?
- table-dialog-item <table-constructors>, <cell-contents-handlers>
  - sequence-dialog-item
- pop-up-menu <handlers>
- scroll-bar <handlers>

Design Guidelines, Research Methodologies, Dialogs

Design guidelines:

Xerox Star innovations
- desktop metaphor
- direct manipulation
- property options
- wysiwyg
- generic commands
- consistency
- few modes
- extensive iterative prototyping

Norman's concepts for design analysis

affordance	properties which cue intuitive uses
constraints	properties which enforce intented uses and limit errors
conceptual model	the users' construction of understanding
mapping	the programs' construction of understanding
visibility	apparentness of the mapping
feedback	apparentness of the conceptual model

Usability guidelines [Nielsen, 1994]
- visibility of system status
- match between system and world
- user control and freedom
- consistency
- error prevention
- recognition rather than recall
- flexibility and ease of use
- aesthetic and minimalist design
- help recognition and recovery from errors
- help and documentation

Design question checklist [Norman, 1988]

How easily can you:
- determine the function of the device?
- tell what actions are possible?
- determine mapping from intention to physical movement?
- perform the action?
- tell if the system is in the desired state?
- determine the mapping from system state to interpretation?
- tell what state the system is in?

Measurement:

Types of measurement

existential	(indicative)	exists or not	
categorical	(nominal)	share some property	+ attribute
ranking	(ordinal)	put in order	+ less than
discrete	(interval)	relate to integers	+ equal steps
comparative	(ratio)	relate to fractions	+ parts and zero
continuous	(real)	relate to infinite	+ compactness
complex	(imaginary)	relate to model	+ i, other unit bases

Sources of variation in measurement

- true differences being measured
- true differences due to some other factor
- transient personal factors
- situational factors
- variation in administration
- sampling of items and experiences
- lack of clarity of measurement instrument
- mechanical factors
- analysis errors

Types of reliability

stability over time, individual and population	(test-retest)
equivalence over instruments	(split-half)
power, relationship between sample size and size of difference	

Types of validity

pragmatic	does it work
construct	does it match the abstract idea
face	does it look like it is expected to (self-evident)
concurrent	does it differentiate
predictive	does it replicate

Research methodologies:

Research steps [Selltiz et al, 62]

- formulate problem
  - concepts and theory, working definitions, results from other studies
- design study
  - exploratory, descriptive, causal, before-after
- collect data
- analyze data
- interpret results of analysis

Research strategies [McGrath, 1994]

- theoretical
  - formal theory

- computer simulation
- experimental
  - laboratory experiment
  - experimental simulation
- field
  - field experiment
  - field study
- respondent
  - sample survey
  - judgment study

Experimental measurement techniques [McGrath, 1994]
- self reports
- trace measures
- observations by a visible observer
- observations by a hidden observer
- public archival records
- private archival records

Data collection methods [Selltiz et al, 62]
- unstructured observation
- structured observation
- interview
- questionnaire
- projective methods
- structured disguised tests
- statistical records
- personal documents
- mass communications
- rating scales
- questions which form scales

Case study techniques
- visual specifications
- iterative design
- rapid prototyping
- behavioral analysis
- empirical evaluation

Evaluation strategies
- heuristic with usability guidelines
- cognitive walkthrough
- usability testing
- usability engineering and metrics
- controlled experiment

Interaction evaluation tools
- state transition diagrams
- statistical analysis of random samples of behavior

exhaustive tracking
protocol analysis
clinical diagnosis and remediation
controlled experiment

Modeling with Graphs

Creating and Obscuring (Winograd)
- When we construct a software tool (or a mathematical model),
- we create within our world-view:
  - a particular collection of representations
  - a blindness to everything not expressible by those representations

Putnam and Functionalism
- Functionalism: the mind is functionally equivalent to a computer
- Putnam invents (1960), then repudiates (1990)
- Why functionalism is false
  - meaning is holistic (requiring even the non-represented aspects of a situation)
  - meaning is normative (defined by context and by negotiation)
  - concepts depend on evolution (defined in historical context and always evolving)
- Basically, mental states (definable shared cognitive objects) do not exist.

