



Human Computer Interaction



Introduction to HCI



The world is full of badly designed things...



oPhoto + caption from Jacob Nielsen's Alertbox June 7, 2004



And well designed things...



<http://www.vision.org>



www.idsa.org



And things that look good but don't work



<http://baddesigns.org>



Does it matter?

If things are badly designed?

You may camp in the wrong place



You may crash your car and get injured.. Or worse!

... in the best case – you might be angry, make mistakes and things will take longer than they should

<http://baddesigns.org>

The 2000 USA Presidential Ballot in Florida

Confusion at Palm Beach County polls

Some Al Gore supporters may have mistakenly voted for Pat Buchanan because of the ballot's design.

Although the Democrats are listed second in the column on the left, they are the third hole on the ballot.

Punching the second hole casts a vote for the Reform party.

ELECTORS FOR PRESIDENT AND VICE PRESIDENT (A vote for the candidates will actually be a vote for their electors.) (Vote for Group)	(REPUBLICAN)	
	GEORGE W. BUSH - PRESIDENT	3 ➤
	DICK CHENEY - VICE PRESIDENT	
	(DEMOCRATIC)	
	AL GORE - PRESIDENT	5 ➤
	JOE LIEBERMAN - VICE PRESIDENT	
	(LIBERTARIAN)	
	HARRY BROWNE - PRESIDENT	7 ➤
	ART OLIVIER - VICE PRESIDENT	
	(GREEN)	
	RALPH NADER - PRESIDENT	9 ➤
	WINONA LaDUKE - VICE PRESIDENT	
	(SOCIALIST WORKERS)	
JAMES HARRIS - PRESIDENT	11 ➤	
MARGARET TROWE - VICE PRESIDENT		
(NATURAL LAW)		
JOHN HAGELIN - PRESIDENT	13 ➤	
NAT GOLDHABER - VICE PRESIDENT		
(REFORM)		
PAT BUCHANAN - PRESIDENT	4 ←	
EZOLA FOSTER - VICE PRESIDENT		
(SOCIALIST)		
DAVID McREYNOLDS - PRESIDENT	6 ←	
MARY CAL HOLLIS - VICE PRESIDENT		
(CONSTITUTION)		
HOWARD PHILLIPS - PRESIDENT	8 ←	
J. CURTIS FRAZIER - VICE PRESIDENT		
(WORKERS WORLD)		
MONICA MOOREHEAD - PRESIDENT	10 ←	
GLORIA La RIVA - VICE PRESIDENT		
WRITE-IN CANDIDATE		
To vote for a write-in candidate, follow the directions on the long stub of your ballot card.		

Sun-Sentinel graphic



How to Design Interactive Technology

- (and future stuff!)





To make better interactive technology We need to

- Know about how people **interact** with things
- Know about what people **can and can't do**
- Know about the **situations** in which people do things
- Know about the **basics** of good design
- Understand people's **goals**





We can't all be designers..

But we can all **learn** things about people and design that will help us create better stuff than we would otherwise have done...

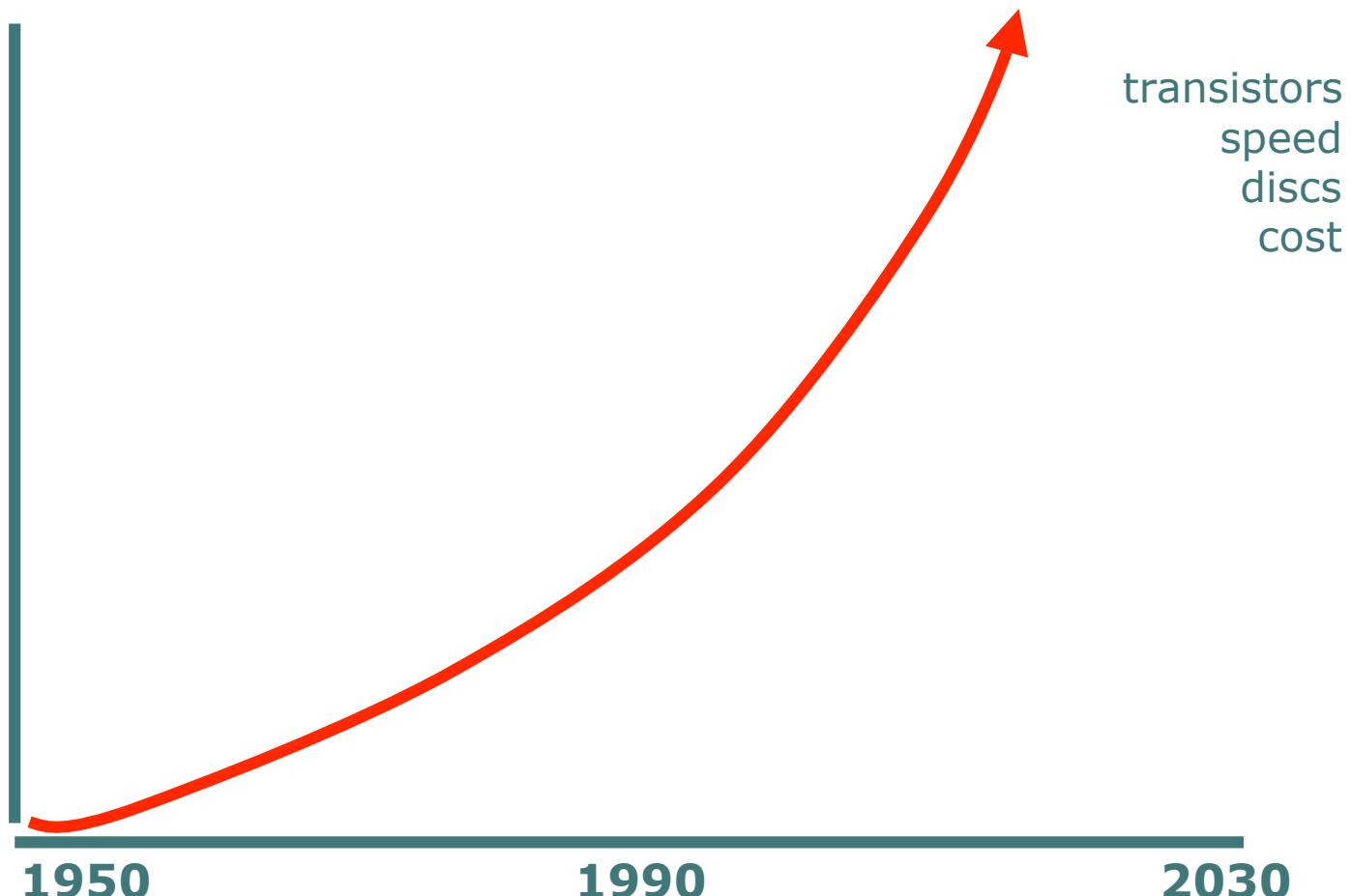


The Computer....



