

HUMAN-COMPUTER INTERACTION THIRD EDITION DIX FINLAY ABOARD BEALE

chapter 21

hypertext, multimedia and the world-wide web

extract for MSc/MRes AISD

dynamic web content

HUMAN-COMPUTER INTERACTION

dynamic web content

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

HUMAN-COMPUTER INTERACTION

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!

HUMAN-COMPUTER INTERACTION

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

HUMAN-COMPUTER INTERACTION

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

HUMAN-COMPUTER INTERACTION

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!

local interaction (at client)

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

examples

dancing histograms are a Java applet

coin race uses JavaScript

search

- create indices off-line
- fast lookup when needed

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

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!

Java applet & JDBC

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

CGI script accesses database

(i) request to server
(ii) CGI script accesses database using SQL/JDBC
(iii) server returns generated pages

- 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

batch generation of web pages

(i) pages generated off-line from database
(ii) pages copied to web server via ftp
(iii) server returns generated pages

- 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?

n-tier architecture

HTML, JSP, XML, JEB, JDBC, database

- one or more intermediate layers
- 'business logic' in layers
- web standard components and protocols

local interaction - remote data

AJAX, Flash, web2.0 ...

(i) user interacts locally
(ii) page loads once
(iii) requests to server
(iv) CGI script generates XML
(v) server returns generated pages

- pros: very rich interaction
- cons: coding 'pull' only, asynchronous ... back, bookmarks, search, ???