Users

C.W. Johnson,

University of Glasgow,
Glasgow, G12 8QQ.
Scotland.
johnson@dcs.gla.ac.uk,
http://www.dcs.gla.ac.uk/~johnson

September 2006

CS-1Q HCI

What is HCI?

- 1940's:

\- Ergonomics focuses on study of work.

- 1950's & 1960's:

\- focus on Human Factors of machine interaction.

- 1980's & 1990's:

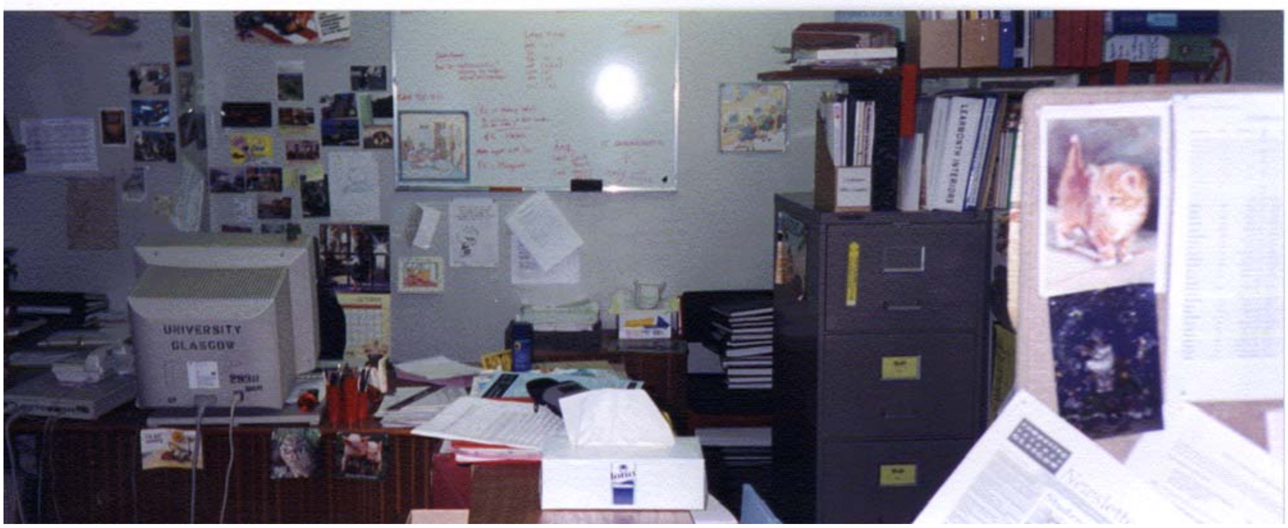
\- focus on Human Computer Interaction.

- 2000 on:

\- back to human factors of machine interaction;
\- *ubiquitous* computers embeded in other devices.

Users

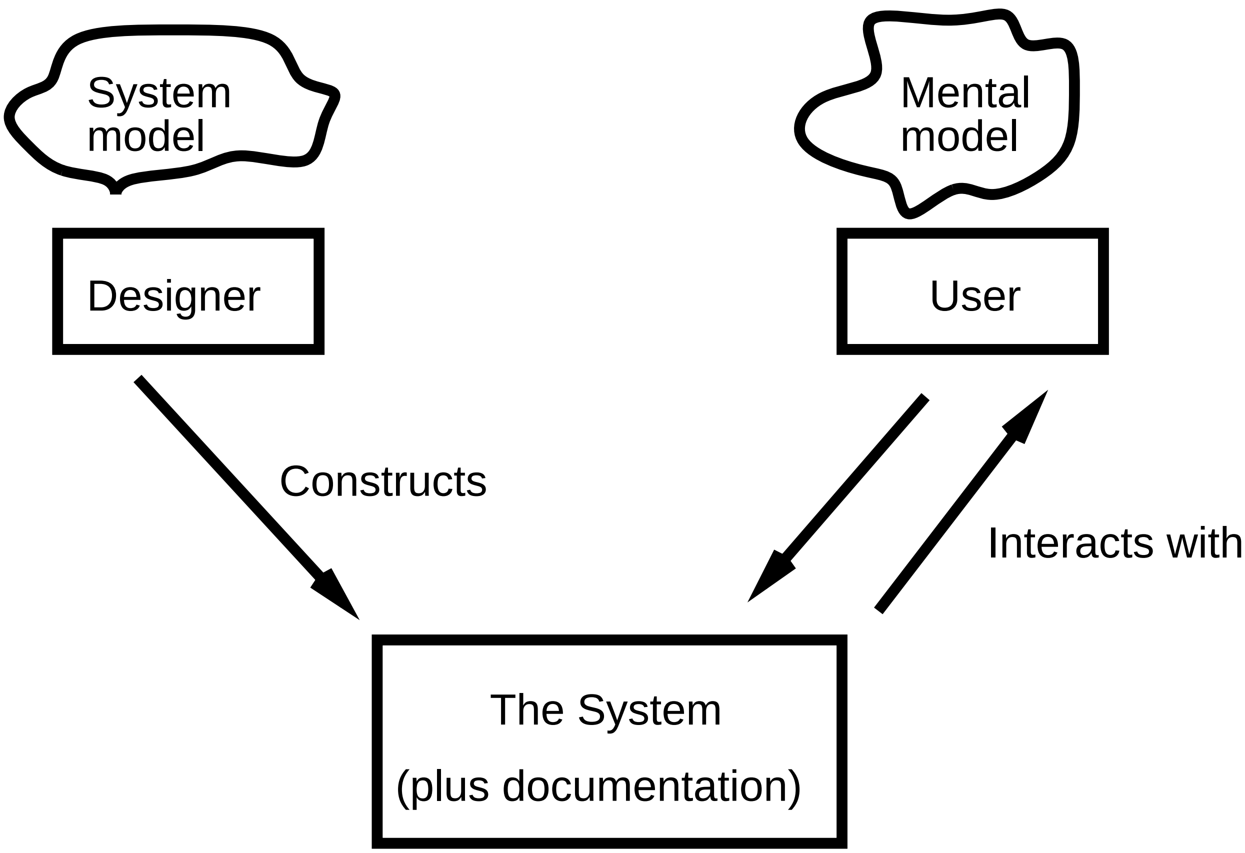
- What is a 'typical' computer user?



- What would make them happy & productive?

Users

- What do you think of when you use a computer?



- Users develop *mental models* of the system.
- This is different from the designers model.
- It also differs from their model of your model.

Perception

- We detect signals in our environment.

- Infra-red eye tracking.

Perception

- What we detect is affected by:

- absolute and relative signal strength;
- anticipation of the signal;
- signal to noise ratio.

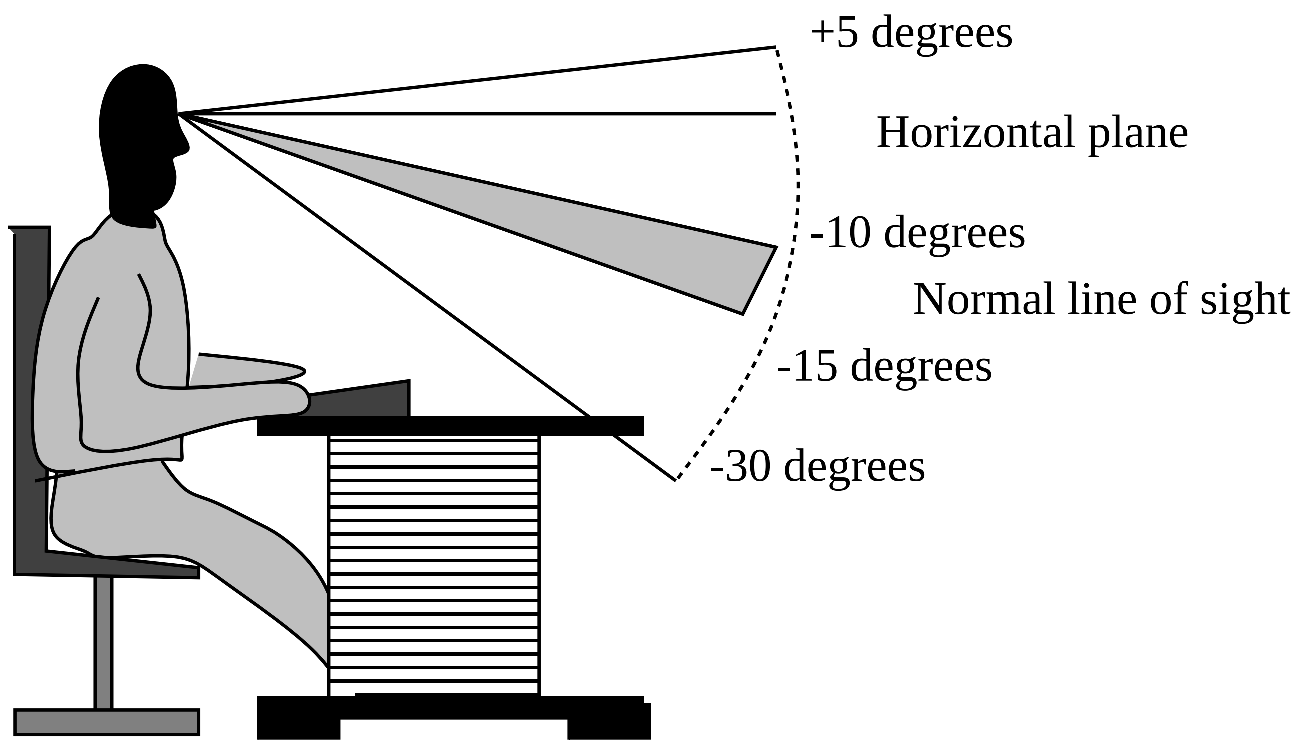
- Links between perception and cognition:

- users often perceive signals;
- but fail to interpret them;
- or understand their *semantics*.

- Anticipation affects perception.

Physiology

- Physical attributes affect ability to use devices.



- RSI, Muscoskeletal injuries etc.

Physiology

- Designers also make assumptions about physiology:

- average person is not 1.8 meters tall
- average person does not weigh 100kg
- nor is their finger span 21cm.

- If we designed for the average who would we exclude?

Cognition

- How do users think about the systems they use?

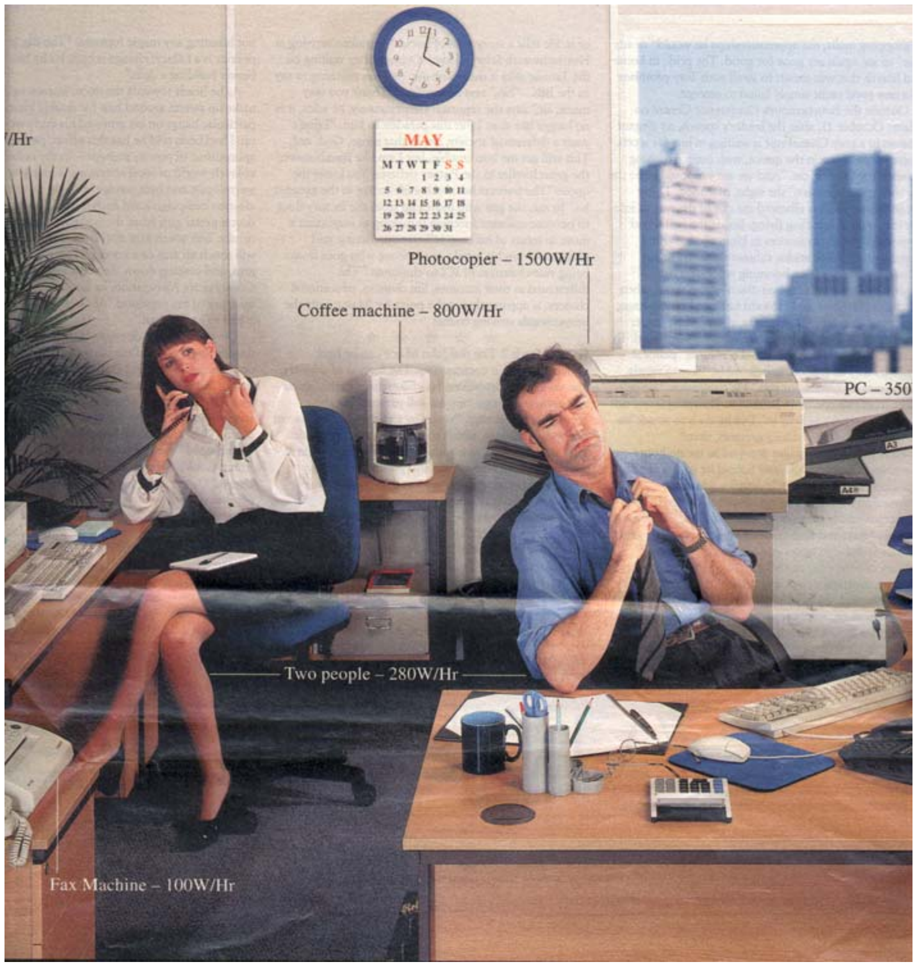
- Different emphasis on *short and long term memory*.

- Experts work at a skilled level;

- if something goes wrong we might use some rules;
- if all else fails try general knowledge.

Environment

- External impact on cognition, perception and physiology.



- What can you control?

- heat, noise, vibration, colleagues?
- don't work next to a printer...

Culture

- Globalisation or fragmentation?

- Japanese version of Microsoft Word:

- took 2-3 years to develop;
- team of local programmers;
- Tokyo usability lab.

- Cultural diversity within a nation (US, Scotland?).

Age

- By 2008, will be more pensioners than schoolchildren.

- *Chronological ageing*:

- natural rate at which we lose our faculties;
- may not be able to see so well etc.

- *Functional ageing:*

- rate beyond the chronological ageing;
- can be associated with workplace injuries.

Gender

- Women have better spatial awareness?

- Gender related terminology?

- ‘kill’ a process or ‘abort’ a print job.

- Men may be better at interrupt handling?

- Thematic vagabonding and encysting.

Disability

- Key concept of *accessibility*.

- Screen readers and braille displays.

- Word prediction systems.

- Scripts based communication.

- Many technologies transfer into the mass market.

Summary

- Users are not designers.

- Physiology, Perception, Cognition.

- Environment, Culture, Age, Gender.

Further Reading

"In the light of the scientific evidence, pregnant women do not need to stop work with VDUs. However, to avoid problems caused by stress or anxiety, women who are working with VDUs should be given the opportunity to discuss their concerns with someone adequately informed of current authoritative scientific information and advice." (Health and Safety Executive)

- Shneiderman on:

- human diversity, pp 18-27;
- individual and society 585-600.

Computers

CS-1Q HCI

Introduction

- Logical and physical I/O devices.

- Dialogue styles:

- text; forms; menus; graphics.

- WYSIWYG, Metaphor and Direct Manipulation.

- Breakdown.

Input Devices

- Several physical devices implement a logical device.

Acknowledgement: Microsoft and Ericsson

- Text input implemented by:

– keyboards, speech recognition, handwriting, gloves.

Input Devices

- Pointing implemented by:

– mice, touch-screens, tracker balls, eye-tracking.

- Cameras also used to provide input:

– automatic facial recognition using feature extraction.

- People are themselves becoming input devices:

– *context aware* systems track the user's location.

Output Devices

- Printed output:

- laser printers; ink-jets; impact and thermal printers.

Acknowledgement: Panasonic and NEC

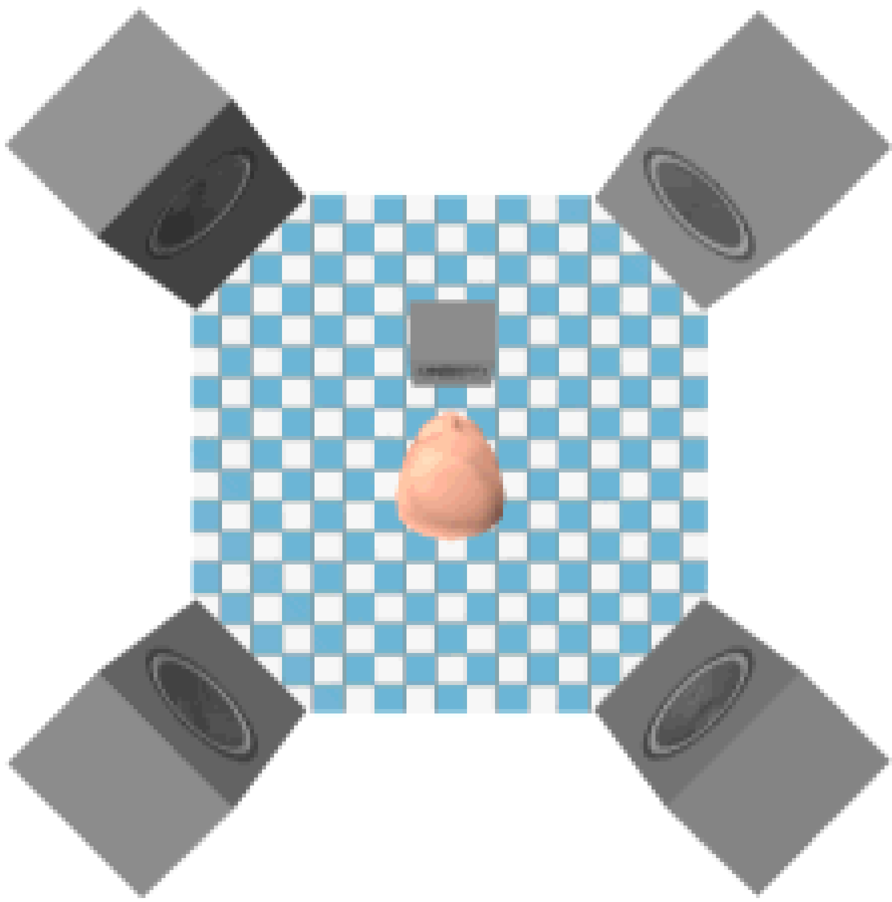
- Graphical output:

- CRT or LCD displays; projectors.

Output Devices

- Audio output:

- stereo/surround speakers; headphones.



- Potential problems:

- earcons or auditory icons?
- pitch/timbre, relationship to music?

Virtual Reality

- Displays: helmet mounted or projection.

- Input devices: head and body tracking.

Acknowledgement: www.vrealities.com

- Many usability issues:

- most systems lack force feedback;
- stresses on body holding a 3D position;
- parallax problems and nausea.

Virtual Reality

- Force feedback or *haptics*:

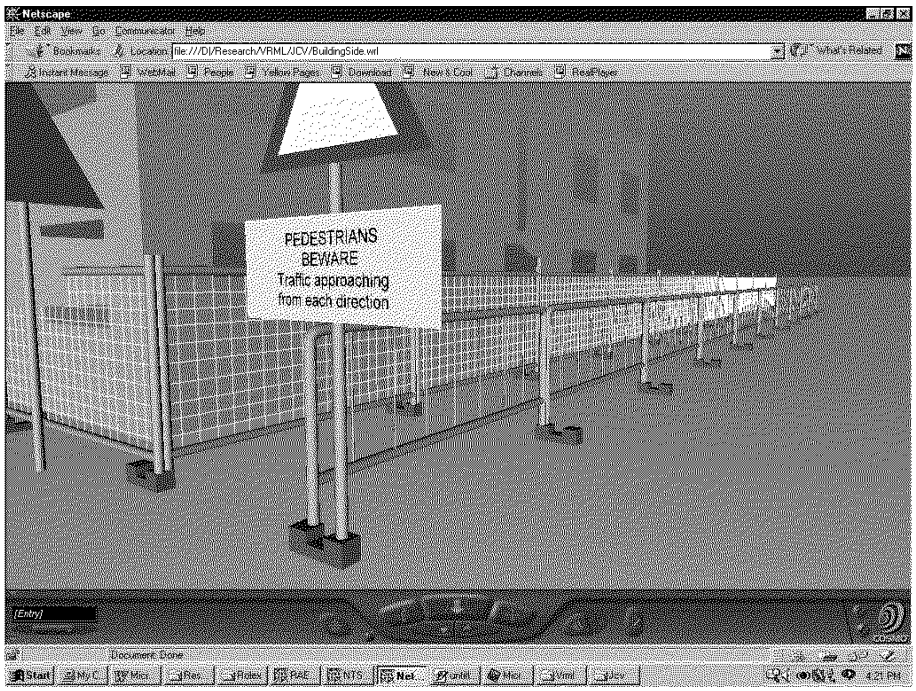
- joysticks; motor-driven feedback; The Bed;
- some of these are input and output devices.

- Financial expense, relaibility etc.

Desktop Virtual Reality

- Technologies:

- QuicktimeVR;
- Virtual Reality Markup Language;
- DirectX and Spatial Audio.



- Many usability issues:

- 3D spatial positioning with 2D devices;
- navigation and way-finding is hard;
- animation of environments requires scripts;
- VRML 'face plant'.

Dialogue Styles

- *Dialogue styles*:

– manner in which information is presented and received.

- The most common are:

– Text (embedded systems, phones);
– Forms (*B2B* systems, call centres);
– Menus (desktop systems, some web interfaces);
– Graphics (desktop, palmtops, the web).

- Most interfaces now mix all of these styles;

– but proportions vary depending on users.

Text

- *Fonts*:

- describe the shape of a character;
- serif vs sans-serif.

