

Index

A Guide to the Index

The index is arranged hierarchically, in alphabetical order, with symbols preceding the letter A. Most second-level entries and many third-level entries also occur as first-level entries. This is to ensure that users will find the information they require however they choose to search for it. Elements are XML Schema elements unless indicated otherwise.

A

- absolute location path (XPath), 327**
- abstract elements**
 - cannot be instantiated in instance documents, 253
- Abstract Syntax Notation One**
 - see ASN.1.
- abstract types, 144**
- active elements (Schematron)**
 - pattern attribute, 501
- actor SOAP-ENV attribute, 541**
- aggregation**
 - kinds of relationship between objects, 410
- all compositor**
 - getting around restrictions imposed by, 261
- all element, 79, 391, 603**
 - attributes, 604
- all group**
 - allows elements to appear in any order, 259
 - restrictions, 261
- annotation element, 27, 28, 490, 604**
- annotations example, 27**
- anonymous complex type, 15, 76, 86**
- any element**
 - attributes, 605
- any element content, 89**
 - times when appropriate, 89
- anyAttribute element**
 - attributes, 606
- anyURI datatype, 54**
- appinfo element, 27, 28, 380, 490, 558, 607**
 - attributes, 607
- Application element, 568**
- arcs, digraphs, 580**
- ASN.1**
 - optimization of BER (Basic Encoding Rule), 532
 - targeted at communications infrastructure and telemetry sectors, 533
- assert element (Schematron)**
 - diagnostics attribute, 500
- association**
 - kinds of relationship between objects, 409
- atomic datatypes, 48**
 - may be primitive or derived, 48

atomicity

- definition, 339

attribute declarations, 23

- always simple types, 23
- global and local, 24
- occurrence, 24
- unordered, 23
- value constraints, 25

attribute element, 363, 607

- attributes, 608
- declaring an attribute, 16
- form attribute, 207

attribute groups, 84

attribute information item, 30

attributeFormDefault attribute

- schema element, 201
- setting value as unqualified, 215

attributeGroup element, 85, 608

- attributes, 609
- name attribute, 84

attributes

- locating, 327
- used to distinguish types of element, 263

authoring tools

- TurboXML (TIBCO Extensibility), 663
- XML Spy 3.5 and 4.0 (Altova), 664

B

base64Binary datatype, 57

block escape, 124, 129

- commonly used Unicode character blocks, 129

boolean datatype, 56

bounds facet, 108

- bounded schema component, 108

built-in derived types, 64

- categories, 64
- string types, 66

built-in primitive types, 53

- date/time datatypes, 59
- decimal types, 69
- encoded binary datatypes, 56
- numeric datatypes, 57
- string types, 54

built-in datatypes, 477

- see *also* constraining facets of built-in datatypes, namespaces, 52
- primitive and derived, 47

built-in simple types, 14

business process analysis, 402

- background information, 403
- cyclical nature of modeling, 402
- identifying business domain objects, 404
- Object Diagram, 404
- system decomposition and abstraction, 407
- rules of thumb, 408
- Use Case Diagram, 404, 405

byte datatype, 70

C

CALS exchange table model

- see OASIS model.

camel case, 230

canonical representation, 46

cardinality

- properties and sub-objects, 409

cardinality facet, 108

- cardinality schema component, 109

category escape, 124

- table of varieties, 127

chameleon design, 290

- good for internal schemas, 292
- industry-wide schemas
- points to consider, 292
- namespace coercion, 290
- qualifying markup, 296

character class escapes, 124

- single character escape, 125

character class expressions

- block escapes, 129
- category escapes, 127
- character groups, 122, 124
- character ranges, 123
- multi-character escape, 126

character class subtraction, 124

character groups, 124

character information item, 30

choice element, 79, 609

- attributes, 610

choice groups

- ideal for representing either/or choices, 252
- more helpful when, 254

cold arcs, 588

- PNML-manipulating stylesheet
- declaring, 584

co-location, 508

combination uniqueness constraints, 331

complex type definitions, 76

complex types, 44

- compared to simple types, 15
- defined, 14
- nesting, 15
- re-use, 238
- example, 239

complexContent element, 138

- attributes, 610

complexType element, 13, 88, 378, 610

- anonymous complex types, 76
- attributes, 611
- block attribute, 364
- final attribute, 364
- named complex type, 76

composition

- kinds of relationship between objects, 410

compositors, 78

- combining, 80
- repeating, 80

concatenated primary key

- definition, 337

constraining facets, 46

- enumeration facet, 113, 639
- fractionDigits facet, 643
- length facet, 636
- length, minLength, maxLength facets, 109
- maxExclusive facet, 641
- maxInclusive facet, 640
- maxLength facet, 637
- minExclusive facet, 641
- minExclusive, minInclusive, maxExclusive, maxInclusive facet, 113
- example of value space bounded above, 114
- exclusive bound, 113
- inclusive bound, 113
- minInclusive facet, 642
- minlength facet, 637
- pattern facet, 112, 638
- totalDigits facet, 642
- totalDigits, fractionDigits facets, 114
- whiteSpace facet, 111, 639

constraining facets of built-in datatypes, 115

- derived datatypes, table, 117
- primitive datatypes, table, 116

constraint, business rule, 424

contacts schema example, 93

- declaring the namespace for XML Schema, 96
- define the root element, 96
- named complex type, 97
- telephone numbers are integer types, 98

content models

- choice group, 252
- creating, 75
- attribute groups, 84
- compositors, 78
- element content specifications, 85
- must be defined globally, 84
- named model groups, 81
- identifying element types, 482
- substitution group, 252