State transition model
- initial state
- final state
- decision points
- rule base for decisions
- problem space is all transitions from initial to final states

Central issues for the meaning of graphs
- formal or intuitive (mathematical or mystical)
- tokens or images
- open or closed system

in/out	perturbation
representational	non-representational
functional	autonomous
formed from outside	formed from inside
integrate languages	maintain organization

- what is a node and what is a link
- what do types of nodes and links mean
- what does connectivity mean
- what is static, what is process
- what is transmitted or exchanged
- what do labels mean

Dialogue

Qualities of a conversational interaction (Nickerson)

- bidirectionality
- mixed initiative
- apparentness of who is in control
- rules for transfer of control
- sense of presence
- nonverbal communication
- intolerance of silence
- helical structure
- characteristic time scale
- wide bandwidth
- informal language
- shared situational context
- common world knowledge
- shared special knowledge
- common history
- peer status of participants

Functions of a dialog manager

- receive and interpret input signals
- filter input errors, provide debugging feedback
- initiate error correction
- negotiate between user and computation about meaning of input (accommodate)
- integrate input into internal representation (assimilate)
- bypass computational levels for efficiency and clarity
- provide explanation, advice, help, justification.

Suchman's description of human-computer interaction

- contingent on unique circumstances which cannot anticipated
- activity is always concrete and embodied
- actions are never planned but are triggered by particular concrete circumstances
- representation cannot form a basis for interaction
- interaction means mutual intelligibility and shared understanding
- representation occurs when transparent activity becomes opaque
- language is indexical, shared meaning is contextual, understanding is collaborative

Design Guidelines and Multimedia

Interaction Description Tools

dialogs and scripts

Dialog: a sequence of information tokens exchanged between two or more agents
Script: a program which controls the exchange of tokens among agents

state transition diagram

a model which maps a token and a current state to a next state
Components:
- finite number of states
- set of transitions   f(current state, token) -> next state
- special state: Start
- special state(s): End

object and process graphs/hierarchies

object-oriented inheritance systems
calling sequences (functional hierarchy)
parse trees
cause and effect chains

concept modeling   (entity-relation graphs)

Entities: data which represents a single person, thing, concept, idea, or event (nouns)
Relations: associations between entities, including structure and organization, constraints, and invariants. Primary examples:
- IsSameAs (equivalence)
- IsA, Generalizes   (typing)
- IsPartOf, Contains   (hierarchy)
- IsMemberOf (sets)

grammars

Components:
- finite set of terminal symbols, representing semantic units
- finite set of non-terminal symbols, representing sub-trees
- set of production rules defining nonterminals