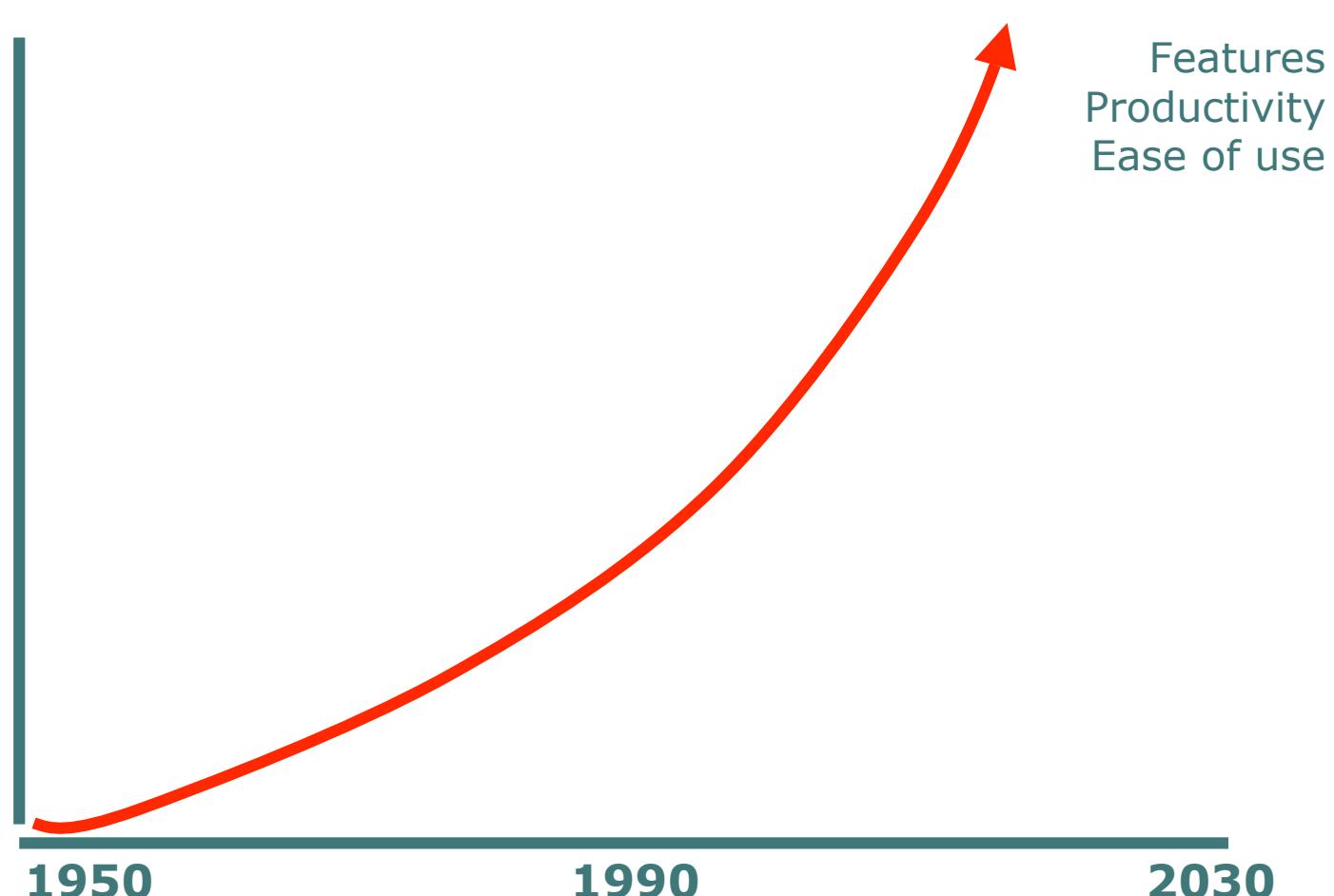


Moore's Law



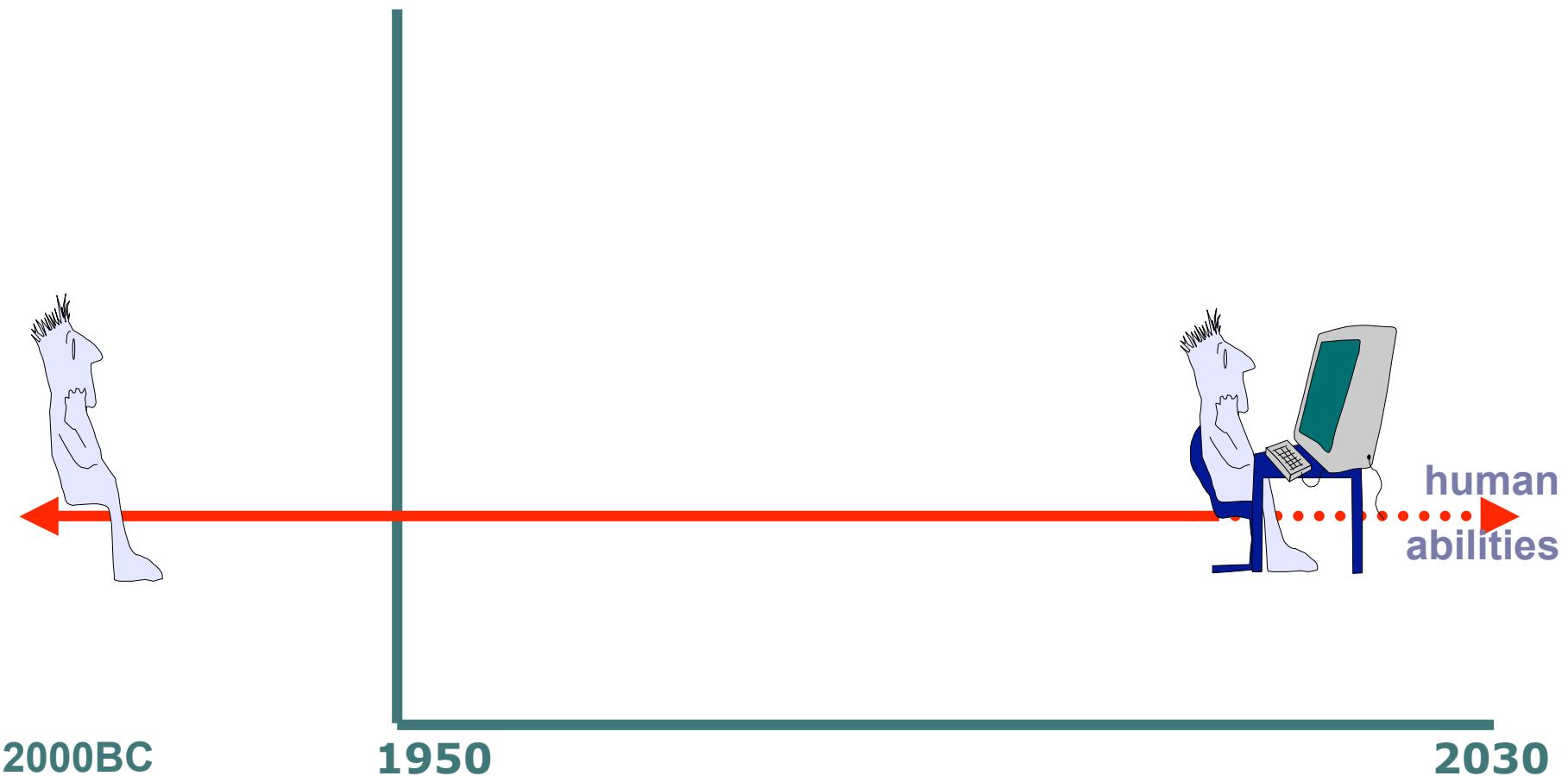


Buxton's Law





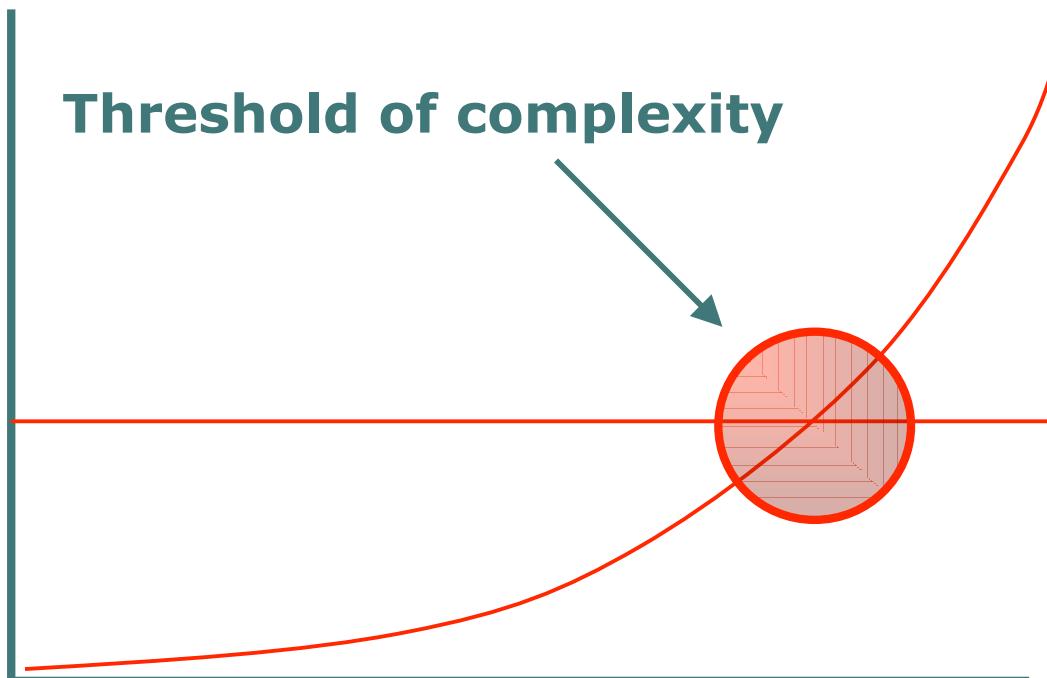
God's Law





Oops!

Threshold of complexity

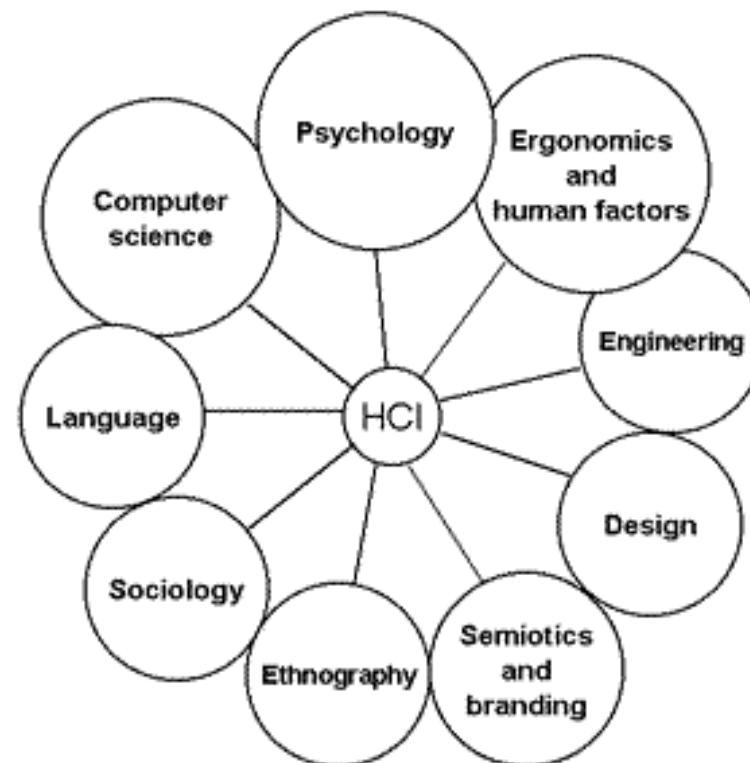




Human Computer Interaction

- Ergonomics
- Sociology
- Psychology
- ...and more

The Field of HCI
(Human Computer Interaction)





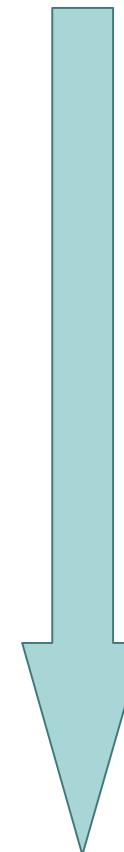
HCI is made up of...

- Theories – learn and apply
- Models – create and use
- Methods – master and apply
- Guidelines – learn and use
- Principles – understand and apply
- Techniques – master and use



HCI is changing..

- Physical things
- GUI interfaces
- Collaborative interfaces
- Internet technologies
- Social technologies
- Ubiquitous technologies
- ?





Understanding the Design Process

- What is the process?
- Where are good / bad decisions made
- Why are good / bad decisions made



Developing Interactive Products

- Stage 1 – design time
 - System developers, with or without user involvement, create environments and tools and sometimes provide complete solutions
- Stage 2 – use time
 - Stakeholders use the system and because their needs contexts and objectives can only be **anticipated** – some changes are generally made to the product at this stage



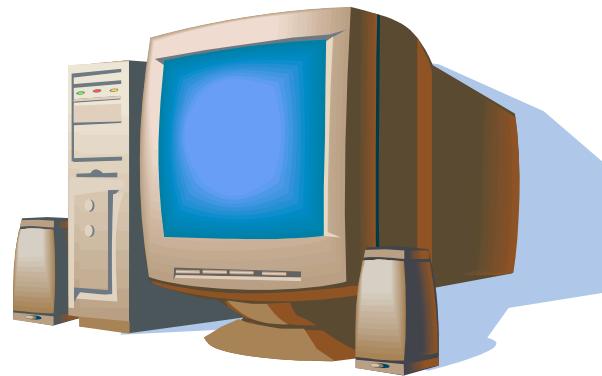
Efficient Development..

- Relies on the systems developer having a GOOD understanding of the stakeholders
 - Needs
 - Objectives
 - Context
- And.. Knowing what the technology can do!



The Focus of the development

- ... can vary..





System Centered Design



E-Mail: SCOTTADAMS@AOL.COM ©1994 United Feature Syndicate



System Centered Design

- What can I easily build on this platform?
- What can I create from the available tools?
- What do I as a programmer find interesting?





User Based...

- Design is based upon a user's
 - abilities and real needs
 - context
 - work
 - tasks
 - values



Methodologies

- Abilities and needs - User Centered Design
- Context - Contextual Design
- Work – Goal Directed Design
- Tasks - Task Centered Design
- Values – Value / Worth Centred Design

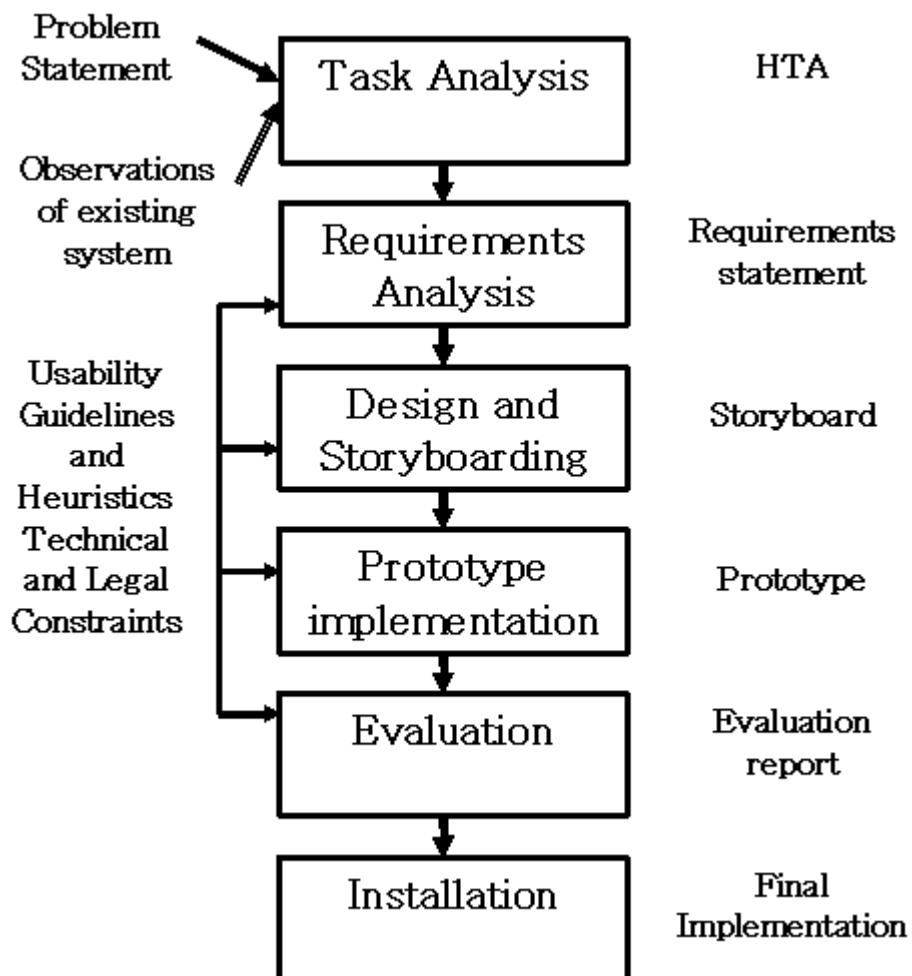


UCSD

- Focus on the people who will use the system, on their preferences and their requirements
- Building models of the users, tasks and systems
- Iterative process
- Prototyping and Evaluation by users



UCSD





Core Concepts in UCSD

- Early stages
 - User model
 - Task models
 - Context
- Later stages
 - Inclusive Design
 - Design for all
 - Universal Access



Our UCSD

- The Early stages

- Focus on users, tasks, contexts..
- But also (in our expanded view) on goals, and values



Who are Users in UCSD?

