
Index

Symbols and Numbers

- { } (braces), constraints and, 236
- < > (angle brackets), XML tags and, 59
- <A> (anchor tag). *See* Anchor tag (<A>)
- <APPLET>. *See* <OBJECT>
- <AREA>, 325
- <BODY>
 - attributes, 319
 - client pages and, 350
 - events, 42
 - as HTML element, 318–319
- «boundary», 222–223
- «build», 239
- «business logic», 194, 210
- <BUTTON>, 335–336
- «client page». *See* Client pages
- «client script object», 273
- «control», 222
- «dynamic page», 348
- «entity», 222
- «extends», 179
- <FORM>
 - attributes, 353–355
 - «form» and, 238
 - HTML forms, 19, 327–329, 351–355
- «form», 238
- «forward», 239, 373
- <FRAME>. *See* <FRAMESET>
- <FRAMESET>, 265–270
 - client pages and, 268
 - conceptual model of, 267–268
 - controversy over use of, 22
 - difficulty of modeling, 265
 - function of, 21, 267
 - as HTML element, 343–345
 - i-frames, 269–270
 - mapping to «frameset», 356
 - mapping to «target», 355–356
 - parameters, 21
 - stereotypes for modeling, 266–267
 - structure of, 265–266
 - targets and, 266–267, 269
- «frameset»
 - mapping to <FRAMESET>, 344
 - UML classes, 356
- <HEAD>, 350
- <HTML> elements
 - <A>, 17–18, 320–325
 - <APPLET>. *See* <OBJECT>
 - <AREA>, 325
 - <BODY>, 318–319
 - <BUTTON>, 335–336
 - <FORM>, 18–21, 327–329

<HTML> elements *continued*

<FRAME>. *See* <FRAMESET>
 <FRAMESET>, 343–345
 <HTML>, 318–319
 <IFRAME>, 346–347
 <INPUT>, 329–333
 <LINK>, 325–327
 <OBJECT>, 336–338
 <PARAM>, 338–339
 <SCRIPT>, 339–343
 <SELECT>, 333–334
 <TEXTAREA>, 334–335
 <TITLE>, 319
 «HTML form»
 logical view and, 238
 mapping <BUTTON> to, 335–336
 mapping <FORM> to, 327
 mapping <INPUT> to, 330–333
 mapping <SELECT> to, 333–334
 mapping <TEXTAREA> to, 334–335
 modeling, 240
 as UML class, 351–355
 «HTTP Resource». *See* HTTP resources
 <IFRAME>, 269–270, 346–347, 360–361
 «iframe», 269–270, 360–361
 «include», 239, 361–363, 373
 «includes», 177, 182
 <INPUT>, 20, 200, 329–333
 «Input form», 200, 210, 357
 «JSP Tag Library», 375–376
 «JSP tag», 282–284
 «JSP»
 glossary application and, 300, 304
 modeling JSP files with, 365–369
 UML JSP profile and, 369–370, 373–375
 <LINK>
 as HTML elements, 325–327
 «link» associations and, 358
 style sheets and, 349
 «link»
 associations, 239, 358–360
 href parameter and, 240
 mapping HTML to UML, 321
 mapping to <LINK>, 325
 screen compartments and, 243

«managed», 210
 <MAP> element, 325
 <META>, 360
 <OBJECT>, 336–338, 360
 «object parameters», 338–339
 «object», 239, 337–338, 360
 associations, 239, 360
 mapping <OBJECT> to, 337–338
 <PARAM>, 338–339
 «physical root», 244–246, 317, 348
 «redirect», 239, 360, 373
 «screen compartment», 204, 243, 357
 «screen»
 navigational maps and, 193
 notations used with, 210
 as UML classes, 356–357
 UX model and, 210
 <SCRIPT>, 339–343, 356
 «script library»
 client-side scripting, 271, 343
 as UML classes, 356
 as UML component, 349
 «script», 343
 <SELECT>, 19, 333–334
 «server page». *See* Server pages
 «style sheet», 349, 357
 «submit», 239, 360
 «Target», 355–356
 «Targeted Link» *n*-ary association, 355
 <TEXTAREA>, 19–20, 334–335
 <TITLE>, 319
 «trace dependencies», 138
 «URL parameters», 358
 «use bean», 373
 «virtual root», 317, 348

A

<A> (anchor tag). *See* Anchor tag (<A>)
 Abstract Windowing Toolkit (AWT), 43
 Abstraction, 4–5
 Access control, 89–90
 Action parameter, <FORM> tag, 19
 Active Server Pages (ASP)
 as example of Web architecture, 7

- as scripted approach, 28–29
- WAE profile and, 363
- ActiveX
 - based on COM model, 45
 - client-side security and, 83–84
 - controls, 149
 - extending client-side functionality, 45
- ActiveX/COM, 45–46
- Activity diagrams
 - controllers, 422
 - mapping analysis classes to use case, 226, 229
 - storyboards, 207–208
 - value of, 179
- Actors
 - defined, 173
 - interactions with system, 173–174
 - relationships with use cases, 177
 - UML representation of, 176
- Address collection interface, HTTP, 50
- Administrator-level access, 75
- ADS (Architecture Description Standard), 134
- Analysis, 117–120, 215–232
 - associations and traceabilities, 121
 - behavioral elements, 225–229
 - classes, 120, 222
 - dependence on requirements set, 120
 - design and, 117, 215–216, 230
 - focus of, 216
 - illustration of, 119
 - iteration and, 216–217
 - mapping to UX model, 230–231
 - master template pattern, 427
 - models produced by, 216
 - organizing based on things rather than
 - actions, 219
 - package hierarchy and, 217–218
 - qualities of packages in, 220
 - relationships, 120
 - software architecture document and, 120, 231
 - structural elements, 222–225
 - structure of, 217–221
 - techniques, 221–222
 - top-level model, 218–221
- Anchor tag (<A>), 17–18
 - associations and, 358
 - functioning as anchor or as link, 320
 - href parameter, 17–18
 - mapping to UML, 320–325
 - target parameter, 18
- Animation, menus and buttons, 42
- Anonymous users
 - attacks by, 74
 - prevalence of, 75
- <APPLET>. *See* <OBJECT>
- Applets. *See* Java applets
- Application servers, thin Web clients, 147
- Architecture, 133–160
 - candidates for, 141–142
 - DCOM, 56
 - design elements, 378–379
 - design viewpoint, 136
 - development process and, 6
 - evolution of Web architectures, 7
 - Number Store application, 380
 - overview of, 133–135
 - presentation tier, 144–145
 - prototypes, 142–143
 - realization viewpoint, 136–137
 - requirements viewpoint, 135–136
 - requirements and, 116
 - software architecture document, 120
 - test viewpoint, 137–138
 - thick Web client, 149–155
 - thin Web client, 145–149
 - use case and, 139–141
 - viewpoint mappings, 138
 - WAE/WAE2 and, 7–8
 - Web application structural patterns, 143–144
 - Web Delivery, 155–159
- Architecture Description Standard (ADS), 134
- <AREA>, 325
- Artifacts, 108–111
 - analysis and, 117–120
 - architectural viewpoints and, 135
 - artifact set, 111–112
 - deployment set, 128–130
 - design set, 120–123
 - as development stepping stones, 108
 - domain set, 113–114
 - implementation set, 123–127