- *Point size*:

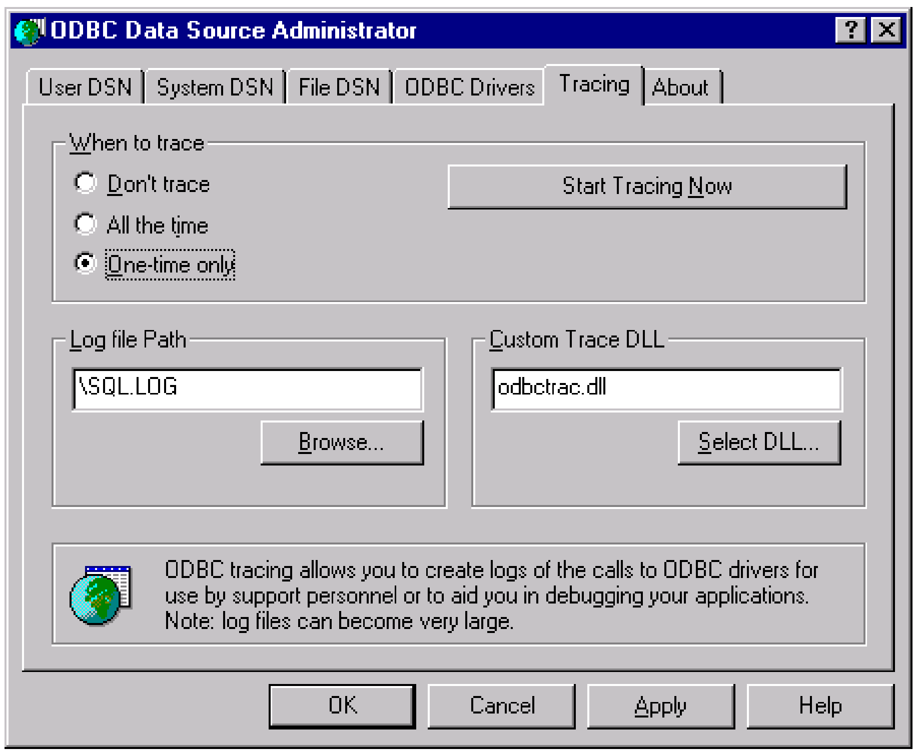
- determines size of a character;
- 1 point is a 72nd of an inch.

- It's the RELATIVE not the absolute size.

Forms

- More advanced forms:

– automatic parsing, context-sensitive.

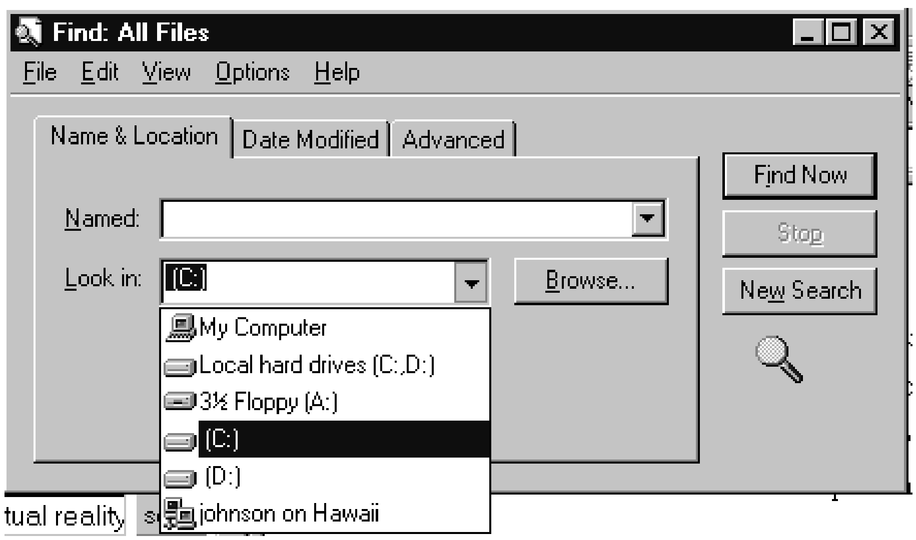


- Must provide title of the form.

Forms

- Navigation:

– tab to move between fields;
– ENTER to submit the form.

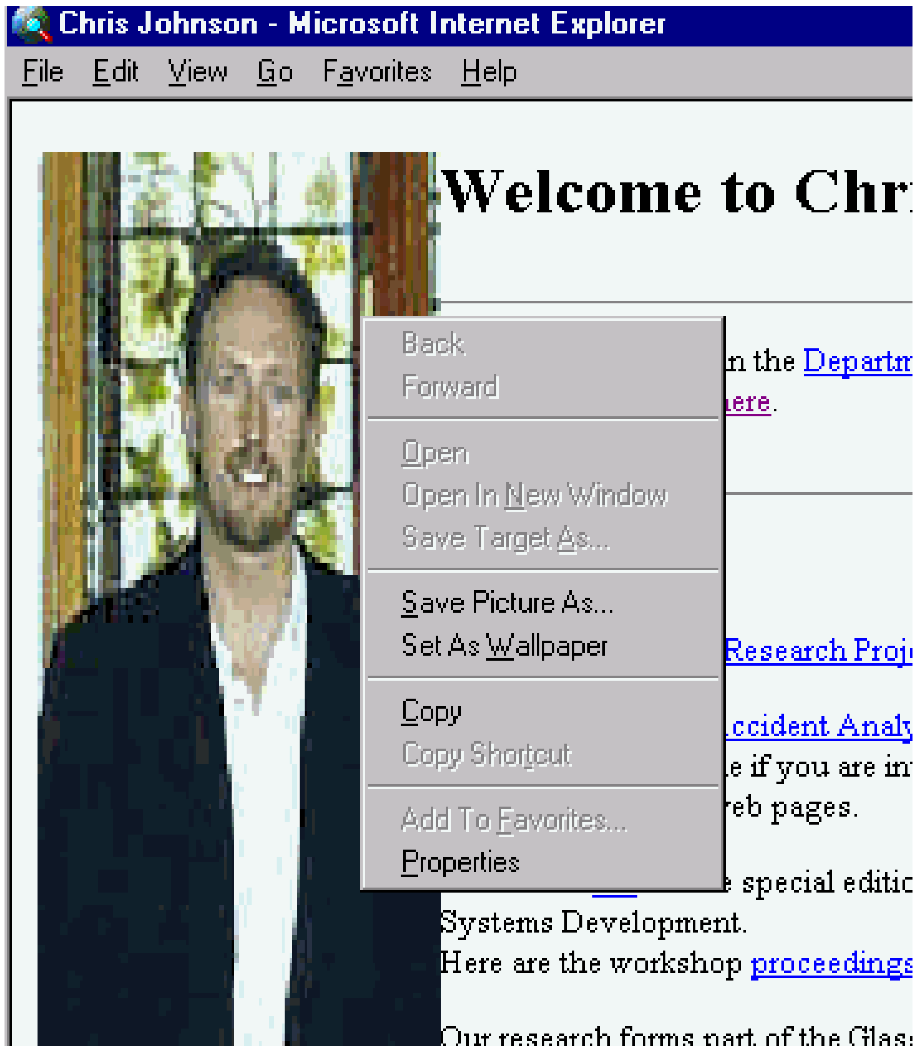


- Must provide a cancel option.

Menus

- Many different types:

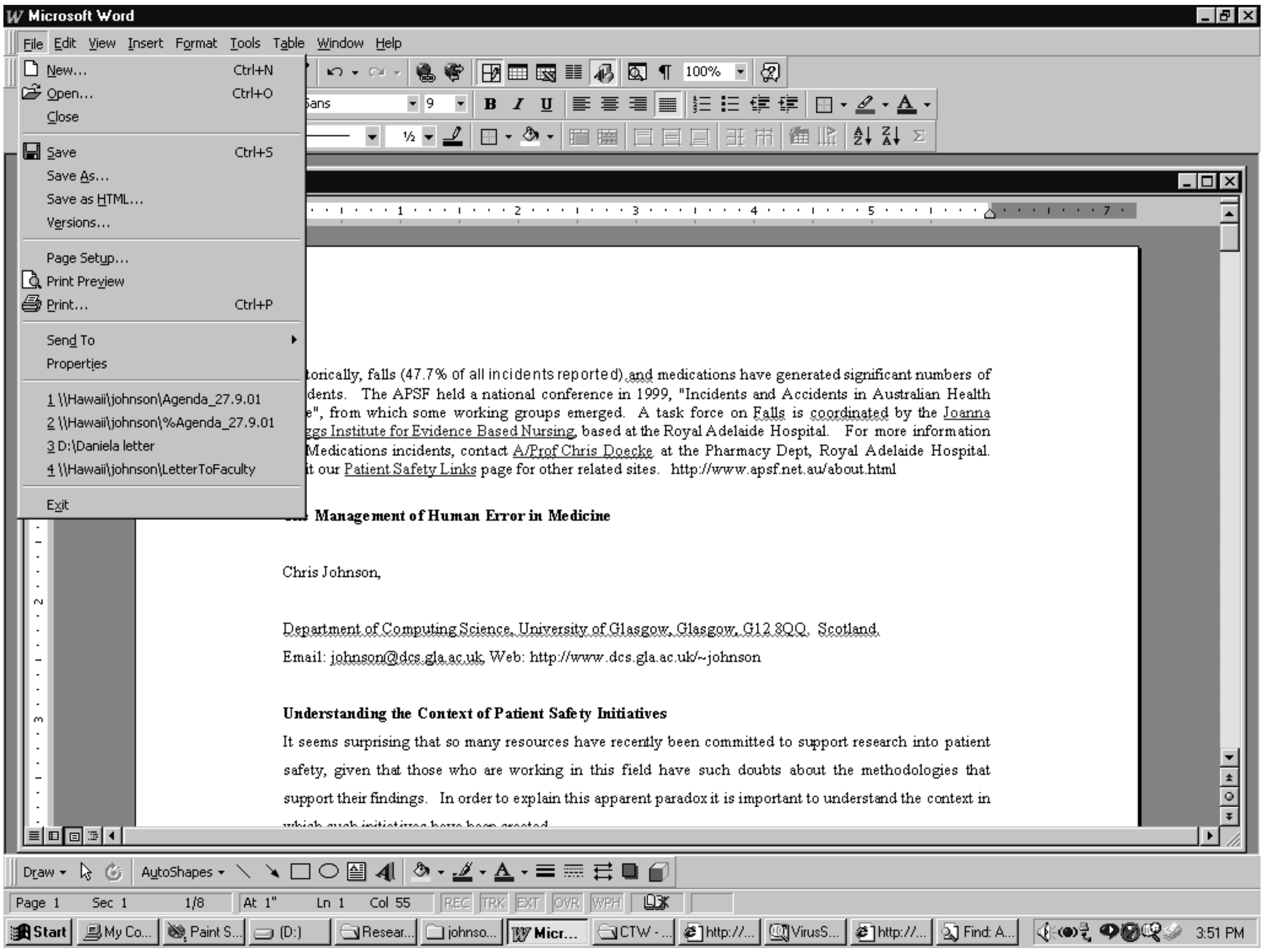
– hierarchical; pull down; tear-off; pop-up.



- The *magic number* is 7 + or - 2.

Menus

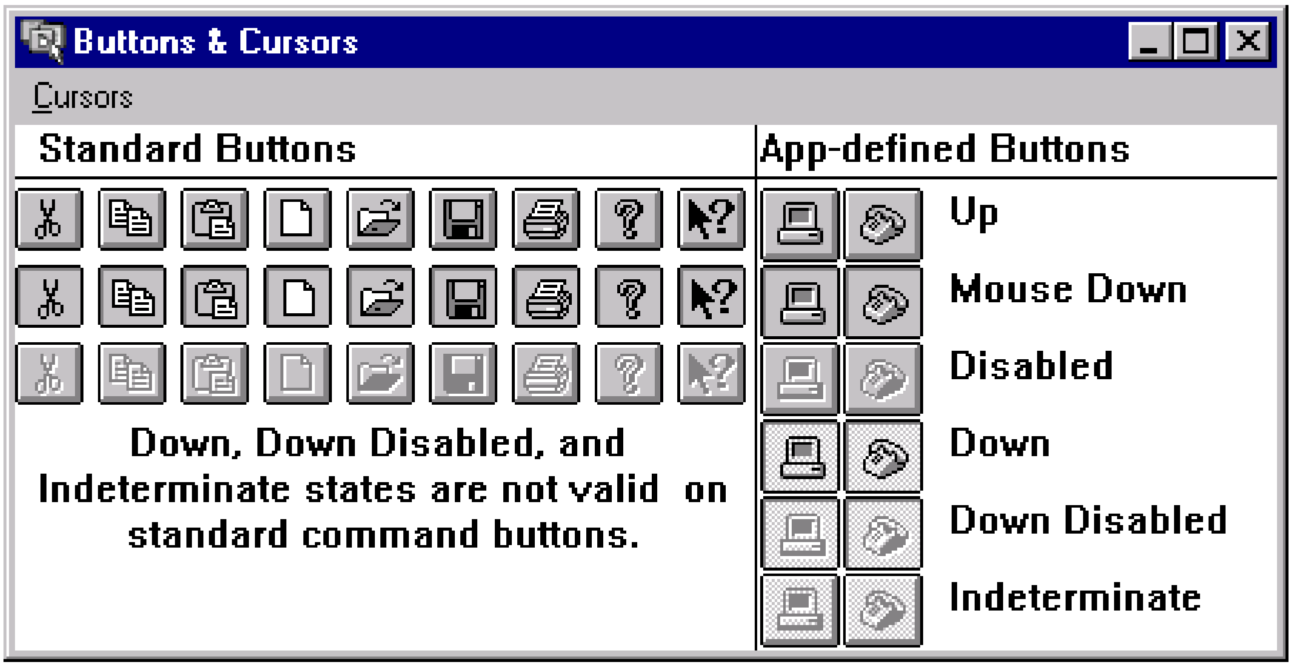
- Low consequence options at the top.



- Dynamic menus reconfigured by item frequency.

Graphics

- Icons help to promote product identity.



- What is the *semantics* of the image?

WYSIWYG

- What you see is what you get.

```
\slideitem{What you see is not what you get.}
```

- Few systems print exactly what you see on the screen.

- Fonts on the printer differ from those on screen.

- Can be deeply frustrating...

Metaphor

- Computers rely on binary signals, people don't.

- Provide an abstract view of underlying complexity.

- Show a desktop with folders and a trash-can.

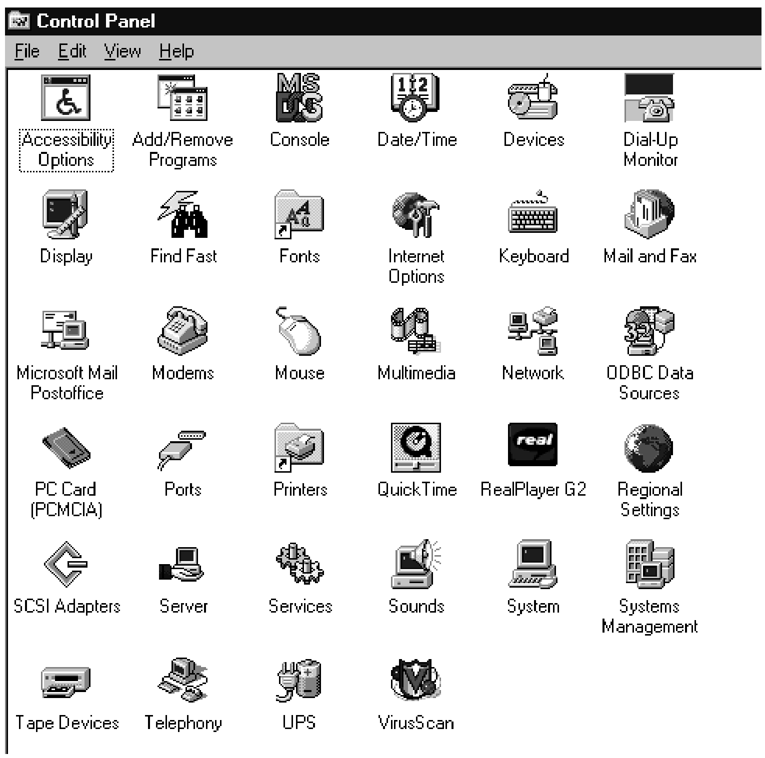
- Do not show complex path information:

`D:\ >Users\profiles\Johnson`

Direct Manipulation

- Good metaphors:

- support skill transfer;
- good metaphors reduce training requirements;
- good metaphors encourage exploration.



- But *direct manipulation* systems must support:

– rapid, incremental and reversible actions.

- What happens when the metaphor breaks down?

Summary

- Logical and physical I/O devices.

- Dialogue styles:

- text; forms; menus; graphics.

- WYSIWYG, Metaphor and Direct Manipulation.

- Breakdown.

Further Reading

- Shneiderman on:

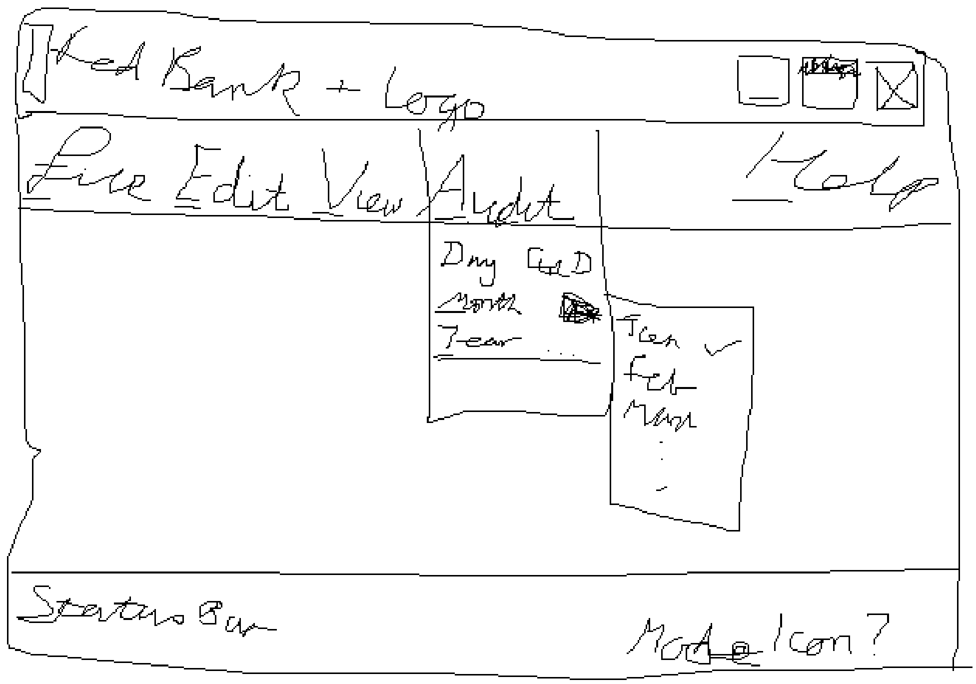
- interaction devices pp305-343;
- direct manipulation pp 185-228.

- Lots more, read it for open assessment!

Elicitation

Introduction

- Participatory and User Centred Design



- Interviews, Questionnaires, Focus Groups.

- Task Analysis.

Recap

- Lecture 1 on mental models.

- User's mental model of the system:

- everything they learned from using it;
- typically, no idea of the implementation.

- Designer's mental model of the system:

- can be ignorant of the application domain;
- but does understand implementation.

- Can designers ever understand users?

What does the Company Want?

- Managers have clear ideas:

- 'new system will reduce staffing';
- 'staff will perform task X in N seconds....'

- Managers have no idea:

- may not understand everyday problems of their staff;
- may urge gratuitous use of particular technologies;
- may disagree amongst themselves.

What do the Users Want?

- Users have little idea of what can be built:

- the easy things can be very hard to imagine;
- the computationally intractable can seem easy.

Acknowledgement: BBC

- Users often find it hard to say what they do:

- over time people build up *expertise*;
- skills become unconscious;
- long-term not short-term memory.

User-Centred Design

- Solution involve users in the development process:

- use of rapid incremental prototyping;
- this supports formative evaluation;
- user testing before making design decisions.

- Problems:

- if tests fail is it the design or unrepresentative users?
- if a test fails how do you generate new solutions?

Participatory Design

- Involve users in the development process:

- more direct involvement than in UCD.

- Users present in design meetings:

- how representative is the user involved?
- often representatives resemble developers!

- Problems:

- very difficult to manage:
- can feel intimidated by development team;
- can feel superior to the development team.

Market Surveys

- Assess users' lifestyles and aspirations:

- Early adopters? Second-wave? Conservatives?


Acknowledgement: BBC

- What competitor systems are successful now?

- Photo diaries and technology trials.

Interviews

- Unscripted sets of questions:

- follow the mood of the interview but...
- can be (mis)led by the interviewee.

- Prescripted sets of questions:

- ask everyone the same things and compare answers;
- might not ask the one really crucial question?

- How to record the responses:

- notes (cheap but may detract from interview);
- audio taping (unintrusive but costly to transcribe);
- video (facial expressions but costly to transcribe).

Interviews

- Questions can give away information.

- Implied criticism:

- ‘Most of the staff here have never used an IX3205B?’

- Poor preparation:

- Interviewer: ‘You work in the paint prepartion shop?’
- Interviewee: ‘No, that was closed last year...’

- Arrogance:

- ‘We’re planning a servlet-based extension to the...’

- Patronising introductions:

‘Now please relax, there really is nothing to be worried about...’

Questionnaires

- Low response rates especially from certain groups?

```
Do you like the existing system?  Yes  No
```

- Multiple choice:

- users may mindlessly tick certain responses;
- may ask trick questions (double negatives etc).

```
What is good about the existing system?
```

- Open ended:

- users may not have time/interest to write much;
- how typical are the users who write lots and lots?