- People like you
- People like your mother!
- People like those you work with
- People that you may never have / will never meet
 - It is better to think in terms of who your users will **NOT** be



Video here....

- <http://www.computing.dundee.ac.uk/projects/UTOPIA/>



Tasks (and Goals)

- Tasks (limited value)
 - Not all systems are task based (eg Games)
 - Not all tasks can be easily specified
- Goals
 - Irrespective of the user tasks, there can be a conflicting or complementary goal – to have fun maybe?



Context

- Where will the system be used?
- What sort of technology will be deployed?
- Will the users be able to get any help?



Values

- What matters to the user
- What is important in the context
- What are the business concerns



Human Computer Interaction

Lecture 2 – Understanding the User



What is a User?

Is
Behaves
Feels





- Norman introduced the idea that product design should address three different levels of cognitive and emotional processing: *visceral*, *behavioral*, and *reflective*.



Visceral Processing

- . The most immediate level of processing, in which we react to visual and other sensory aspects of a product that we can perceive before significant interaction occurs. Visceral processing helps us make rapid decisions about what is good, bad, safe, or dangerous.



Behavioral Processing

- The middle level of processing that lets us manage simple, everyday behaviors, which according to Norman, constitute the majority of human activity. Norman states, “historically, interaction design and usability practices have primarily addressed this level of cognitive processing”.



Reflective Processing

- . The least immediate level of processing, which involves conscious consideration and reflection on past experiences. Reflective processing can enhance or inhibit behavioral processing, but has no direct access to visceral reactions. This level of cognitive processing is accessible only via memory, not through direct interaction or perception.



The 'is' of being Human

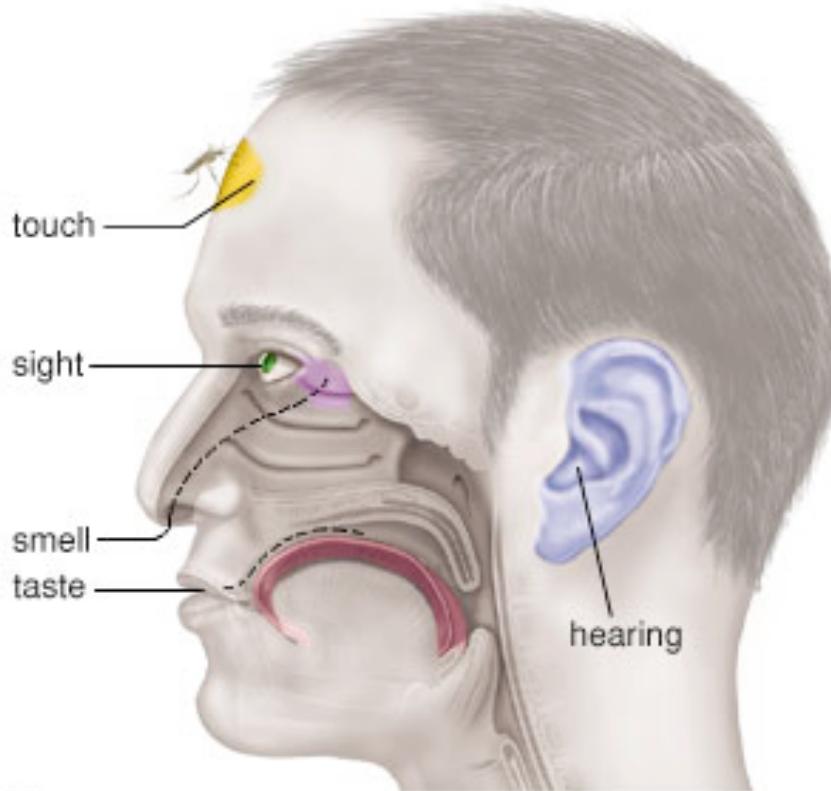
- Senses
- Body
- Thinking
- Memory



766 209 487



The Human as an Input Device



© 2006 Encyclopædia Britannica, Inc.

- How we make ‘sense’ of the world around us... inputs and understanding



Each Sense has...

- A tool – e.g.. Eye, skin, ear etc
- A process – nerves, electricity etc
- Limitations – pitch, brightness etc
- And there is the added complexity of individual differences in sensory perception

766 209 487

<http://www.hagenstoons.com>





Sight - Interpreting Images

(1)

- Size and depth
 - visual angle indicates how much of view object occupies
 - cues like overlapping help perception of size and depth
- Brightness
 - affected by luminance of object
 - measured by just noticeable difference
 - visual acuity increases with luminance as does flicker
- Colour
 - blue acuity is lowest
 - 8% males and 1% females colour blind

766 209 487



Interpreting images (2)

- The visual system compensates for:
 - movement
 - changes in luminance
- Context is used to resolve ambiguity
- Optical illusions sometimes occur due to over compensation

12
A B C

14

766 209 487



Hearing

- Provides information about environment:
distances, directions, objects etc.
- Physical apparatus:
 - outer ear, middle ear, inner ear –
- Key Sound Variations
 - pitch – sound frequency
 - loudness – amplitude
 - timbre – type or quality



766 209 487



Hearing (cont)

- Humans can hear frequencies from 20Hz to 15kHz
 - less accurate distinguishing high frequencies than low.
- Auditory system filters sounds
 - can attend to sounds over background noise.
 - for example, the cocktail party phenomenon.



Touch

- Provides important feedback about environment.
- May be key sense for someone who is visually impaired.
- Stimulus received via receptors in the skin:
 - thermoreceptors – heat and cold
 - nociceptors – pain
 - mechanoreceptors – pressure
- Some areas more sensitive than others e.g. fingers.
- Kinesthesia - awareness of body position
 - affects comfort and performance.





Smell and Taste

- Not much used in computer interfaces but olfactory interfaces using smell are currently being developed... as we will not use these senses we are not discussing them here!



The Human as a Store

- Humans have the capacity to remember and retrieve information... this affects the way they use technology



<http://www.cooperativeindividualism.org>



Three Different ‘Stores’

- Sensory buffers: momentary stores for stimuli received by the senses. This information, unless encoded in the short-term memory, is quickly lost.
- Short-term memory (or working memory): short-term memory acts as a store for information required fleetingly.
- Long-term memory: this forms the main resource for memory.



Short Term Memory

- An example of this would be recalling a telephone number long enough to write it down. Short-term memory degrades quickly, and has a limited capacity.
- Quick access time – 70ms
- Short term storage – 200ms (10 – 20 second decay time)
- Limited capacity
 - Length of sequence remembered in order = 7 ± 2 ([Miller, 1956](#))
chunks of data are similar
recency effect
- Maintained and increased with rehearsal
- Nowadays referred to as ‘Working memory’



Chunking and STM

- Short-term memory holds information that is actively being used (thought about, reasoned with).
- A chunk can be thought of as a single object that conveys a larger amount of information (like a Chinese ideogram).
- Examples of these include words, shapes and colours. However, the information decays in seconds as items are displaced by new items coming in.
- Icons are an example of chunked information on a desktop which allows users to distinguish between the various programs available to them.



LTM – Networked chunks

Here we store everything we ‘know’. Long-term memory is characterised by huge capacity, slow access time and relative accuracy over time.

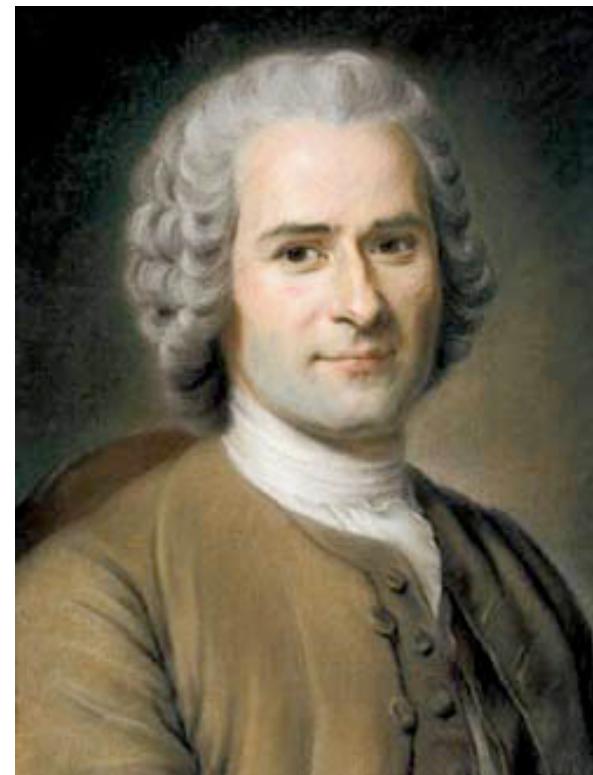
It is organised in an Episodic way
events and experiences in sequential order
.....and a Semantic way
facts, concepts and skills that we have acquired

- Storage
 - Structure, familiarity and concreteness Forgetting
 - Decay, interference
- Retrieval
 - **Recall** - reproduced
 - **Recognition** – clue given



The Human as a Philosopher

- Philosophers make sense of the world by thinking – traditionally they wonder about great things but in a micro level – humans think about small things



<http://www.realhistories.org.uk>



What is a Human - the Cartesian View

- 400 years old!
- Determinism - Cause and Effect
- Reductionist – to a machine
- Homogenous – all the same



<http://www.creationism.org>



How humans solve problems

- Deductive
- Inductive
- Abductive
 - reasoning





Deductive Reasoning

- Deduction:

derive logically necessary conclusion from given premises.

e.g. If it is Friday then she will go to work
It is Friday
Therefore she will go to work.

- Logical conclusion not necessarily true:

e.g. If it is raining then the ground is dry
It is raining
Therefore the ground is dry



Inductive Reasoning

- Induction:
 - generalize from cases seen to cases unseen
 - e.g. all elephants we have seen have trunks
 - therefore all elephants have trunks.
 - Unreliable:
 - can only prove false not true
- ... but useful!



Abductive reasoning

- reasoning from event to cause
 - e.g. Sam drives fast when drunk.
If I see Sam driving fast, assume drunk.
- Unreliable:
 - can lead to false explanations



Impact on HCI



- Understanding the limits of human abilities and the scope of their abilities is essential for good design..

<http://www.offthemarkcartoons.com>



Nugget

- Foraging Theory



Class announcements

- Time for coffee....



Models of Users

.. It's all about empathy...



Modelling Users using Personas

- Widely promoted by Alan Cooper as an integral component of Goal – Centred (Directed) design
- (Refer to www.interactionbydesign.com/presentations/olsen_persona_toolkit.pdf -)
- Personas are user archetypes that we construct from behavioural data gathered during *user interviews* and *field observations*..



The Power of Personas

- The personas are given realistic names, faces, and personalities to foster **user empathy** within a product team.
- Personas can be used later to evaluate the interactive product
- a critical element of personas—beyond capturing typical user behavior patterns and roles—is capturing user motivations in the form of specific goals.



Scenarios

- programmers to also design and program **test cases** for their own programs, sometimes before even writing the program itself.
- HCI approaches this in a similar fashion by **creating scenarios** they expect users to encounter or run through. One major distinction is that scenarios cover the **most likely** of cases and only sometimes include the edge cases. They will never, ever, cover every possible scenario.