Artifacts continued

- models as, 109–110
 - navigational paths, 192–193
 - overview of, 188–189
 - project management set, 111–113
 - reasons for creating, 109
 - requirements set, 114–117
 - screens, 189–192
 - storyboards, 192
 - test set, 127–128
 - vision document and requirements and, 161
- ASP. *See* Active Server Pages (ASP)
- Associations. *See also* Relationships
- navigation path modeling, 199
 - UML. *See* UML associations
 - user input class and, 200–201
 - WAE, 239
- Assumptions, use cases, 174
- Attackers
- administrator-level access and, 75
 - anonymous vs. in-house, 74
 - detecting, 74
 - exploitation of bugs and improper configuration, 77–78
 - monitoring network traffic, 87
 - use of network sniffers by, 76–77
- Attributes
- <FORM>, 353–355
 - HTML core, 319
 - requirements, 166
- Audience, architectural design and, 133
- Auditing, security, 89
- Authentication
- levels of, 74
 - passwords and, 75, 85–86
 - security models and, 89
- Authors, use cases and, 174
- AWT (Abstract Windowing Toolkit), 43

B

- Behaviors (operations)
 - analysis and, 225–229
 - behavioral diagrams, 225
 - screens, 196, 198

- Berners-Lee, Tim, 9
- Best practices, security policy, 88–89
- Binding information, UDDI Web services, 65
- Black box testing, 127
- Blue Screen of Death (BSOD), 83
- <BODY>
 - attributes, 319
 - client pages and, 350
 - events, 42
 - as HTML element, 318–319
- Booch, Grady, 3, 97
- <boundary>, 222–223
- Browsers, 14–17
 - compatibility with JavaScript, 36–37
 - cookies and, 23
 - directory browsing, 78
 - HTML and, 14–17
 - HTTP and, 10
 - multiple, 261
 - thick Web clients and, 152
 - thin Web clients and, 146
 - Web application design and, 261–262
- BSOD (Blue Screen of Death), 83
- Bugs
 - exploitation by attackers, 77–78
 - JavaScript, 80
 - security risk and, 74
- <build>, 239
- Business information, UDDI Web services, 65
- Business logic
 - role of clients in execution of, 31
 - system models and, 4
 - Web applications and, 22–23
- <<business logic>>, 194, 210
- Business objects, 246–247
- Business requirements, 116
- <BUTTON>, 335–336
- Buttons, animating, 42

C

- Carriage return/line-feed (CR/LF), 10
- CAs (Certificate Authorities), 86, 87
- Cascading style sheets (CSS), 57–58
- CASE tools, 5, 288

- CERN (European Laboratory for Particle Physics), 9
- Certificate Authorities (CAs), 86, 87
- Certificates, 86
- cgi-bin, 26
- CGI (Common Gateway Interface), 26–27
- Change management, 112–113, 217
- Class diagrams
 - client pages, 255
 - glossary application, 300
 - master template pattern, 425
 - search functionality and, 278
 - UML models and, 5
 - Web page control mechanism, 290
 - Web page design and, 252–253
- Class identifiers (CLSIDs), 55–56
- Class-Responsibility-Collaboration (CRC) cards, 221
- Class stereotypes, 210–212
- Classes, UML. *See* UML classes
- Client nodes, 149
- «client page». *See* Client pages
- Client pages
 - «client page» UML class, 350–351
 - dynamic, 253
 - HTML framesets and, 268
 - logical view and, 237–238
 - mapping to HTML elements, 318–319
 - modeling, 240–242
- «client script object», 273
- Client/server architecture, 141
- Client-side programming, 42
- Client-side risks, 79–85
 - ActiveX, 83–84
 - cookies and, 79–80
 - Java applets, 81–83
 - JavaScript, 80
 - plug-ins and MIME types, 84–85
- Client-side scripting
 - design and, 253–257
 - extending thin clients, 149
 - script libraries, 271–272
 - script objects, 273
- Client-side validation, 31–32
- Client state management, 23–26
 - cookies, 23–25
 - need for, 23
 - sessions, 25–26
- Client tier, 392, 431
- Clients. *See also* Dynamic clients
 - browsers, 146
 - dynamic. *See* Dynamic clients
 - environments, 164–167
 - security risk and, 72
 - thick Web. *See* Thick Web clients
 - thin Web. *See* Thin Web clients
- CLSIDs (class identifiers), 55–56
- CMS. *See* Content management systems (CMS)
- Cockburn, Alistair, 173
- Code. *See also* Implementation
 - dangers of coding prior to creating artifacts, 108
 - mapping design into, 287
- Collaboration diagrams, 204–206, 225–227
- cols parameter, HTML frameset, 21
- COM. *See* Component Object Model (COM)
- Common Gateway Interface (CGI), 26–27
- Communication
 - artifacts and, 109
 - realization viewpoint and, 137
 - role of models in, 5
 - thick Web clients and, 152
- Companies, tailoring development to, 98–99
- Compiled-module approach, 27–28
- Complexity, managing, 50, 219, 377
- Component diagrams, 279, 405–406
- Component Object Model (COM)
 - ActiveX based on, 45, 83
 - DCOM as extension of, 55
 - interoperability of components, 63
 - JavaBeans and, 63
- Component packages. *See* UML component packages
- Component view, 242–246
 - dynamic page extensions, 244
 - glossary application, 441–443
 - Number Store application, 401–406
 - physical root extensions, 244–246
 - static page extensions, 242–244