context nodes, 328

Controller

- implementation, 570
- implements linkage between View and Model, 567
- initializing, 577
- majority of code never changes, 575

copy element, 586

countries

- see *also* custom datatypes example
- country codes (ISO 3166), 151
- table of currencies, ISO codes and languages, 151
- three character codes, 151
- two character codes, 151

country description example, 156

section schemes, 479**Currency Codes (ISO 4217), 153**

examples, 153

custom datatypes examples, 150

countries

- Country Codes (ISO 3166), 151
- Currency Codes (ISO 4217), 153
- Language Codes (ISO 639), 155
- length and pattern facets, 152

country description example, 156

geodetic locations, 178

internet datatypes, 157

- domain names, 159

- IP addresses, 158

- URIs and URLs, 160

people, 163

- gender, 164

- names, 163

- telephone numbers, 165

postal addresses, 168

- NAFTA and British Addresses example, 175

- postal codes, 172

- postaladdress datatype - revised, 173

- regions, 170

- street addresses, 169

simple basic types, 150

- fractions, 150

- single character, 150

D**Data Dictionary, create, 411****data integrity, 339**

domain integrity, 342

entity integrity, 340

referential integrity, 342

user-defined integrity, 343

data model testing, 438**data modeling, 426**

creating schema and instance documents, 427

global declarations, 435

glossary of terms, 395

modeling process schematic, 396

namespaces, schemas, and instance documents, 426

objects compared to instances, 396

types of state, 397

principles, 394

simplified approach, 394

data points

elements or attributes, 448

data redundancy

definition, 337

database terms

clarifying, 337

data-centric XML documents

differences to document-centric XML documents, 459

datatype derivation

block attribute, 146

blockDefault attribute, 146

final attribute, 145

finalDefault attribute, 146

fixed attribute, 145

datatype inheritance, 135

deriving complex types, 138

deriving simple types, 137

type hierarchy, 136

- derive new complex types, 136

- derive new simple types, 136

datatype properties, 45**datatypes, 43, 149**

advantages, 43

built-in types compared to user-defined, 48

deriving by restriction, list or union, 47

different kinds, 48

datatypes in XML overview, 44**date datatype, 61****date/time datatypes, 54, 59****dateTime datatype, 60**

acceptable formats, 60

decimal datatype, 57**decimal types, 69****declarative programming, 564**

formal methods, 565

SBP, 564

SQL is most common example, 564

declaring order and occurrence

fixed order, all elements required, 242

fixed order, optional elements, 242

requiring both elements or none, 244

requiring elements in a mixed model, 244

requiring repeating sequences of elements, 243

definition business rule, 424**delivery receipt schema example**

validation with XSV (XML Schema Validator), 36

derivation business rule, 425**derived datatypes, 47, 105**

complex types, 105

defined in terms of an existing datatype, 47

simple types, 105

deriving complex types, 101

abstract types, 144

deriving by extension, 139

deriving by restriction, 142

restrictions on element content, 138

- complexContent element, 138

- extension element, 139

- restriction element, 138

- simpleContent element, 138

substitution groups, 143

diagnostic element (Schematron), 500**diagnostics element (Schematron), 500****Digital Object Identifier**

see DOI.

digraphs, 579**dir element (Schematron), 503****disabled transitions, 585**

PNML-manipulating stylesheet

- declaring, 584

document analysis

mapping structure of document set, 463

- analyzing structures, 463

- identifying data types, 466

document class requirements

normalizing XML, 347

document fragments

constructing a dummy document, 353

context, 353

positioning schemas' identity constraints, 351

problems with multiple schemas, 352

problems with references and uniqueness, 350

- cross-references, 350

- ID and IDREF datatypes, 350

- key and keyref relationships, 350

validation into a two-step process, 352

document information item, 30

document management, 459

- developing a document schema, 462
- managing document schema development, 462
- Physics Press goes online case study, 462

document schemas compared to data schemas

- ratio of tags to content, 460
- what XML documents model and what they leave out, 461
- XML data example, 460
- XML document example, 460

document scope

- determining content, 443
- representational or transactional, 443

Document Type Definitions

- see DTFDs.

document() function, 363, 376

documentation

- creating, 358
- navigating between components, 365
- navigating schemas, 364
- retrieving the target namespace, 362
 - creating templates, 363
- schema information, 360
- summarizing components, 361
- summarizing list attributes, 363

documentation element, 27, 28, 358, 490

- attributes, 612

document-centric XML documents

- differences to data-centric XML documents, 459

documents

- breaking into fragments, 349

DOI, 466

- standard numbering scheme, 472

domain integrity, 342

- appropriateness of data for specific field, 342

domain key normal form

- rule, 347

domain names

- internet datatypes, 159

Dot.Notation, 230

double datatype, 58

DTD cardinality operators

- functionality replicated by occurrence restraints, 20

DTFDs, 10

Dublin Core Metadata Initiative

- meta data elements for published works, 477

duration datatype, 59

E

e-commerce case study, 537

- architecture, 545
 - advantages of passing documents not parameters, 545
 - advantages of using SOAP, 546
 - communication between bank and partner, 545
 - communication with the customer, 545
- automation, 538
- data structures, 549
 - database and tables, 549
- design, 546
 - conversation sequence diagram, 547
 - transactions and messages, 548
- message schema, 550
 - sample messages, 555
 - schemaLocation attribute embedded in each message body, 557