Questionnaires

- Asking more subtle questions...

```
It is easy to make mistakes?  Disagree 1 2 3 Agree


                        terrible      wonderful
Were the lectures helpful? 1  2  3  4  5  6     N/A
```

- Scalar responses (Likert scales):

- can be difficult for users to choose between values.

```
Rank the following from 1 to 3
The most important thing about this lecture is:
----that it finishes in an hour       _____
----that it helps me to pass the exam _____
----that it never mentions Java       _____
```

- Ranked responses:

- what if there are equal preferences?

Focus Groups

- Questionnaires open to bias and influence.

- Peer groups complete questionnaires together?

- May miss shared attitudes and beliefs.

- How do you interpret responses?

- partial answers may need more detail;
- responses may contradict views of managers etc.

- Focus groups to follow-up questionnaires.

Focus Groups

- You act as facilitator for a discussion:

– introduce a topic and let them talk about it;
– prompt and direct towards key topics.

- Record and account for different perspectives.

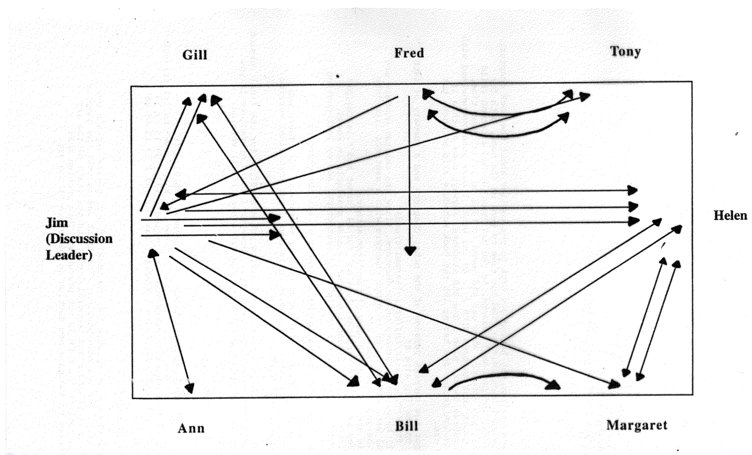
- Problems:

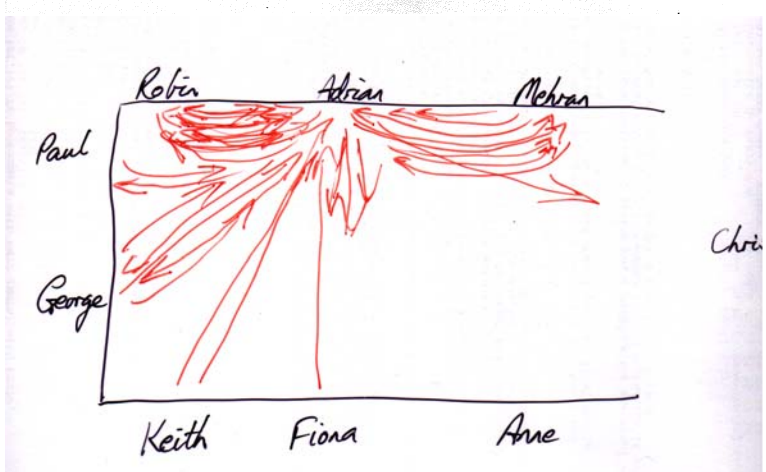
– can be ‘hi-jacked’ by opinionated participants;
– care must be taken with group dynamics.

Participation Grids

- Who contributed to a focus group?

– minor manager talks but project sponsor is silent;
– might come away believing minor view?





- Participation grids:

- draw an arrow when A talks to B;
- repeat for each major topic.

Requirements Documents

- What do you do with the information?

- Construct a requirements document.

- Describes *what* must be done:

- provide automated ordering facilities for all staff;
- staff should complete first order with 1 day training.

- Does not describe *how* to do it:

- use a Pentium III running NT, written in Java...

- Can be based on usage *scenarios*.

Hierarchical Task Analysis

- What do you do with requirements document?

- Spilt a high-level task into sub-tasks.

```
0.  in order to complete an order
      0.1 take customer's product selection
            0.1.1 ask for reference number
            0.1.2 enter reference number in system
      0.2 take customers contact details
            0.2.1 ask for customers post code
            0.2.2 enter postcode into system
            0.2.3 complete any blank fields
      0.3 take customers payment details
```

- Good points:

- builds a conceptual model of users' view;
- can identify knowledge requirements with each step.

- Simple plans but what if things go wrong?

Current Problems: Plans and Situated Actions

- Lucy Suchman criticises much of this.

- Plans evolve within a complex working environment.

- People:

- are more opportunistic;
- must adapt to interruptions.

- So designer must:

- consider users' model of the system;
- consider the effect that context has on the model.

Conclusions

- Participatory and User Centred Design

- Interviews, Questionnaires, Focus Groups.

- Task Analysis.

Further Reading

- Shneiderman on:

- design process - pp. 95-117;
- evaluation - pp. 124-150.

- He combines elicitation and evaluation.

Evaluation

CS-1Q HCI

Introduction

- Formative evaluation:

- heuristic evaluation, cooperative evaluation.

- Summative evaluation:

- lab-based techniques, diaries, ethnomethodology.

- Current problems:

- mobile systems; fun and games.

Formative Evaluation

- Helps to form design decisions.

- Should we use a menu here or icons?

– build pencil and paper prototypes of both;

– do some user testing, throw one away.

- Will results from prototypes accurately predict

- performance with the final computer-based system?

Heuristic Evaluation

- Check to see design meets guidelines.

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

Shneiderman's *Designing the User Interface*, Chapter 2, Page 74-75.

- Can be difficult to agree in specific cases.

Co-operative Evaluation

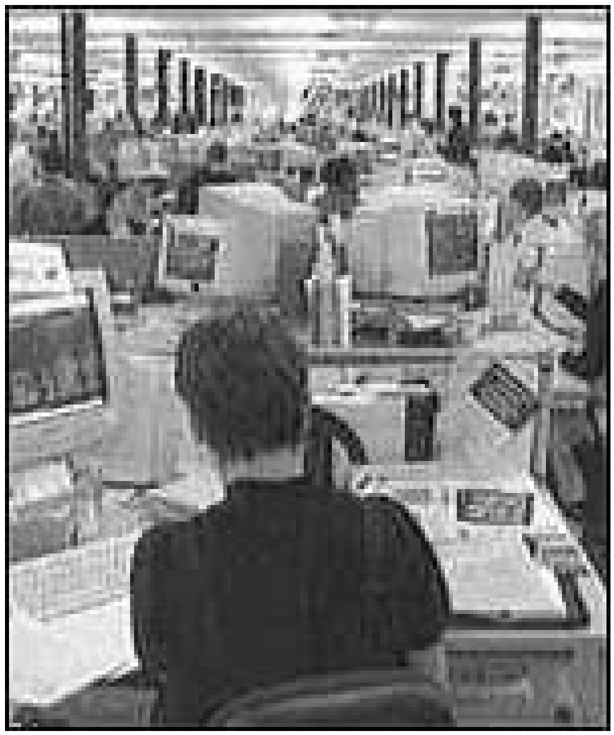
- Involves real users...

- Relatively simple procedure:

- ask users to perform a specified task;
- only intervene to help them if they get stuck;
- if they get stuck this indicated need for redesign;
- get them to 'think aloud' as they use the system.

Co-operative Evaluation

- Involves real users...



Acknowledgement: BBC

- Problems:

- 'thinking aloud' can be unnatural;
- 'thinking aloud' can interrupt thought processes;
- users may perform well by guessing and hitting lucky.

Summative Evaluation

- Takes place at the end of the design process.

- Check to see if interface meets requirements.

- From lecture 3:

- provide automated ordering facilities for all staff;
- staff should complete first order with 1 day training.

- Need some resources left if problems are found?

Lab-Based Experimentation

- Experimental method:

- clearly defined hypothesis;
- appropriate method to support hypothesis;
- results described accurated;
- conclusions connexct results to hypothesis.

Acknowledgement: IBM

- Must constrain the environment:

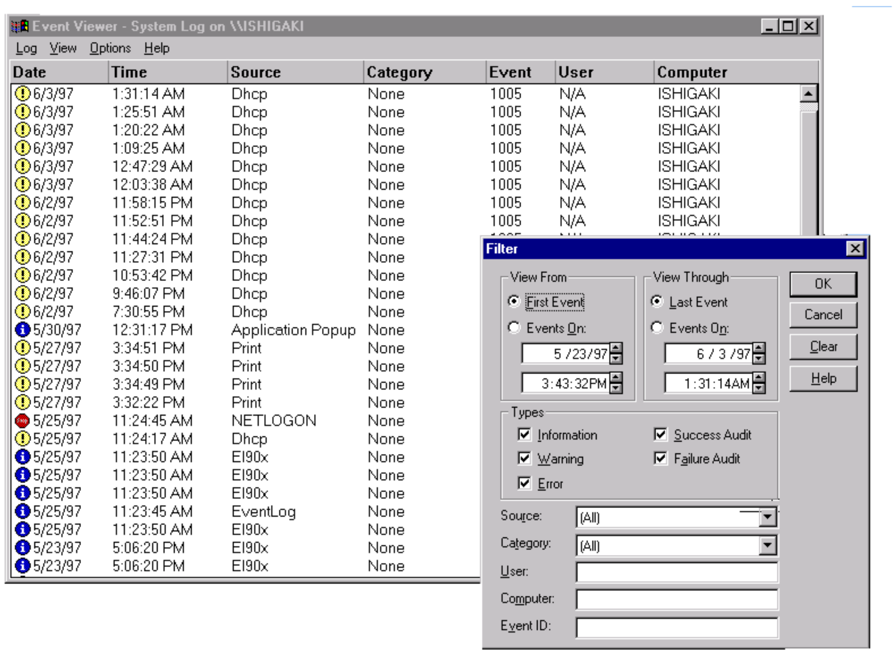
- counter-balancing of tasks for learning effects;
- appropriate sample of potential users;
- exact replication of conditions between tests...

- Is this representative of the real world?

Usage Diaries and Logs

- When system is almost ready to deliver:

- provide few users with advance copy;
- ask them to keep usage diaries of any problems;
- collect results prior to final debugging etc.



- Are users robust enough to cope with early release?

- if so, are they representative of the eventual users?

- Less formal approach via beta-releases.

Problems

- Observation affects the observed.

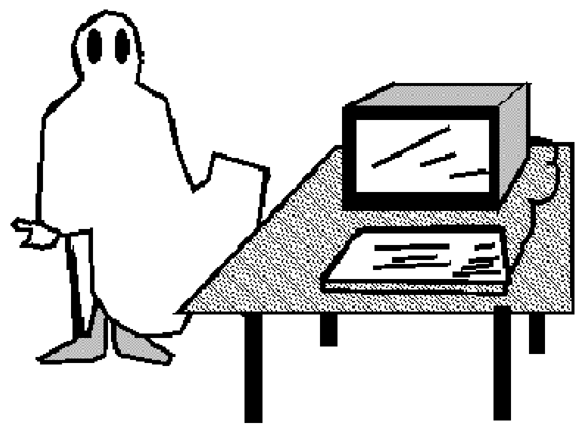
Acknowledgement: BBC

- *Hawthorne Effect*:

- if you know you're being watched you act differently;
- people are more careful under experimental conditions?

Ethnomethodology

- No predetermined hypotheses.



Just Pretend I'm Not Here...

- Observe patterns of use in working environment.
- May later try to interpret meaning of those patterns.
- Highly skilled, little understood, not widely used.
- Ethnography or ethnomethodology?

Problems

- Don't know effect of system until it's used.

Acknowledgement: Nokia

- *Hermeneutics*:

- new systems designed to support existing tasks but
- new systems change existing tasks and create new ones;
- eg on-line shopping changes nature of shopping.

Current Problems: Mobile Evaluation

- Lab-based techniques aren't very mobile.

- Observational techniques involve chasing people.

- Tendency to field trial first:

- high costs and potentially big losses;
- will users buy the service or product?

Current Problems: Fun and Games

- Can people have fun in a lab setting?

- Can you take the time to evaluate?

- Highly subjective issue:

- extreme responses depending on user;
- some will love a game that others hate.

- More general problems with Web:

- can you get people to ‘browse’ under observation?
- is it ethical to log performance routinely?

Conclusion

- Formative evaluation:

- heuristic evaluation, cooperative evaluation.

- Summative evaluation:

- lab-based techniques, diaries, ethnomethodology.

- Current problems:

- mobile systems; fun and games.

Further Reading

- Same as for previous lecture!

- Shneiderman on:

- design process - pp. 95-117;
- evaluation - pp. 124-150.

- He combines elicitation and evaluation.

Distributed Interaction

CS-1Q HCI

Distributed Interaction

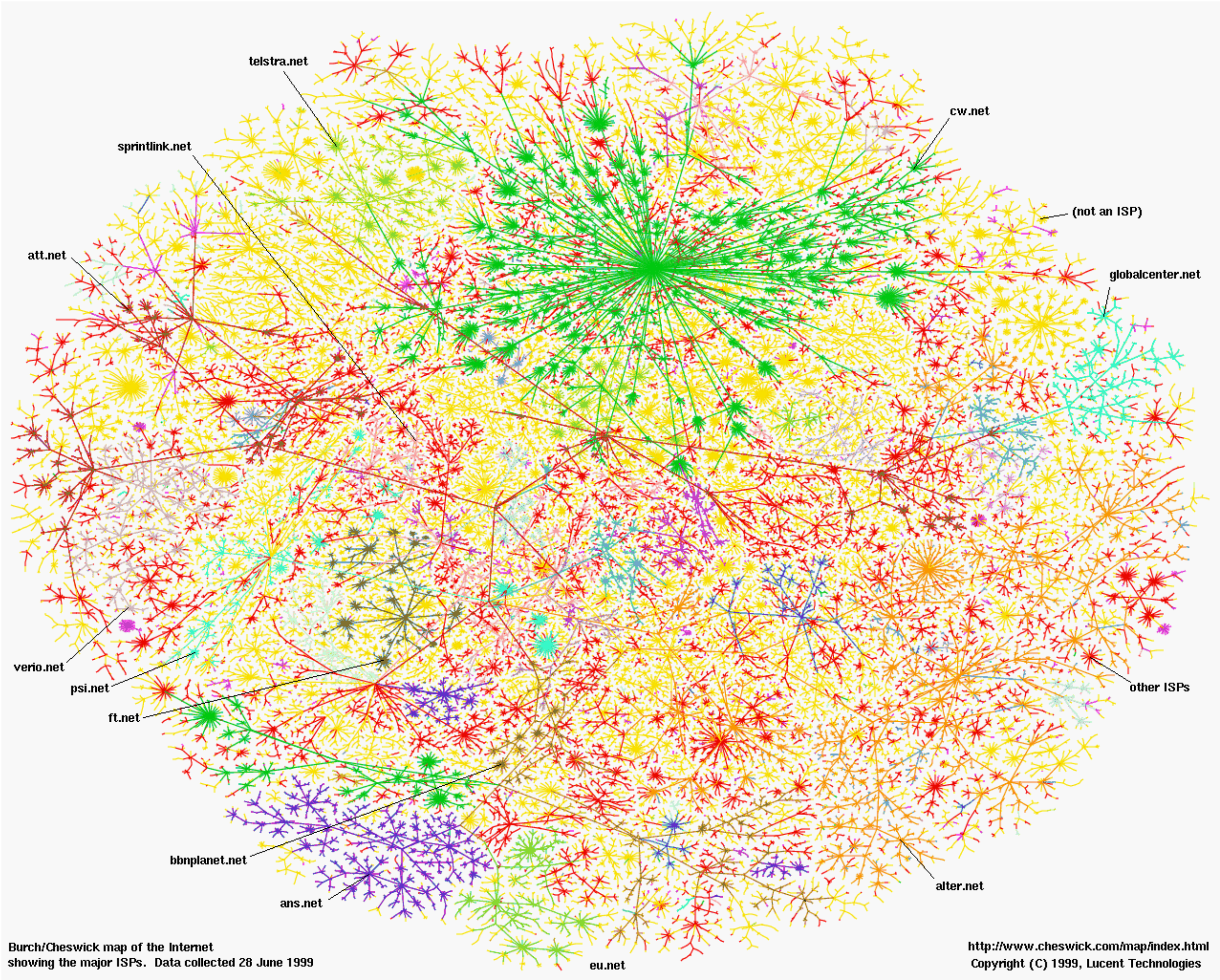
- Infrastructure issues:

- Internet protocols, the Web and beyond.

- Usability issues:

- browsing and task directed search;
- information saturation and redundancy;
- delays, unpredictability and security.

Distributed Interaction



Acknowledgement: http://www.cs.bell-labs.com/who/ches/map/gallery/is
ss.gif

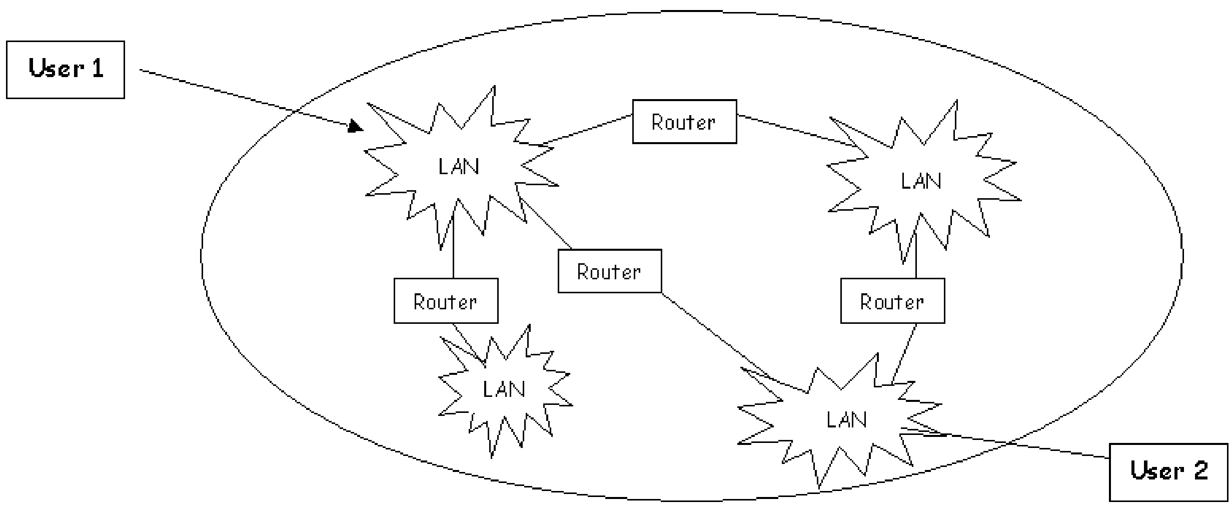
- The Internet:

- 1971 23 hosts; 1980 100 hosts; 1990 1,000,000...

Infrastructure Issues

- What do we need to connect machines:

- an addressing scheme ('where are you out there?');
- transfer protocols ('how much info can I pass you?').



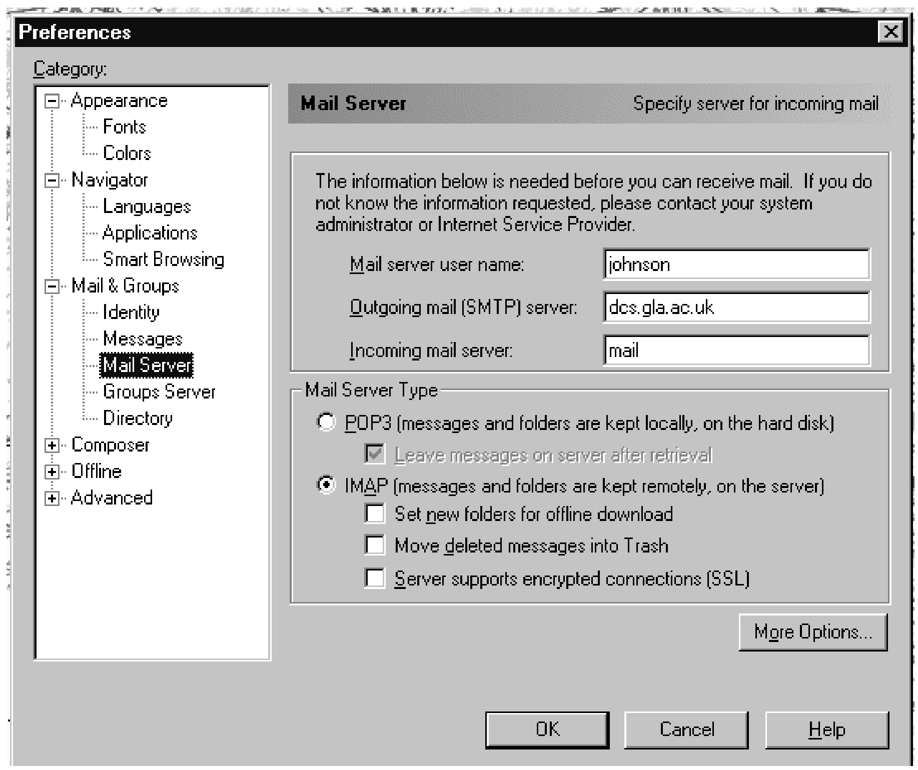
- Internet Protocol underlies it all:

- IP address of sender and recipient plus information;
- routers read the address of packets and forward them on.