- Components. *See* UML components
 - Computer-aided software engineering (CASE)
 - tools, 5, 288
 - Configuration, security risk and, 74, 77–78
 - Connectionless protocols, fault tolerance and, 13
 - Constantine, Larry, 173
 - Constraints, UML extensions and, 261–262
 - Contained classes, user input, 200–201
 - Content management systems (CMS)
 - business logic and, 191–192
 - classes, 194, 210
 - design and, 260–261
 - dynamic content, 190–191, 194
 - screens, 190–192, 194–196
 - Content, static vs. dynamic, 189–190. *See also*
 - Dynamic content
 - «control», 222
 - Controllers. *See also* Model View Controller (MVC) pattern
 - activity diagrams, 422
 - adding, 413
 - analysis diagrams, 417
 - analysis model classes, 415–417
 - analysis model collaborations, 418
 - implementation and, 290
 - Number Store application, 395–398
 - removing, 414
 - use case view, 411–414
 - ControlTemplate. *jsp* source, 295, 395
 - Cookies, 23–25
 - function of, 23
 - limitations of, 25
 - managing session state, 25–26, 140
 - parameters, 23–24
 - security and, 79–80
 - Set-Cookie header, 24
 - Coordinators, controllers, 415–416
 - CORBA IIOP, 52
 - Countermeasures, security, 89, 91
 - Country-specific domains, 12
 - CR/LF (carriage return/line-feed), 10
 - Crackers. *See* Attackers
 - CRC (Class-Responsibility-Collaboration) cards, 221
 - Cryptology. *See* Encryption
 - CSSs (cascading style sheets), 57–58
 - The Cuckoo's Egg* (Stoll), 74
- ## D
- Data structures, UDDI, 65–66
 - Data tier, 401, 438–441
 - Database servers, 147
 - DCE (distributed computing environment), 56
 - DCOM. *See* Distributed COM (DCOM)
 - Decentralized processing, Web services, 63
 - Delivery. *See* Web delivery
 - Deployment, 128–130
 - associations and traceabilities, 129
 - illustration of, 130
 - plan, 128–129
 - requirements, 166
 - Design, 120–123, 233–263
 - analysis and, 117, 215–216, 230
 - associations and traceabilities, 124
 - characteristics of designers, 122–123
 - client-side scripting and, 253–257
 - component view, 242–246
 - glossary application, 431–434
 - guidelines, 261–262
 - identification of Web pages, 247–253
 - illustration of, 122
 - integrating with CMS, 260–261
 - logical view, 237–242
 - mapping to code, 287
 - mapping to UX model, 257–260
 - overview of, 136, 233–236
 - process view, 120
 - subsystem details, 120
 - thick Web clients, 152–154, 246–247
 - thin Web clients, 146–149
 - UML extensions and, 236–237
 - Web applications, 246
 - Web delivery applications, 155–157, 247
 - Design, advanced, 265–286
 - client-side scripting, 271–273
 - HTML frames, 265–270
 - HTTP resources, 273–279
 - JSP custom tags, 281–285

- Design, Number Store, 385, 392–401
 - architecture, 380
 - client tier, 392
 - controllers, 395–398
 - data tier, 401
 - entity tier, 398–401
 - page requests, 394–395
 - presentation tier, 392–394
 - top-level model, 385
 - UX mapping, 398
- Design team, 120, 122–123
- Designers, characteristics of, 122–123
- Develop Software use case, 101–105. *See also*
 - Software development process
 - activity diagram for, 102
 - artifact version management, 101
 - business analysis, 101
 - domain model, 103
 - problem analysis, 103
 - project plan, 103
 - risk assessment and, 104
 - vision document, 103
- Developers, infrastructure independence and, 54
- Development environment, 137, 248
- Development process. *See* Software development process
- DHTML (Dynamic HTML), 36
- Diagrams
 - activity, 179, 207–208, 226, 229
 - behavioral, 225
 - class, 252–253
 - collaboration, 204–206, 225–226
 - interaction, 177
 - navigation, 192–193
 - package, 184
 - participant, 208–209
 - sequence, 179, 181, 225–226, 252–253
 - use case, 177–179
- Digitally signed files, 82
- Directory browsing, 78
- Distributed COM (DCOM), 55–57
 - architecture of, 56
 - class identifiers (CLSIDs) and, 55–56
 - disadvantages of, 57
 - object deployment with, 56–57
 - overview of, 55
 - Web delivery and, 156
- Distributed computing environment (DCE), 56
- Distributed objects. *See also* Distributed COM (DCOM); Remote Method Invocation (RMI)
 - advantages of, 50–51
 - examples of, 51–52
 - Web architecture and, 49–50
- DNS (domain name server), 11
- !DOCTYPE keyword, 60
- Document elements, WSDL, 66–67
- Document identification, URLs, 10–11
- Document Object Model (DOM), 33–36
 - goals of, 34–35
 - illustration of, 34
 - JavaScript and, 37–38
 - relationship of components in, 33–34
 - vendor and platform neutrality, 35
- Document objects, JavaScript, 37
- Document type definitions (DTDs), 60–61, 62
- doGet(), 290
- DOM. *See* Document Object Model
- Domain name server (DNS), 11
- Domains, 113–114
 - as artifact of domain set, 113
 - domain model and, 113
 - glossary and, 113–114
 - names, 11–12
 - in Number Store application, 380–381
 - requirements viewpoint and, 135
 - UML and, 103
- doPost(), 290
- Dot notation, JavaScript, 37
- DTDs. *See* Document type definitions
- Dynamic clients, 31–47
 - ActiveX/COM, 45–46
 - DOM and, 33–36
 - Java applets and, 43–44
 - JavaScript events, 40–42
 - JavaScript objects, 37–40
 - overview of, 31–33
 - scripting technologies and, 36–37
- Dynamic content
 - content management and, 190–191
 - contrasted with static content, 189–190