Creating Scenarios

- “*A scenario is a concise description of a persona using a product to achieve a goal*” (Cooper).
 - **concise** : short but complete; breadth instead of depth
 - **product** : assume the product (software or physical device) exists, even if it doesn’t
 - **goal** : the reason why we perform a task



Why Use Scenarios?

- Scenarios help us validate our design
- Scenarios help us check our assumptions
- Successful Scenarios help us transfer theoretical/conceptual design to “wire frame” design
- Like Personas, Scenarios create a shared understanding of the end users –for the entire design team (including designers, marketing folk, engineers, executives, etc).



Example



- Steve texts using multitap, he is sending a message to a number he doesn't have in his contacts, it is a girl he met on the train and he wants to meet up with her - he has the number on a scrap of paper. He is walking down towards New Street Station, it is getting quite dark, he is texting without checking the screen,,,



Writing Good Scenarios

- Brainstorm, within the context of our problem domain, the goals our Personas will have
- Write the Scenarios for a specific Persona
- Go for breadth rather than depth – it is more important to describe things from start to finish rather than in exhaustive detail



Further Work

- In next weeks class we will look at the first part of the Coursework – this will include the use of personas and scenarios
- Research several web sites to see what experts say about personas and scenarios – post ‘good finds’ to Janet’s wiki..



Next Week

- We will consider the gathering of requirements – this will be a practical class with only a 15 minute introductory lecture
- Bring a phone with a camera if you have one!



Human Computer Interaction

Lecture 3 – Gathering Requirements



Gathering Requirements

People
Places
Processes



You can learn a lot...

- Watching People
 - In their workplaces
 - Doing tasks
 - Using technology
- Asking People
 - About their difficulties
 - About how they do things
 - About their goals and wishes..
-



Using Probes in HCI

- Cultural Probes

- Leave things lying around and see what happens to them
- A map or a postcard...

- Technology Probes

- Put technology in peoples hands and see what is captured



Probes and Interliving

- Context – cross generational communication
- Interesting – people in families



Methods

- Critical Incident Technique
- Relationship Map
- Cultural Probe
- Questionnaire
- Use Scenario



Human Computer Interaction

Inspecting Interfaces



Why Evaluate?

- In HCI we evaluate interfaces to:
 - Determine how usable they are for different user groups
 - Identify good and bad features to inform future design
 - Compare design choices to assist us in making decisions



Evaluation Methods

- Inspection methods (no users needed!)
 - Heuristic evaluations
 - Walkthroughs
 - Other Inspections
- User Tests (users needed!)
 - Observations
 - Usability tests



Inspection Methods

- Methods that are based on inspections
 - **Heuristic evaluation**
 - Heuristic estimation
 - **Cognitive walkthrough**
 - Feature inspection
 - Standards inspection
 - Pluralistic walkthrough
 - Consistency inspection
 - Formal usability inspection

One at a time!



Heuristic evaluation (*what is it?*)

- Method for finding usability problems
- Popularized by Jakob Nielsen
- “Discount” usability engineering
 - Use with working interface or scenario
 - Convenient
 - Fast
 - Easy to use



Heuristic evaluation

- Systematic inspection to see if interface complies to guidelines
- Method
 - 3-5 inspectors
 - usability engineers, end users, double experts...
 - inspect interface in isolation (~1–2 hours for simple interfaces)
 - compare notes afterwards
 - single evaluator only catches ~35% of usability problems
 - 5 evaluators catch 75%
- Works for paper, prototypes, and working systems



Points of Variation

- Evaluators
- Heuristics used
- Method employed during inspection



Evaluators

- These people can be novices or experts
 - “novice evaluators”
 - “regular specialists”
 - “double specialists” (- Nielsen)
- Each evaluator finds different problems
- The best evaluators find both hard and easy problems



Heuristics

- Heuristics are rules that are used to inform the inspection...
- There are many heuristic sets



Nielsen's Heuristics

1. Visibility of system status
2. Match between system & real world
3. User control and freedom
4. Consistency & standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility & efficiency of use
8. Minimalist design
9. Help error recovery
10. Help & documentation



1. Visibility of system status





What is “reasonable time”?

- 0.1 sec: Feels immediate to the user.
No additional feedback needed.
- 1.0 sec: Tolerable, but doesn’t feel
immediate. Some feedback needed.
- 10 sec: Maximum duration for keeping
user’s focus on the action.
- For longer delays, use % done
progress bars.



2. Match between the system and the real world

Socrates: *Please select command mode*

Student: Please find an author named
Octavia Butler.

Socrates: *Invalid Folio command: please*

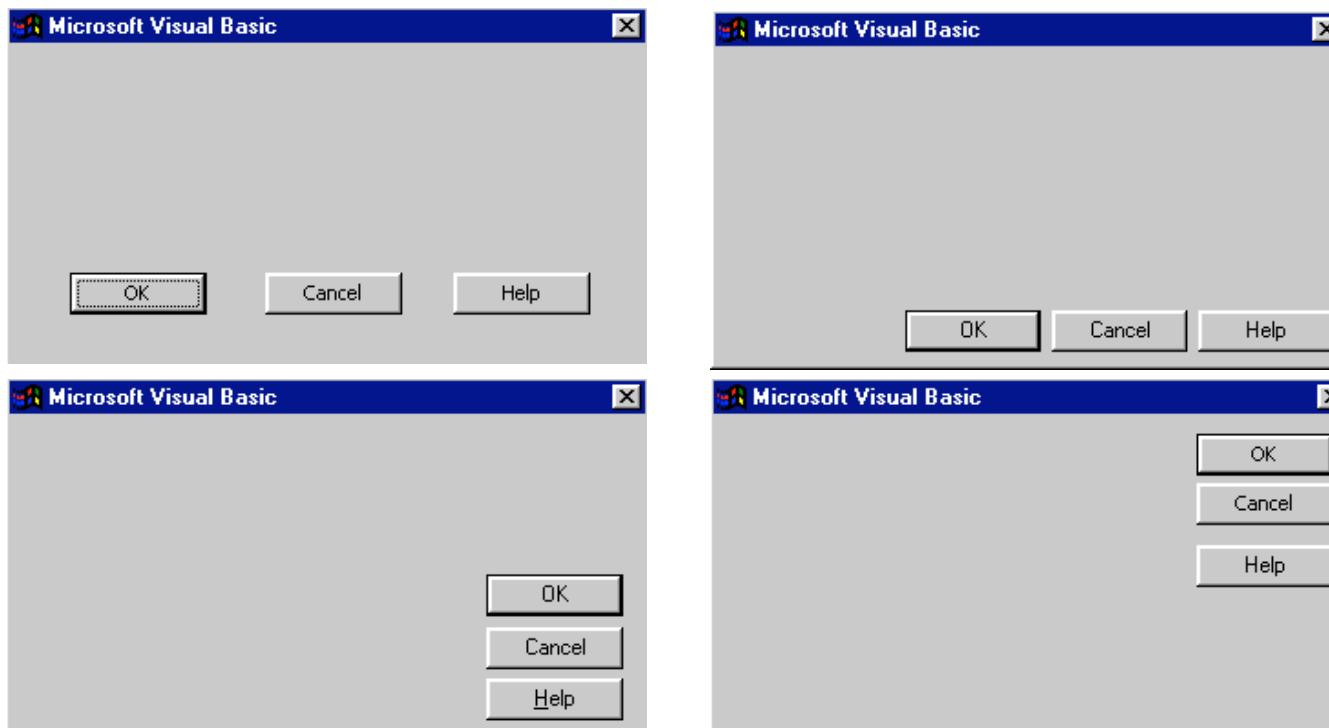


3. User control and freedom

- Provide exits for mistaken choices
- Enable undo, redo
- Don't force users to take a particular path



4. Consistency and standards





5. Error prevention

People make errors.

Yet we can try to prevent them.

*How might you go about trying
preventing errors?*



6. Recognition rather than recall

Ex: Can't copy info from one window to another

Violates: *Minimize the users' memory load*



7. Flexibility and efficiency of use

- Provide short cuts
- Enable macros

8. Aesthetic and minimalist design

Form Title -- (appears above URL in most browsers and is used by WWW search Q&D Software Development Order Desk		Background Color: FFFFBF0 <input type="button" value="..."/>
Form Heading -- (appears at top of Web page in bold type) Q&D Software Development Order Desk		Text Color: 000080 <input type="button" value="..."/>
E-Mail responses to (will not appear on dversch@q-d.com)	Alternate (for mailto forms only)	Background Graphic: <input type="button" value="..."/>
Text to appear in Submit button Send Order	Text to appear in Reset button Clear Form	<input type="radio"/> Mailto <input checked="" type="radio"/> CGI
Scrolling Status Bar Message (max length = 200 characters) ***WebMania 1.5b with Image Map Wizard is here!!***		
<input type="button" value="<< Prev Tab"/>		<input type="button" value="Next Tab >>"/>



9. Help users recognize, diagnose, and recover from errors

**SEGMENTATION VIOLATION! Error
#13**

**ATTEMPT TO WRITE INTO
READ-ONLY MEMORY!**

Error #4: NOT A TYPEWRITER



10. Help and documentation





We should wonder.....

- If this is a sensible heuristic set
 - Coverage
 - Uniqueness
 - Ease of use



Phases of a heuristic evaluation

1. Pre-evaluation training - give evaluators needed domain knowledge and information on the scenario
2. Evaluate interface independently
3. Rate each problem for **severity**
4. Aggregate results
5. Debrief: Report the results to the interface designers



Severity ratings

Each evaluator rates individually:

- 0 - don't agree that this is a usability problem
- 1 - cosmetic problem
- 2 - minor usability problem
- 3 - major usability problem; important to fix
- 4 - usability catastrophe; imperative to fix

Consider both impact and frequency.



Styles of Heuristic evaluation

- Problems found by a single inspector
- Problems found by multiple inspectors
- Individuals vs. teams
- Goal or task?
- Structured or free exploration?



Problems found by a single inspector

- Average over six case studies
 - 35% of all usability problems;
 - 42% of the major problems
 - 32% of the minor problems
- Not great, but
 - finding some problems with one evaluator is *much* better than finding no problems with no evaluators!