- project requirements, 543

- design goals, 545

- main use, 543

- project scenario, 539

- SOAP, 537

element content

- defined as allowed to have child elements, 86

element content specifications, 85

- any content, 89, 90
- element content, 86, 87
- empty elements, 89
- mixed content, 88
- nil values, 92

element declarations, 16

- element occurrence indicators, 20
- global and local, 17
- value constraints on element content
 - default and fixed content, 22

element element, 13, 363, 378, 387, 390

- attributes, 612
- block attribute, 364
- default attribute, 22
- final attribute, 363
- fixed attribute, 22
- form attribute, 207
- name attribute, 16

element information item, 30

element naming, 478

element occurrence indicators, 20

- examples, 21

element only content, 482

element type definitions

- example, 16

elementFormDefault attribute

- schema element, 201, 204
- setting value as qualified, 215

e-mail addresses (mailto:

- internet datatypes), 162

emph element (Schematron), 503

empty elements

- add attributes, 89
- ideal for place holders, 89

enabled transitions, 585

- PNML-manipulating stylesheet
- declaring, 584

encoded binary datatypes, 53, 56

entities, definition, 337

ENTITIES datatype, 69

ENTITY datatype, 69

entity integrity, 340, 345

- enforcement with primary key, 340
- unique identifier, 340

enumerating a simple Type, 250

enumeration facet, 113,

- attributes 639

equality facet, 106

Examplotron

- attributes and import element, 530
- compiles XSL validators from XML documents, 528
- reasons to use, 531

extending types

- points to remember, 302

extends element (Schematron), 503

- abstract attribute, 503

extensible content models

- substitution groups, 309
- disadvantages, 311

extensible content models (continued)

- using unknown content
 - any or anyAttribute wildcard, 311
 - controlling, 313
 - processContents attribute, 312

extension element, 139, 614

- attributes, 615

F**facets - datatype properties, 45****facets - datatypes properties, 46****facets of datatypes, 105**

- constraining facets, 109
- fundamental facets, 106

fact, business rule, 425**failed-assert element (Schematron), 498****field element**

- attributes, 615

field, XPath expression, 327**first grade XML normalization**

- document class requirements, 347
- structural normalization suggestions, 347

first normal form

- functional dependencies, 345
- maintain entity integrity, 345
- rule, 344

Flat Catalog Design

- see Salami Slice Design.

float datatype, 58

- canonical form, 58

for-each element, 595

- XSLT elements, 575

foreign key

- definition, 337

form attribute

- attribute element, 207
- element element, 207

form creation, 375

- accessing schemas from instances, 375

form generation, 376

- creating form fields, 381
- creating labels, 380

formal methods, 565

- declarative programming, 565

Foundation Classes

- designed around MVC, 566

fractionDigits facet, 114, 643

- attributes, 643
- values, 114

functional dependencies, 345

- definition, 338

fundamental facets, 46

- bounds facet, 108
- cardinality facet, 108
- equality facet, 106
- numeric/non-numeric facets, 109
- order facet, 107

future proofing, 474**G****gaining tokens, 585**

- PNML-manipulating stylesheet
 - declaring, 584

gDay datatype, 63**generate-id() function, 365****generic elements, 475****geodetic locations, 178****global declarations, 435**

- global attribute declarations, 24, 435
- global element declarations, 17, 435
- complex types can use by creating reference, 17

gMonth datatype, 63**gMonthDay datatype, 63****granularity**

- level of mark up-required, 467

group element, 81, 391, 615

- attributes, 616
- minOccurs and maxOccurs attributes, 83

grouping and hierarchy

- advantages of grouping for processing, 248
- creating hierarchical structures, 249
- grouping related items of information, 246
- grouping same information items, 247
- key uses, 246

gYear datatype, 62**gYearMonth datatype, 62****H****heterogeneous namespace design, 285**

- effects on instance documents, 288

hexBinary datatype, 57**hierarchical structures**

- creating, 249

holding states, 584

- PNML-manipulating stylesheet, declaring, 584

homogeneous namespace design, 288**hot arcs, 584, 588**

- PNML-manipulating stylesheet, declaring, 584

HTML elements

- script element, 575

I**ID datatype, 67**

- limitations, 332

identifying data types

- content which falls in particular patterns, 466

identity constraints, 321

- defining, XPath expressions required, 327
- definitions must be unique within a schema, 331
- documents representing relational database structures, 336
- key and keyref mechanism, 322
- key and keyref mechanisms, 100
- locating nodes with XPath, 325
- specifying, 326
- stafflist example, 322
- unique values, 329
- uniqueness constraints, 100, 322
- XML ID type, 329

IDREF datatype, 68

- limitations, 332

IDREFS datatype, 68

- limitations, 332

import element, 373, 616

- attributes, 617
- namespace attribute, 280, 371
- schemaLocation attribute, 280
- XSLT elements, 575

import mechanism, 280

include element, 273, 374

- attributes, 617
- schemaLocation attribute, 369

include mechanism, 274

info set

- see information set

information modeling

- business rules, 423
- cardinality, 409
- components, 408
- custom datatypes, 423
- Data Dictionary, 411
- defining properties, 419
 - distinguishing business rules from application rules, 420
- kinds of relationship between objects, 409
 - standard notation for diagrams, 410
- primary object list, 410
 - class diagram, 413
 - create definitions, 411
 - future proofing and scope of model, 414
- secondary object list, 414
- sequence diagrams, 416
 - messages and properties, 417

information set

- analogous to the term tree, 30
- consists of information items, 30
- contributions to namespace qualification, 199
- element and attribute information item
 - properties for namespace resolution, 200
- local name and namespace properties, 200
 - must match with values in schema, 200
- provide a consistent set of definitions, 30

inheritance

- kinds of relationship between objects, 409

instance document

- adding namespaces, 199
- validating against schema, 77
- validity, 197

int datatype, 70

integer datatype, 69

intelligent content, 462

Intentions element, 568, 575

- collection of codeBlock elements, 570
- possesses procedural code, 570

internet datatypes

- domain names, 159
- e-mail addresses (mailto:), 162
- IP addresses, 158
- URLs and URLs, 160
- URLs, 161

