


 **HUMAN-COMPUTER INTERACTION** THIRD EDITION  DIX FINLAY ABOWD BEALE



chapter 21

hypertext, multimedia and the world-wide web

 **HUMAN-COMPUTER INTERACTION** THIRD EDITION  DIX FINLAY ABOWD BEALE



hypertext, multimedia and the world-wide web

- [understanding hypertext](#)
 - text escapes linearity, words and the page
- [finding things](#)
 - navigating hyperspace
- [web technology](#)
 - how it all works
- web content
 - [static](#): unchanging pictures and text
 - [dynamic](#): interaction and applications on the web

 **HUMAN-COMPUTER INTERACTION** THIRD EDITION  DIX FINLAY ABOWD BEALE

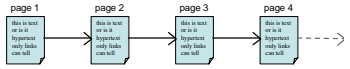
understanding hypertext

what is the hyper?
rich content: graphics, audio, video,
computation and interaction






Text

- imposes strict linear progression on the reader

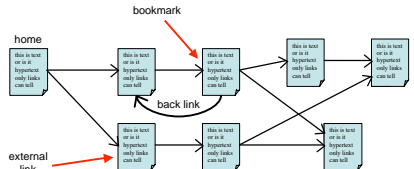




- the author's ideas of what is best
- often good :-)
... but not always!

Hypertext - not just linear



- non-linear structure
 - blocks of text (pages)
 - links between pages create a mesh or network
 - users follow their own path through information





Hypermedia - not just text

- hypertext systems + additional media
 - illustrations, photographs, video and sound
- links/hotspots may be in media
 - areas of pictures
 - times and locations in video
- also called multimedia
 - but term also used for simple audio/video



animation

- adding motion to images
 - for things that change in time
 - digital faces – seconds tick past or warp into the next
 - analogue face – hands sweep around the clock face
 - live displays: e.g. current system load
 - for showing status and progress
 - flashing caret at text entry location
 - busy cursors (hour-glass, clock, spinning disc)
 - progress bars



animation (ctd)

- for education and training
 - let students see things happen ... as well as being interesting and entertaining images in their own right
- for data visualisation
 - abrupt and smooth changes in multi-dimensional data visualised using animated, coloured surfaces
 - complex molecules and their interactions more easily understood when they are rotated and viewed on the screen
- for animated characters
 - wizards and help



video and audio

- now easy to author
 - tools to edit sound & video and burn CDs & DVDs
- easy to embed in web pages
 - standard formats (QuickTime, MP3)
- still big ... but getting manageable
 - memory OK ... hand held MP3 players, TiVo etc.
 - but download time needs care – tell users how big!
- very linear
 - hard to add 'links' often best as small clips or background



audio issues

- formats
 - raw sound samples
 - huge ... used for mixing and editing
 - MIDI
 - just which notes played and when
 - MP3
 - uses psychoacoustics - how the ear hears
- issues
 - annoying if unwanted
 - even more annoying for others!