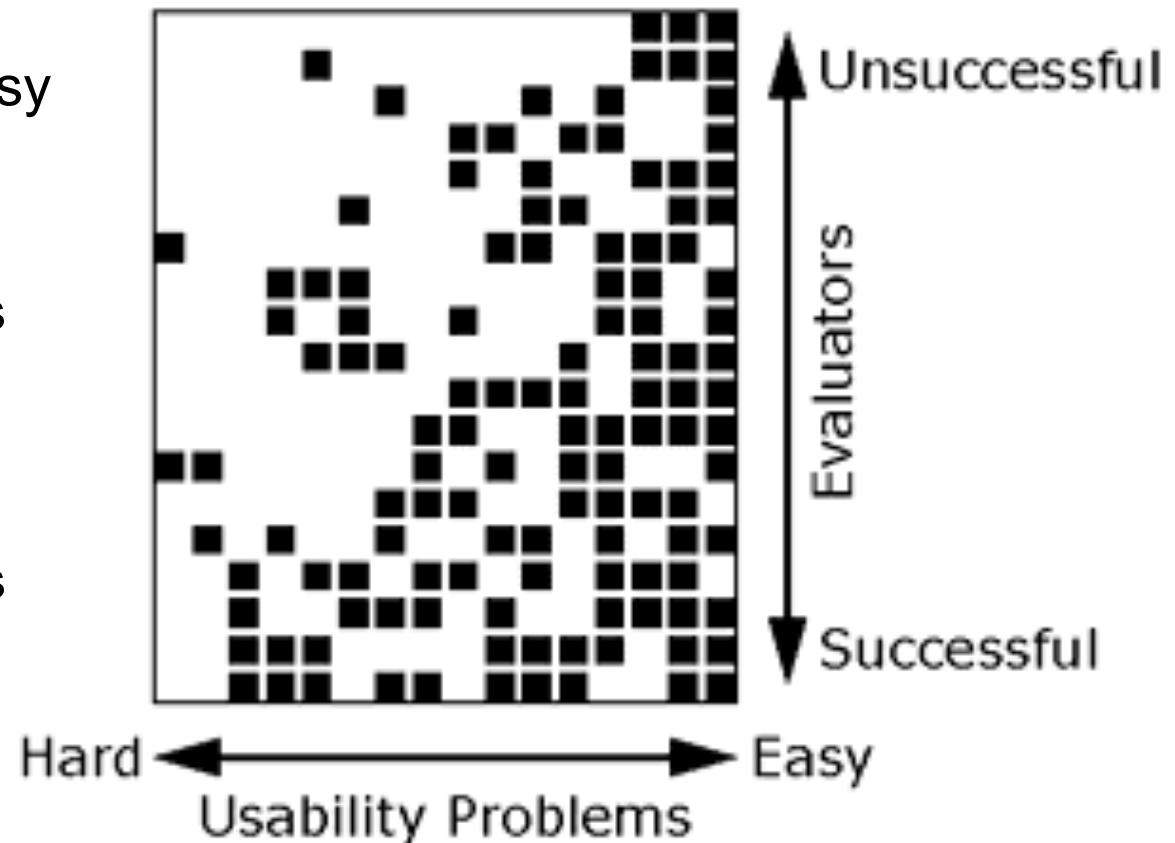
Problems found by a single inspector

- Varies according to
 - difficulty of the interface being evaluated
 - the expertise of the inspectors
- Average problems found by:
 - novice evaluators - no usability expertise - 22%
 - regular specialists - expertise in usability - 41%
 - double specialists - experience in both usability and the particular kind of interface being evaluated - 60%
 - also find domain-related problems
- Tradeoff
 - novices poorer, but cheaper!



Problems found by a single inspector

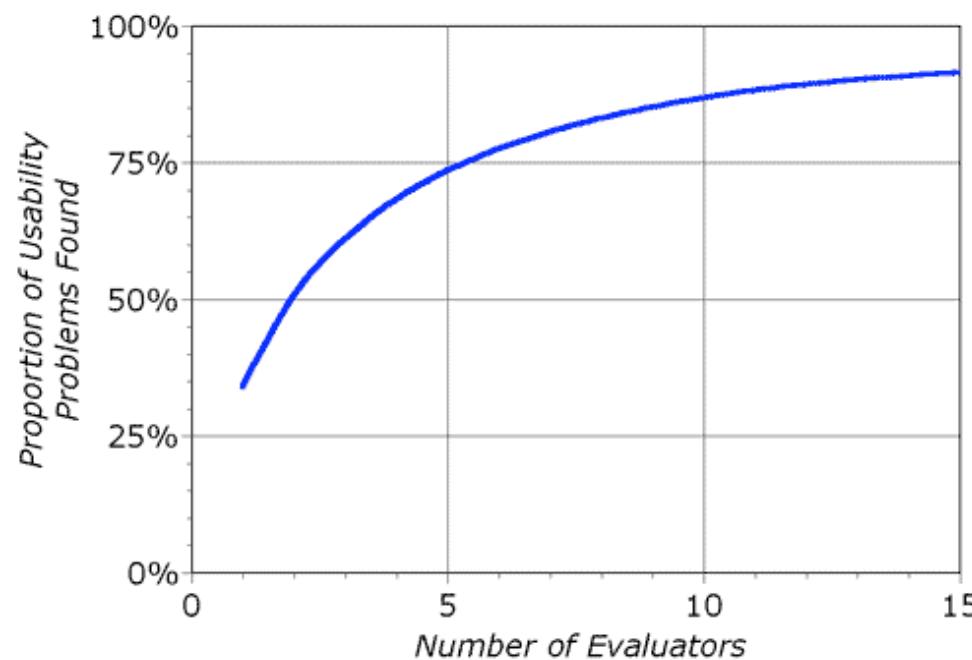
- Evaluators miss both easy and hard problems
 - ‘best’ evaluators can miss easy problems
 - ‘worse’ evaluators can discover hard problems





Problems found by multiple evaluators

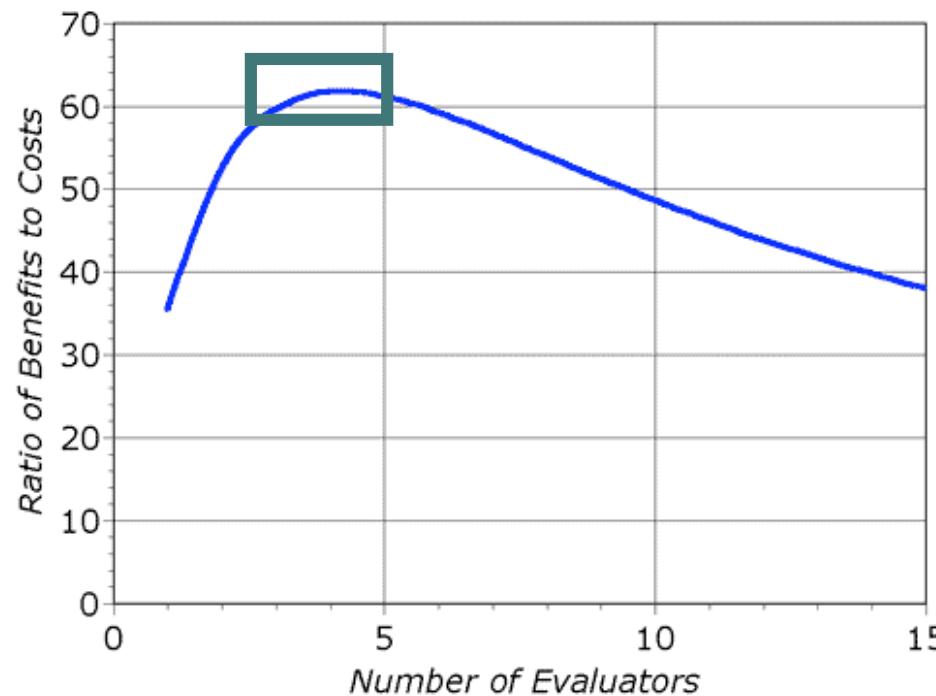
- 3-5 evaluators find 66-75% of usability problems
 - different people find different usability problems
 - only modest overlap between the sets of problems found





Problems found by multiple evaluators

- Where is the best cost/benefit?





Individuals vs. teams

- Nielsen
 - recommends individual evaluators inspect the interface alone
- Why?
 - evaluation is not influenced by others
 - independent and unbiased
 - greater variability in the kinds of errors found
 - no overhead required to organize group meetings



Self Guided vs. Scenario Exploration

- Self-guided
 - open-ended exploration
 - Not necessarily task-directed
 - good for exploring diverse aspects of the interface, and to follow potential pitfalls
- Scenarios
 - step through the interface using representative end user tasks
 - ensures problems identified in relevant portions of the interface
 - ensures that specific features of interest are evaluated
 - but limits the scope of the evaluation - problems can be missed



How useful are they?

- Inspection methods are *discount* methods for *practitioners*. They are not rigorous scientific methods.
 - All inspection methods are subjective.
 - No inspection method can compensate for inexperience or poor judgement.
 - Using multiple analysts results in an inter-subjective synthesis.
 - However, this also
 - a) raises the false alarm rate, unless a voting system is applied
 - b) reduces the hit rate if a voting system is applied!
 - Group synthesis of a prioritized problem list seems to be the most effective current practical approach.



Human Computer Interaction

Designing Systems that work for
People



design rules

Designing for maximum usability
– the goal of interaction design

- Principles of usability
 - general understanding
- Standards and guidelines
 - direction for design



types of design rules

- principles
 - abstract design rules
 - low authority
 - high generality
 - standards
 - specific design rules
 - high authority
 - limited application
 - guidelines
 - lower authority
 - more general application
- increasing generality
- Guidelines
- Standards
- increasing authority



Principles to support usability

Learnability

the ease with which new users can begin effective interaction and achieve maximal performance

Flexibility

the multiplicity of ways the user and system exchange information

Robustness

the level of support provided the user in determining successful achievement and assessment of goal-directed behaviour



Principles of learnability (1)

Predictability

- determining effect of future actions based on past interaction history
- operation visibility

Synthesizability

- assessing the effect of past actions
- immediate vs. eventual honesty



Principles of learnability (2)

Familiarity

- how prior knowledge applies to new system
- guessability; affordance

Generalizability

- extending specific interaction knowledge to new situations

Consistency

- likeness in input/output behaviour arising from similar situations or task objectives



Principles of flexibility (1)

Dialogue initiative

- freedom from system imposed constraints on input dialogue
- system vs. user pre-emptiveness

Multithreading

- ability of system to support user interaction for more than one task at a time
- concurrent vs. interleaving; multimodality

Task migratability

- passing responsibility for task execution between user and system



Principles of flexibility (2)

Substitutivity

- allowing equivalent values of input and output to be substituted for each other
- representation multiplicity; equal opportunity

Customizability

- modifiability of the user interface by user (adaptability) or system (adaptivity)



Principles of robustness (1)

Observability

- ability of user to evaluate the internal state of the system from its perceivable representation
- browsability; defaults; reachability; persistence; operation visibility

Recoverability

- ability of user to take corrective action once an error has been recognized
- reachability; forward/backward recovery; commensurate effort



Principles of robustness (2)

Responsiveness

- how the user perceives the rate of communication with the system
- Stability

Task conformance

- degree to which system services support all of the user's tasks
- task completeness; task adequacy



Standards

- set by national or international bodies to ensure compliance by a large community of designers
standards require sound underlying theory and slowly changing technology
- hardware standards more common than software high authority and low level of detail
- ISO 9241 defines usability as effectiveness, efficiency and satisfaction with which users accomplish tasks



Guidelines

- more suggestive and general
- many textbooks and reports full of guidelines
- abstract guidelines (principles) applicable during early life cycle activities
- detailed guidelines (style guides) applicable during later life cycle activities
- understanding justification for guidelines aids in resolving conflicts



Golden rules and heuristics

- “Broad brush” design rules
- Useful check list for good design
- Better design using these than using nothing!
- Different collections e.g.
 - Nielsen’s 10 Heuristics
 - Shneiderman’s 8 Golden Rules
 - Norman’s 7 Principles



Shneiderman's 8 Golden Rules

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



Norman's 7 Principles

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



Universal Design

- equitable use
- flexibility in use
- simple and intuitive to use
- perceptible information
- tolerance for error
- low physical effort
- size and space for approach and use



Multi-Sensory Systems

- More than one sensory channel in interaction
 - e.g. sounds, text, hypertext, animation, video, gestures, vision
- Used in a range of applications:
 - particularly good for users with special needs, and virtual reality
- Will cover
 - general terminology
 - speech
 - non-speech sounds
 - handwriting
- considering applications as well as principles



Usable Senses

The 5 senses (sight, sound, touch, taste and smell) are used by us every day

- each is important on its own
- together, they provide a fuller interaction with the natural world

Computers rarely offer such a rich interaction

Can we use all the available senses?

- ideally, yes
- practically – no

We can use • sight • sound • touch (sometimes)

We cannot (yet) use • taste • smell



Multi-modal vs. Multi-media

- Multi-modal systems
 - use more than one sense (or mode) of interaction

e.g. visual and aural senses: a text processor may speak the words as well as echoing them to the screen
- Multi-media systems
 - use a number of different media to communicate information

e.g. a computer-based teaching system:may use video, animation, text and still images: different media all using the visual mode of interaction; may also use sounds, both speech and non-speech: two more media, now using a different mode



Speech

Human beings have a great and natural mastery of speech

- makes it difficult to appreciate the complexities

but

- it's an easy medium for communication



Structure of Speech

phonemes

- 40 of them
- basic atomic units
- sound slightly different depending on the context they are in, these larger units are ...

allophones

- all the sounds in the language
- between 120 and 130 of them
- these are formed into ...

morphemes

- smallest unit of language that has meaning.