IP addresses

- internet datatypes, 158

K

key element

- attributes, 617
- name attribute, 333

key element (Schematron)

- allows accessing of XSLT keys, 505
- path and name attributes, 505

key mechanism, 322

key() function, 366

key-keyrefs relationships

- modeling the logical schema relationship, 449

keyref element

- attributes, 618
- refer attribute, 333

keyref mechanisms, 322

keys and key references, 332

- naming conventions, 334
- specify scope of, 332
- syntax, 333

L

language codes (ISO 639)

- older two-character language codes, 156

Language Codes (ISO 639), 155

- examples, 155

language datatype, 66

leaf elements

- advantage of XML Schema over DTDs, 477

length facet, 109

- attributes, 636

lexical space - datatype properties, 45, 46

list datatypes

- always derived types, 49
- base type known as itemType, 49

list element

- attributes, 618

local element declarations, 17

local schema-validity, 31

long datatype, 70

losing tokens, 585

- PNML-manipulating stylesheet, declaring, 584

M

magazine publisher goes XML example, 397

- establishing system requirements, 397
 - functions XML applied to, 398
 - identify key players, 399
 - requirements definition and analysis, 400
- publisher background, 399

mailto_Name datatype

- pattern value, 162

markup languages

- history, 459

maxExclusive facet, 113

- attributes, 641

maxInclusive facet, 113

- attributes, 640

maxLength facet, 109, 382, 383

- attributes, 637

maxOccurs

- element occurrence indicators, 20

minExclusive facet, 113

- attributes, 641

minInclusive facet, 113, 642

- attributes, 642

minLength facet, 109, 637

- attributes, 637

minOccurs

- element occurrence indicators, 20

- mixed content, 88, 482**
 - combination of child elements and character data, 88
 - Model**
 - core data in application, 566
 - core functionality, 566
 - state information, 566
 - implementation, 568
 - Model - View - Controller**
 - see MVC.
 - Model element, 568**
 - declaration, 569
 - model group definition**
 - creating, 81
 - model group navigation, 384**
 - model group schema components, 81**
 - modular schemas**
 - creating, 269, 315
 - dependencies, 317
 - documenting, 317
 - functionality gained, 269, 270
 - granularity of schema components, 316
 - include and redefine mechanism, 279
 - practical for internal schemas, 319
 - modularization of XHTML, 269**
 - core modules, 269
 - optional modules, 269
 - points of conformance, 269
 - MSXML 4 extensions, 516**
 - Schematron example, 517
 - MSXML4 and XDK (Microsoft)**
 - working with, 661
 - XML Schema conformance, 660
 - multi-character escape, 124**
 - table of pre-defined sequences, 126
 - multiple schemas**
 - how many namespaces should be used, 284
 - chameleon design, 290
 - design approach comparison, 291
 - heterogeneous namespace design, 285
 - homogeneous namespace design, 288
 - mustUnderstand SOAP-ENV attribute, 541**
 - MVC**
 - Controller, 567
 - implementation, 570
 - Model, 566
 - implementation, 568
 - seperates state and behavior from visual interface, 565
 - View, 567
 - implementation, 569
- N**
- NAFTA and British Addresses example, 175**
 - Simple Child Element Form – Canadian (CAN), British (GBR), and Mexican (MEX) Addresses, 176
 - US address - detailed street data attribute form, 177
 - name collisions, 297**
 - example, 297
 - Proxy Schemas, 297
 - name datatype, 67**
 - name element (Schematron), 499**
 - named complex type, 15, 76, 87**
 - named model groups, 81, 87**
 - creating re-usable building blocks, 82
 - defined globally, 82
 - repeating model groups, 83
 - occurrence operators, 83
 - re-use, 238
 - example, 239
 - Venetian Blind Model
 - advantages of using, 224
 - namespace coercion, 273**
 - namespace information item, 30**
 - namespace prefix, 186**
 - qualifying elements, 18
 - namespace qualification, 198**
 - namespace support, 101**
 - purposes, 101
 - namespaces, 183**
 - declaring, 186
 - reserved attributes, 186
 - declaring and using in XML document, 185
 - defaulting, 189
 - examples, 190
 - defining for markup, 195
 - creating vocabularies that do belong to a namespace, 196
 - creating vocabularies that do not belong to a namespace, 195
 - xs:noNamespaceSchemaLocation* attribute, 196
 - elements and attributes qualification, 201
 - hide or expose, 293
 - different models, 293
 - name collisions, 297
 - should default namespace be the targetNamespace or XML Schema namespace, 294
 - infoset contributions to namespace qualification, 199
 - local control of namespace qualification, 207
 - models for local versus global declarations, 208
 - Russian Doll Design, 209
 - Salami Slice Design, 210
 - Venetian Blind Model, 214
 - namespace qualification, 198
 - no namespace, 191
 - rules to remember, 206
 - scope, 187
 - using multiple namespaces in document, 187
 - support with DTDs is difficult, 183
 - uniqueness of attributes, 192
 - use no default namespace declaration, 294
 - way of identifying where markup has come from, 185
 - namespaces for built-in data types, 52**
 - declaring, 53
 - naming conventions, 229**
 - abbreviated names, 229
 - consistency, 230
 - descriptiveness, 230
 - distinguish type and group names, 231
 - intuitiveness, 229
 - overloaded names, 230
 - Dot.Notation, 231
 - examples, 231, 232
 - readability, 230
 - naming elements, 478**
 - navigating between schemas**
 - incorporating other schemas, 369
 - navigating schemas**
 - navigating between components
 - key definitions, 365
 - linking in documentation, 368
 - resolving definitions, 367
 - NCName datatype, 67**
 - negative character group, 124**