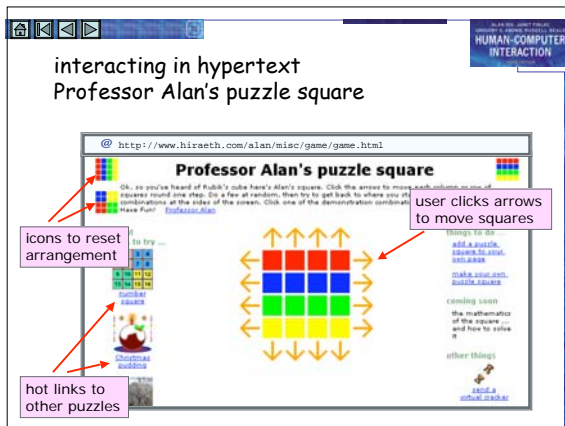
using animation and video

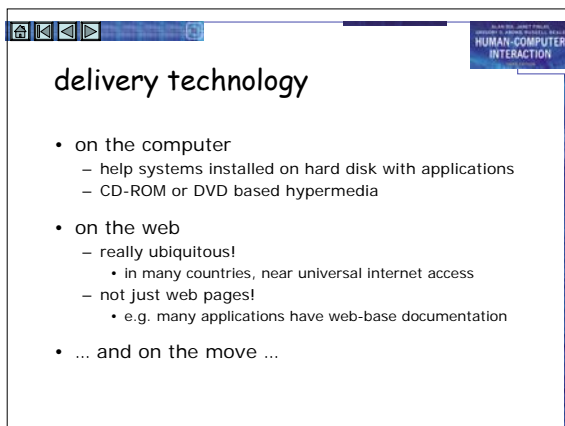
- potentially powerful tools
 - note the success of television and arcade games
- but ...
 - how to harness the full possibilities of such media
 - different from 'standard' interfaces
 - this technology when we have much more experience.
- SO ...
 - need to learn from film makers, dramatic theory, cartoonists, artists, writers

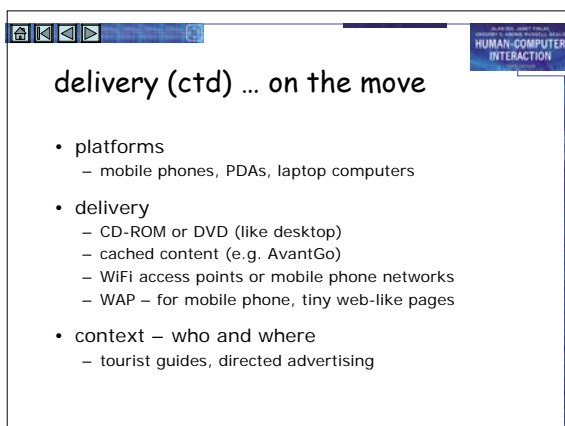



computation, intelligence and interaction

- computers??
 - don't just *show* things ... *do* things
- examples:
 - search – the HCI book web site
 - not just exercises, table of contents ... also search
 - interaction
 - embedded applications (e.g, puzzle square)
 - adaption:
 - e-commerce sites suggest other things to buy







HUMAN-COMPUTER INTERACTION

application areas

- rapid prototyping
 - create live storyboards
 - mock-up interaction using links
- help and documentation
 - allows hierarchical contents, keyword search or browsing
 - just in time learning
 - what you want when you want it (e.g. technical manual for a photocopier)
 - technical words linked to their definition in a glossary
 - links between similar photocopiers

HUMAN-COMPUTER INTERACTION

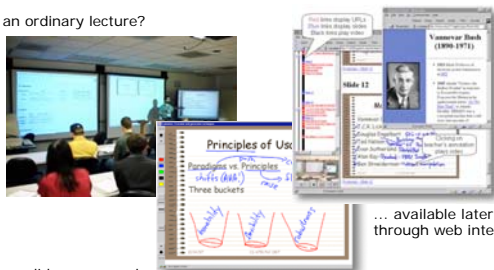
application areas (ctd)

- education
 - animation and graphics allow students to see things happen
 - sound adds atmosphere and means diagrams can be looked at while hearing explanation
 - non-linear structure allows students to explore at their own pace
 - e-learning
 - letting education out of the classroom!!
 - e.g. eClass

HUMAN-COMPUTER INTERACTION

eClass (formerly Classroom 2000)

an ordinary lecture?



slides, pen marks, video are 'captured'

... available later through web interface

Navigation icons: Home, Back, Forward, Stop, Refresh, Search

HUMAN-COMPUTER INTERACTION

finding things

lost in hyperspace
structure and navigation
history and bookmarks
indices, directories and search

Navigation icons: Home, Back, Forward, Stop, Refresh, Search

HUMAN-COMPUTER INTERACTION

lost in hyperspace

- non-linear structure
 - very powerful ...
 - but potentially confusing
- two aspects of lostness
 - cognition and content
 - fragmentary information – no integration ... confusion
 - navigation and structure
 - hyperlinks move across structure – where am I?
- no easy solutions
 - but good design helps!

Navigation icons: Home, Back, Forward, Stop, Refresh, Search

HUMAN-COMPUTER INTERACTION

designing structure

- ideas for structure
 - task analysis to for activities and processes
 - existing paper or organisational structures
- going non-linear
 - paper and organisation single structure
 - hypertext – multiple structures
 - problems with common material, inconsistencies etc.
 - clarity of cross structure links v. important
- scent
 - do hot spots for links make it clear where they are going to??