Infrastructure Issues

- On top of IP we can build:

- Transmission Control Protocol - reliable connections;
- User Datagram Protocol - unreliable messages.



- On top of TCP/IP we can build:

- ftp - file transfer protocol;
- SMTP - mail transfer protocol;
- NNTP - net news transfer protocol;
- http - hypertext transfer protocol.

Infrastructure Issues

- The Web:

- CERN and Tim Berners-Lee (among others);
- 1993 Nat. Centre for Supercomputer Applications' Mosaic;
- Netscape commercial successor then Microsoft IE.

- Remembers:

- Web is not only form of hypertext system;
- Web is not the same as the Internet...

- Many other ideas and models:

- Ted Nelson's Xanadu transclusion ideas;
- 'The Grid', E-Science - lots of hype...

Usability Issues: Accessibility

- So what does distribution buy users?

- Can access remote resources:

- computational resources;
- information resources;
- human resources (next lecture).

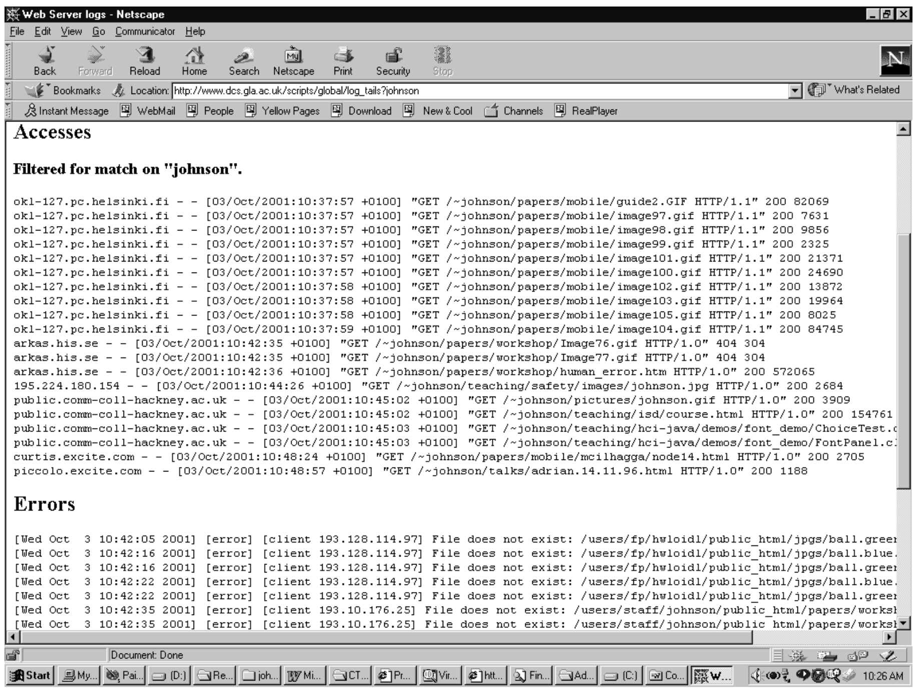
- Two principle modes of activity:

- browsing, undirected, ad hoc and opportunistic;
- direct search, clear task, often time limited.

Usability Issues: Browsing

- As designers, can you attract user?

- avoid scrolling and gratuitous animation;
- banner advertising can be off-putting;
- high download latencies (see later).

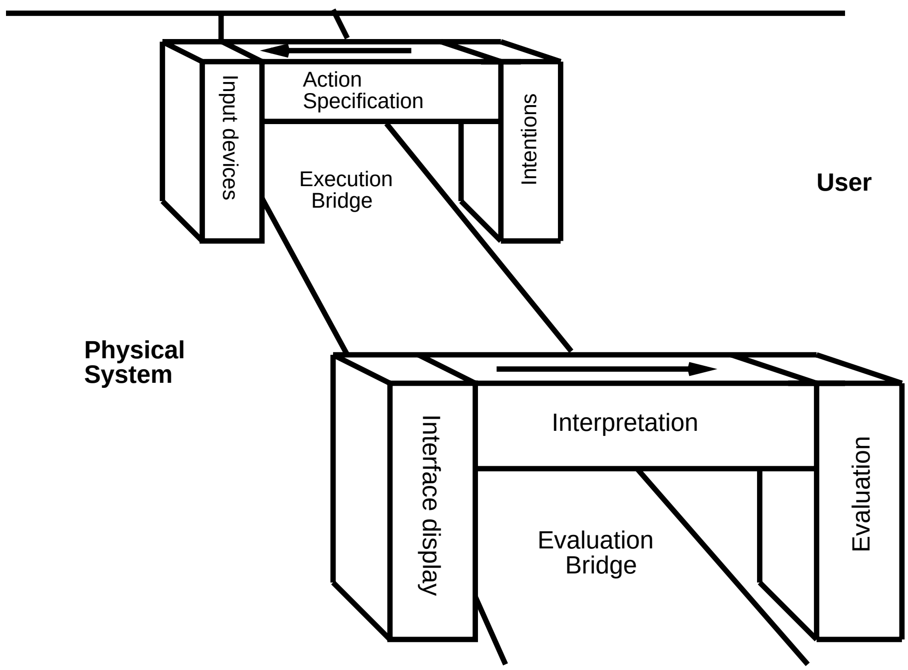


- Dwell time is a key issue:

- examine server logs for abandoned requests;
- may need remote usability testing.

Usability Issues: Directed Search

- Task directed interaction.



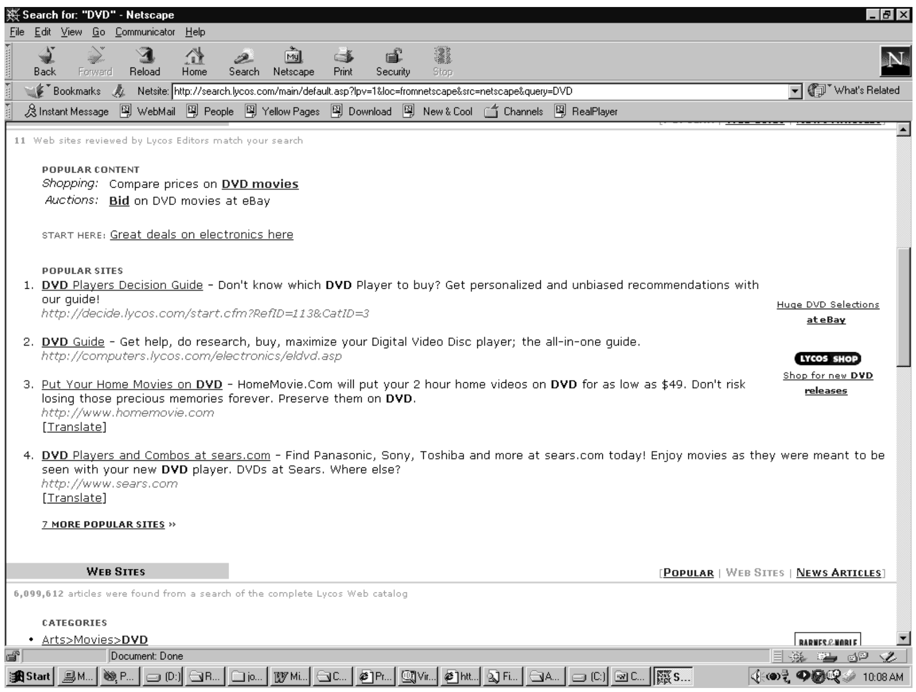
- Information retrieval:

1. Form intention to find information;
2. Translate intention into query language;
3. Evaluate results of search request;
4. If not found, reform intention or query...

Usability Issues: Information Saturation

- May get a huge number of hits;

- few of them may actually be relevant to your task.



- Search on DVD gives 6 million+ hits.

Usability Issues: Information Saturation

- Precision:

$$\frac{number\ of\ relevant\ hits}{total\ number\ of\ documents\ returned}$$

- Recall:

$$\frac{number\ of\ relevant\ hits}{total\ number\ of\ relevant\ documents\ in\ collection}$$

- Relevance feedback:

- user indicates which 'hits' are relevant;
- system uses this to improve next search.

- Remember also the problem of search IN a page.

Usability Issues: Redundancy

- Multiple sources of information:

- if one server fails then there are others;
- camera bought in Glasgow but drivers in Singapore.

- Increase in global competition:

- B2B (business to business) transactions;
- if your site fails I can find another.

- Usability is suddenly very important.

- 2 click techniques (eg Amazon).

Usability Issues: Delays

- Distributed systems speed interaction:

– cacheing holds information close to point of use;
– mirror sites replicate a server close to user.

- So where does a web page come from?

- Always try to minimise file sizes.

- Avoid gratuitous graphics and animations.

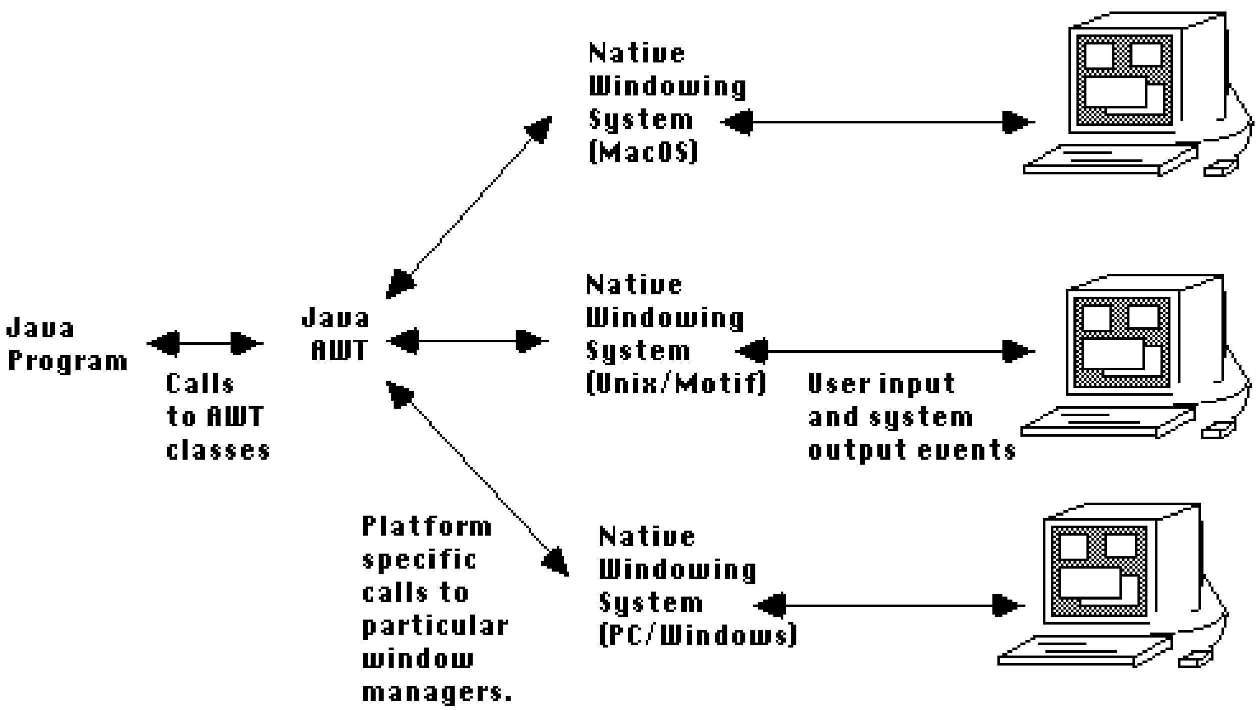
- Give users warning about ‘expensive’ resources.

- Myth of the Infinitely Fast Machine.

Usability Issues: Portability and Java

- Portability:

- don't just move data between machines;
- also move code and applications.

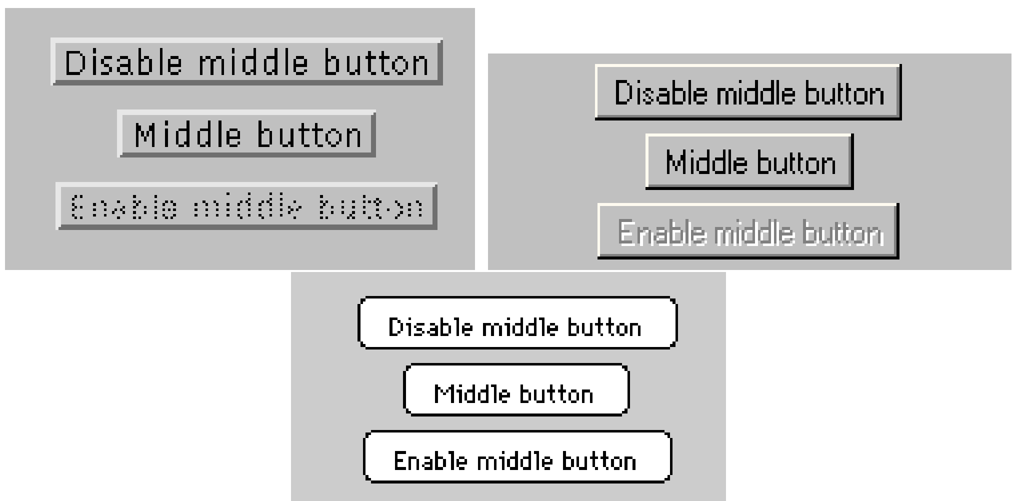


- Local execution of remote code:

- Java bytecode downloaed from web;
- Java virtual machine executes on client;
- security issues??

Usability Issues: Portability and Java

- AWT makes interface look like host machine.



- Swing makes interface look same on all machines.

Usability Issues: Unpredictability

- Summary:

- don't know where your information is coming from;
- don't know where code is being executed.

- Loading on remote servers varies:

- with time of the day (US daylight hours);
- with process profile (lots of computations?);
- with mirror/cacheing support.

- Web performance is very unpredictable.

Usability Issues: Unpredictability

- Solutions to unpredictability.

- Give users an idea of the possible delay:

- indicate file sizes for remote resources.

- Indicate quality of the resource:

- thumb-nail images of videos show production qualities.

- Alternatively, make all delays predictable:

- technically not easy to do but some suggest adding delays;
- everything would take 5 minutes...
- users would develop *coping strategies*;
- slightly wacky - not sure...

Usability Issues: Security

- Users often don't know about this.

- Openly share passwords.

- Use the same password everywhere 8(

- Social issues - do you trust the web?

Final Caveats

- US census 1990:

- total population 230,445,777;
- 198,600,798 only speak English;
- 31,844,979 primarily non-English speaking.

Very Well	Well	Not Well	Not at All
17,862,477	7,310,301	4,826,958	1,845,243

Table 1: 1990 US Census Data for Self-Reported Ability in English

- US census 2000:

- 54,000,000 (51%) of households had 1 or more computers;
- this was an increase of +42% from 1998.

- US census 2000:

- 45,000,000 (42%) use Internet at home;
- it was 26% in 1998 and 18% in 1997.

- Don't get carried away by growth of Internet!

Summary

- Infrastructure issues:
- Internet protocols, the Web and beyond.

- Usability issues.

- Accessibility.

- Browsing and task directed search.

- Information saturation and redundancy.

- Portability, Java and RMI.

- Delays, unpredictability and security.

Further Reading

- A lot of detailed material to cover.

- Shneiderman on:

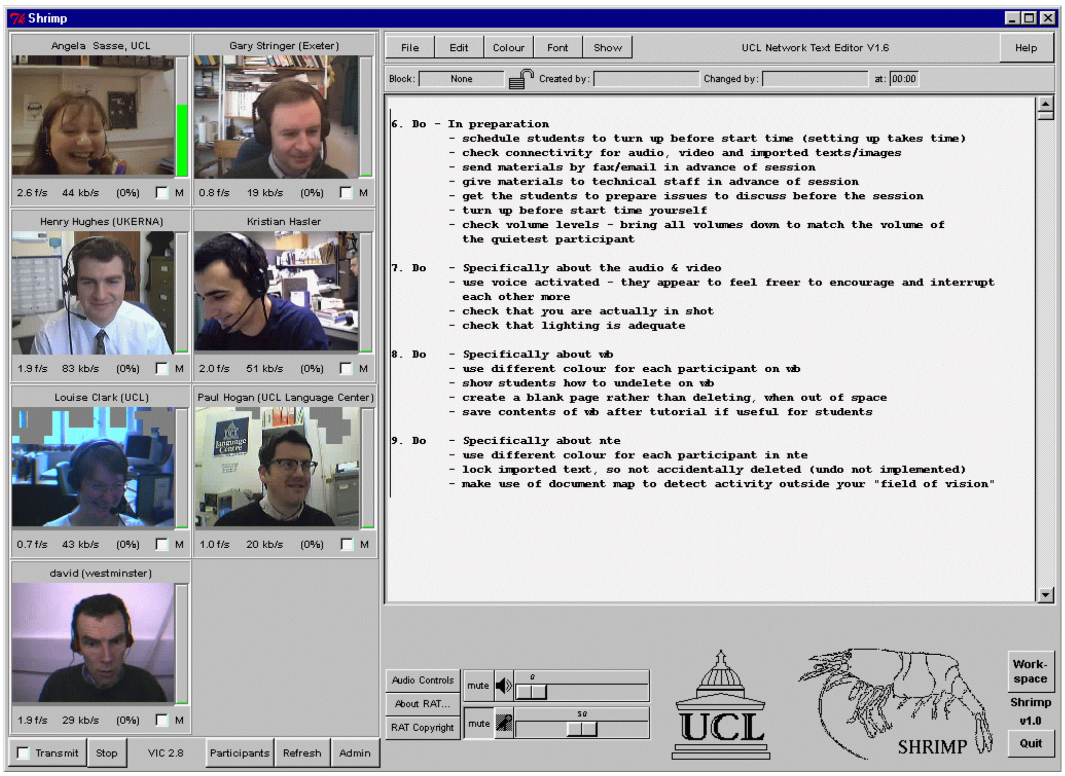
- response time - pp. 351-366;
- the web - pp. 551-579.

Team-Based Interaction

CS-1Q HCI

Team-Based Interaction

- Computer-Supported Cooperative Work.



Acknowledgement: Shrimp project

- Synchronous vs Asynchronous interaction.

- Clark, Brennan and Common Ground.

Problems of Groupwork

- Why is groupwork difficult?

- Distraction:

- individual interrupts colleague's tasks.

- Group coordination failures:

- overhead of coordinating group actions impairs group.

- Group planning and management failures:

- groups create unnecessary tasks.

- Excessive influence of the leader:

- high status leader stifle contrary opinions.

- Group polarisation and groupthink:

- group persuaded by dillusions of its own invulnerability.

Problems of Groupwork

- Computers make things worse.

- Free-riders: “it’s lost in the mail”.

- Distractions and group planning:

- encysting can be a problem.

- Influence of the leader:

- can ‘freeze’ people out of video-conferences.

- Group coordination much worse:

- “can you all hear me?”.

Computer-Supported Cooperative Work (CSCW).

- Face to face: same place, same time.

- Synchronous: different place, same time.

- Asynchronous (1): different place, different time.

- Asynchronous (2): same place, different time.

Face to Face

- Applications:

- share a computer to record design decisions;
- use computer to share visualisations.

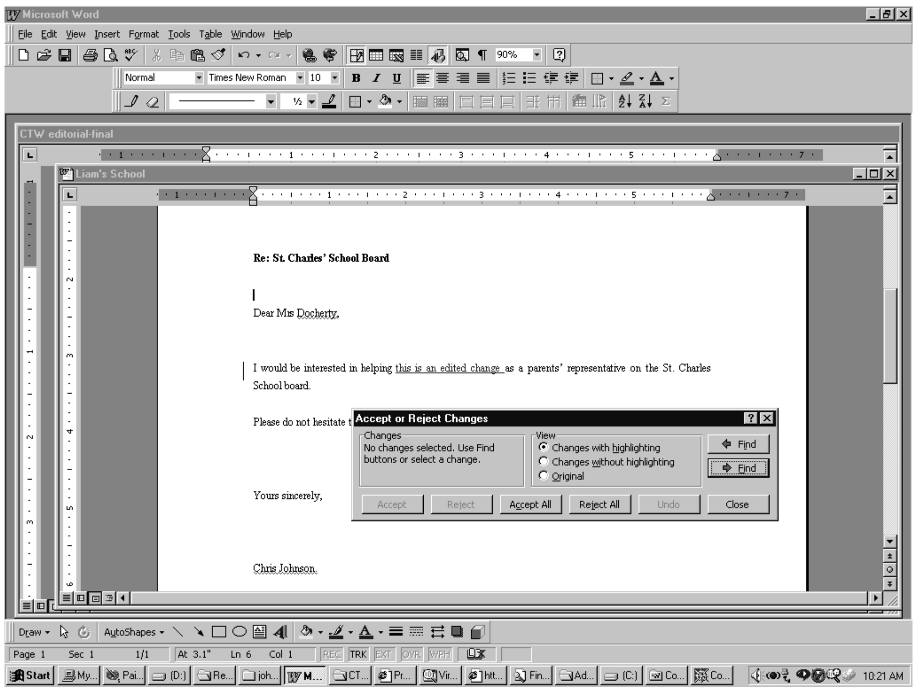
Acknowledgement: Faculty of Science, Loughborough Univ.

- Fighting over access to the input devices?

- One person thinks while the other types 8(

Asynchronous CSCW

- Relatively simple add-on to existing systems.