- negativeInteger datatype, 70**
- nil values, 92**
 - analogous to the use of 0 versus NULL in SQL, 92
- NMTOKEN datatype, 68**
- NMTOKENS datatype, 68**
- nodes**
 - locating with XPath, 325
- noNamespaceSchemaLocation attribute, 29**
- nonNegativeInteger datatype, 69**
- nonPositiveInteger datatype, 69**
- normal forms, 343**
 - problems solved by, 343
 - rules do not always hold true for XML data, 343
- normalization**
 - simplifies structure of data, 336
- normalizedString datatype, 66**
- normalizing XML, 347**
 - document class requirements, 347
 - document fragments, 349
 - advantages, 349
 - structural requirements, 347
- NOTATION datatype, 55**
- notation element, 55**
 - attributes, 619
- numeric datatypes, 53, 57**
- numeric/non-numeric facets, 109**

O

- OASIS model, 477**
- occurrence constraints, 79**
- order facet, 107**
 - ordered schema component, 107
 - total or partial order, 107
- orphaned elements, 450**
- overloaded names, 230**
 - retrieval of information using SAX or DOM, 231

P

- p element (Schematron), 503**
- P/T nets**
 - graphical representation, 580
 - reasons to use, 580
- parallel design**
 - makes structures easier to learn, 235
 - see *also* consistent structure, 235
- param element**
 - XSLT elements, 574
- parent and child items**
 - one-to-one mapping desirable, 231
- Parsers and XML Development Kits (XDKs)**
 - MSXML4 and XDK (Microsoft), 660
 - xerces-c (C++) and xerces-j (Java), 659
 - XML Schema Processor and XDKs (Oracle), 661
 - XML4C and XML4J (IBM), 660
- pattern elements (Schematron)**
 - id attribute, 501
- pattern facet, 112, 638**
 - attributes, 638
 - regular expressions, 118

- Perl regex module**
 - regex syntax comparison, 133
- permissive schemas, 228**
- Petri Net markup language**
 - see PNML.
- Petri Nets**
 - digraphs, 579
 - mathematical graphs, 579
 - P/T net, 580
- phase element (Schematron), 501**
- Physics Press goes online case study, 461**
 - domain-specific markup
 - mathematical equations, 468
 - future proofing, 474
 - granularity required, 467
 - identifying data types
 - built-in date datatypes, 466
 - intelligent content, 462
 - managing document schema development, 462
 - managing schema design process
 - dividing the task, 481
 - re-purposing content, 462, 472
 - turning the design into a schema, 482
- place/transition net**
 - see P/T net.
- places**
 - digraphs, 580
- PNML**
 - suited to the representation of simple Petri Nets, 581
- PNML document**
 - general structure, 581
 - arcs, 583
 - place element, 582
 - transitions, 583
- positive character group, 124**
- positiveInteger datatype, 70**
- possibleHouseForSale.xsd schema**
 - creating documentation, 358
 - turn schema into HTML page, 359
- prescriptiveness of schemas, 470**
 - meta-data, 471
 - optional parts, 470
- primary components, 102**
- primary key**
 - definition, 337
- primitive datatypes, 47**
 - cannot be defined from smaller components, 47
- process model, 425**
 - iterative development, 426
- processing XML documents, 228**
- properties of fields**
 - definition, 338
- protecting markup - controlling re-use, 313**
 - enforcing namespace qualification, 313
 - preventing derivation of types, 314
 - block for control in instance documents, 315
 - final types, 314
 - fixed attribute for simpleTypes, 314

Q

- QName datatype, 56**
- qualification**
 - namespaces, 198
- quantifiers, 130**

R

- Rational Rose**
 - modeling UML, 402
- RDBMS**
 - maintaining stability, security, and accuracy of data, 336
 - normalization, 336
- redefine element, 277, 303, 370, 374**
 - attributes, 619
- redefine mechanism, 276**
 - altering schema constructs, 303
 - schema constructs, altering
 - example, 304
- referencing elements and attributes**
 - re-use, 238
 - example, 239
- referential integrity, 342**
- regex syntax comparison, 131**
 - Perl, 133
 - table of differences between Perl and XML Schema, 133
 - unicode, 132
- regexes**
 - see regular expressions and XML Schema regexes
- regular expressions, 118**
 - XML Schema regexes, 119
 - character classes, 122
- REgular LAnguage for XML**
 - see Relax NG.
- relational database schema**
 - logical schema, 442
- relative location path (XPath), 327**
- Relax NG, 523**
 - compared to XML Schemas and DTDs, 524
 - modularity, 526
 - namespace support, 525
 - structural validation, 524
 - validation, 526
 - example schema, 523
 - namespace, 524
 - why use, 527
- repeating model groups**
 - group element
 - minOccurs and maxOccurs attributes, 83
- report element (Schematron)**
 - diagnostics attribute, 500
- re-purposing content, 462, 472**
- reserved attributes, 186**
- restriction element, 138, 302, 303, 381, 382, 620**
 - attributes, 622
- re-use of schema constructs**
 - complex types, 238
 - cross-schema re-use, 237
 - example, 238
 - internal re-use, 237
 - named model groups, 238
 - referencing elements and attributes, 238
- rich tokens, 587**
- Russian Doll Design, 209**
 - characteristics, 212
 - child declarations, decoupled, 212
 - child elements and attributes are opaque, 212
 - compared to Salami Slice Design, 214
 - localized scope, 212