- Dynamic content *continued*
 - input forms and, 202
 - UX model and, 193–194
 - Dynamic HTML (DHTML), 36
 - «dynamic page» class, 348
 - Dynamic pages
 - component view, 244
 - «dynamic page» class, 348
 - requests, 150
 - thin Web client components, 147
 - Dynamic relationships, 177
- E**
- EJB (Enterprise JavaBeans), 145, 401
 - Elements
 - HTML. *See* <HTML> elements
 - XML, 58–59
 - Enabling technologies, 26–29
 - CGI, 26–27
 - client-side, 33
 - compiled-module approach, 27–28
 - ISAPI and NSAPI, 27
 - scripted approach, 28–29
 - Encryption, 86–88
 - disadvantages of, 88
 - key pairs and, 87
 - protecting network traffic, 76–77, 86–87
 - technologies, 87
 - VPNs and, 77
 - Engineering team, 258–259
 - Enterprise JavaBeans (EJB), 145, 401
 - «entity» class, 222
 - EntryList tag, JavaServer Pages (JSP), 303–310, 435, 437–438
 - EntryLoop tag, JavaServer Pages (JSP), 303, 311, 313–314, 435, 437–438
 - Entry tag, JavaServer Pages (JSP), 310–313, 435, 437–438
 - Environments
 - client, 164–167
 - development, 137, 248
 - distributed, 56
 - runtime, 381
 - European Laboratory for Particle Physics (CERN), 9
 - Event handlers, JavaScript, 41
 - Events, JavaScript, 40–42
 - «extends», 179
 - Extensible Markup Language. *See* XML
 - Extensible Stylesheet Language (XSL), 57–58
 - eXtreme Programming (XP), 97
- F**
- Façade architectural pattern, 144
 - Fault tolerance, 13–14
 - Field validation, dynamic clients, 31
 - File systems, thin Web clients, 147
 - Firewalls, 85, 146
 - <FORM>
 - attributes, 353–355
 - «form» and, 238
 - HTML forms, 19, 327–329, 351–355
 - Form elements, HTML, 18–21
 - <FORM>, 19
 - <INPUT>, 20
 - overview of, 18–19
 - <SELECT>, 19
 - <TEXTAREA>, 19–20
 - Form validation, 31
 - «form», 238
 - «forward», 239, 373
 - <FRAME>. *See* <FRAMESET>
 - <FRAMESET>, 265–270
 - client pages and, 268
 - conceptual model of, 267–268
 - controversy over use of, 22
 - difficulty of modeling, 265
 - function of, 21, 267
 - as HTML element, 343–345
 - i-frames, 269–270
 - mapping «frameset» to, 356
 - mapping «target» to, 355–356
 - parameters, 21
 - stereotypes for modeling, 266–267
 - structure of, 265–266
 - targets and, 266–267, 269

«frameset», 344, 356
 mapping to <FRAMESET>, 344
 UML classes, 356
 function statement, JavaScript, 340
 Functional decomposition, avoiding, 179–183
 Functional requirements
 defined, 165
 as focus of analysis process, 216
 requirements viewpoint and, 135
 software development and, 116
 Funding, vision document and, 164

G

GET values, HTML forms, 19
 Glossary, project
 component view, 441–443
 data tier, 438–441
 design model, 431–434
 development process and, 113–114
 entity tier, 438
 function of, 167–168
 implementation and, 300–301
 presentation tier, 435–438
 requirements and use case model, 429–431
 screens, 444–448
 user experience model, 431
 Goal statements, 174

H

Hackers. *See* Attackers
 Hardware requirements, 166
 <HEAD>, 350
 HI (human interaction) skills, 188
 Home.jsp source, 299–300
 Host names, 11
 href parameter, 17–18, 240
 <HTML> elements
 <A>, 17–18, 320–325
 <APPLET>. *See* <OBJECT>
 <AREA>, 325
 <BODY>, 318–319
 <BUTTON>, 335–336
 <FORM>, 18–21, 327–329

<FRAME>. *See* <FRAMESET>
 <FRAMESET>, 343–345
 <HTML>, 318–319
 <IFRAME>, 346–347
 <INPUT>, 329–333
 <LINK>, 325–327
 <OBJECT>, 336–338
 <PARAM>, 338–339
 <SCRIPT>, 339–343
 <SELECT>, 333–334
 <TEXTAREA>, 334–335
 <TITLE>, 319
 «HTML form»
 logical view and, 238
 mapping <BUTTON> to, 335–336
 mapping <FORM> to, 327
 mapping <INPUT> to, 330–333
 mapping <SELECT> to, 333–334
 mapping <TEXTAREA> to, 334–335
 modeling, 240
 as UML class, 351–355
 HTML (Hypertext Markup Language), 14–22
 core attributes, 319
 elements. *See* <HTML> elements
 form elements, 18–21
 formatting commands, 15–16
 frames, using for screen compartments, 203
 framesets. *See* <FRAMESET>
 function of, 14
 input controls, 200
 limitations of, 14
 mapping to UML, 315–316
 multimedia tags and, 17
 as output of CGI modules, 27
 overview of, 14–17
 separating content from presentation, 16–17
 storyboards and, 192
 style sheets and, 16
 tags, 14–15
 URL resolution, 316–318
 Web architecture and, 49
 XML compared with, 57–58
 HTTP (Hypertext Transfer Protocol)
 disadvantages of, 49–50
 document identification and, 10–11

HTTP (Hypertext Transfer Protocol) *continued*
 domain names and, 11–12
 fault tolerance and, 13–14
 limitations of, 155
 resource identifiers and, 13
 state management, 23. *See also* cookies
 thin Web client connections, 147
 Web architecture and, 49
 «HTTP Resource». *See* HTTP resources
 HTTP resources, 273–279
 class stereotype, 277
 glossary application and, 442
 mapping to URLs, 273
 physical, 275, 317, 348
 as UML component, 349
 UX model and, 281
 virtual, 275–276, 348
 HTTPS (Secure HTTP), 14
 Human interaction (HI) skills, 188
 Hypertext Markup Language. *See* HTML
 Hypertext Transfer Protocol. *See* HTTP

I

IA (Information architect), 100, 188
 ICANN (Internet Corporation for Assigned
 Names and Numbers), 12
 ICONIX Unified Process, 97
 IE (Internet Explorer), 84
 <IFRAME>, 269–270, 346–347, 360–361
 «iframe», 269–270, 360–361
 IIOP (Internet Inter-Orb Protocol), 52
 Implementation, 123–127, 287–314
 associations and traceabilities, 126
 control mechanism, 290
 ControlTemplate, 295
 EntryListTag, 303–310
 EntryTag, 310–313
 glossary page, 300–301
 home page, 299–300
 illustration of, 125
 index page, 301–303
 overview of, 287–289
 page requests, 291
 PresentationTemplate, 296–299

 programming languages and technologies,
 123, 125
 RequestProcessor, 292–295
 ScreenDefinitions, 295–296
 unit testing and, 123
 «include», 239, 361–363, 373
 «includes», 177, 182
 Index.jsp source, 301–303
 Information architect (IA), 100, 188
 <INPUT>, 20, 200, 329–333
 «Input form», 200, 210, 357
 Input forms
 multiple, 202
 types of, 200
 Interaction diagrams, 177
 Internet Corporation for Assigned Names and
 Numbers (ICANN), 12
 Internet Explorer (IE), 84
 Internet Inter-Orb Protocol (IIOP), 52
 Internet Protocol (IP), 11
 Internet Server API (ISAPI), 27
 Intrusion detection, 89
 Iteration
 activity diagram for, 106
 analysis and, 216–217
 plans, 103, 111–112
 Software Iteration use case, 105–108
 workflow and, 104–105