Speech (cont'd)

Other terminology:

- prosody
 - alteration in tone and quality
 - variations in emphasis, stress, pauses and pitch
 - impart more meaning to sentences.
- co-articulation
 - the effect of context on the sound
 - transforms the phonemes into allophones
- syntax – structure of sentences
- semantics – meaning of sentences



Speech Recognition Problems

- Different people speak differently:
 - accent, intonation, stress, idiom, volume, etc.
- The syntax of semantically similar sentences may vary.
- Background noises can interfere.
- People often “ummm....” and “errr....”
- Words not enough - semantics needed as well
 - requires intelligence to understand a sentence
 - context of the utterance often has to be known
 - also information about the subject and speaker

e.g. even if “Errr.... I, um, don’t like this” is recognised, it is a fairly useless piece of information on its own



Speech Recognition: useful?



- Single user or limited vocabulary systems
 - e.g. computer dictation
- Open use, limited vocabulary systems can work satisfactorily
 - e.g. some voice activated telephone systems



- general user, wide vocabulary systems ...
 - ... still a problem



- Great potential, however
 - when users hands are already occupied
 - e.g. driving, manufacturing
 - for users with physical disabilities
 - lightweight, mobile devices



Speech Synthesis

The generation of speech

Useful

- natural and familiar way of receiving information

Problems

- similar to recognition: prosody particularly

Additional problems

- intrusive - needs headphones, or creates noise in the workplace
- transient - harder to review and browse



Speech Synthesis: useful?

Successful in certain constrained applications
when the user:

- is particularly motivated to overcome problems
- has few alternatives

Examples:

- screen readers
 - read the textual display to the user utilised by visually impaired people
- warning signals
 - spoken information sometimes presented to pilots whose visual and haptic skills are already fully occupied



Non-Speech Sounds

boings, bangs, squeaks, clicks etc.

- commonly used for warnings and alarms
- Evidence to show they are useful
 - fewer typing mistakes with key clicks
 - video games harder without sound
- Language/culture independent, unlike speech



Non-Speech Sounds: useful?

- Dual mode displays:
 - information presented along two different sensory channels
 - redundant presentation of information
 - resolution of ambiguity in one mode through information in another
- Sound good for
 - transient information
 - background status information

e.g. Sound can be used as a redundant mode in the Apple Macintosh; almost any user action (file selection, window active, disk insert, search error, copy complete, etc.) can have a different sound associated with it.



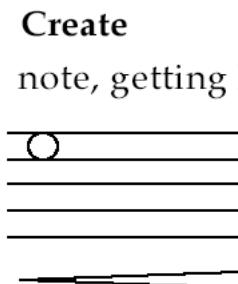
Auditory Icons

- Use natural sounds to represent different types of object or action
- Natural sounds have associated semantics which can be mapped onto similar meanings in the interaction
 - e.g. throwing something away
 - ~ the sound of smashing glass
- Problem: not all things have associated meanings
- Additional information can also be presented:
 - muffled sounds if object is obscured or action is in the background
 - use of stereo allows positional information to be added

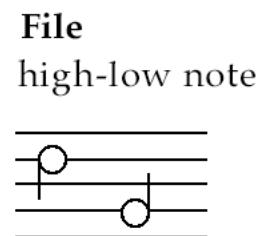


Earcons

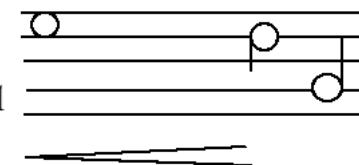
- Synthetic sounds used to convey information
- Structured combinations of notes (motives) represent actions and objects
- Motives combined to provide rich information
 - compound earcons
 - multiple motives combined to make one more complicated earcon



30



Create file
create icon followed
by file icon





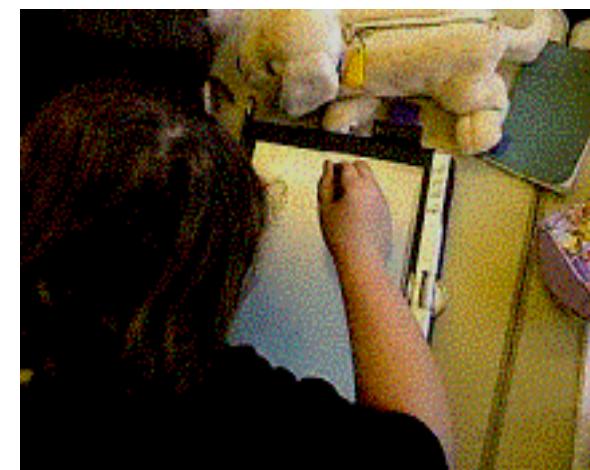
touch

- haptic interaction
 - cutaneous perception
 - tactile sensation; vibrations on the skin
 - kinesthetics
 - movement and position; force feedback
- information on shape, texture, resistance, temperature, comparative spatial factors
- example technologies
 - electronic braille displays
 - force feedback devices e.g. Phantom
 - resistance, texture



Writing and Text Input

- Writing – (1) Process - a creative act that generally results in original text (2) Product - the output from a pen activity
- Text Input – a function that generally results in text appearing in a machine readable form





Research Questions

- Should Children be taught Keyboarding?
- When is the RIGHT (if there is a right) age?
- Can Computers assist in Literacy?
- Whither the Word Processor?
- What would be NATURAL?





4. Whither the Word Processor?

- GOOD AT

- Making work beautiful
- Font manipulation
- Rearranging layout
- Alerting to grammar and spelling

- POOR AT

- Incorporating drawings
- Correcting grammar
- Correcting spellings
- Individualisation





5. What would be natural?

- For Writing
 - Speech
 - ‘Text’ and illustration
- For Text Input
 - Speech





.....Enter Stage Right..





Digital Ink

Refers to technology that digitally represents handwriting in its natural form. In a typical digital ink system, a digitizer is laid under or over an LCD screen to create an electromagnetic field that can capture the movement of a special-purpose pen, or stylus, and record the movement on the LCD screen. The effect is like writing on paper with liquid ink. The recorded handwriting can then be saved as handwriting or converted to typewritten text using handwriting recognition technology.

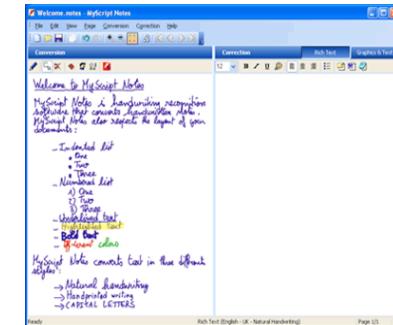


Hardware





Software



- Text Manipulation tools (IOTools, Journal)
 - Keep the digital ink as ink
 - Allow erasing, insertion, movement and searching
- Recognition tools (MyScript, Microsoft)
 - Change the digital ink into ASCII text
 - Allow more accurate spelling checking



Research Questions

- Does using digital ink affect keyboard progress?
- Can children use digital ink?
- Can the handwriting recognition work?
- Should the handwriting recognition work?
- What is digital ink well suited to?



Research Questions

- Does using digital ink affect keyboard progress? U/K
- Can children use digital ink? YES
- Can the handwriting recognition work?
POTENTIAL
- Should the handwriting recognition work?
DEPENDS
- What is digital ink well suited to? See the
next slide



Handwriting Recognition

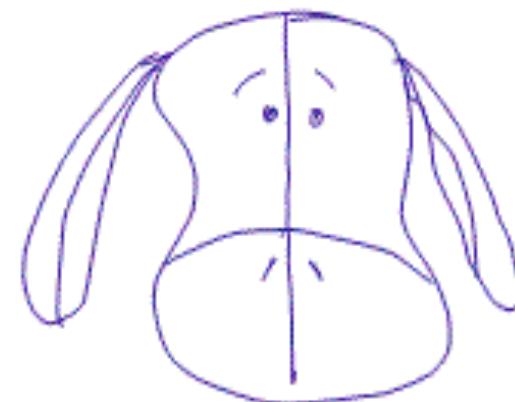
Once I went to the shop and bought some sandwich sweets, drink and ice cream. I went back home and told my mum that I'm going up stairs I went upstairs and I kept on eating my sweets sweets. Then I went down stairs and my mum said "its tea time" I said oh no I'm not hungry.



Useable and Used?

- Location
- Orientation
- Personalisation

I
♥
O.B
4eva

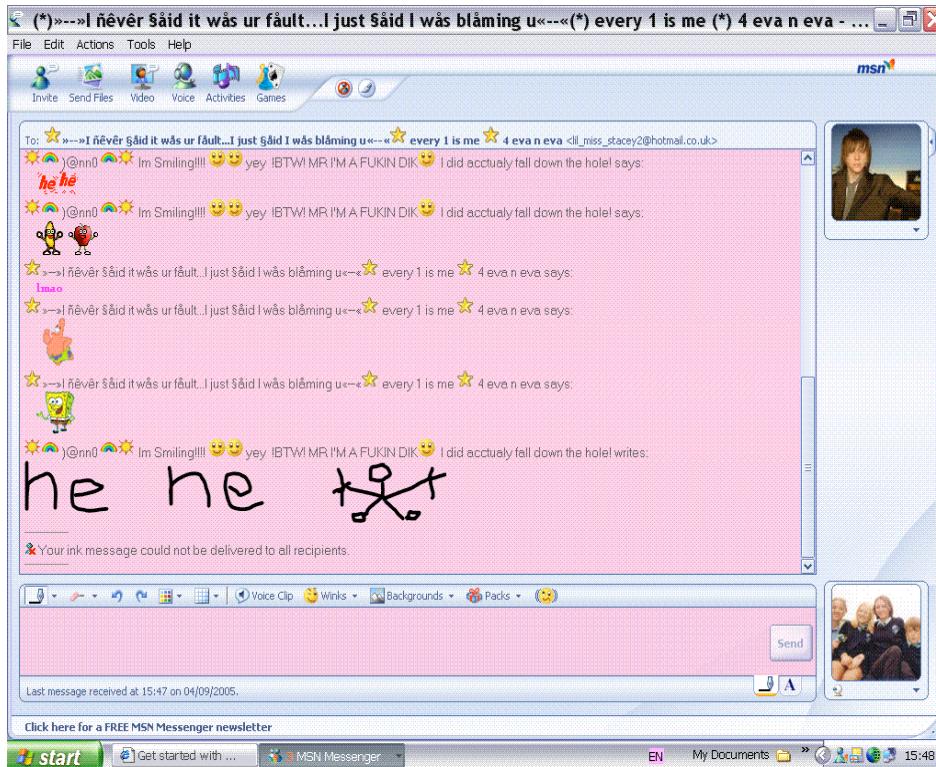


All work and no play
Did you have a good time!

- Digital doodling



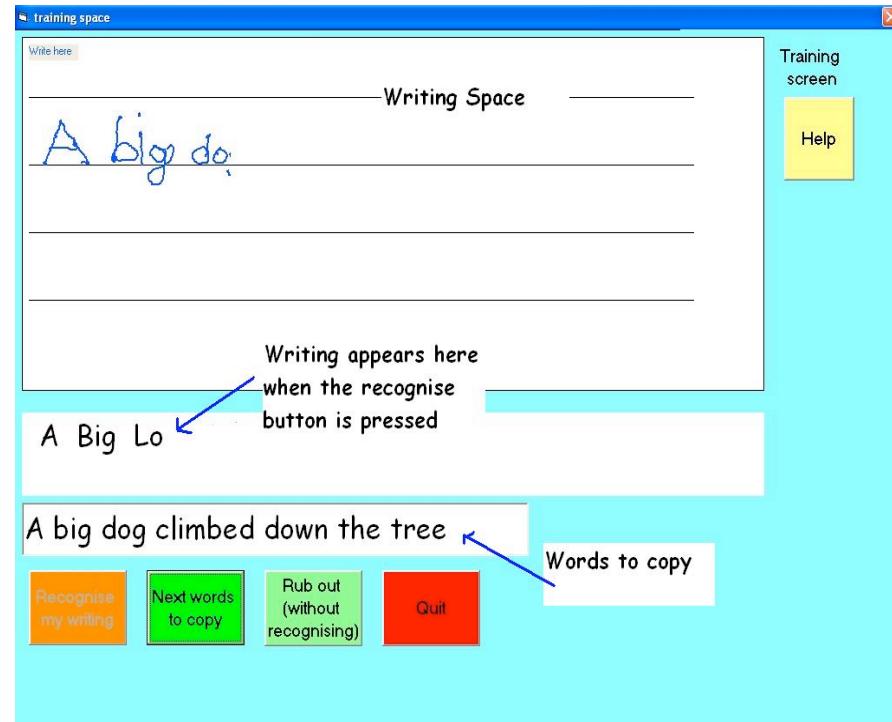
Personalisation





Handwriting Recognition

- Speech failed
- 2000, HR just going global
- Early studies used tablets
- Could children use it for text input?