S

- Salami Slice Design, 210**
 - characteristics, 213
 - child elements and attributes are transparent, 213
 - declarations are considered coupled, 213
 - differences to Russian Doll Design, 214
 - elementFormDefault and attributeFormDefault settings
 - have no effect, 213
 - global scope, 213
 - use elements by reference, 213
- SBP (Schema Based Programming)**
 - declarative programming, 564
 - research project, 563
- SBP application implementation, 565, 572**
 - event processing, 578
 - notify () function, 578
 - initial download actions, 573
 - Controller activation, 574
 - shell user interface creation, 573
 - scheme of processing, 572
- SBP future directions, 599**
 - development tools, 599
 - expressive potential of Petri Nets, 600
- SBP Schema, 567**
 - Application element, 568
 - Controller, 570
 - event elements, 568
 - Intentions element, 568
 - Model element, 568, 569
 - required code and data, 567
 - view element, 569
 - view elements, 568
- Schema Based Programming**
 - see SBP.
- Schema checkers**
 - XML Schema Quality Checker (IBM), 662
 - XML Schema Validator, 662
- schema components, 102**
 - helper components, 103
 - primary components, 102
 - secondary components, 102
- schema constructs, altering, 299**
 - deriving new types and element content models, 300
 - extension, 300
 - restriction, 302
 - forcing substitution with abstract elements and types, 308
 - methods of altering markup that we import and include, 300
 - redefine, 303
 - substitution groups, 306
- schema conversion tools, 666**
- Schema datatypes reference, 629**
 - constraining facets, 636
- schema design**
 - guidelines, 436
- schema design fundamentals, 227**
 - choice of structures, 251
 - choices of content models, 252
 - limiting choice, 252
 - choices for simple content using enumeration, 250
 - enumerating a simple type, 250

schema design fundamentals (continued)

- consistency, order and occurrence, 233, 236
 - consistent structure, 233
 - document structures, 234
 - effects on processing requirements, 236
 - parallel design, 235
 - re-use of schema constructs, 237
- creating enumerated simple Type, 250
- creating hierarchical structures, 249
- declaring order and occurrence, 241
- elements and their content models
 - choosing between, 254
 - few choices and small content models, 254
 - several choices or larger content models, 256
 - using substitution groups, 257
- fixed order, flexible occurrence, 258
- flexible order and occurrence, 258, 259
 - allcompositor, 261
- grouping and hierarchy, 245
- naming conventions, 229
- orders, 235
 - applying, 235
- prescriptive versus descriptive models, 228
- restricting choices, 249
 - fixed values, 250
- structure should follow users experience, 234
- structures kept together in logical sequence, 252

schema design process

- creating HTML-based, hyper-linked documentation, 490
- documentation, 490
- managing, 481
 - dividing the task, 481
- managing ongoing development, 491
- overview, 463
- strategic issues, 467
 - choosing an appropriate level of granularity, 467
 - choosing many specific or few general elements, 473
 - coping with legacy data, 469
 - documentation and usage guidelines, 476
 - domain-specific markup, 468
 - enumerating the different types of cited object, 474
 - extensibility and future proofing, 474
 - generic elements, 475
 - prescriptiveness, 470
 - rendering, 468
 - smallest licensable units, 472
 - use of inappropriate elements, 475
- tactical issues
 - built in data types, 477
 - creating data types, 478
 - data types, 477
 - element naming, 478
 - section schemes, 479
 - standard modules, 476
- testing the schema, 489
- turning design into process, 482

schema development group

- developing a document schema, 462

schema element, 373

- attributeFormDefault attribute, 201, 363
- attributes, 622
- blockDefault attribute, 146
- elementFormDefault attribute, 201, 204, 363
- finalDefault attribute, 146, 363
- targetNamespace attribute, 196, 362, 367
 - populating, 198

Schema technologies

- ASN.1, 531
- attributes and import element, 530
- Examplotron, 528

- XML Spy, 530

schemaLocation attribute, 29

schemas, creating from multiple namespace

- examples, 270

Schematron, 493

- ability to call external documents or web transactions, 559
- and XPath 2.0, 516
- and XSLT 2.0, 516
- co-location
 - example, 509
 - extracting Schematron schema, 512
 - reasons for, 509
- co-location limitations
 - different names, same type, 515
 - namespace re-mapping, 516
 - same name, different type, 514
- co-location with XML Schema, 508
 - validation process, 508
- compiling a basic validator, 496
 - using Saxon, 497
- compiling and running an HTML validator, 498
- compiling and running an XML validator, 497
 - failed-assert element, 498
- complementary features
 - allows schema to specify root elements, 507
 - designed to be layered over other schemas, 507
 - runs in any environment that supports XSLT, 507
 - Schematron allows arbitrary logical constraints, 506
 - supports XSLT/XPath constraints of any complexity, 508
 - XML Schema is grammar-based - Schematron is rule-based, 506
- complementary to XML Schema, 494
- declare namespace, 558
- error message specification, 494
- formatting/documentation, 503
- inheritance, 503
- phases, 500
 - example, 501
- rule based schema language, 494
 - gives greater flexibility, 494
- usage schemas
 - can be layered on top of defitional schemas, 493
- using with XML schema, 557
 - non-XML schema constraints, 557
- validation process, 496
- writing a pattern containing a rule, 558

