

Web Design

Pressman, Chapter 12 (parts)

Instructor: Peter Baumann

email: p.baumann@jacobs-university.de

tel: -3178

office: room 88, Research 1



Web Design

“artistic ideal: express yourself

vs.

engineering ideal: solve problem for a customer”

-- Jakob Nielsen

Design & WebApp Quality

- Security
 - Rebuff external attacks
 - Exclude unauthorized access
 - Ensure the privacy of users/customers
- Availability
 - time that WebApp is available
- Scalability
 - significant variation in user / transaction volume
- Time to Market

WebApp Interface Design

- Where am I?
 - indication of WebApp accessed
 - inform user of location in content hierarchy
- What can I do now? ...help user understand current options
 - what functions are available?
 - what links are live?
 - what content is relevant?
- Where have I been, where am I going? ...facilitate navigation
 - Provide “map”: where user has been, what paths available
 - Easy to understand

Interface Design Principles-I

- **Anticipation** — anticipate user's next move
- **Communication** — communicate status of any activity initiated by user
- **Consistency** — use of navigation controls, menus, icons, and aesthetics (e.g., color, shape, layout)
- **Controlled autonomy** — facilitate user movement, but enforce navigation conventions
- **Efficiency** — optimize user's work efficiency, not efficiency of Web engineer or client-server environment

Aesthetic Design

- Don't be afraid of white space
- Emphasize content
- Organize layout from top-left to bottom-right
- Group navigation, content, and function geographically within page
- Don't extend your real estate with the scrolling bar
- Consider resolution and browser window size when designing layout

Web Design: Home Page Variants

- „front door“ home page approach
 - nice & appealing impression first, information area later
- „information rich“ home page approach
 - minimal mouse clicks
- Mixed approaches



New Style

[www.scala-lang.org/]



[DOCUMENTATION](#) [DOWNLOAD](#) [COMMUNITY](#) [CONTRIBUTE](#) 

Object-Oriented Meets Functional

Have the best of both worlds. Construct elegant class hierarchies for maximum code reuse and extensibility, implement their behavior using higher-order functions. Or anything in-between.

[LEARN MORE](#)

DOWNLOAD

Getting Started


 *Milestones, nightlies,
etc.*
 *All Previous Releases*



Scala
2.10.4

API DOCS

API: Current | Nightly

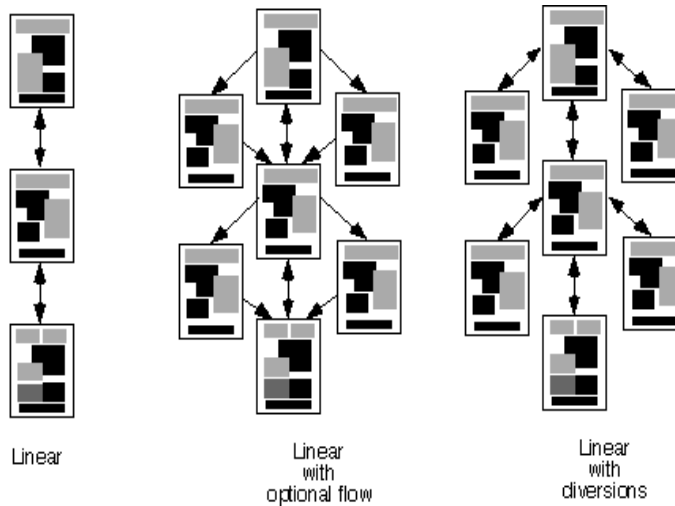
 *Scala Documentation*
 *Language Specification*

Content Design

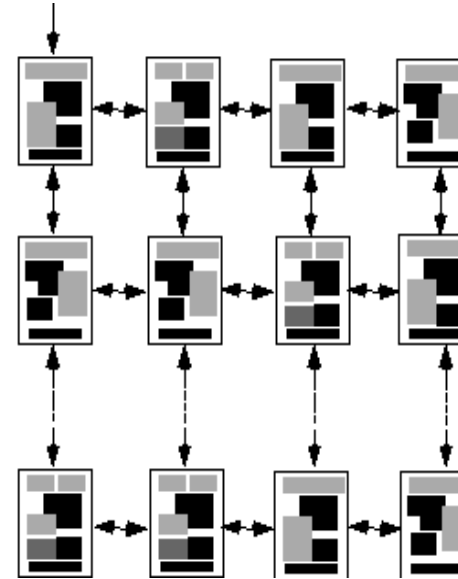
- design representation for content objects
 - content object ~ data object for conventional software
- mechanisms required to instantiate relationships
 - Cf. relationship: analysis classes + design components
- content-specific & implementation-specific information