Navigation icons: Home, Back, Forward, Stop, Refresh, Print, Search

HUMAN-COMPUTER INTERACTION

making navigation easier

- maps
 - give an overview of the structure
 - show current location – you are here!
- recommended routes
 - guided tour or bus tour metaphor
 - linear path through non-linear structure
- levels of access
 - summary then progressive depth
- supporting printing!
 - needs linearised content, links back to source

Navigation icons: Home, Back, Forward, Stop, Refresh, Print, Search

HUMAN-COMPUTER INTERACTION

history, bookmarks, etc.

- revisiting
 - ‘hub and spoke’ access – click-back-click-back
 - lots of revisiting of pages
 - ‘back’ is 30% of all browser navigation
 - but multi-step back and history used less
 - bookmarks and favourites for longer term revisiting
- deep links
 - bookmarks and external links – into heart of site
 - are pages self explanatory? what site? where in it?
 - e.g. breadcrumbs for context
- frames
 - difficult to bookmark, search and link to
 - but some good reasons for use (see [/e3/online/frames/](#))

Navigation icons: Home, Back, Forward, Stop, Refresh, Print, Search

HUMAN-COMPUTER INTERACTION

indices, directories and search

- index
 - often found in help, documentation, ... even books
 - selective: not an exhaustive list of words used
- directories
 - on web index would be huge! so hand chosen sites
 - e.g. [open directory project](#), [Yahoo!](#)
- web search engines
 - ‘crawl’ the web following links from page to page
 - build full word index (but ignore common ‘stop’ words)
 - looks up in index when you enter keywords to find pages

HUMAN-COMPUTER INTERACTION

complex search

- too many pages for single word ... need to be more selective:
- Boolean search
 - combine words with logic: e.g. 'engine AND NOT car'
- link structure
 - Google uses richness of in and out links to rank pages
- recommender systems
 - use other people's choices to guide other people
- being search engine friendly
 - add 'Meta' tags, relevant title, keywords, description
 - hard to index generated pages ... the hidden web

HUMAN-COMPUTER INTERACTION

finding research literature

- special portals and search sites:
 - e.g. citeseer <citeseer.nj.nec.com>
 - searches web for papers
 - scans the papers for bibliography
 - uses this to build up citation index

The diagram consists of two parts. On the left, labeled 'bibliography backwards in time', a central blue box has four red arrows pointing to four yellow boxes, representing older papers cited by a newer paper. On the right, labeled 'citation forwards in time', a central blue box has four red arrows pointing to four pink boxes, representing newer papers that cite the central paper.

HUMAN-COMPUTER INTERACTION

web technology and issues

protocols and browsers
 web servers and clients
 networking

HUMAN-COMPUTER INTERACTION

web basics ...

- the 'web' – protocols and standard
 - HTTP – to carry information over the internet
 - HTML, XML and graphics formats for content
 - browsers to view the results ... plus plug-ins
- changing use
 - initially research (CERN - high energy physics)
 - now corporate, government, commerce and entertainment, advertising, community
- challenges
 - lost in hyperspace, information overload

HUMAN-COMPUTER INTERACTION

web servers and clients

- the web is distributed
 - different machines far across the world
 - pages stored on servers
 - browsers (the clients) ask for pages
 - sent to and fro across the internet

```

graph LR
    subgraph Client [web client (browser)]
        U[User]
        B[Browser]
    end
    subgraph Server [web server (stores pages)]
        S[Server]
    end
    U -- 1 user clicks link --> B
    B -- 2 browser sends request "GET /a3/author.html HTTP/1.1" --> S
    S -- 3 server finds page --> S
    S -- 4 server sends page back --> B
    B -- 5 browser displays it --> U
    B <--> |communicate with HTTP| S
  
```

HUMAN-COMPUTER INTERACTION

network issues - timing

- QoS (quality of service)
 - bandwidth
 - how much information per second
 - latency
 - how long it takes (delay)
 - jitter
 - how consistent is the delay
 - reliability
 - some messages are lost
 - ... need to be resent ... increases jitter
 - connection set-up
 - need to 'handshake' to start

bandwidth, latency and jitter

design implications

- bandwidth ⇒ think about download time
 - e.g. 100K image: 1 sec – broadband, 18 secs – 56K modem
 - save graphics in appropriate format and size
 - reuse the same graphics
 - in the browser cache after first load
- connection time
 - one big file may be better than several small ones
 - beware of 'fit on one screen' rule – scrolling is fast!
 - think before breaking big graphic into bits
- latency ⇒ think about feedback

feedback and feedthrough

- network delays too slow! so ...
- feedback – think:
 - immediate local feedback – something has happened
 - slower semantic feedback – what has happened
- feedthrough between users:
 - hard – cannot avoid network