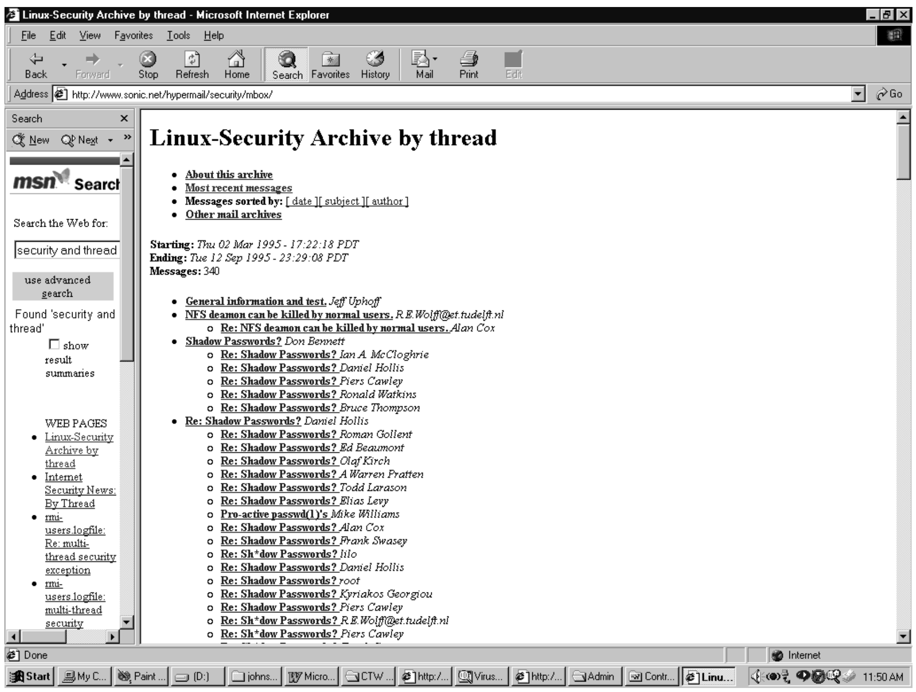


- Need for version control on shared objects.

Asynchronous CSCW

- Applications:

- electronic mail, new and bulletin boards;
- increasingly used to provide 24 hour cover.

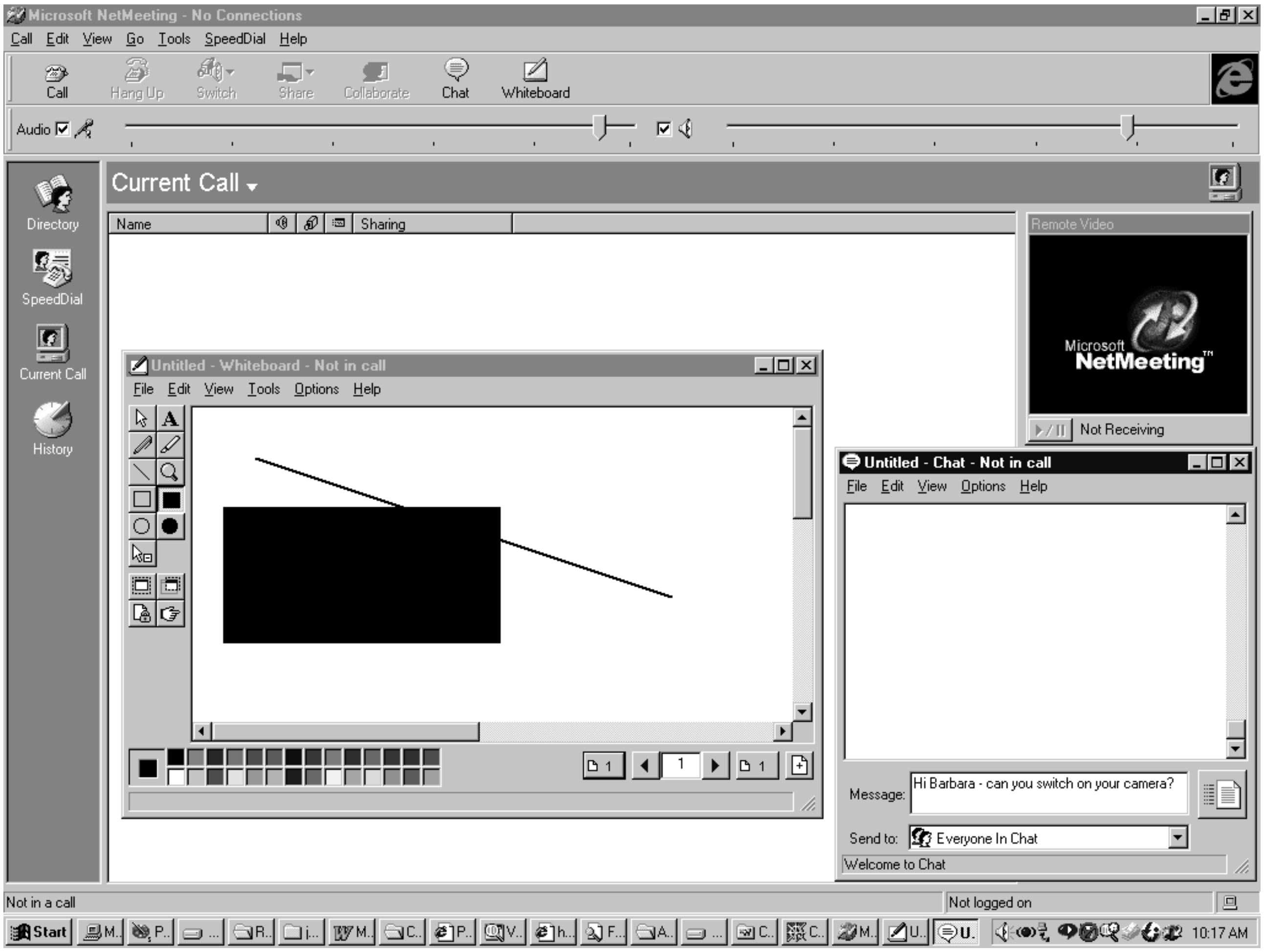


- Need to establish the context of messages:

- remember that messages will arrive out of order;
- threads in postings and use of `Re:` in mail.

Synchronous CSCW

- Integrate different modes of communication.

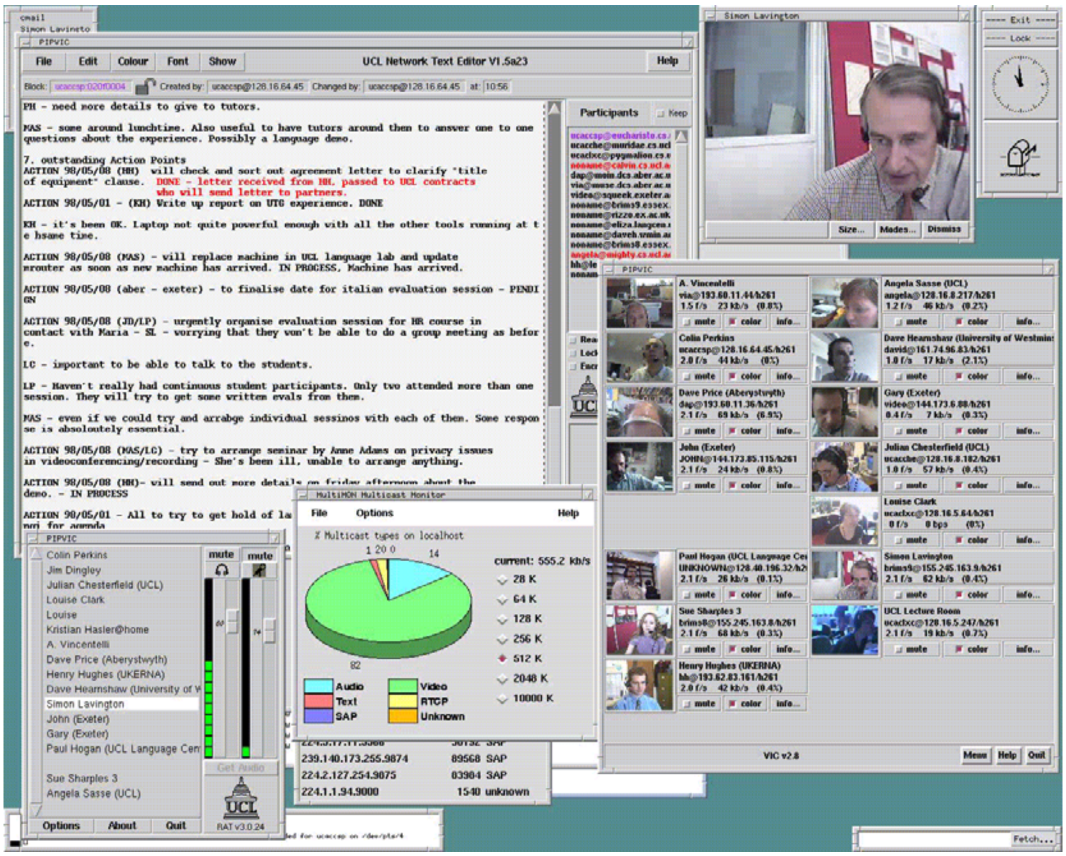


- Applications:

- NetMeeting and video conferencing;
- shared editing tools and CAD/CAM systems;
- games and MUDS (multi-user dungeons).

Synchronous CSCW

- Need access control mechanisms.



Acknowledgement: Shrimp Project

- Two types:

- social convention ('after you...');
- technological (locking systems).

Synchronous CSCW

- Access rates:

- better links/equipment give better response;
- this can have social/interaction effects.

- Frsutration over delays:

- jitter and quality of service.

- Need for conflict resolution:

- lock object while you are working on it?
- allow concurrent edits then resolve conflicts?

Common Ground

- Clark and Brennan:

- conversations to establish common ideas.

Cost	Description
Formulation	formulate and reformulate utterances
Production	producing the utterance
Reception	receiving a message
Understanding	understanding a message
Start-up	starting a new discourse
Delay	planning and revising before execution
Asynchrony	timing of discourse exchanges
Speaker change	changing speakers
Display	presenting an object of the discourse
Fault	producing a mistake
Repair	repairing a mistake

- If you say something but are misunderstood then

- you have to initiate a repair conversation;
- this is the cost of establishing common ground.

Common Ground

- Analyse transcripts to support design.

- Problems with speaker change:

- consider dialogue control measures?
- look again at locking techniques?

- Lots of repair activities:

- provide greater view of colleague's work?
- possibly add video to audio communications?

- Problems with delay:

- must see changes made while planning last message?

Summary

- Problems of groupwork.

- Computer-Supported Cooperative Work:

- face to face interaction;
- synchronous interaction;
- asynchronous interaction.

- Clark, Brennan and Common Ground.

Further Reading

- Shneiderman on:

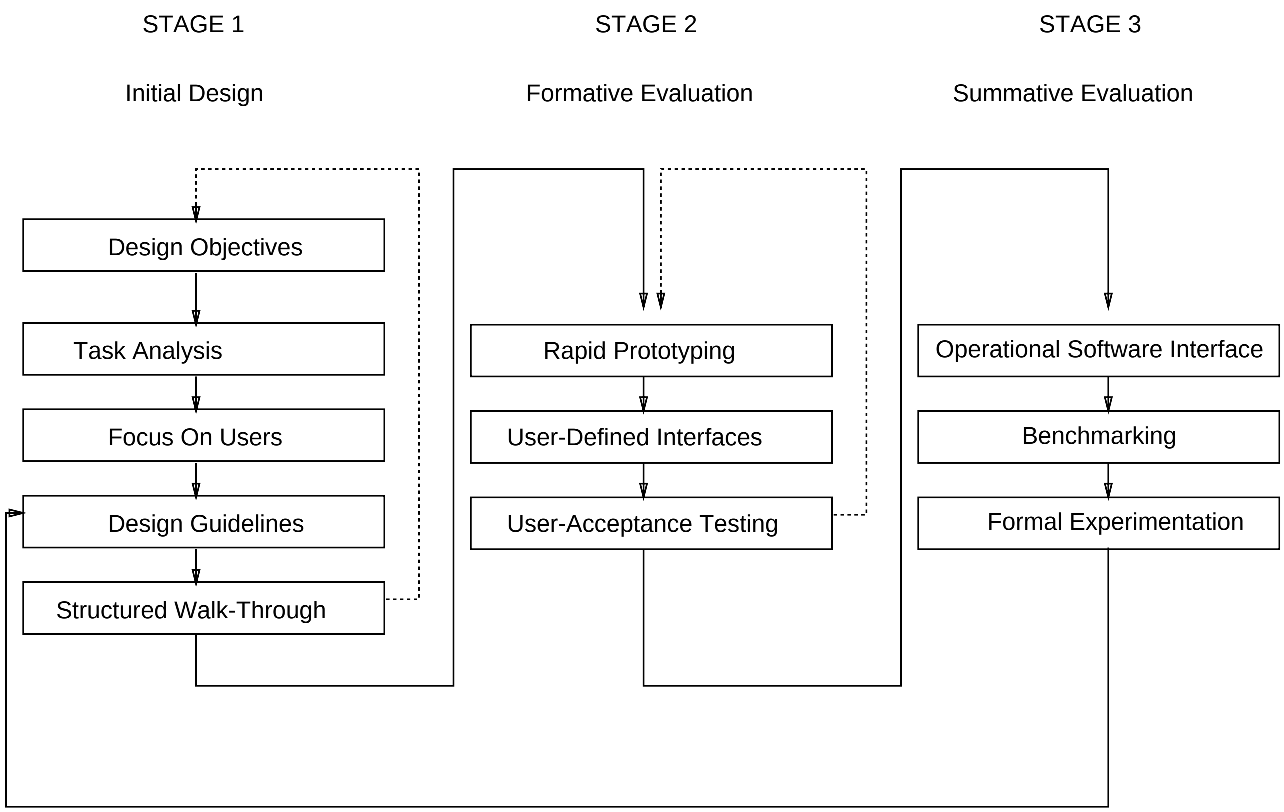
- CSCW - pp. 477-502.

Desktop Interaction

CS-1Q | HCI

Desktop Interaction

- How do we implement a desktop system?



- Iterative development and user-centred design

Desktop Interaction

1. consider users and environment;
2. design and specification;
3. select dialogue style;
4. implementation issues;
5. documentation issues;
6. evaluation issues.

1. Consider Users and Environment

- Home:

- cluttered with books, coffee cups etc;
- distractions from radio, other people etc.

Acknowldgement: BBC

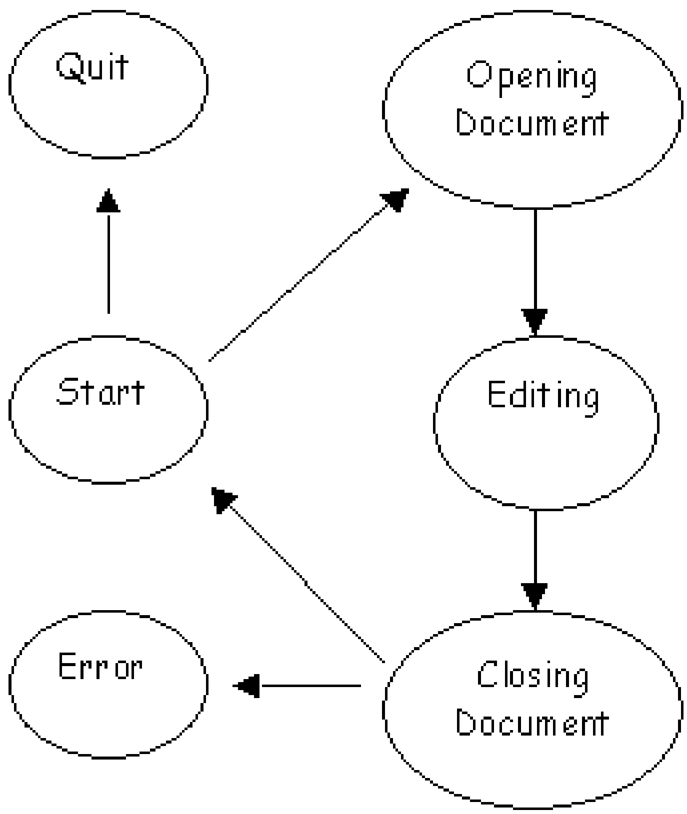
- Office:

- time pressure to complete tasks;
- need to achieve *closure*;
- heat, noise, interruptions, multiple tasks.

2. Design and Specification

- Task analysis looks from users' point of view.

- Specifications begin to look at system behaviour:
- focus on *what* to do not *how* to do it.



- State-transition diagrams often used.

2. Design and Specification

- States:

- situations where system continues to perform activity;
- change from a state is triggered by a transition;
- represented by nodes in the network.

- Transitions:

- edges represent change between states;
- usually triggered by user actions;
- can be labelled by frequency information.

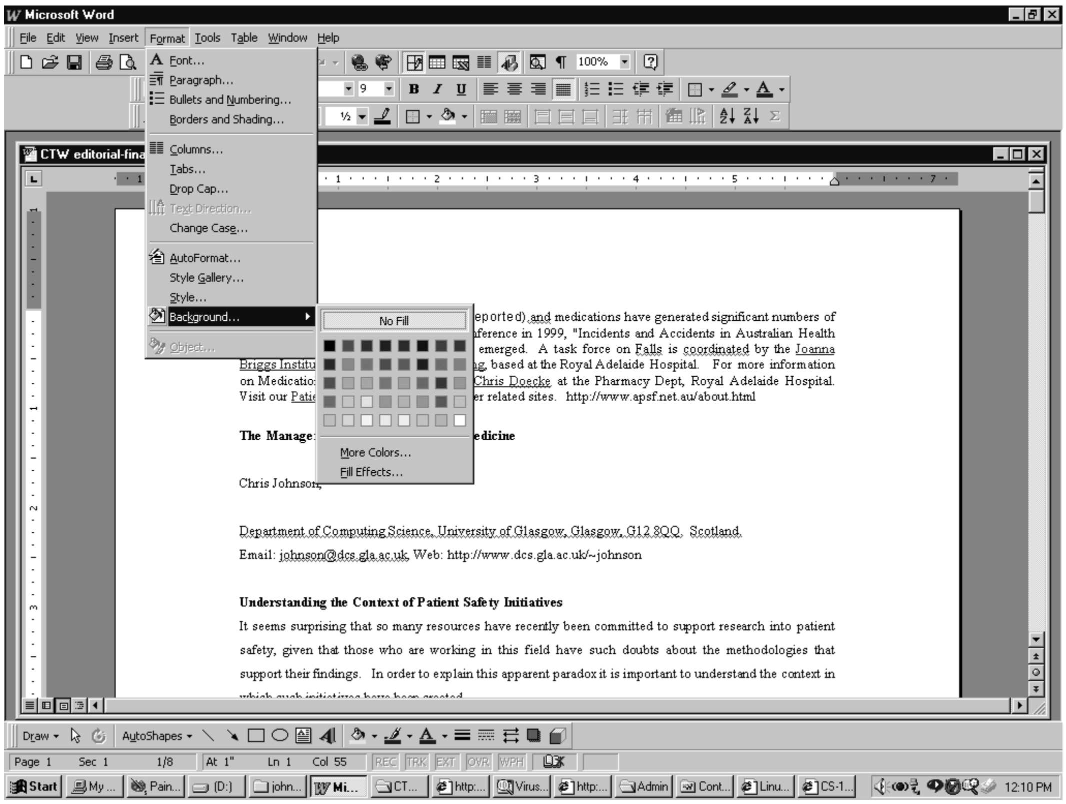
- Can do static analysis:

- can you get from state A to B (reachability);
- maximum of N transitions from A to D (spanning).

3. Select Dialogue Style

- Dialogue styles:

- text, forms, menus, graphics, mixtures...



- Dialogue styles rely on widgets:

- lists, choices and buttons;
- checkboxes and radio buttons;
- pull-down, tear-off, pop-up, scrolling, hybrid menus

3. Select Dialogue Style

- Windows:

- titles, borders and frames, scroll bars.

- Windowing actions:

- open, move, close, resize, select, bring forward,
- feedback is crucial to all of these actions.

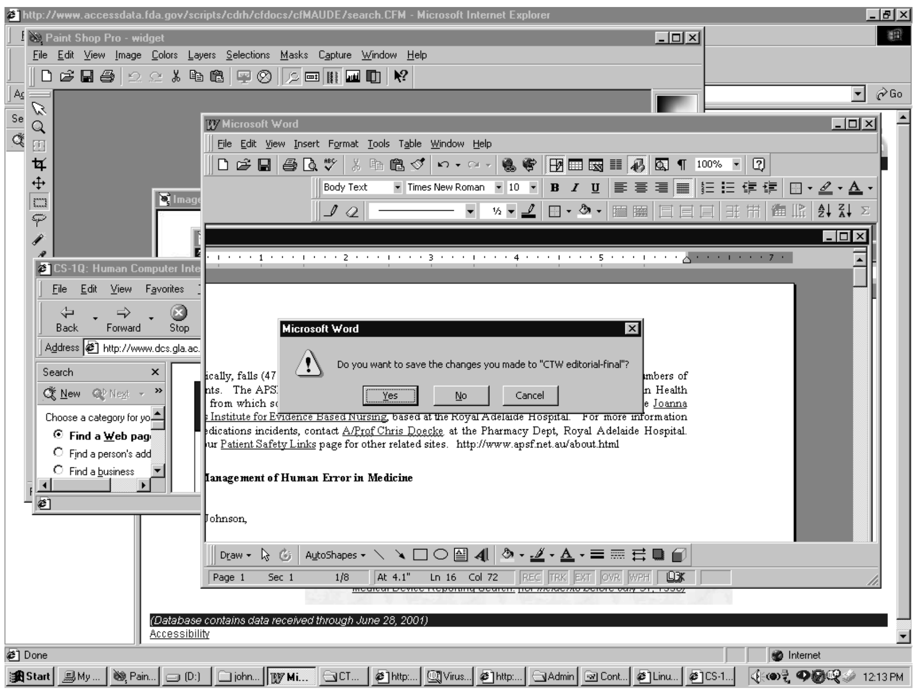
- Some very complex web-behaviours:

- spontaneously opening a linked window with a web page;
- ‘where on earth did that come from?’
- lots of security issues here - unsigned applets.