Standard form: *BNF*   Example:

```
<integer>  ::=   <digit> | <digit><digit>*
<digit>  ::=  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

context-free when nonterminal substitution has no dependency on adjacent nonterminals

rules, constraints, and inference

Components: a graph with two types of nodes
- slots: types and values for the attributes of an object
- rules: a transformation which generates a slot value from other slot values

Types of rules:
- upper or lower bound constraint
- enumeration of acceptable values constraint
- procedure invocation
- selection between several slot values
- function for calculating new slot value

multiple agents and communication models

shared memory: single records with regulated access
event handlers: continuous processes which respond to input events
event executive: process which prioritizes event handling (conflict resolution)
multithreaded dialogs
petri nets (information locations, synchronized transitions, arcs)

behavior modeling

task analysis: mapping the component steps/processes in a task
protocol analysis: mapping the component activities while doing a task
self-report: end user description and narration while doing a task
clinical diagnosis and remediation: single subject trouble shooting
controlled experiment: factoring the task into manipluated and measured variables

Visual Design (Mullet and Sano, Designing Visual Interfaces)

Elegance and simplicity

Qualities
- approachability, recognizability, immediacy, usability

Principles
- unity, refinement, fitness

Common errors
- clutter and visual noise
- interference between competing elements
- using explicit structure as a crutch
- belaboring the obvious
- overly literal translation
- excessive detail and embellishment
- gratuitous dimensionality

Techniques
- reduce a design to its essence
- regularize the elements of the design
- combine elements for maximum leverage

Scale, contrast, and proportion

Qualities
contrast, proportion, differentiation, emphasis, activity, interest
Principles
clarity, harmony, activity, restraint
Common errors
insufficient contrast
excessive contrast
visual interference
spatial tension
overextension
awkward dimensions
Techniques
squint test
establish perceptual layers
sharpen visual distinctions
integrate figure and ground

Organization and visual structure

Qualities
unity, integrity, readability, control
Principles
grouping, hierarchy, relationship, balance
Common errors
haphazard layout
conflicting symmetries
ambiguous internal relationships
aligning labels but not controls
alignment within but not across controls
false structure
excessive display density
all of the above
Techniques
use symmetry to ensure balance
use alignment to establish visual relationships
optical adjustment for human vision
shape the density with negative space

Module and program

Qualities
structure, predictability, efficiency
Principles
focus, flexibility, consistent application
Common errors
arbitrary component positions

arbitrary component dimensions
random window sizes and layouts
unrelated icon sizes and imagery
inconsistent control presentations
inconsistent visual language

Techniques

reinforce structure thorough repetition
establish modular units
create grid-based layout programs

Image and representation

Qualities

identification, expression, communication

Principles

immediacy, generality, cohesiveness, characterization, communicability

Common errors

misleading syntax
poorly integrated structure
dominant secondary elements
using type as image
using images for abstract concepts
images based on obscure allusions
culture or language dependencies
offensive or suggestive imagery

Techniques

selecting the right vehicle
refinement through progressive abstraction
coordination to ensure visual consistency

Style

Qualities

emotion, connection, context

Principles

distinctiveness, integrity, comprehensiveness, appropriateness

Common errors

unwarranted innovation
combining unrelated elements
partial fulfillment
internal and external inconsistency
incompatible concepts

Techniques

mastering a style
working across styles
extending and evolving a style

Cyberspace, Hypertext and the Web (R. Horn, Information Mapping)

Paper metaphors for hypertext

library card catalogues
footnotes
cross-reference
sticky notes
commentaries
indexes
quotes
anthologies

Computer metaphors for hypertext

linked note cards
popup notes
linked screens or windows
stretch text and outlines
semantic nets
branching stories
relational databases
simulations

Hypertext Links

system-supplied
- command and control pathways
- table of contents
- history tracking
- automated profiling

user-created
- detours and shortcuts
- notes, commentary, reminders
- analogical links
- new text
- links to other knowledge bases

author-created
- links to prerequisite knowledge
- hierarchical classification
- chronological structures

Kinds of links

hierarchical	building a tree
keyword	building an array
referential	building a pointer list
cluster	building a struct

Wayfinding in cyberspace (these don't work very well)

show all connections
go back to the beginning
show history of behavior

Node sizes

one sentence
text of arbitrary size (article, monograph)
index card size
screen size
scroll of any length
variable record sizing
variable size, precisely and flexibly chunked

Information types

structure
concept
procedure
process
classification
principle
fact

Information Blocks

chunking	small, manageable hunks (blocks, maps)
relevance	one main point per chunk, based on purpose or function to reader
consistency	similar words, labels, formats, organization