Schematron and XSLT 2.0

- sample using MSXML 4 extensions, 516
- validation, 518

Schematron elements

- active element
 - pattern attribute, 501
- assert element
 - diagnostics attribute, 500
- diagnostic element, 500
- diagnostics element, 500
- dir element, 503
- emph element, 503
- extends element, 503
 - abstract attribute, 503
- failed-assert element, 498
- key element, 505
- name element, 499
- p element, 503
- pattern elements
 - id attribute, 501
- phase element, 501
- report element
 - diagnostics attribute, 500

Schematron elements (continued)

- span element, 503
- value-of element
 - select attribute, 500

Schematron namespace

- elements, 496

Schematron Schema example

- Purchase Order, 495

schematron technologies

- Relax NG, 523

script element, HTML, 575**s-e range, 123****second grade XML normalization**

- document class requirements, 348
- structural normalization suggestions, 348

second normal form

- rule, 345

secondary components, 102**section schemes**

- nesting, 479

selector element

- attributes, 623

selector, XPath expression, 327**separating characters, 230****sequence element, 14, 79, 86**

- attributes, 623
- maxOccurs attribute, 388
- minOccurs attribute, 388

short datatype, 70**Simple Type Definition, 47****simple types, 44**

- defined, 14
- differences to complex types, 14

simpleContent element, 138

- id attribute, 624

simpleType element, 47, 378, 379, 381

- attributes, 624
- final attribute, 364

single character escape, 124

- table of pre-defined sequences, 125

single character reference, 123**single normal character, 123****smallest licensable units, 472****SOAP**

- advantages over Perl, 546
- links, 542
- message structure, 539
 - Body element, 541
 - envelope element, 540
 - Header element, 540
 - SOAP Fault element, 542
- overview, 539
- provides transport layer, 537

SOAP Body element, 541**SOAP Envelope element, 540****SOAP Fault element, 542****SOAP Header element, 540**

- SOAP-ENV attributes, 541
- terminology, 540

SOX, 44**span element (Schematron), 503****stafflist example, 322**

- four things demonstrated, 325
- locating nodes with XPath, 325
- relationships and uniqueness constraints, 334

- schema, 323
- specifying identity constraints, 326
- structure, 322

Standard Generalized Markup Language

- see SGML.

standard modules, 476

- advantages, 477
- SGML defacto standards, 477
- should be used where possible, 476
- will soon be developed, 476

string datatypes, 54

- derived, 66
- primitive, 53, 54

structural requirements

- normalizing XML, 347

stylesheet element

- XSLT elements, 574

substitution groups, 309

- allows substitution of elements, 253
- caveats, 306
- handling choice between elements, 257
- head element, 143
- limitations, 253
- more extensible, 253
- more helpful when, 254
- one element acts as the head, 253

T**targetNamespace**

- use as default namespace for the schema document
 - example, 295
- use as default namespace for the schema document
 - option, 294

templates

- link mode
 - \$qname parameter, 368
 - to retrieve target namespace, 363

text only content, 482**text-only element content models**

- re-use, 240

third normal form

- document class requirements, 348
- rule, 346
- structural normalization suggestions, 348

time datatype, 62**token datatype, 66****tokens, digraphs, 580****Topologi**

- interactive Schematron development tool
 - graphical validation tool, 520
 - Schematron development tool, 520

totalDigits facet, 114

- attributes, 642
- values, 114

transform element

- XSLT elements, 574

transitions, digraphs, 580**traversal path**

- complex example, 453
- example, 445
- importance of, 444

TurboXML (TIBCO Extensibility)

- XML Schema conformance, 664

U

UDDI

directory services, 560

UML

model categories, 401
Rational Rose, 402
see Unified Modeling Language.
strengths, 400
types of diagrams, 401

UML and XML modeling, 400

UML reference

classes and objects, 649
components, 654
inheritance, 652
multiple inheritance, 652
object interactions, 653
relationships, 649
aggregation, 651
association, 650
composition, 651
dependency, 650
multiplicity, 651
naming an association, 652
states, 652

unicode

levels of support for regex engines, 132

unicode regex features

XML Schema support, 132

Uniform Resource Identifier

see URL.

Uniform Resource Locator

see URL.

union datatypes, 50

union element, 625

attributes, 625

union types

base types known as member types, 50
example, 50

unique element

attributes, 626

unique identifier, 340

unique values, 329

ensuring unique values within a document instance, 329
flexibility, 329

uniqueness constraint mechanism, 100

uniqueness constraints, 322

defining is three-step process, 330
example, 330

Universal Discovery, Description and Integration

see UDDI.