3. Select Dialogue Style

- Window managers between applications.

- Layout managers within an application.



- Placement issues:

\- multiple overlapping windows (how to select a window?);
\- tiling strategies (eg powerpoint slide view);
\- zooming (eg Acrobat page size);
\- card cascades etc.

4. Implementation Issues

- How do you get a widget in your code?

- Toolkits such as AWT, Swing (or JEWL).

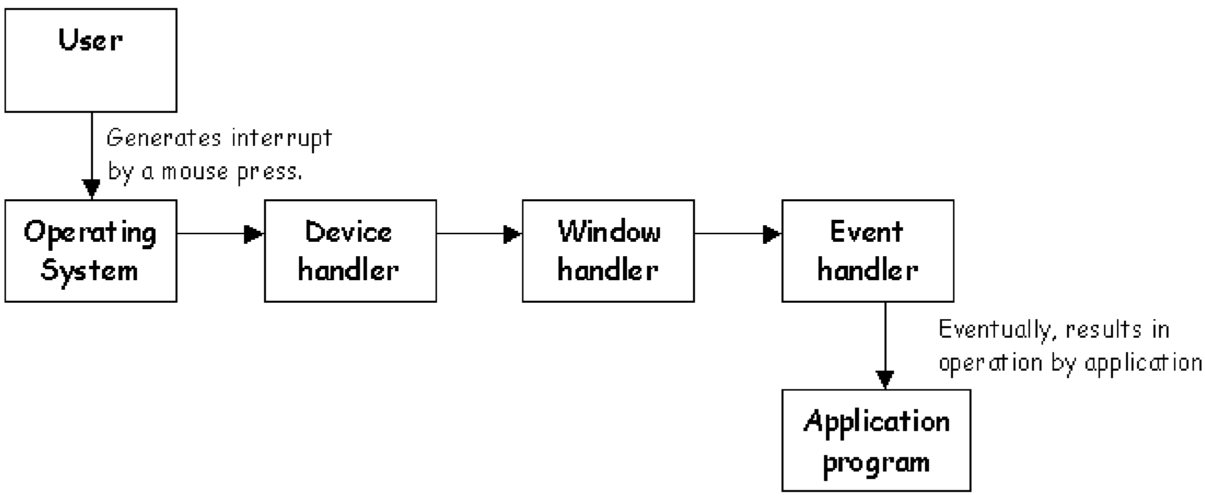
```
/* do NOT memorise this code! */
/* Try to understand what it is doing */

b1 = new Button(); // create a new button

b1.setLabel("Disable middle button"); // put a piece of text in it

b1.setActionCommand(DISABLE); // associate a command with it
```

- Toolkits are libraries of programs:

- others have written them to implement common widgets;
- you don't have to write code to draw a button etc;
- can be difficult to change these pre-coded widgets.

4. Implementation Issues

- What happens when button is clicked?




```
public void actionPerformed(ActionEvent e) {
    String command = e.getActionCommand();
    if (command == DISABLE)
        {} // b1 was pressed so do whatever you need to
}
```

4. Implementation Issues

- User interface programming is skilled:

- can take 3-4 months to learn the basics.

- Lots of tools to simplify the process.

- Direct manipulation interface builder:

- select widget and place it on 'screen';
- system automatically generates code you need;
- can be inflexible and inefficient.

- Microsoft Visual Basic, Borland Delphi.

- Also scripting tools such as Tcl/Tk.

5. Documentation Issues

- Printed manuals:

- seldom read, intimidating and go missing;
- must stay open at correct page while typing;
- must leave room for the keyboard and mouse!

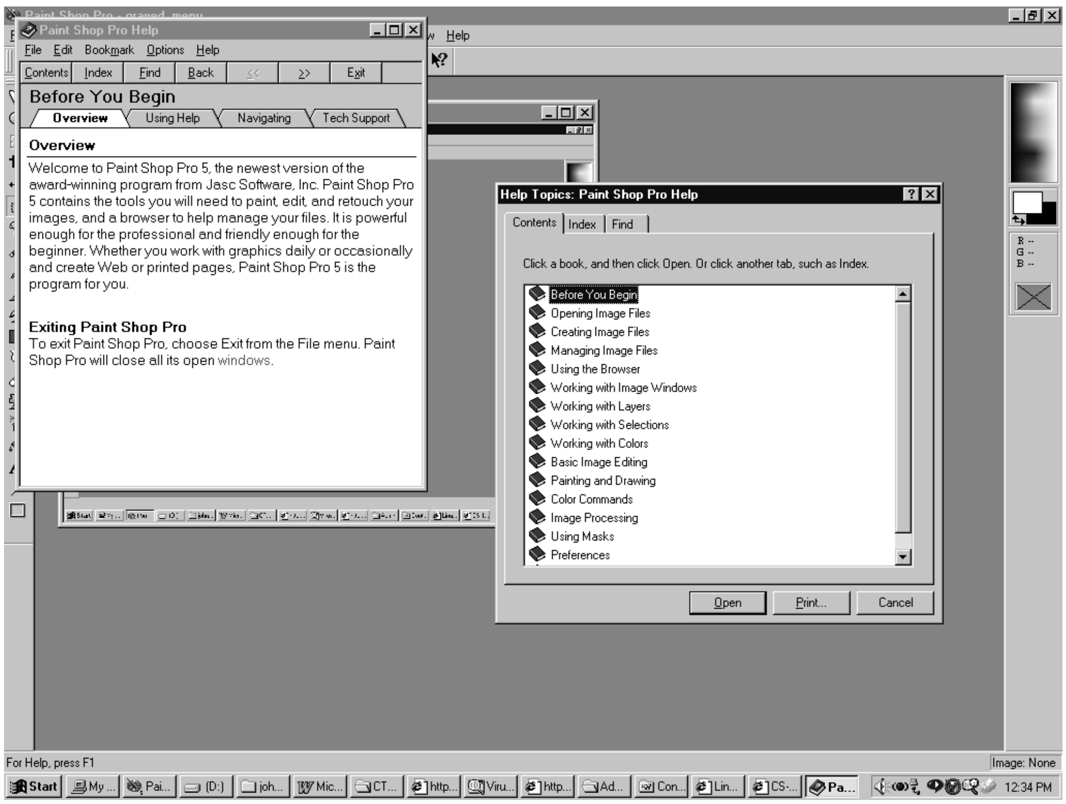
- Key idea of *minimal manuals*:

- short 'get you started' summaries;
- lists of available commands and short cuts.

5. Documentation Issues

- On-line tutorials and manuals:

- can be quite basic (Unix `apropos`/`man`)

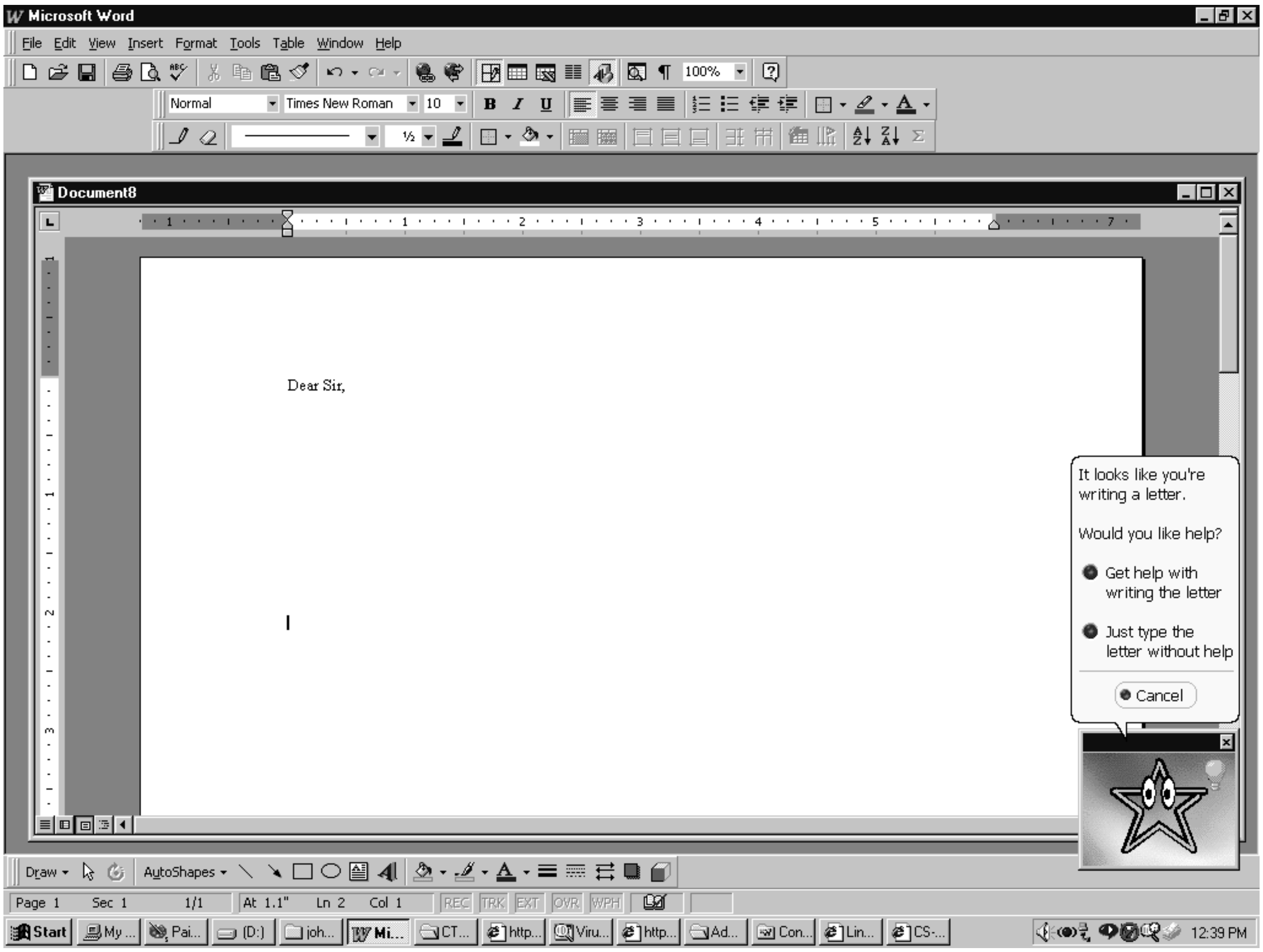


- On-line and context-sensitive help:

- can be very difficult to navigate;
- can be out of synch with software installation;
- can be pitched at too simple/complex a level.

5. Documentation Issues

- But screen reading is tiring and error prone.



- The infamous 'Office Assistant':

- does anthropomorphism always support users?

- 'Tip of the day' - can be patronising.

6. Evaluation Issues

- Recap lecture 4:

- what is formative evaluation?
- what is summative evaluation?

- Recap lecture 4:

- name one formative technique?
- name one summative technique?

Summary

1. consider users and environment;
2. design and specification;
3. select dialogue style;
4. implementation issues;
5. documentation issues;
6. evaluation issues.

Further Reading

- Shneiderman on:

- software tools - pp. 155-179.

Mobile Interaction

CS-1Q HCI

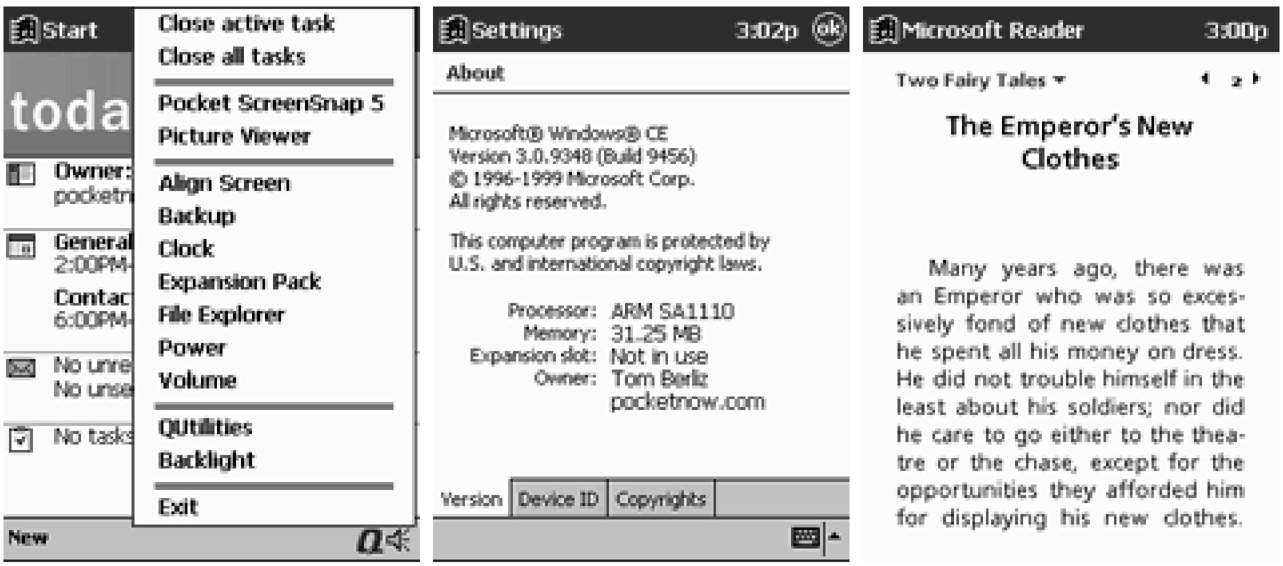
Mobile Interaction

- Input and output issues.
- Processor, battery and memory issues.
- WAP, G3 and Bluetooth.
- Context Awareness and GCS.
- Reality Checks...

Input and Output

- Display limitations:

- flicking through decks of cards;
- tiling, click to maximise;
- also 3D audio, tactile output.



Acknowledgement: T. Berlitz, www.pocketnow.com

- Input limitations:

- character recognition;
- automatic word completion;
- speech recognition;
- gloves and cameras.

Processor, Memory and Battery Limitations

- HP Jornada 710:

– 206MHz Strong ARM processor;
- 32MB RAM, 32 MB Flash Memory,
- Pocket Microsoft Windows Office.

Acknowledgement: www.hp.com

- 12-14 hours battery life.

Processor, Memory and Battery Limitations

- Compaq iPaq H3650:

- StrongArm 200 mhz
- 12 bit (4,096) color
- 32 MB RAM.

Acknowledgement: T. Berlitz, www.pocketnow.com

- What can you do with them?

Wireless Application Protocol

- Similarities to desktop web:

- phone has a browser like on a PC;
- no special server is needed.

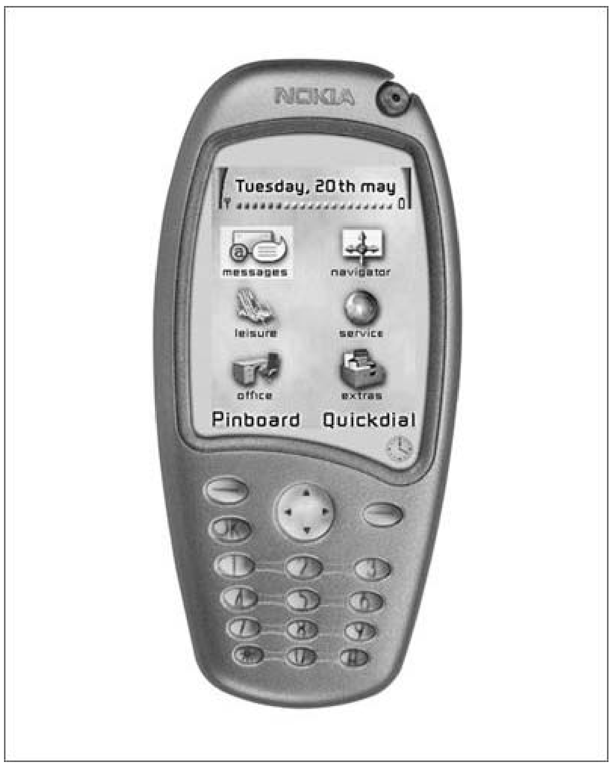
```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml.xml">
<wml>
<card id="Card1" title="Card 1">
<p align="center"> Happy Hippy's WML Demo<br/>
<big>This is card 1</big>
</p>

<do type="accept" label="Card 2">
<go href=">"
</do> </card>
<card id="Card2" title="Card 2">
<p align="center">
Happy Hippy's WML Demo<br/>
<big>This is card 2</big> </p>
<do type="accept" label="Card 1">
<go href=">"
</do>
</card>
</wml>
```

- But:

- HTML is now Wireless Markup Language;
- rules are more rigid to support small display.

Third Generation (3G)

“Peter Bodor, public relations manager at Ericsson:
”WAP’s disappointment was caused by industry failure to manage expectations, and the main problem was its slowness. This won’t be a problem with 3G. The 3G Internet experience will be as good as surfing from home, with the added benefit of location-based services making the experience more personal.” `http://news.zdnet.co.uk/`



- What is 3G and why pay so much for licenses?

Third Generation (3G)

- Simplest 3G phones for talking and

- will store all their information on the network.

- Second type video-streaming, news and web.

- Third type will be 'information centres',

- more like conventional top-end PDAs.

- They will be equipped with *Bluetooth*.

Third Generation (3G)

- More technical information:

- 1st generation provide analogue voice telephony
- 2nd (current) generation add some data (fax and email)
- 3rd will provide data rates of up to 2 Mb per second.

- UMTS (Universal Mobile Telecommunications System)

- 3G standard being developed across globe;
- International Telecommunications Union (ITU);
- family of standards to switch between, not one

- Expectation that 3G will be here in 2002;

- based on EU's UMTS Decision for licensing schedule.

Bluetooth

- Local Area Network radio systems:

- IEC 802.11 (Apple AirPort, Lucent Orinoco;
- can be up to 10Mb/s depending on card;
- cover 50-200 meter cells depending on walls etc.

- But - need for shorter connections:

- connect laptop to your phone to your TV to your fridge.

- Infra-red (IrDA) communications:

- operate over a few meters, line of sight;
- difficult to maintain and slow data rates.

- Bluetooth:

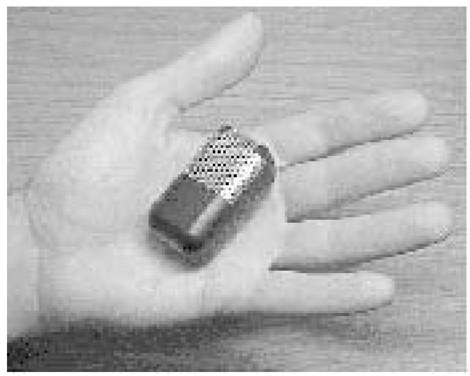
- ow-cost, short range radio links;
- 1Mb/s with an actual data rate of 728 Kb/s.

- BUT - IEC802.11 now rivals Bluetooth!

Context Awareness

- Why are cell sizes important?

- inside GPS receivers will not work;
- so find out what cell a user is in;
- follow-me applications such as phone switching.



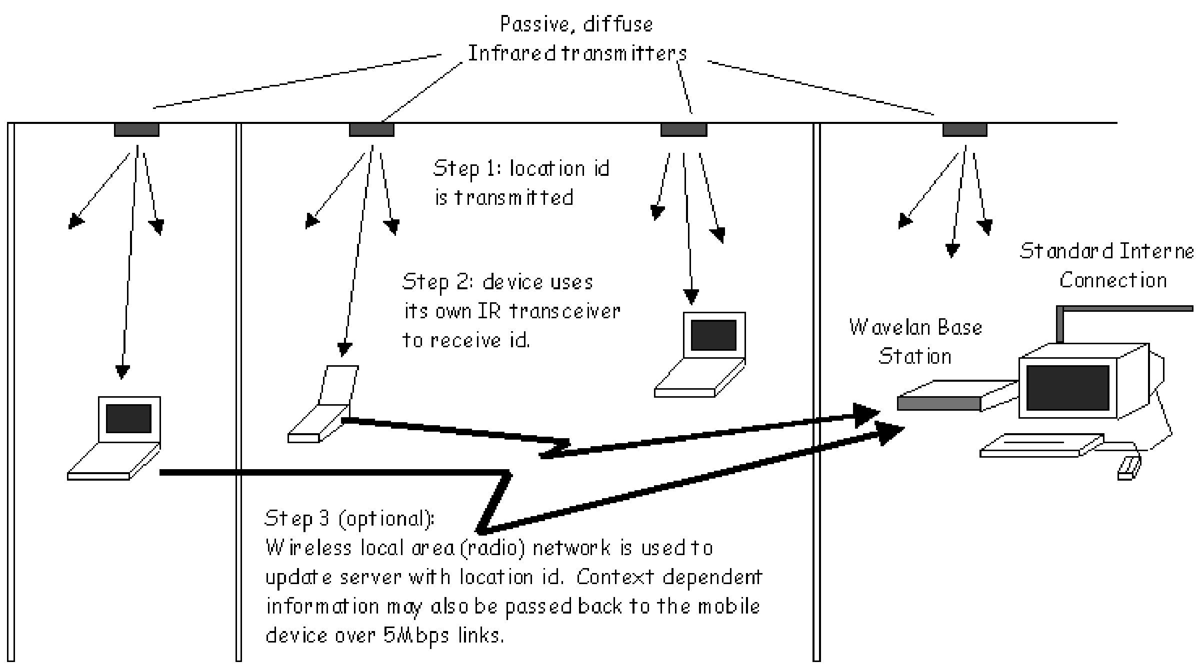
Acknowledgement: Olivetti and iButton.