Usability Problems and Solutions

- Errors
 - Spelling – phonic spellchecking software and speak out
 - Execution errors – tablet PC, training, time lag, cursive writing
 - Recognition (software) – better algorithms!
- Rubbing out / scribbling
- Starting in the top left hand corner
- Fixing bad writing
- Adding missing things



Users with disabilities

- visual impairment
 - screen readers, SonicFinder
- hearing impairment
 - text communication, gesture, captions
- physical impairment
 - speech I/O, eyegaze, gesture, predictive systems (e.g. Reactive keyboard)
- speech impairment
 - speech synthesis, text communication
- dyslexia
 - speech input, output
- autism
 - communication, education



... plus ...

- age groups
 - older people e.g. disability aids, memory aids, communication tools to prevent social isolation
 - children e.g. appropriate input/output devices, involvement in design process
- cultural differences
 - influence of nationality, generation, gender, race, sexuality, class, religion, political persuasion etc. on interpretation of interface features
 - e.g. interpretation and acceptability of language, cultural symbols, gesture and colour



Human Computer Interaction

Prototyping



Overview

- Prototyping and construction
- Conceptual design
- Physical design
- Generating prototypes
- Tool support



Prototyping and Construction

- What is a prototype?
- Why prototype?
- Different kinds of prototyping
 - low fidelity
 - high fidelity
- Compromises in prototyping
 - vertical
 - horizontal
- Construction



What is a Prototype

- In other design fields a prototype is a small-scale model:
 - a miniature car
 - a miniature building or town





What is a Prototype

- In interaction design it can be (among other things):
 - a series of screen sketches
 - a storyboard, i.e. a cartoon-like series of scenes
 - a Powerpoint slide show
 - a video simulating the use of a system
 - a lump of wood (e.g. PalmPilot)
 - a cardboard mock-up
 - a piece of software with limited functionality written in the target language or in another language



Why Prototype

- Evaluation and feedback are central to interaction design
- Stakeholders can see, hold, interact with a prototype more easily than a document or a drawing
- Team members can communicate effectively
- You can test out ideas for yourself
- It encourages reflection: very important aspect of design
- Prototypes answer questions, and support designers in choosing between alternatives



What to Prototype

- Technical issues
- Work flow, task design
- Screen layouts and information display
- Difficult, controversial, critical areas



Lo-Fi Prototyping

- Uses a medium which is unlike the final medium, e.g. paper, cardboard
- Is quick, cheap and easily changed
- Examples:
 - sketches of screens, task sequences,
 - 'Post-it' notes
 - storyboards
 - 'Wizard-of-Oz'



Why Use Low-Fi Prototypes

- Traditional methods take too long
 - Sketches → prototype → evaluate → iterate
- Can simulate the prototype
 - Sketches act as prototype
 - Designer “plays computer”
 - Other design team members observe & record
 - Might sound silly, but is surprisingly effective
- Kindergarten implementation skills
 - Allows non-programmers to participate
- Widely used in industry



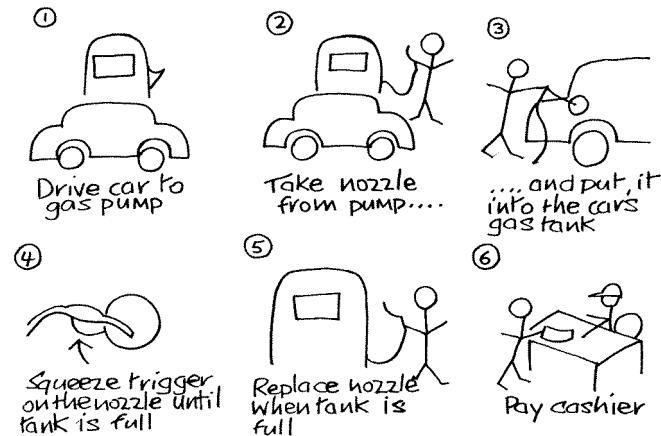
Storyboards

- Often used with scenarios, bringing more detail, and a chance to role play
- It is a series of sketches showing how a user might progress through a task using the device
- Used early in design



Sketching

- Sketching is important to low-fidelity prototyping
- Don't be inhibited about drawing ability. Practice simple symbols





Card-based prototypes

- Index cards (3 X 5 inches)
- Each card represents one screen or part of screen
- Often used in website development

Travel
Organiser

23 August 2006

WELCOME HELEN

Where do you want to go?

What date do you want to travel?

Which form of transport do you want? TRAIN

Do you need accommodation? YES

Travel
Organiser

23 August 2006

Train timetable from Milton Keynes Central
to York
on 16.09.06

Depart	09:09	10:09	same	22:09
Arrive	12:30	13:30	mins past hour	01:30

Accommodation Hotel B&B
£40 to £150 £20 to £60



Wizard of Oz

- The user thinks they are interacting with a computer, but a developer is responding to output rather than the system.
- Usually done early in design to understand users' expectations
- What is 'wrong' with this approach?



Hi-Fi Prototyping

- Uses materials that you would expect to be in the final product.
- Prototype looks more like the final system than a low-fidelity version.
- For a high-fidelity software prototype common environments include Macromedia Director, Visual Basic, and Smalltalk.
- Danger that users think they have a full system.....see compromises



Compromises

- All prototypes involve compromises
- For software-based prototyping maybe there is a slow response? sketchy icons? limited functionality?
- Two common types of compromise
 - ‘horizontal’: provide a wide range of functions, but with little detail
 - ‘vertical’: provide a lot of detail for only a few functions
- Compromises in prototypes mustn’t be ignored. Product needs engineering



Hi-Fi Disadvantages

- Distort perceptions of the tester
 - Formal representation indicates “finished” nature
 - People comment on color, fonts, and alignment
- Discourages major changes
 - Testers don’t want to change a “finished” design
 - Designers don’t want to lose effort put into creating the hi-fi design

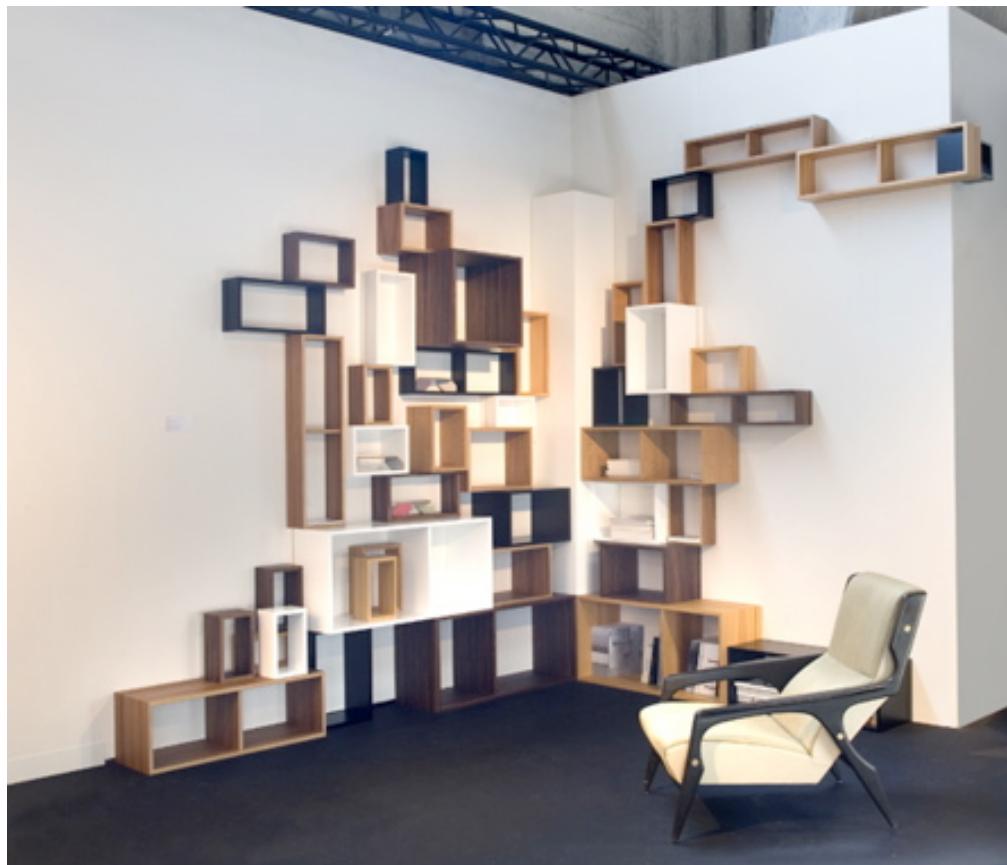


Construction

- Taking the prototypes (or learning from them) and creating a whole
- Quality must be attended to: usability (of course), reliability, robustness, maintainability, integrity, portability, efficiency, etc.
- Product must be engineered
 - Evolutionary prototyping
 - ‘Throw-away’ prototyping



So how do I design?





Conceptual design – R to D

- Transform user requirements/needs into a conceptual model
- “a description of the proposed system in terms of a set of integrated ideas and concepts about what it should do, behave and look like, that will be understandable by the users in the manner intended”
- Don’t move to a solution too quickly. Iterate, iterate, iterate
- Consider alternatives: prototyping helps



Interface Metaphors

- Interface metaphors combine familiar knowledge with new knowledge in a way that will help the user understand the product.
- Three steps: understand functionality, identify potential problem areas, generate metaphors
- Evaluate metaphors:
 - How much structure does it provide?
 - How much is relevant to the problem?
 - Is it easy to represent?
 - Will the audience understand it?
 - How extensible is it?



Interaction Types

- Which interaction type?
 - How the user invokes actions
 - Instructing, conversing, manipulating or exploring
- Do different interface types provide insight?
 - WIMP, shareable, augmented reality, etc



Expanding the model

- What functions will the product perform?
 - What will the product do and what will the human do (task allocation)?
- How are the functions related to each other?
 - Sequential or parallel?
 - Categorisations, e.g. all actions related to telephone memory storage
- What information needs to be available?
 - What data is required to perform the task?
 - How is this data to be transformed by the system?