labeling	label every chunk based on specific criteria

Common types of information blocks

analogy	example	parts table
block diagram	fact	prerequisite
checklist	flow chart	principle
classification table	flow diagram	procedure table
classification tree	formula	purpose
comment	input-procedure-output	rule
cycle chart	non-example	stage
decision table	notation	synonym
definition	objectives	theorem
description	outlines	when to use
diagram	parts-function table	worksheet

Types of hypertrail, path

- prerequisite
- classification
- chronological
    - sequence of events
    - storyline
    - natural development
- geographic
- project
- structural
- decision
- definition
- example

How readers behave

- novices stop reading too soon
- novices are mislead by superficial features
- novices rarely seek non-linear information
- readers construct a hierarchical mental representation
- readers remember the top level of information better
- readers depend on repetition of keywords

Reading cues

- hierarchical text organization
- explicit transitions
- sequence signals
- contrast and similarity cues
- pronouns as cohesiveness cues
- metaphors
- content schemas

Document titles

- just right: not too general, too specific, too long, too short
- common language for the intended audience
- itemize all possible readers and use lowest common denominator
- no cuteness or silliness
- no vague, mislabeled topic headers
- same words in contents, titles, pages, and references

Virtual Reality (W. & M. Bricken)

The VR Paradigm Shift

We adapt to digital processes ==> digital processes adapt to us.

The VR shift from formalism to friendliness

physiological naturalness	responsive to human physiology
cognitive ease	responsive to human thinking patterns
environmental familiarity	transparent, responsive, interactive
whole body involvement	multisensory interface
embedded functionality	task-oriented affordances
behavioral information	spatial and experiential information

Design Paradigm Shifts (M. Bricken, No Interface to Design)

Interface to inclusion
Mechanism to intuition
User to participant
Visual to multimodal
Metaphor to virtuality

Varieties of Meaning

physical semantics:
map between digital representation and activity in the physical world
virtual semantics:
map between digital representation and perceived virtual world activity
natural semantics:
hiding the digital layer, map between physical activity and virtual consequence

Component Technologies

behavior transducers	map behavior onto computation and back
inclusive computation	software for management of environments
intentional psychology	integrate information, cognition and behavior
experiential design	unifying inclusion and intention feels good

VR functional integration of

realtime operating systems
sensor fusion
dynamic adaptive control
distributive and parallel processing
dynamic database management
coordination and communication techniques
biological/environmental modeling
physical dynamics

arbitrary interactivity
physiological and cognitive modeling
design of experience

VR operating system requirements

realtime interactive programming
multiple participants
parallel decomposition model
distributed, heterogeneous processing
arbitrary i/o mappings

System-oriented programming extends oop

Every entity is an *autonomous operating system*, controlling its own
- attributes
- resources (memory, processes, i/o)
- communication
- timing

Entities follow *biological/environmental models*, using commands such as
- enter
- perceive
- react
- persist

Spaces and environments are first-class
- behavior is situated and contextual

Viewpoint transformations

turn head
fly (interactive, 3-space functional curve, constrained pathway)
jack into location (instantly transport)
ride vehicle
inhabit
grasp world
multiple concurrent views
projection (over dimensions, abstractions)

The Wand

A position sensor on a laser pointer. The virtual form changes with function.

Viewpoint control
- sight (attach ray to head orientation)
- move faster/slower
- jack (teleport)
- scale (travel in size)

Object manipulation
- grasp
- normal (make object perpendicular to ray)
- come (bring object to participant)
- connect (construct a port on the object)
- cut (the ray is a knife)

feel (tactile feedback ray)

Information gathering

identify pointed at object
measure distance
count/compute environmental complexity

Other uses

Draw	select
light (the ray is a flashlight beam)	baton (direct sound events)

Divergent Worlds

Physical reality

Experience is unique for each person.
We perceive only instances.
Matter dictates consensus.
We negotiate differences.

Virtual reality

Form is unique for each participant.
We perceive possibilities.
Choice dictates consensus.
We negotiate communality.

Multiple participant group space

build mutual context rather than global truth
each participant is unique
credibility rather than validity
comprehension rather than consistency
inconsistency maintenance and uniqueness enforcement

VR bumper stickers

Psychology is the Physics of VR.
Our body is our interface.
Computation is in your hands.
One experience is worth a trillion bits.
The virtual is more than real.
VR is the first empirical tool of metaphysics.

Issues

ownership	information wants to be free
ethics	how is access and behavior controlled
fluid self	our virtual body is ethereal
intoxication	cognitive remodeling and dreaming in polygons
consistency	unique, private, interpenetrating worlds
post-symbolism	semantics takes a back seat
embedded virtuality	enhanced sensorium and private filters
rights of programs	self-reference and autonomy
actual or virtual	the line is very fuzzy