J

J2EE
 API, 7
 Number Store example, 379
 Web page management, 248–249
 Jacobson, Ivar, 3, 97, 173
 JAR (Java Archive) files, 44, 82
 Java applets
 <OBJECT> tag and, 43–44
 class libraries, 43
 client-side security and, 81–83
 downloading and caching on client, 44
 extending thin clients with, 149
 passing parameters to, 44
 RMI and, 55

- Java Archive (JAR) files, 44, 82
 - Java Development Kit (JDK), 52, 81–82
 - Java Foundation Classes (JFC), 43
 - Java Pet Store, 7
 - Java programming language, 36, 43
 - Java Remote Method Protocol (JRMP), 52
 - Java servlets, 27
 - Java technology, 43
 - Java Virtual Machine (JVM), 83
 - JavaBeans
 - COM objects and, 63
 - compared with scripting, 32
 - JavaScript
 - <SCRIPT> element and, 339
 - animating menus and buttons, 42
 - browser compatibility and, 36–37
 - client-side security and, 80
 - custom objects, 39–40
 - document objects, 37
 - DOM and, 37–38
 - events, 40–42
 - function statement and var keyword, 340
 - Java programming language and, 36
 - script libraries, 271–272
 - script objects, 273
 - variables, 342
 - JavaServer Pages (JSP)
 - custom tags, 281–285
 - as example of Web architecture, 7
 - JSP components in glossary application, 443
 - JSP files, 365–369
 - mapping to UML, 363–369
 - scripted approach of, 28–29
 - stereotypes, 282–283
 - Web applications and, 363–365
 - web.xml file, 363
 - JavaServer Pages Model 2 Architecture, 249–250
 - JDK (Java Development Kit), 52, 81–82
 - JFC (Java Foundation Classes), 43
 - JRMP (Java Remote Method Protocol), 52
 - .js file, 271–272, 349
 - JSP. *See* JavaServer Pages (JSP)
 - JSP files, 365–369
 - «JSP Tag Library», 375–376
 - «JSP tag», 282–284
 - «JSP»
 - glossary application and, 300, 304
 - modeling JSP files with, 365–369
 - UML JSP profile and, 369–370, 373–375
 - JVM (Java Virtual Machine), 83
- ## K
- Kruchten, Philippe, 133
- ## L
- Languages, UML, 3
 - <LINK>
 - as HTML elements, 325–327
 - «link» associations and, 358
 - style sheets and, 349
 - «link»
 - associations, 239, 358–360
 - href parameter and, 240
 - mapping HTML to UML, 321
 - mapping to <LINK>, 325
 - screen compartments and, 243
 - Lockwood, Lucy, 173
 - Logical view, 237–242
 - client page extension, 237–238
 - design viewpoint and, 136
 - HTML forms extension, 238
 - master template pattern, 425–426
 - modeling client pages, 240–242
 - modeling HTML forms, 240
 - modeling server pages, 240–241
 - relationships between stereotypes, 239
 - server page extension, 237
 - Web components and, 280
- ## M
- «managed», 210
 - <MAP> element, 325
 - Master template pattern, 423–427
 - analysis model, 427
 - logical view, 425–426
 - use case view, 423–424
 - MDE (Microsoft Development Environment), 248

Menus

- animating, 42
- glossary application and, 434
- <META>, 360
- method parameter, <FORM> tag, 19
- Microsoft DCOM. *See* Distributed COM (DCOM)
- Microsoft Development Environment (MDE), 248
- Microsoft Transaction Server (MTS), 145
- MIME (Multipurpose Internet Mail Extensions)
 - types, 84–85
- Model View Controller (MVC) pattern, 411. *See also* Controllers
- Modeling
 - benefits of, 5
 - by-products vs. purposes of, 5–6
 - development process and, 109–110
 - levels of abstraction in, 4–5
 - model-driven development, 3
 - secure systems, 89–91
 - updates and, 287
- Mouse-over events, 42
- MTS (Microsoft Transaction Server), 145
- Multipurpose Internet Mail Extensions (MIME)
 - types, 84–85
- MVC (Model View Controller) pattern, 411. *See also* Controllers

N

- Namespaces, XML, 61–62
- National Center for Supercomputing Applications (NCSA), 16
- navigate to(), 196
- Navigational map
 - associations and, 199
 - diagrams, 192–193
 - glossary application and, 432
 - notations used with, 210
 - Number Store application and, 387–388
 - pathways between screens, 192–193
 - principal paths, 196, 200
 - top-level mapping of Web applications, 210
 - user input, 201
- NCSA (National Center for Supercomputing Applications), 16

.NET

- Web page management, 248–249
- Web services and, 63
- Netscape Server API (NSAPI), 27
- Network security, 72
- Nonfunctional requirements
 - categories of, 165–166
 - requirements viewpoint and, 135–136
 - software development and, 116
- Notation, navigational, 210
- NSAPI (Netscape Server API), 27
- Number Store application. *See* Software architecture document, Number Store

O

- <OBJECT>
 - as HTML elements, 336–338
 - «object» and, 360
- Object-oriented principles, 3
- «object parameters», 338–339
- «object»
 - associations, 239, 360
 - mapping <OBJECT> to, 337–338
- Objects, JavaScript, 37–40
- Open Systems Interconnection (OSI), 13
- Operating system security, 79
- Operations (behaviors). *See* Behaviors (operations)
- Organizations, tailoring development to, 98–99
- OSI (Open Systems Interconnection), 13
- Outstanding issues, use cases and, 174

P

- Package diagrams, 184
- Packages
 - hierarchy in analysis model, 217–218
 - «JSP Tag Library» package, 375–376
 - Number Store application, 378
 - ownership of, 219–220
 - qualities in analysis model, 220
 - UML components, 348
- Page composition, 144, 417
- Page requests
 - controlled controllers and, 412