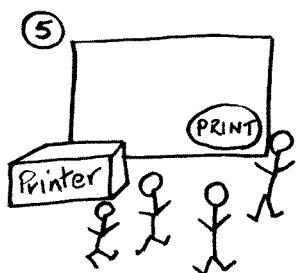
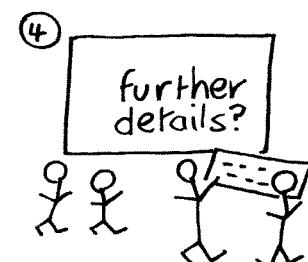
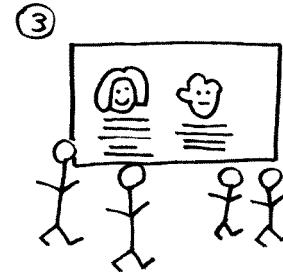
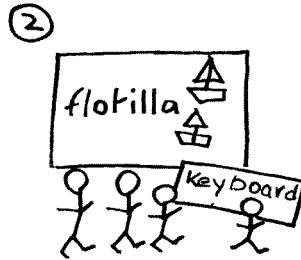
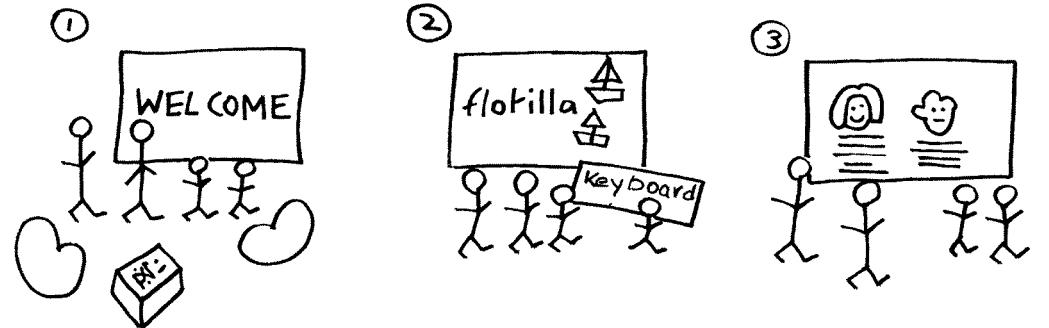


Using Scenarios

- Express proposed or imagined situations
- Used throughout design in various ways
 - scripts for user evaluation of prototypes
 - concrete examples of tasks
 - as a means of co-operation across professional boundaries
- Plus and minus scenarios to explore extreme cases



Generating Storyboards





Generating Cards...

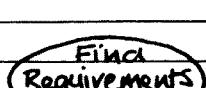
TRAVEL INFORMATION	
Visa requirements	
Vaccination Recommendations	
What to pack before you go	

VISA REQUIREMENTS	
Destination Country	<input checked="" type="checkbox"/>
Traveller's Nationality	<input checked="" type="checkbox"/>
<i>Final Requirements</i>	

VISA REQUIREMENTS FOR (COUNTRY)	
<i>Print</i>	

- Generate card-based prototype from use case

- TRAVEL INFORMATION
- Visa requirements
- Vaccination Recommendations
- What to pack before you go

VISA REQUIREMENTS	
Destination Country	<input type="checkbox"/>
Traveller's Nationality	<input checked="" type="checkbox"/>
 Finca Requirements	

VISA REQUIREMENTS FOR (COUNTRY)



Summary

- Different kinds of prototyping are used for different purposes and at different stages
- Prototypes answer questions, so prototype appropriately
- Construction: the final product must be engineered appropriately
- Conceptual design (the first step of design)
- Consider interaction types and interface types to prompt creativity
- Storyboards can be generated from scenarios
- Card-based prototypes can be generated from use cases



Down-sides to informal design

- Clients

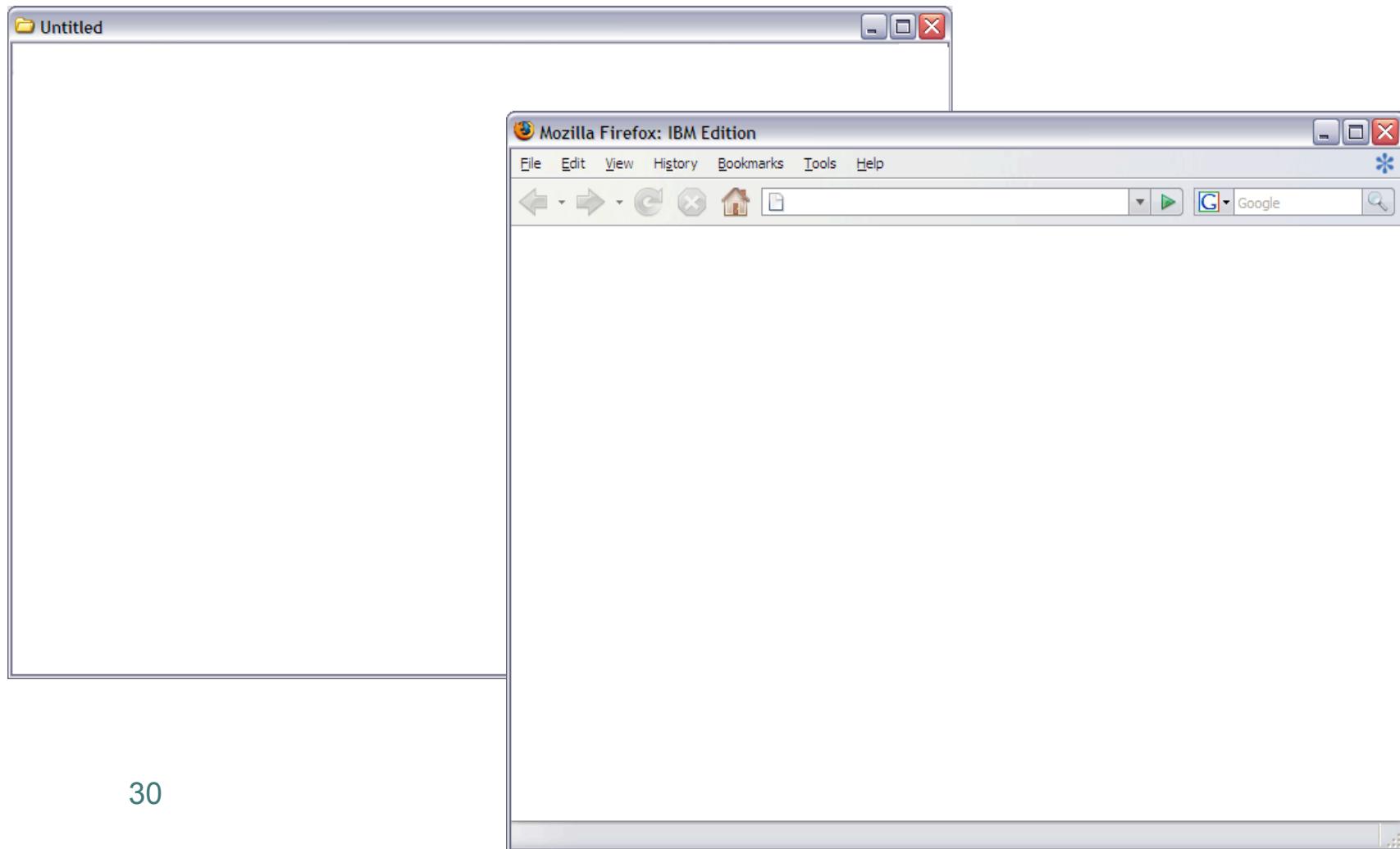
- Often see the fidelity of the interface as an indication of development effort
- Often hard to involve them as subjects
- Talk to them early and often!
- Explain the process and set expectations up front!



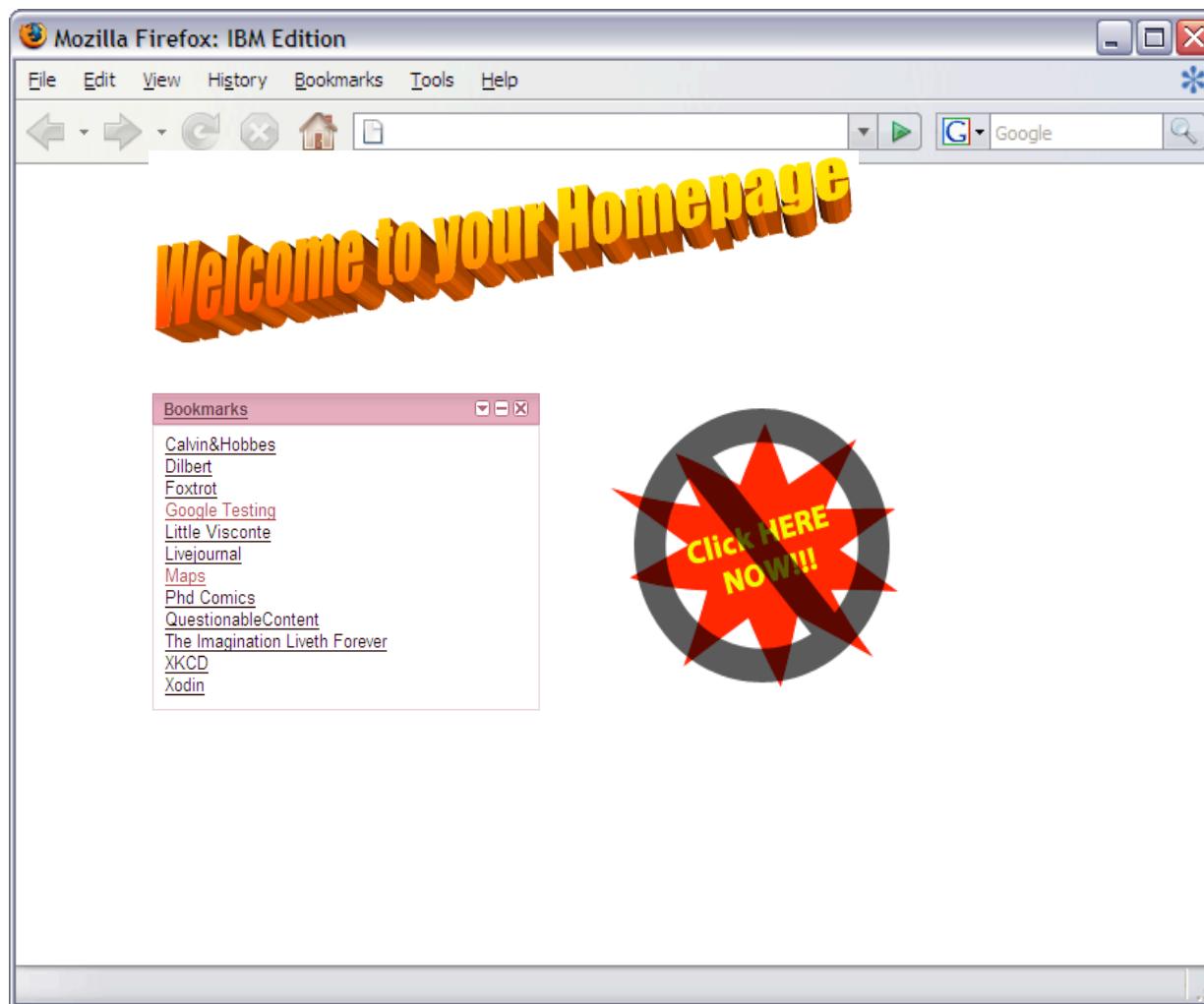
Screenshotting



Start with a Blank Page



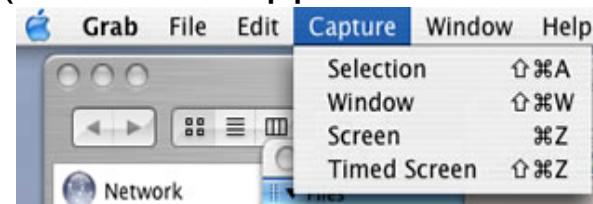
Use a drawing program to insert items you want





Useful Tricks

- Print Screen and then cut out elements you like
 - Windows: Alt-PrtScn captures the active window into the clipboard
 - Mac Control - Command (Apple)-Shift-4 captures the “grabbed” area into the clipboard
 - Mac OSX Grab Utility
 - Open Grab (located in Applications/Utility).



- Use a simple graphics editor like Paint to stitch the images together
 - Copy & Paste many elements for different looks/feels



Web-Based

- Provides the illusion of interactivity
- Can employ screenshots to show parts of your webpage
- Can be used as just a “click-through” of screenshots
- Good if you are considering adding features to an existing website



Resources

- <http://www.krisjordan.com/2008/09/07/10-minute-mock-prototyping-tips-for-powerpoint/>
- <http://www.istartedsomething.com/20071018/powerpoint-prototype-toolkit-01/>