- Alter information to users location.

Glasgow Context Server

- Key principles:

- passive location detection;
- ‘off the shelf’ hardware;
- uses infra-red and radio LAN (IEC 802.11).



- Currently working on human PAC-MAN.

Reality Check 1

- What will all these devices be used for?

- Nokia 9000 Communicator:

- developed, marketed and sold;
- all before anyone knew how it would be used.

- Market opportunity not user-centred design:

- but Nokia then do close observational studies;
- inform subsequent development of product;
- after initial market is established.

Reality Check 2

- Mobile devices pose huge challenges.

- Physical constraints demand new widgets.

- User tasks are difficult to predict.

- Users move and so are difficult to observe.

- Marketing hype often claims early adopters;
- difficult to anticipate longer term usability issues.

Summary

- Input and output issues.
- Processor, battery and memory issues.
- WAP, G3 and Bluetooth.
- Context Awareness and GCS.
- Reality Checks...

Further Reading

- Shneiderman isn't very good on this area.

`http://www.cs.strath.ac.uk/ mdd/mobilehci/procs/`

- Read a couple of articles instead?

Error

CS-1Q HCI

Errors

- Slips, lapses, mistakes and violations.

- Error tolerant design

- Error detection and recovery.

- Organisational factors.

Slips, Lapses and Mistakes

- Junior trader sells $16m of German bond futures.

Acknowledgement: BBC (Nov. 1998)

- Trader thought it was a training screen.

- Whose fault is this? Trader or Company?

Slips, Lapses and Mistakes

- Error:

- unwitting deviation of actions from intentions.

- Violation:

- deliberate deviation of actions from regulations.

- Slip:

- visible failure in the execution of a plan;
- a slip of the tongue is observable.

- Lapse:

- invisible failure in the execution of a plan;
- forgetting someone's name.

- Mistake:

- a failure of intention;
- trying to use Word to maintain complex accounts.

Why do Errors Occur?

- Fatigue and circadian rhythms:

- mistakes are very likely last thing on a Friday!

- Stress (light, heat, noise, domestics):

- environmental factors and distractions induce mistakes.

Why do Errors Occur?

- Alcohol and drugs:

- long office lunches don't help interaction

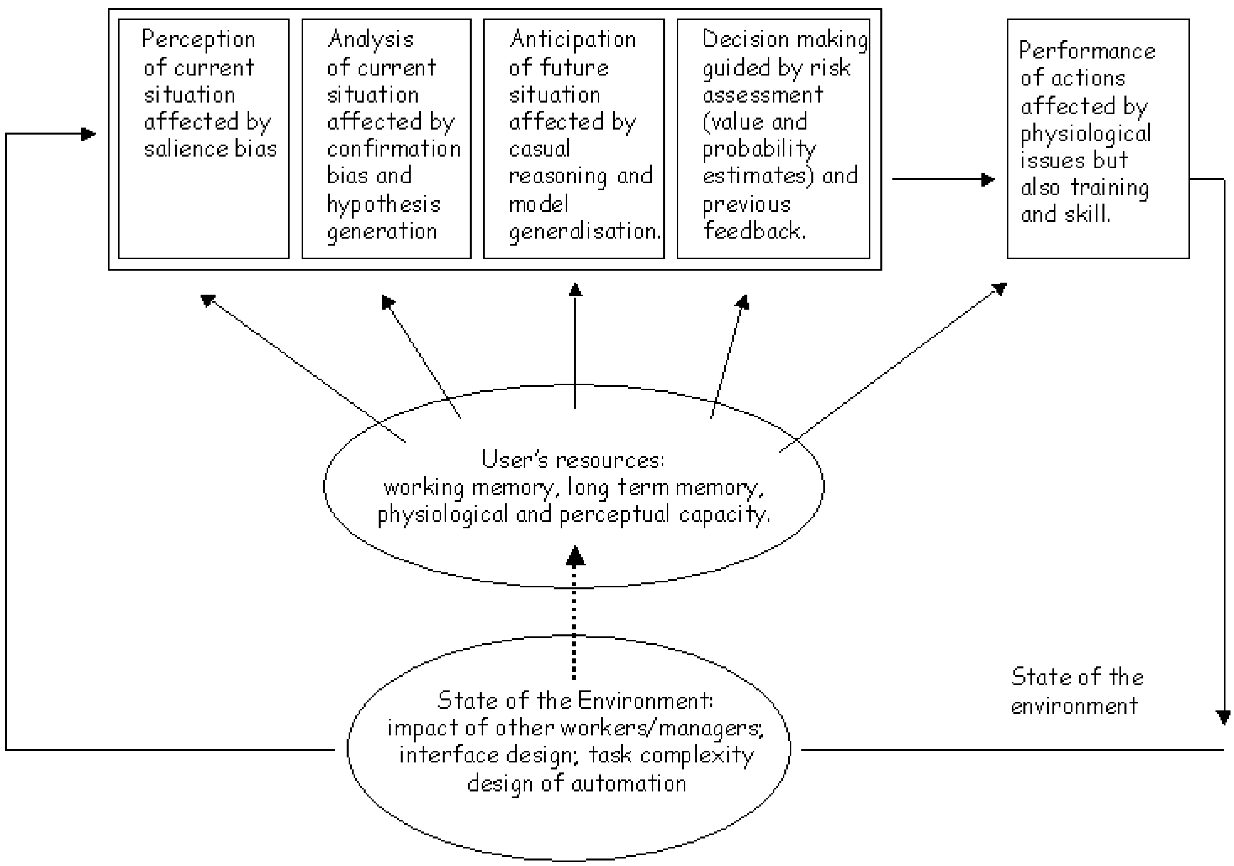
- Workload (physical, mental etc):

- time pressures can impair performance.

- Individual differences:

- some people actually do make more mistakes than others;
- spell-checking indicator - correct now or at end?

Perception, Cognition, Physiology



Situation Awareness

- Level 1: perception of elements in environment.

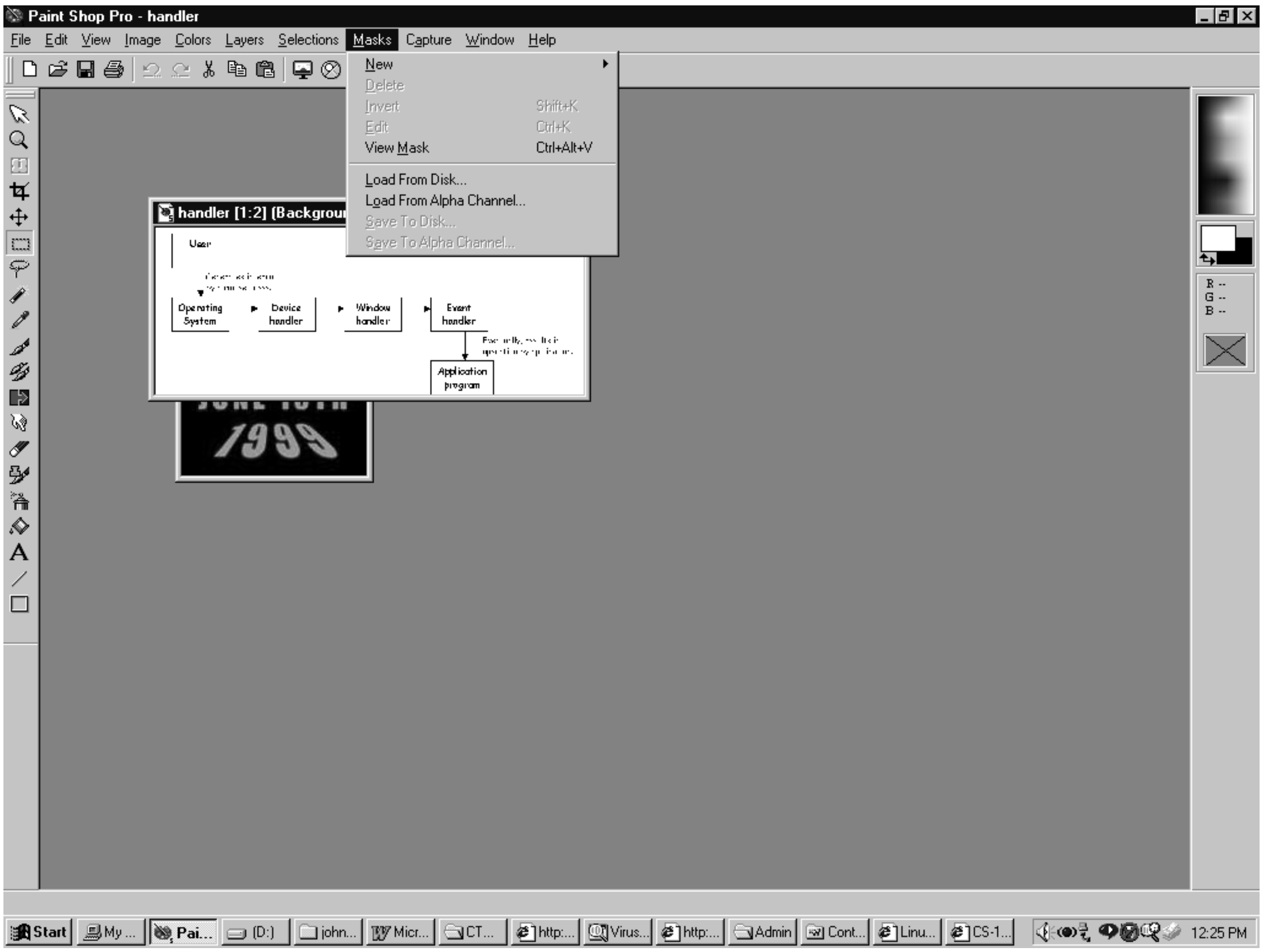
- Level 2: comprehension of current situation.

- Level 3: projection of future states.

Error Tolerant Design

- Greying out menu items:

- users can't select inappropriate item.

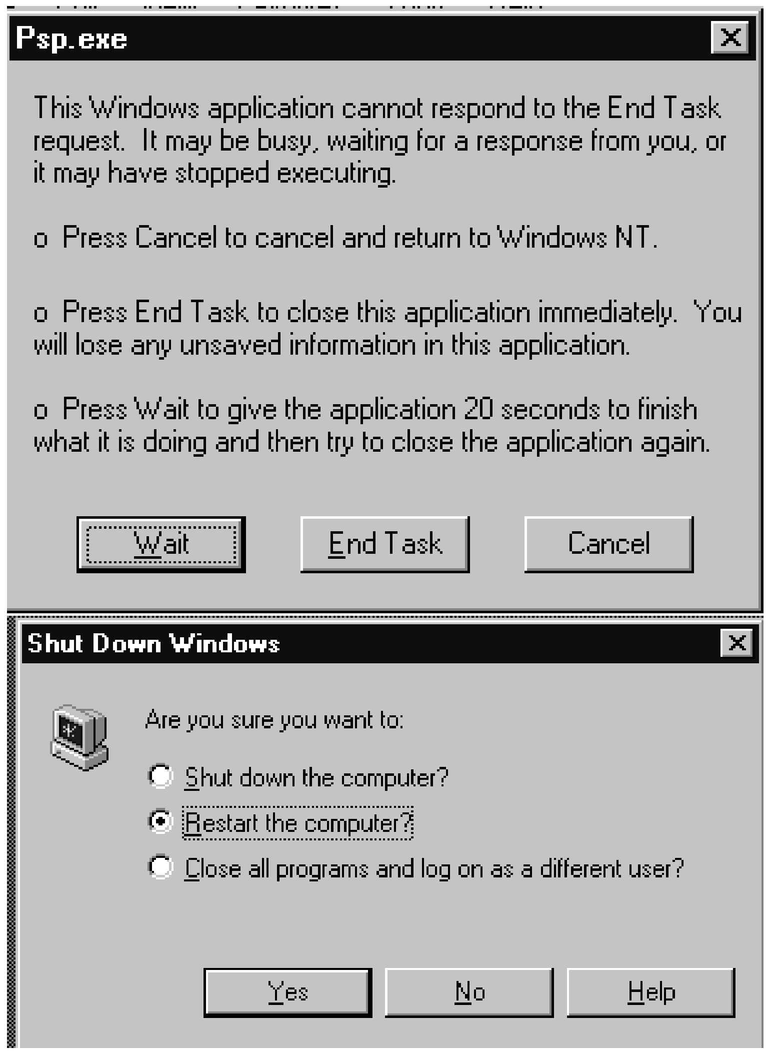


- Training wheels:

- prevent users from making an error.

Error Tolerant Design

- What happens if you undo and undo?

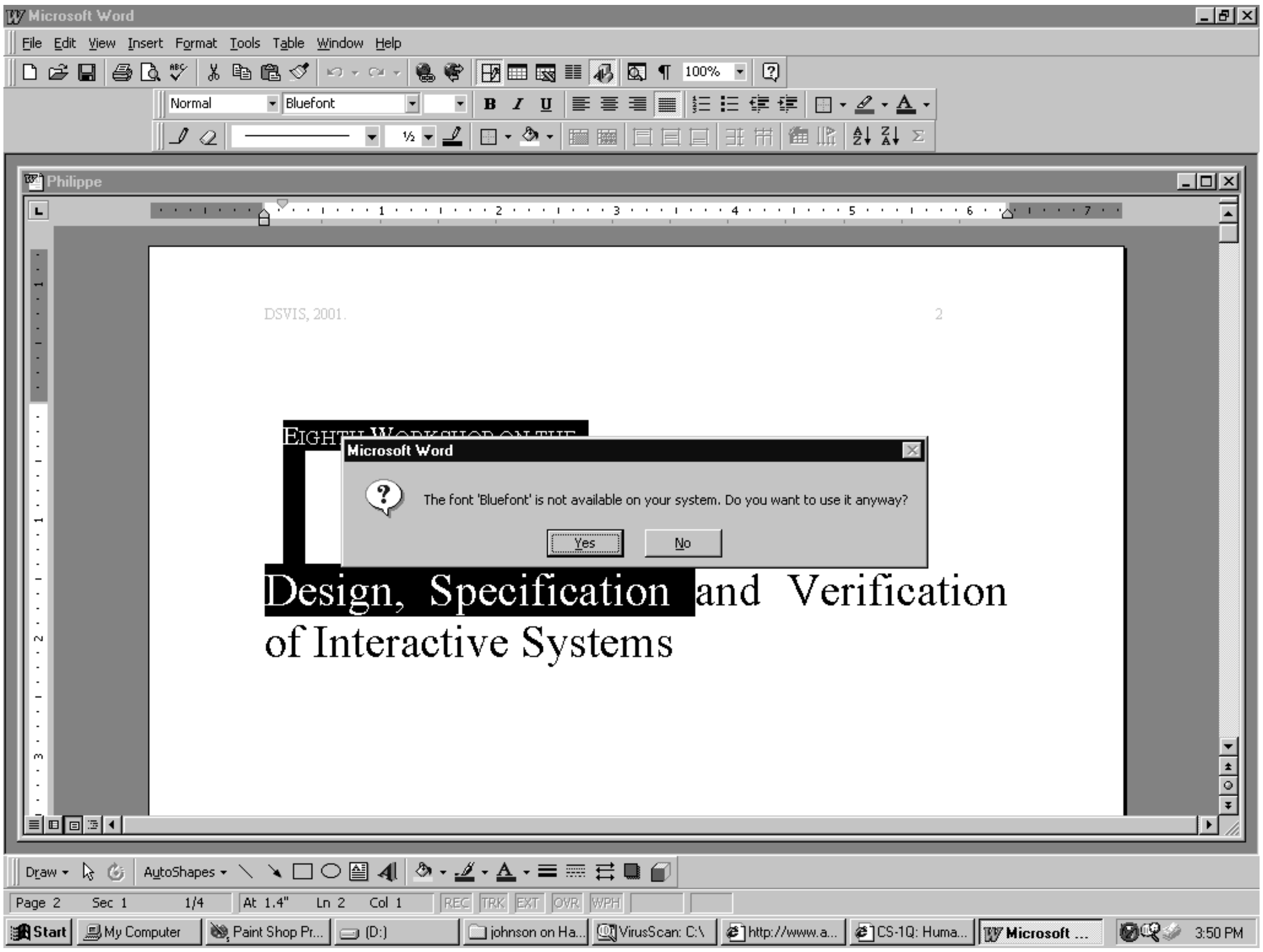


- Confirmation of irreversible actions:

- do you really want to reformat this disk?
- do you really want to end this task?

Error Tolerant Design

- Validation of input prior to entry.



- As soon as possible to reduce frustration.

Error Detection

- Error messages:

- recap on Winograd and Flores' breakdown.

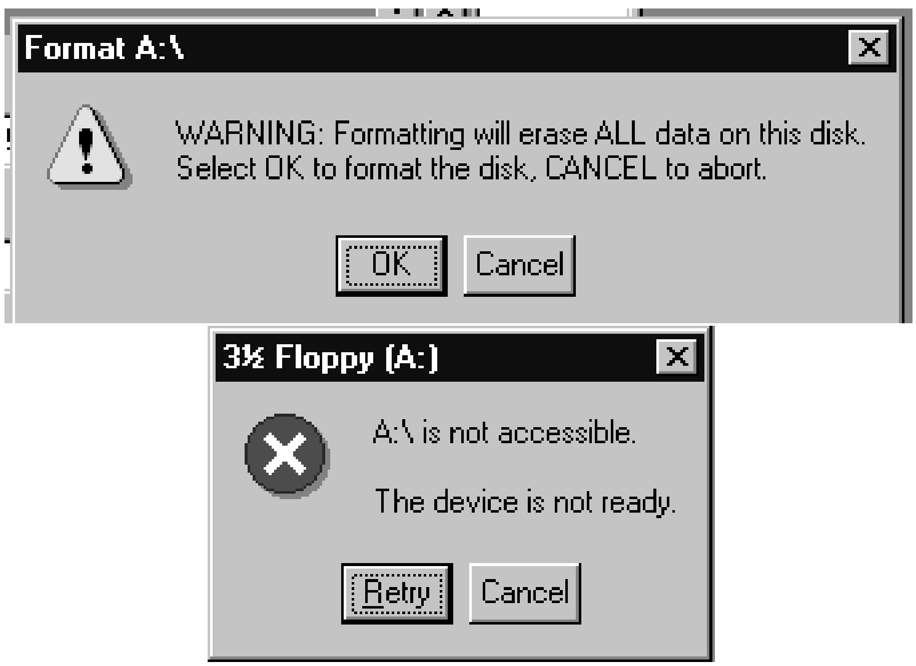
- Must consider both frequency and consequences:

- a rare error message may need more explanation;
- users must receive indication of seriousness.

Error Detection

- Interface design:

- some users will fail to observe error messages;
- some cannot interpret the meaning of the message.



- Focus for user testing: 'now get out of that'.

Observing Errors

- Will user testing reproduce errors?

- people strive to please investigator;
- people know they are being watched (Hawthorne effect)

- Logging and tracking of user's behaviour.

- Problems:

- will logs distinguish between slip and mistake?
- will logs help to detect lapses at all?
- ethical issues and legal issues...

Organisational Factors

- James Reason:

- Human Error (1990);
- Managing the Risks of Organizational Accidents (1997).

- Who causes the error:

- the user, the designer, the manager?

Summary

- Slips, lapses, mistakes and violations.

- Error tolerant design

- Error detection and recovery

- Organisational factors.

Further reading

- Again Shneiderman skims this issue.

- Shneiderman on error messages - 373-379.

- Try to read Reason's Human Error?

- Yes, seriously...

Fun

CS-1Q HCI

Fun and Games

- Why Learn from Games?

- How to Transfer Strengths of Games:

- steal superficial design features;
- derive heuristics from successful games?
- understand the psychology of play?

- Physiology and Affective Computing...

Fun and Games

- Use strengths of games in other interfaces?

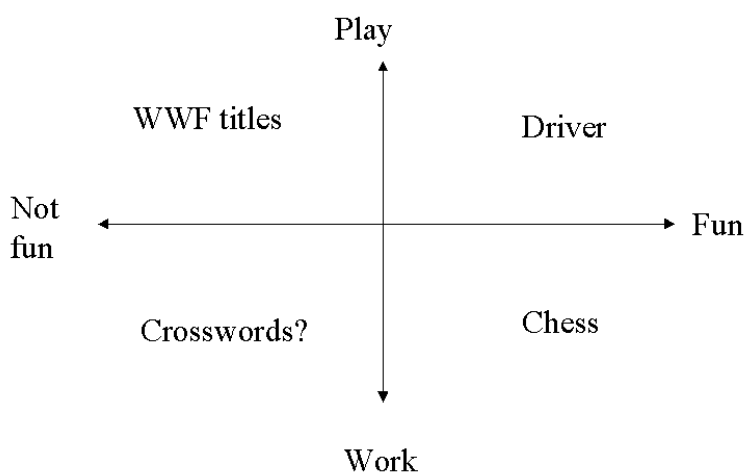


- MS Word as much fun as TombRaider?

- Holy grail of HCI?

Fun and Games?