- handling, 394–395
- implementation scenarios, 291
- mechanisms, 289
- Number Store application and, 396–398
- <PARAM>, 338–339
- Participants diagram, 208–209
- Passwords
 - basic authentication, 75
 - creating policy for, 85–86
 - guidelines for choosing, 76
 - programs for cracking, 75–76
- Performance
 - requirements, 165
 - testing, 127
- Perl (practical extraction and reporting language), 26
- PHP, 28–29
- «physical root», 244–246, 317, 348
- Physical view. *See* Component view
- Plug-ins, client-side security and, 84–85
- POST values, HTML forms, 19
- Postconditions, use cases, 174
- Practical extraction and reporting language (Perl), 26
- Presentation logic, vs. business logic, 31
- Presentation tier
 - glossary application, 435–438
 - Number Store application, 392–394
 - patterns, 143–145
- PresentationTemplate.jsp source, 296–299
- Prioritization of requirements, 171–173
- Privacy, cookies and, 25, 80
- Private/public keys, 87–88
- Procedural risk, contrasted with technical risk, 73
- Process. *See* Software development process
- Process view, 120, 136
- Project management, 111–113
- Project plans, 103, 111–112
- Project scope, 162–163
- Prototypes
 - architectural, 142–143
 - JavaScript, 40
 - as output of UX team, 188
- Public/private keys, 87–88

Q

- Quality assurance (QA), 99, 127

R

- Rational Rose, as modeling tool, 4
- Rational Unified Process (RUP), 97, 105
- Realization viewpoint
 - overview of, 136–137
 - thick Web client, 154
 - thin Web client, 149
 - Web Delivery, 157
- «redirect», 239, 360, 373
- Regression testing, 127
- Relationships. *See also* Associations
 - analysis and, 120
 - dynamic, 177
 - between requirements, 166–167
 - between stereotypes, 239
 - use case model, 177
- Reliability requirements, 165
- Remote Method Invocation (RMI), 52–55
 - IIOP as transport protocol, 52
 - interaction of applets and remote objects, 55
 - layered architecture of, 53
 - setting up remote object server, 54
 - stub and skeleton objects and, 52–53
 - Web delivery and, 156
- Remote object server, 54, 156
- Remote objects, 156
- Remote procedure calls (RPCs), 56, 63
- Remove object transfer protocol, 156
- RequestProcessor
 - controllers and, 415
 - implementation and, 292–295
 - Number Store application, 395
- Requirements, 164–167
 - associations and traceabilities, 118
 - attributes, 166
 - dependence of analysis on, 120
 - example document, 168–170
 - functional and nonfunctional, 116, 165–166
 - gathering, 168–169
 - glossary application and, 429–431

Requirements *continued*

- hierarchy of, 166
 - illustration of, 115
 - Number Store application and, 378–381
 - overview of, 135–136
 - prioritizing, 171–173
 - relationships between, 166–167
 - requirements team, 114, 168
 - reviewing and amending, 170
 - screens as artifacts of, 192
 - as system constraints, 164–165
 - testing, 165
 - thick Web clients, 151–152
 - thin Web clients, 145–146
 - use cases, 116–117, 175
 - user experience document and, 117
 - vision document and, 116, 163
 - Web Delivery, 155
 - writing requirements statements, 171
- Requirements team, 114, 168
- Resource identifiers, 13
- Reverse-engineering
- modeling and, 5–6
 - updating models and, 288
- Risk
- addressing early, 217
 - client-side, 79–85
 - development process and, 104
 - server side, 77–79
 - technical, 73–77
 - types of, 72–73
- RMI. *See* Remote Method Invocation (RMI)
- Role playing, analysis process and, 221
- Round-trip engineering, 4, 5
- Rows parameter, HTML frameset, 21
- RPCs (remote procedure calls), 56, 63
- Rumbaugh, Jim, 3, 97
- Runtime environment, 381
- RUP (Rational Unified Process), 97, 105

S

- SARs (significant architectural requirements), 135, 380
- Scalability strategies, 137

Schedules

- development process and, 104
 - iteration and, 107–108
- Schemas, XML, 62–63
- SCM (service control manager), 56
- Scope, vision document, 162–163
- «screen compartment», 204, 243, 357
- Screen flow, 196, 200. *See also* Navigational map
- «screen»
- navigational maps and, 193
 - notations used with, 210
 - as UML classes, 356–357
 - UX model and, 210
- ScreenDefinitions.jsp source, 295–296, 395
- Screens, 189–200
- behaviors (operations), 196, 198
 - compartments, 203–204
 - content management and, 190–192, 194–196
 - contrasted with Web pages, 189
 - flow, 196, 200
 - glossary application, 444–448
 - navigational map of, 192–193
 - Number Store application, 406–410
 - properties, 190
 - as requirement artifact, 192
 - «screen» class and, 193
 - storyboards and, 192
 - as user interface, 189
- <SCRIPT>, 339–343, 356
- «script library»
- client-side scripting, 271, 343
 - as UML classes, 356
 - as UML component, 349
- «script», 343
- Scripts. *See also* JavaScript
- applying client-side business logic, 31–32
 - enabling technologies and, 28–29
 - extending thin clients, 149
 - libraries, 271–272
 - objects, 273
 - security of, 80
 - WAE profile and, 363
- Search engines, 22–23
- Secure Electronic Transaction (SET), 87
- Secure HTTP (HTTPS), 14