HUMAN-COMPUTER INTERACTION

WAP - web on the phone

- very **small screen**
 - scrolling painful => small 'pages'
 - GSM connection **slow** => big chunks
- WML (wireless mark-up language)
 - content delivered in 'stacks' of 'cards'
 - cards are the 'pages' the user views
 - but navigation within the stack fast

N.B. larger screens and faster connections mean WML giving way to small HTML pages

server sends whole stack using WAP protocol

navigation within stack fast

server stores 'stacks' of cards

email: book says stacks of notes, cards is correct

HUMAN-COMPUTER INTERACTION



static web content

medium and message
text, graphics, movies and sound

HUMAN-COMPUTER INTERACTION



the message and the medium

- "content is king"
 - the catch phrase of dot.com era ... but widely ignored
- the message ... content should be
 - appropriate to the audience, timely, reliable,
 - generally worth reading !
- the medium ... page and site design
 - good design - essential to attract readers ... but won't hide bad material!
 - bad design - may mean good material never seen
 - printable!






text



- text style
 - generic styles universal: serif, sans, `fixed`, **bold**, *italic*
 - specific fonts too, but vary between platforms
 - cascading style sheets (CSS) for fine control
... but beware older browsers and fixed font sizes
 - colour ... often abused!
- positioning
 - easy .. left, right justified or centred
 - precise positioning with DHTML ... but beware platforms ...
 - screen size
- mathematics ... needs special fonts, layout, ... arghhhh

graphics


- use with care ...
 - N.B. file size and download time ...  this image = 1000 words of text
 - affected by size, number of colours, file format
 - backgrounds ... often add little, hard to read text
- speeding it up
 - caching – reuse same graphics
 - progressive formats:
 - image appears in low res and gets clearer



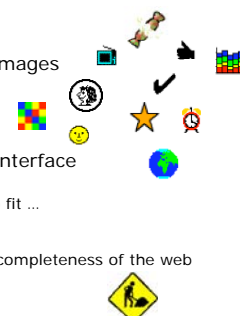
graphics (ctd)

- formats
 - JPEG – for photos
 - higher compression but 'lossy'
 - get 'artefacts'
 - GIF for sharp edges
 - lossless compression
 - PNG supported by current web browsers
- and action
 - animated gifs for simple animations
 - image maps for images you can click on



icons

- on the web just small images
 - for bullets, decoration
 - or to link to other pages
 - lots available!
- design ... just like any interface
 - need to be understood
 - designed as collection to fit ...
- under construction
 - a sign of the inherent incompleteness of the web
 - or just plain lazy ??

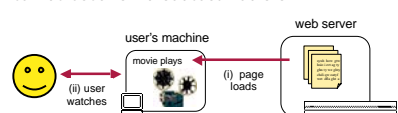



web colour

- how many colours?
 - PC monitors – millions – 24 bits per pixel ... but the 'same' colour may look very different
 - N.B. usually only 72–96 dpi
 - older computers, PDAs, phones ...
 - perhaps only 16 bits or 8 bits per pixel ... 256 colours
 - or even greyscale
- colour palettes
 - choose useful 256 colours
 - different choices, but Netscape 'web safe' 216 are common
 - each GIF image has its own palette – use for fast download

movies and sound


- problems
 - size and download... like graphics but worse!
 - may need special plug-ins
 - audio not so bad, some compact formats (MIDI)
- streaming video
 - play while downloading
 - can be used for 'broadcast' radio or TV






dynamic web content

what happens where
 technology and security
 local interaction, search
 remote & batch generation
 dynamic content





the active web

- early days of the web
 - static pages ... mostly text
 - some gateways (ftp, gopher)
 - usability ... easy - one simple model (except frames break the model!)
- dynamic content
 - what is the model/metaphor ???
 - passive pages or active interface
 - each leads to different user understanding
 - no easy answers!





what happens where?