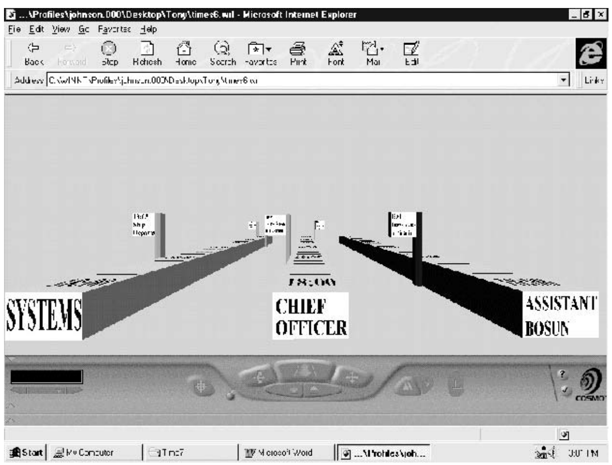
- Not all games are the same...

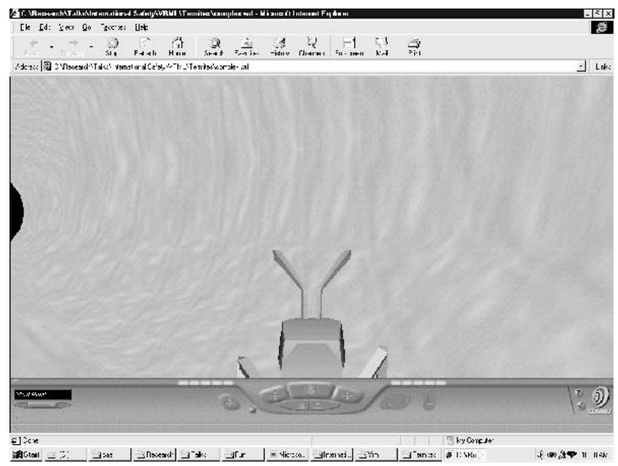


- We don't understand much about games...

Why Learn from Games?

- Technology is now easy to access (eg VRML)



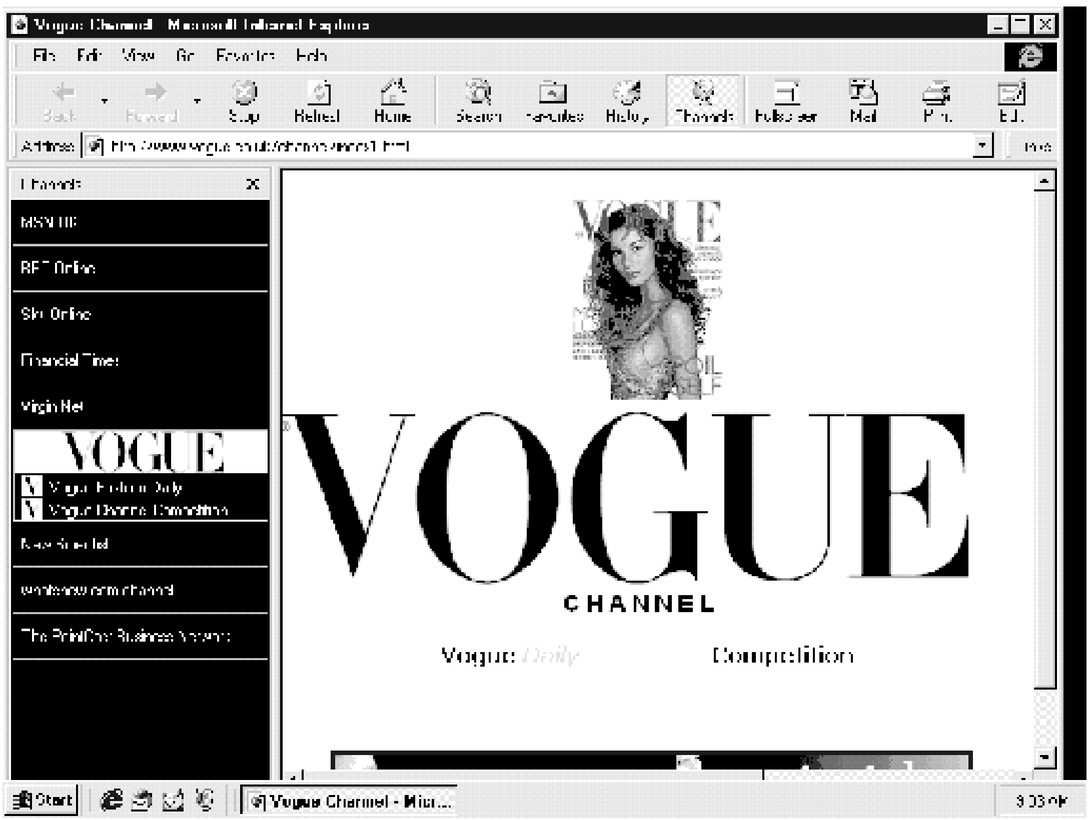


- Increase creativity in interface design;

- go beyond the usual menu based structures.

Why Learn from Games?

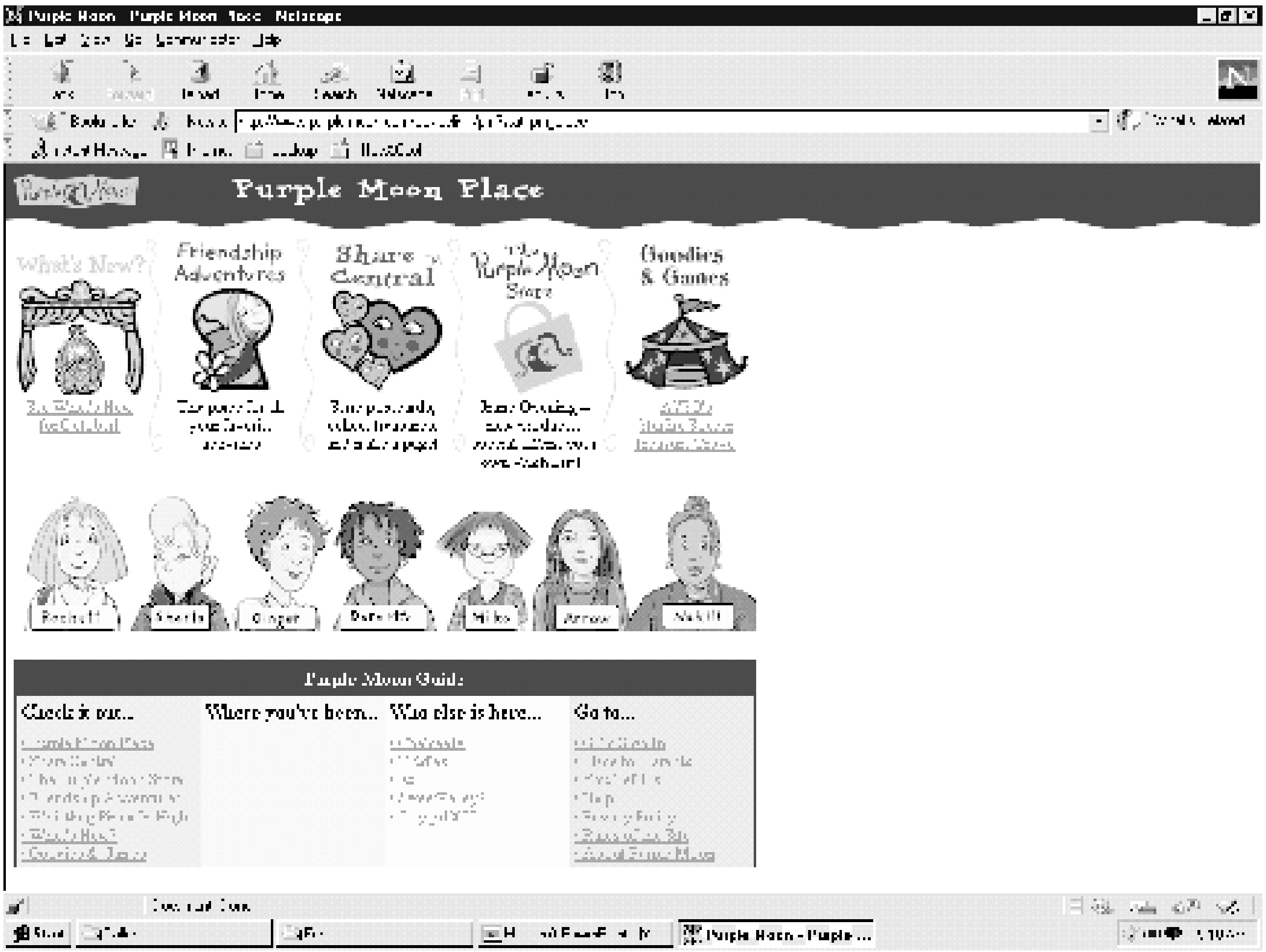
- Subjective satisfaction is very subjective.



- Task analysis does not always help here.

Why Learn from Games?

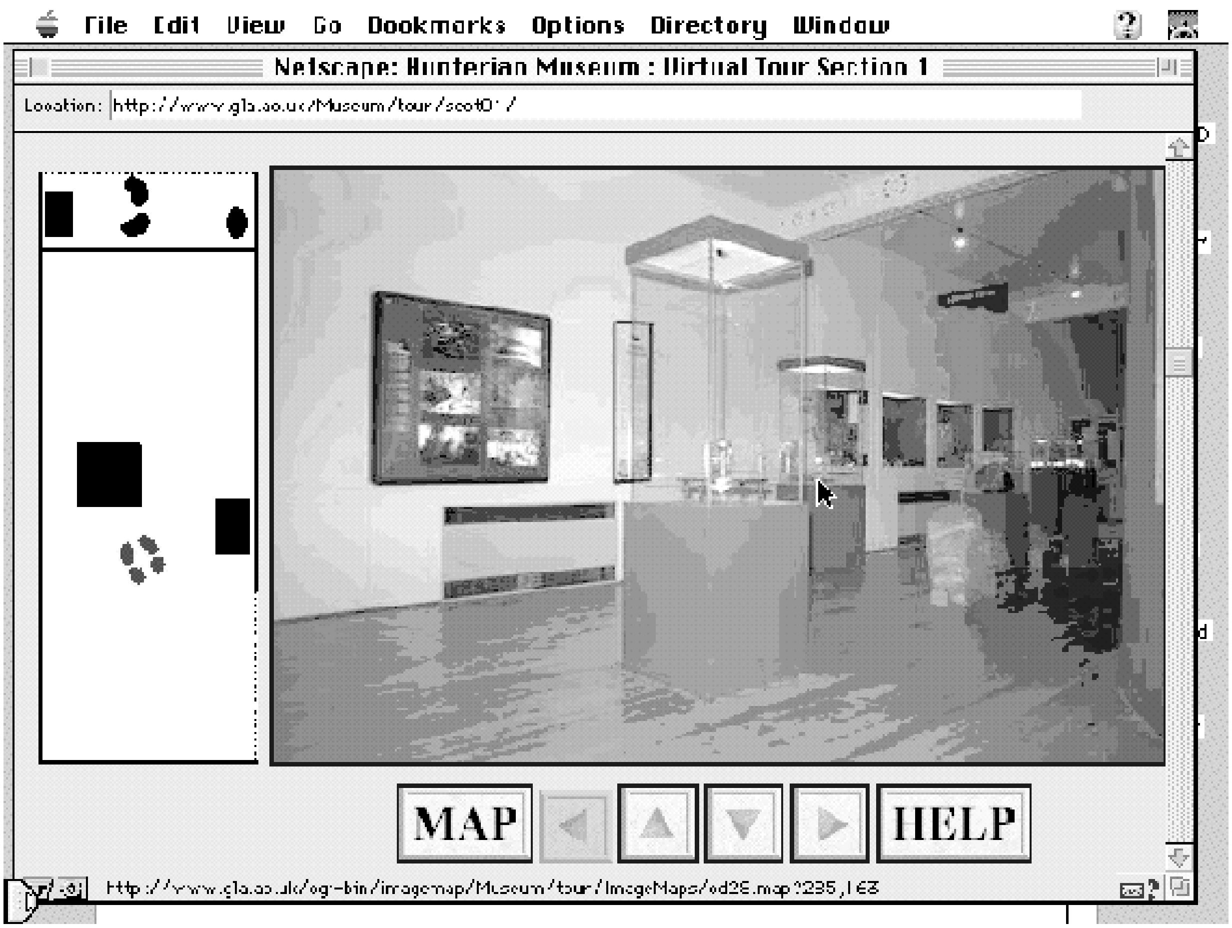
- Games open up new markets.



- Target new users, young, old etc.

How Can We Transfer Ideas from Games?

- Steal superficial screen components.



- But:

- unlikely to radically increase motivation;
- what can we steal? Scores? Punishments?
- how can we predict which features will work?

How Can We Transfer Ideas from Games?

- Lisa Neal (1990) looks for generic ideas.

- A sense of control.

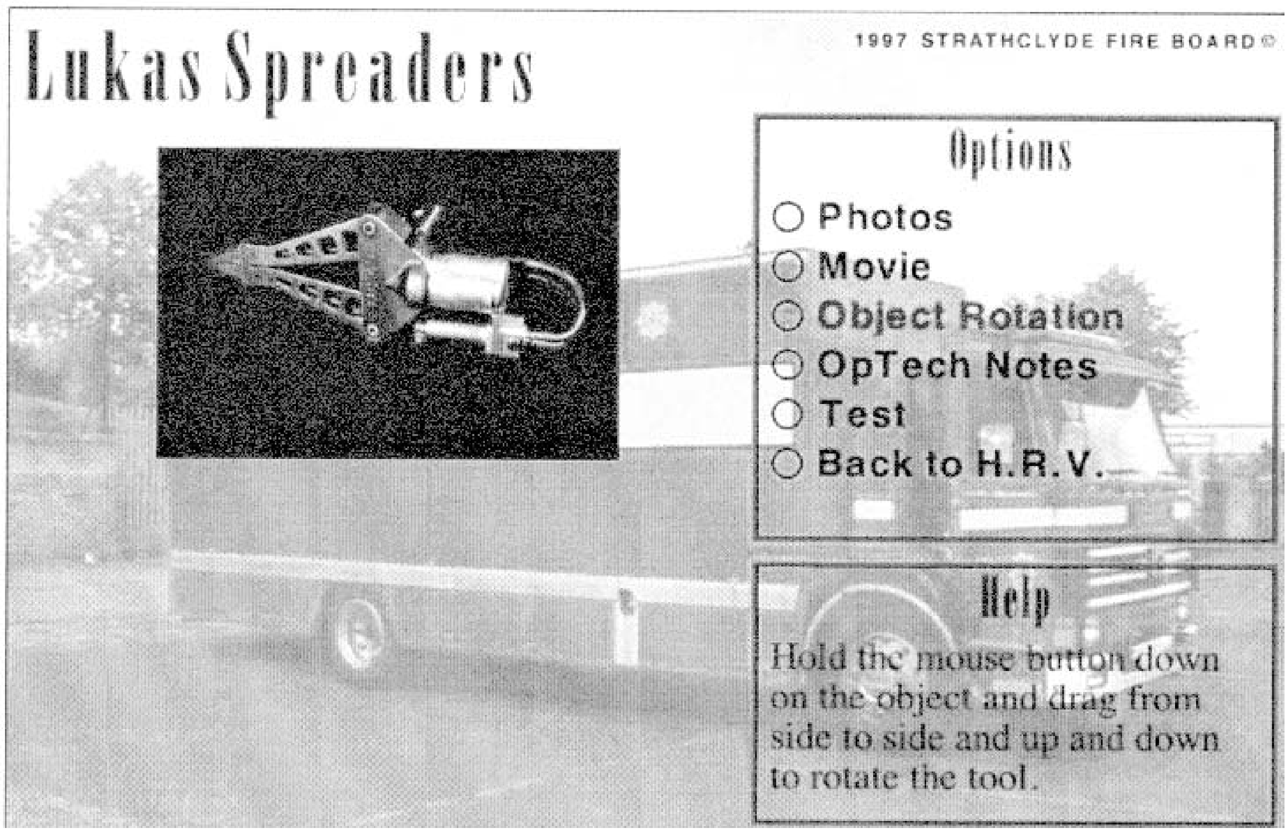
- Opportunity for discovery.

- Non-goal activities support learning?

- some users just want to have fun...

How Can We Transfer Ideas from Games?

- Strathclyde Regional Fire Brigade.

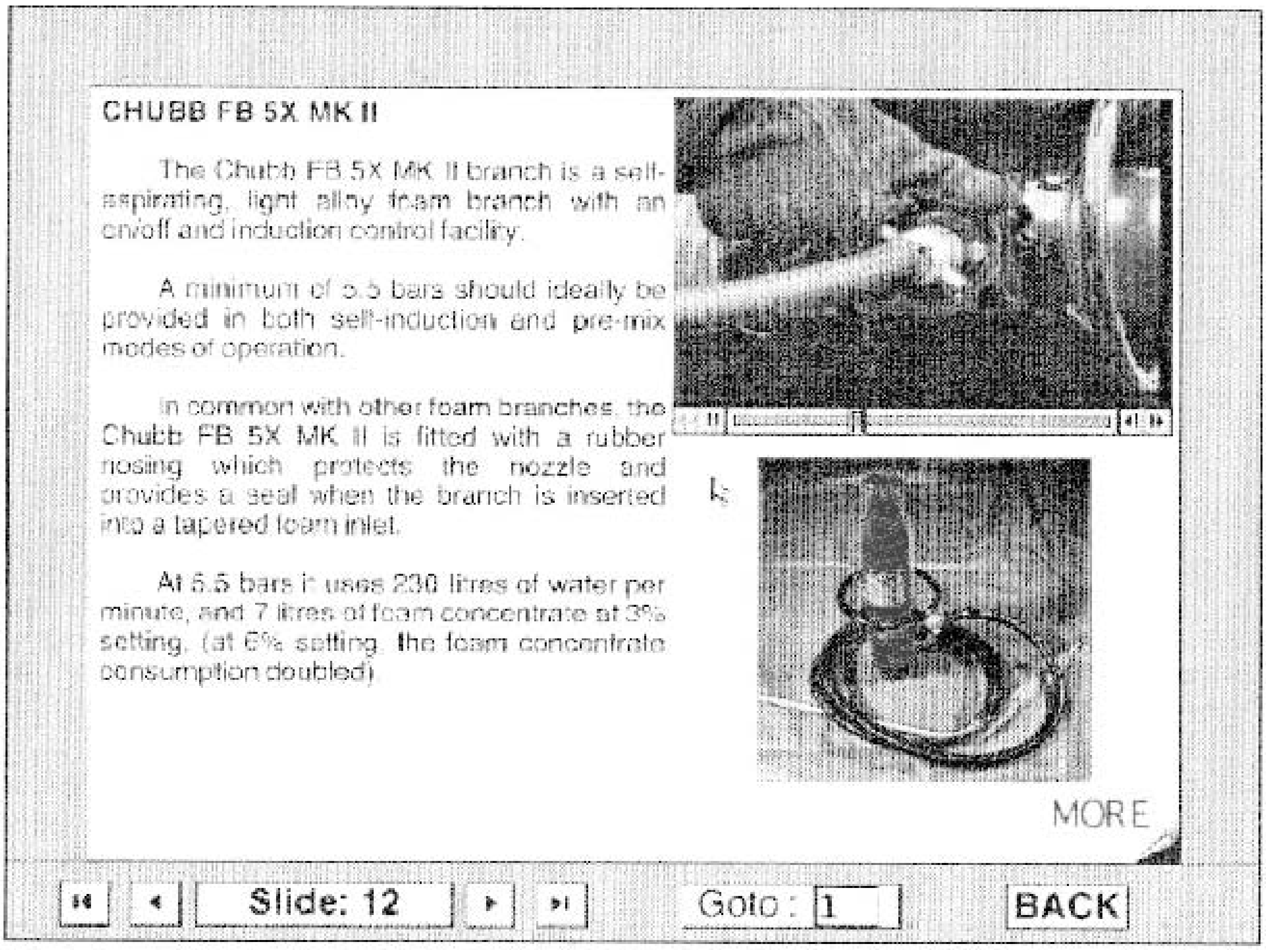


- Heavy rescue Vehicle (Mathers et al.);

- uses desktopVR to explore inside the vehicle.

How Can We Transfer Ideas from Games?

- Sometimes heuristics don't work.



- Foam training procedures have strict sequence;

- can't explore in an undirected way and learn sequence?

Psychology of Play

- Neal still looking at surface issues;

- need to understand the underlying psychology.

- Morris and ‘mood congruence’

- behaviour and mood is linked to self-image.

How Can We Transfer Ideas from Games?

- Csikzentmihalyi and 'flow theory'

- optimal challenge and control (risk?).

- Change levels of challenge to user expertise.

How Can We Transfer Ideas from Games?

- Mood congruence:

- Picard and affective computing;
- computers might adapt to your moods;
- ‘you do it, I can’t be bothered.

- Flow theory:

- force users to explore new functionality;
- ok so you did it this way last time, so now...

- Great interest in physiological computing;

- sense user’s pulse, galvanic skin resistance etc.

Summary

- Why Learn from Games?

- How to Transfer Strengths of Games:

- steal superficial design features;
- derive heuristics from successful games?
- understand the psychology of play?

- Physiology and Affective Computing...

Further reading

- Shneiderman doesnt look at this issue.

http://www.hcibib.org

- Look up Fun or Games on the Perlman database.

- Some brief papers by Rosalind Picard.