- Secure Sockets Layer (SSL), 87
- Security, 71–92
 - ActiveX and, 83–84
 - best practices, 88–89
 - client-side risks, 79–85
 - cookies and, 79–80
 - encryption, 86–88
 - Java applets, 81–83
 - JavaScript, 80
 - modeling secure systems, 89–91
 - plug-ins and MIME types, 84–85
 - requirements, 166
 - risk types, 72–73
 - server-side risks, 77–79
 - strategies, 85–86
 - technical risks, 73–77
 - vision document and, 164
 - zones in Internet Explorer (IE), 84
- <SELECT>, 19, 333–334
- Sequence diagrams
 - Browse Catalog and, 181, 227
 - client pages, 257
 - vs. collaboration diagrams, 225–226
 - value of, 179
 - Web page design and, 252–253
- «server page». *See* Server pages
- Server pages
 - logical view stereotypes, 237
 - modeling, 240–241
 - «server page» class, 349
 - UML JSP profile, 370–373
 - user interface construction, 262
- Server security, 72
- Server-side includes (SSI), 16–17, 78
- Server-side risks, 77–79
- Service control manger (SCM), 56
- Session state, 25–26, 140
- Set-Cookie header, 24
- SET (Secure Electronic Transaction), 87
- SGML (Standard Generalized Markup Language), 14, 58
- Significant architectural requirements (SARs), 135, 380
- Simple Object Access Protocol. *See* SOAP
- Skeleton objects, RMI, 52–53
- Sniffers, 76–77
- SOAP (Simple Object Access Protocol), 64–65
 - combining with WSDL and UDDI, 67–68
 - message parts, 64
- Software architecture document, 120, 231
- Software architecture document, Number Store, 379–406
 - component view, 401–406
 - design view. *See* Design, Number Store
 - requirements view, 379–381
 - use case view, 382
 - user experience view, 382–385
- Software bugs, 74
- Software development process
 - analysis set, 117–120
 - artifacts, 108–111
 - deployment set, 128–130
 - design set, 120–123
 - Develop Software use case, 101–105
 - domain set, 113–114
 - implementation set, 123–127
 - managing complexity in, 377
 - overview of, 96–100
 - project management set, 111–113
 - requirements set, 114–117
 - roles in, 96
 - Software Iteration use case, 105–108
 - test set, 127–128
 - UML model as basis of, 100–101
 - vision document as basis of, 161
 - Web architectures and, 6
- Software Iteration use case, 105–108
 - activity diagram for, 106–107
 - review and refinement process and, 107–108
 - slipping schedules and, 107
- Source code, 137
- Specification information, UDDI Web services, 65–66
- SSI (server-side includes), 16–17, 78
- SSL (Secure Sockets Layer), 87
- Stakeholders, 101, 161–162
- Standard Generalized Markup Language (SGML), 14, 58
- State diagrams, 401

- State management, 23
 - Static content, screens, 189–190
 - Static pages
 - extensions, 242–244
 - requests, 148
 - «static page» class, 348
 - thin Web client components, 147
 - Stereotypes
 - component view, 242–246
 - HTML framesets, 266–267
 - JSP tags, 282–283
 - logical view stereotypes, 237–242
 - UML extensions and, 261–262
 - Stoll, Clifford, 74
 - Storyboards, 204–209
 - checkout storyboard, 197–198
 - collaboration diagrams and, 204–206
 - glossary application and, 433
 - navigational flow, 207–208
 - Number Store application and, 389–391
 - participants diagram, 208–209
 - screens and, 192
 - Structural elements, analysis, 222–225
 - Structural patterns, Web applications, 143–144
 - Stub objects, 52–53, 156–157
 - «style sheet», 349, 357
 - Style sheets, HTML, 16
 - Subdomains, 11
 - «submit», 239, 360
 - Swing components, Java applets, 43
 - Symbolic links, 78
 - System
 - constraints. *See* Requirements
 - development process and, 109–110
 - interaction with actors in, 173–174
 - nodes, 137
- T**
- Tag library stereotype, 282
 - Tag stereotype, 283
 - Tagged values, UML extensions, 261–262
 - Tags
 - HTML for document text, 14–16
 - JSP custom, 281–285
 - special HTML tags for multimedia, 17
 - XML, 59
 - target parameter, anchor tag (<A>), 18
 - «Target», 355–356
 - «Targeted Link» *n*-ary association, 355
 - Targets, 266–267, 269
 - TCP (Transmission Control Protocol), 13
 - Teams
 - architectural, 139, 141–142
 - design, 120, 122–123
 - preferences and work habits of, 99
 - relationship between engineering and UX, 258–259
 - requirements, 168
 - skill level of, 99
 - test, 127
 - UX, 100, 122–123
 - Technical risk, 73–77
 - authentication and, 74–76
 - configuration and bugs and, 74
 - contrasted with procedural risk, 73
 - encryption and, 76–77
 - Tei stereotype, 283
 - Template pattern. *See* Master template pattern
 - Templated pages, 144
 - Test team, 127, 138
 - Tests
 - compared with quality assurance, 127
 - illustration of, 128
 - overview of, 137–138
 - plan, 138
 - requirements, 165
 - thick Web client, 154–155
 - thin Web client, 149
 - types of, 127
 - Web Delivery, 157
 - Web pages, 261
 - <TEXTAREA>, 19–20, 334–335
 - Thick Web clients, 149–155
 - defined, 144–145
 - deploying, 154
 - design viewpoint, 152–154
 - designing, 246–247
 - illustration of participants, 152
 - overview of, 149–151

- realization viewpoint, 154
 - requirements viewpoint, 151–152
 - test viewpoint, 154–155
 - Thin Web clients, 145–149
 - defined, 144
 - deploying, 151
 - design viewpoint, 146–149
 - illustration of participants, 148
 - partitioning objects for, 246
 - realization viewpoint, 149
 - requirements viewpoint, 145–146
 - test viewpoint, 149
 - Third-level domains, 12
 - <TITLE>, 319
 - Top-level domains (TLDs), 11–12
 - Top-level model
 - analysis and, 218–221
 - glossary application, 432
 - Number Store application, 378, 385, 392
 - «trace dependencies», 138
 - Traceability
 - analysis, 121
 - deployment, 129
 - design, 124
 - development process and, 109, 111
 - implementation, 126
 - project management and, 112
 - requirements, 118
 - Transmission Control Protocol (TCP), 13
- U**
- UDDI. *See* Universal Description, Discovery, and Integration
 - UML associations
 - «iframe», 360–361
 - «include», 361–363
 - «link», 358, 358–360
 - «object», 360
 - «redirect», 360
 - «submit», 360
 - UML classes
 - «boundary», 222–223
 - «client page», 350–351
 - «control», 222
 - «entity», 222
 - «frameset», 356
 - «HTML form», 351–355
 - «input form», 200, 357
 - «screen compartment», 204, 357
 - «screen», 193, 356–357
 - «script library», 356
 - «server page», 349
 - «style sheet», 357
 - «target», 355–356
 - «targeted link» *n*-ary association, 355
 - UML component packages
 - «physical root», 348
 - «virtual root», 348
 - UML components
 - «dynamic page», 348
 - «HTTP resource», 349
 - «script library», 349
 - «static page», 348
 - «style sheet», 349
 - UML JSP profile, 369
 - «forward», 373
 - «include», 373
 - «JSP tag», 373–375
 - «JSP», 369–370
 - «JSP tag library», 375–376
 - «redirect», 373
 - «server page», 370–373
 - «use bean», 373
 - UML (Unified Modeling Language). *See also* Web Application Extension for UML (WAE)
 - creation of, 3
 - design extensions, 236–237
 - mapping to/from HTML, 315–316, 347
 - mapping to JSP, 363–369
 - modeling XML with, 63
 - scripting and, 340
 - software development process and, 100–101
 - use case model and, 176
 - user experience. *See* User experience (UX)
 - Uniform resource identifiers (URIs), 13, 316
 - Uniform resource locators (URLs)
 - document identification and, 10–11
 - HTTP parameters, 322–323
 - mapping to HTTP resources, 273–279, 316–318