- architectural design is about what happens where
- this affects:
 - feedback
 - seeing results of one's own actions
 - feedthrough
 - seeing effects of other people's actions
 - also affects complexity of implementation and hence maintenance

user view



- what changes?
 - media stream, presentation, content
- by whom?
 - automatic, site author, user
 - other users - feedthrough
- how often?
 - pace of change: days, months, seconds

technology

where does it happen

- client
 - applets , Flash, JavaScript & DHTML
- server
 - CGI scripts, Java servlets , JSP, ASP, PHP, etc,
- another machine
 - author's machine, database server, proxy
- people
 - socio-technical solutions

security

- for computation
 - code and data at same place!
- problem
 - data - needs to be secure
 - web-server - least secure machine
 - client machine even worse

... and networks!

HUMAN-COMPUTER INTERACTION

local interaction (at client)

```

    graph LR
      User((User)) -- "(ii) user interacts locally" --> Machine[User's machine]
      Machine -- "script / Java running in client" --> Machine
      Server[Web server] -- "(i) page loads once" --> Machine
      Machine --> Server
  
```

- fixed content
- use Java applets, Flash, JavaScript+DHTML
- pros: rapid feedback
- cons: only local, no feedthrough
- after interaction ... what does 'back' do ??

HUMAN-COMPUTER INTERACTION

examples

two horse races

coin race uses JavaScript

dancing histograms are a Java applet

interactive stacked histograms

HUMAN-COMPUTER INTERACTION

search

```

    graph LR
      User((User)) -- "(i) user fills field in form" --> Machine[User's machine]
      Machine -- "web page with text field for search words" --> Machine
      Machine -- "(ii) search results returned" --> Server[Web server]
      Server -- "CGI script looks up words in index" --> Server
      Index[index file pre-computed] --> Server
  
```

- create indices off-line
- fast lookup when needed

see <http://www.hcibook.com/e3/search/>

HUMAN-COMPUTER INTERACTION

automatic generation

- dilemma;
 - hand crafting ... leads to web stasis!!
 - so need database driven sites
- early days ad hoc, now many tools
- options:
 - client-end applet or Flash access remote DB
 - server-end CGI driven by web forms (limited UI)
- hybrid solutions
 - CGI generated pages can contain JavaScript etc.
 - JavaScript can 'write' web pages on the fly!

HUMAN-COMPUTER INTERACTION

Java applet & JDBC

```

    graph LR
      User((User's machine)) <--> Applet[Java applet]
      Applet -- JDBC accesses database --> DB[(Database)]
      DB --- Server[Web server]
  
```

- pros: interactive DB access
- cons: bandwidth, security

HUMAN-COMPUTER INTERACTION

CGI script accesses database

```

    graph LR
      User((User's machine)) -- "(i) request to server" --> Server[Web server]
      subgraph Server
        CGI[CGI script]
        DB[(Database)]
        Pages[generated pages]
        CGI -- "(ii) CGI script accesses database using SQL/JDBC" --> DB
        DB -- "(iii) server returns generated pages" --> Pages
      end
      Pages --> User
  
```

- pros: up-to-date, use existing DB
- cons: not proxy/index friendly

batch generation

- for slow varying data
 - update local database
 - periodically generate pages and upload
- many technologies
 - C, Java, HyperCard, Visual Basic

```

Set db = openDatabase("C:\test.mdb");
sql = "select Name, Address from Personnel";
Set query = db.OpenRecordset(sql)
Open "out.html" For Output As #1

Print #1, "<h1>Address List</h1>"
query.MoveFirst
While Not query.EOF
Print #1, "<cp>" & query("Name") & " * " & query("Address")
query.MoveNext
Wend

Close #1
query.Close
  
```

batch generation of web pages

```

graph LR
    A["(i) pages generated off-line from database"] -- "(ii) pages copied to web server via ftp" --> B["web server"]
    B -- "(iii) server returns generated pages" --> C["user's machine"]
  
```

- pros: indexable, secure
- cons: slower turnaround

dynamic content

- really 'active' web pages ...
 - data updated as well as presented on the web
- presentation
 - any of the previous means: CGI, applet-JDBC
- update
 - web form/interface -> server script -> update db
 - e.g. book theatre seats
- issues
 - authentication and security
 - multiple transactions due to 'back' button
 - right pace/control – do we want human in the loop?