Content Architecture

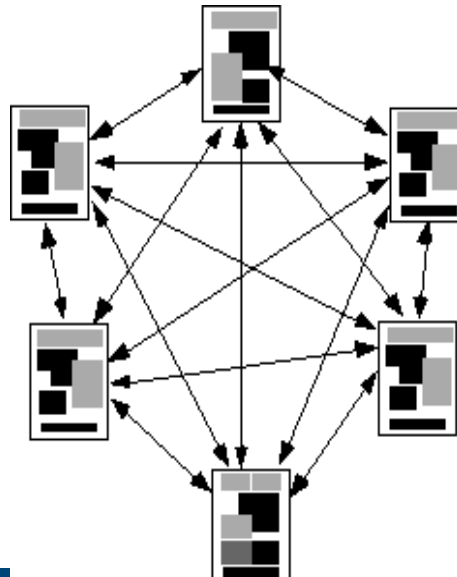
Linear
structure



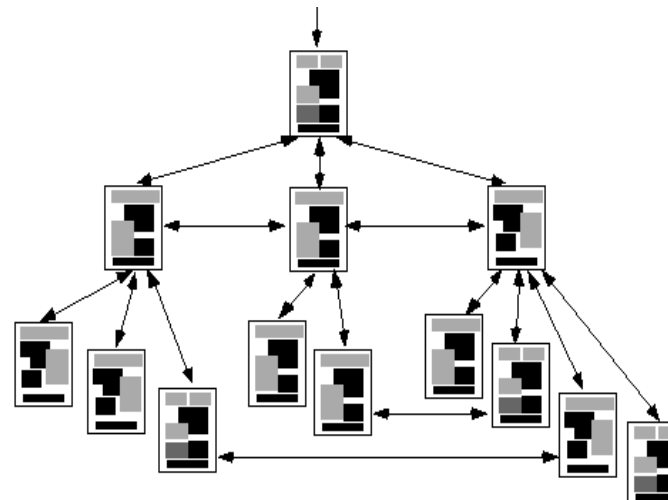
Grid
structure



Network
structure



Hierarchical
structure



Quality Dimensions for End-Users I

■ Time

- Last change when?
- how to highlight changed parts?

■ Structural

- parts hold together
- inside & outside links working?
- images working?
- parts not connected?

■ Content

- critical pages: content matches expectation?
- key phrases continually in highly-changeable pages?
- critical pages maintain quality content from version to version?

Quality Dimensions for End-Users II

■ Accuracy and Consistency

- Are today's copies of the pages downloaded the same as yesterday's? Close enough?
- data accurate enough? How do you know?

■ Response Time and Latency

- Does the Web site server respond to a browser request within certain parameters?
- E-commerce context: end-to-end-response time after SUBMIT?
- parts too slow for users?

■ Performance

- Browser-Web-Web site-Web-Browser connection quick enough?
- variance by time of day, by load, by usage?
- adequate for E-commerce applications?

Hypermedia Design Patterns-I

- Architectural patterns
 - design of content & WebApp architecture
 - many available (*Java Blueprints*: java.sun.com/blueprints/)
- Component construction patterns
 - combining WebApp components
(e.g., content objects, functions)
- Navigation patterns
 - navigation links, overall navigation flow, ...

Hypermedia Design Patterns-II

- Presentation patterns
 - user interface control functions
 - relationship between interface action & affected content objects
 - effective content hierarchies
- Behavior/user interaction patterns
 - informs user of consequences of a specific action
 - expand content, based on usage context & user desires
 - describe link destination
 - status of on-going interaction
 - ...

Patterns Repositories

- Hypermedia Design Patterns Repository
 - <http://www.designpattern.lu.unisi.ch/>
- Improving Web Information Systems with Navigational Patterns
 - <http://www8.org/w8-papers/5b-hypertext-media/improving/improving.html>
- An HTML 2.0 Pattern Language
 - <http://www.anamorph.com/docs/patterns/default.html>
- Common Ground - A Pattern Language for HCI Design
 - http://www.mit.edu/~jtidwell/interaction_patterns.html
- Indexing Pattern Language
 - <http://www.cs.brown.edu/~rms/InformationStructures/Indexing/Overview.html>