- Uniform resource locators (URLs) *continued*
 - referencing remote objects, 53
 - resolution for HTML, 316–318
 - URL redirection for session management, 26
 - URL uniqueness, 323–325
 - «virtual root» and, 348
 - Uniform resource names (URNs), 13, 317
 - Unique ID, 174
 - Unique keys, 26
 - Unit testing, 123, 287
 - Universal Description, Discovery, and Integration (UDDI), 65–66
 - application of, 66
 - combining with WSDL and SOAP, 67–68
 - data structure types, 65–66
 - publishing Web services, 65
 - Updating models, 287
 - URIs (uniform resource identifiers), 13, 316
 - «URL parameters», 358
 - URLs. *See* Uniform resource locators (URLs)
 - URNs (uniform resource names), 13, 317
 - Usability requirements, 165
 - «use bean», 373
 - Use cases, 173–185
 - analysis, 120, 216
 - controllers, 411–414
 - Develop Software, 101–105
 - diagrams, 177–179
 - examining and prioritizing, 139–141
 - functional decomposition and, 179–183
 - glossary application, 430
 - interaction between actors and systems, 173–174
 - key information captured with, 174
 - Master Template Pattern, 423–424
 - multiple scenarios and, 175–176
 - Number Store application, 382
 - relationships, 177
 - requirements, 116–117
 - risk assessment, 104
 - Software Iteration, 105–108
 - structure of, 175, 183–185
 - UML representation of, 176
 - User account security, 88
 - User experience (UX), 187–213
 - class stereotypes, 210–212
 - document, 117, 185
 - glossary application, 431
 - HTTP resources and, 281
 - mapping analysis to UX model, 230–231
 - mapping design to UX model, 257–260
 - navigational map, 192–193, 210
 - Number Store application, 382–385, 394, 398
 - overview of, 188–189
 - screen compartments, 203–204
 - screen flow, 196–200
 - screens, 189–196
 - storyboards, 192, 204–209
 - user input, 200–203
 - User experience (UX) team
 - artifacts responsible for, 188–189
 - artistic and creative aspect of projects and, 100
 - characteristics of, 122–123
 - engineering team and, 258–259
 - information architect (IA) as member of, 188
 - responsible for look-and-feel, 187
 - user experience document and, 117
 - User IDs, 75. *See also* Passwords
 - User input, 200–203
 - association and contained classes and, 200–201
 - dynamic content and, 202
 - input forms, 200
 - multiple input forms, 202
 - navigational flow and, 201
 - User interface
 - construction of, 262
 - screens as, 189
 - Users, 173
 - UX. *See* User experience (UX)
 - UXG (UX guidelines) document, 185
- ## V
- Validation
 - client-side, 31–32
 - thick Web clients and, 151–152
 - var keyword, JavaScript, 340
 - Version control, 110
 - Viewpoints, architectural
 - design viewpoint, 136
 - mapping, 138

- overview of, 134
 - realization viewpoint, 136–137
 - requirements viewpoint, 135–136
 - test viewpoint, 137–138
- Virtual private networks (VPNs), 77
- «virtual root», 317, 348
- Vision document, 161–164
- artifact process and, 108
 - as basis of development process, 103, 161
 - elements of, 162
 - features and requirements, 163
 - funding and, 164
 - Number Store application and, 377
 - project scope and, 162–163
 - requirements set and, 116
 - security and client environments, 164
- VPNs (virtual private networks), 77
- ## W
- W3C (World Wide Web Consortium)
- HTML standard, 14
 - XML namespaces, 61
 - XML schemas, 62–63
 - XML standard, 57
- Web Application Extension for UML (WAE). *See also* UML
- applicability to Web architectures, 7
 - association stereotypes, 239
 - class stereotypes, 316
 - defined, 379
 - expressing in terms of stereotypes, tagged values, and constraints, 236
 - extending UML for Web modeling, 4
 - version 2, 8
- Web applications
- client state management, 23–26
 - design guidelines, 261–262
 - designing, 246
 - enabling technologies, 26–29
 - JavaServer Pages (JSP) and, 363–365
 - Web sites vs. Web applications, 22–23
- Web architecture
- distributed objects and, 49–50
 - HTML and, 49
 - HTTP and, 49
 - limitations of, 7
- Web delivery, 155–159
- defined, 145
 - deploying, 159
 - design viewpoint, 155–157
 - designing applications for, 247
 - overview of, 155
 - realization viewpoint, 157
 - remote object execution scenario, 158
 - requirements viewpoint, 155
 - test viewpoint, 157
- Web pages
- class diagrams, 252
 - control mechanisms, 289
 - design, 247–253
 - J2EE and .NET and, 248–250
 - Number Store shopping functionality and, 399
 - page request mechanisms, 289
 - screens contrasted with, 189
 - search engine example, 250–251
 - sequence diagrams, 252–253
 - UX design elements, 248
- Web servers, 146
- Web services, 63–64
- compared with RPCs, 63
 - decentralized processing and, 63
 - publishing with UDDI, 65
 - SOAP, 64–65
 - UDDI, 65–66
 - WSDL, 66–68
- Web Services Description Language. *See* WSDL
- Web sites, vs. Web applications, 22–23
- web.xml file, 289, 363
- Word form analysis, 222
- Workers, 96
- Workflows
- development process and, 96–97
 - as dynamic relationship, 177
 - iterative nature of, 97–98
 - RUP and, 105
- World Wide Web Consortium. *See* W3C (World Wide Web Consortium)

WSDL (Web Services Description Language),
66–68
combining with UDDI and SOAP, 67–68
document elements, 66–67
Web services and, 66

X

XML (Extensible Markup Language), 57–63
advantages of, 58

compared with HTML, 57–58
document elements, 58–59
DTDs, 60–61
namespaces, 61–62
roots in SGML, 58
schemas, 62–63
successful use of, 59
tags, 59
XP (eXtreme Programming), 97
XSL (Extensible Stylesheet Language), 57–58