Unobtanium sample application, 589

Controller

define an event, 596
Initialize function, 596
writing the event handler, 597

Model

devise P/T net, 591
schematic of the process, 590

View

prototype of user interface, 593

unqualified elements

borrowing constructs from different target namespace, 283
borrowing constructs from the same namespace, 278

unsignedByte datatype, 70

unsignedInt datatype, 70

unsignedLong datatype, 70

unsignedShort datatype, 70

URLs

internet datatypes, 161

user-defined datatypes

based on existing datatypes, 48

user-defined integrity, 343

validating an instance document, 29

indicating location of schema, 29

V

validation

globally declared elements can be used as starting point, 199

validation tools

validation with XSV (XML Schema Validator), 31

validation with XSV (XML Schema Validator), 31

delivery receipt schema example, 36
noNamespaceSchemaLocation attribute, 36
error messages, 33
validating a simple schema, 32
validating an instance document, 34

value constraints

attribute declarations, 25
element content, 22

value space - datatype properties, 45

value-of element (Schematron)

select attribute, 500

Venetian Blind Model, 214

compared to Salami Slice Design, 215
complex types and model groups as building blocks, 221
encourages use of complexTypes, 214
example, 215
issues with complex types, 219
adding attributes, 220
associate each element name with only one type, 219
must be careful when defining, 220
named model groups
advantages of using, 224
schema example, 216
allowing re-use of schema components, 218
components, 217

View

implementation, 569
as HTML, 569
implementation of visual interface, 567
linked to Model, 567

view element

declaration, 569

W

W3C XML Schema

bridges divide between documents and data, 229
define different content models can be defined for elements with same name, 231

W3C XML Schema namespace

use as default namespace for the schema document, 296
use as default namespace for the schema document option, 294

W3C XML Schema Recommendation

- advantages over DTDs, 12
- three parts, 12
- validation specification, 30
 - infoset, 30
 - local schema-validity, 30

W3C XML Schema Working Group

- aims for defining structure of documents, 11
- aims in offering primitive data typing, 11

waiting states, 584

- PNML-manipulating stylesheet
 - declaring, 584

Web Services Description Language

- see WSDL.

web services links, 561**Web Services View**

- UDDI, 560
- WSDL, 560

whiteSpace facet, 111

- attributes, 639
- values, 111

WSDL, 560**X****xerces-c (C++) and xerces-j (Java)**

- specifying a schema, 660
- working with, 660
- XML Schema conformance, 659

XHTML

- modularization, 269

XHTML Namespaces

- example, 188

XLink

- not yet a Recommendation, 575

XML data example, 460**XML document example, 460****XML domain/key norm**

- document class requirements, 348
- structural normalization suggestions, 348

XML ID type, 329**XML Information Set**

- see infoset.

XML Schema built-in datatypes

- derived types, 633
- primitive types, 630

XML Schema Datatypes Namespace, 193**XML Schema elements**

- all element, 391, 603
- annotation element, 27, 28, 490, 604
- any element, 605
- anyAttribute element, 606
- appInfo element, 27, 28, 380, 490, 558, 607
- attribute element, 16, 363, 607
 - form attribute, 207
- attributeGroup element, 84, 85, 608
- choice element, 609
- complexContent element, 138, 610
- complexType element, 13, 76, 88, 378, 610
 - block attribute, 364
 - compositors must be specified inside, 14
 - final attribute, 364
- documentation element, 27, 28, 358, 490, 612
- element element, 13, 16, 363, 378, 387, 390, 612
 - block attribute, 364

- default attribute, 22
- final attribute, 363
- fixed attribute, 22
- form attribute, 207
- extension element, 139, 614
- field element, 615
- group element, 81, 391, 615
- import element, 373, 616
 - namespace attribute, 280, 371
 - schemaLocation attribute, 280
- include element, 273, 374, 617
 - schemaLocation attribute, 369
- key element, 617
 - name attribute, 333
- keyref element, 618
 - refer attribute, 333
- list element, 618
- notation element, 55, 619
- redefine element, 277, 303, 370, 374, 619
- restriction element, 138, 302, 303, 381, 382, 620
- schema element, 366, 373, 622
 - attributeFormDefault attribute, 201, 363
 - blockDefault attribute, 146
 - elementFormDefault attribute, 201, 204, 363
 - finalDefault attribute, 146, 363
 - targetNamespace attribute, 196, 198, 362, 367
- selector element, 623
- sequence element, 14, 86, 623
 - known as a compositor, 14
 - maxOccurs attribute, 388
 - minOccurs attribute, 388
- simpleContent element, 138, 624
- simpleType element, 47, 378, 379, 381, 624
 - final attribute, 364
- union element, 625
- unique element, 626

XML Schema elements and attributes

- quick reference, 603

XML Schema for Instance Documents namespace

- noNamespaceSchemaLocation attribute, 29
- schemaLocation attribute, 29

XML Schema instance attributes, 626**XML Schema Instance Namespace**

- attributes, 194
- xsi:nil attribute, 194
- xsi:noNamespaceSchemaLocation attribute, 194
- xsi:schemaLocation attribute, 194
- xsi:type attribute, 194

XML Schema Namespace, 193**XML Schema Processor and XDKs (Oracle), 661****XML Schema Quality Checker (IBM)**

- XML Schema conformance, 662

XML Schema regexes

- atoms, 121
- basic grammar, 119
- character class escapes, 124
- character class expressions, 122
- meta characters, 121
- quantifiers, 130

XML Schema tools, 657

- authoring tools, 663
- Parsers and XML Development Kits (XDKs), 657
 - table of comparisons, 658
- schema checkers
 - table of comparison, 662
- schema conversion tools, 666

XML Schema Validator, 662

- XML Schema conformance, 662

XML Schemas

- advantage over DTDs
 - leaf elements, 477
- advantages over XML IDs and IDREFs, 332
- all element, 79
- annotations, 27
- attribute declarations, 23
- background, 10
- borrowing constructs from different target namespace, 280
 - import mechanism, 280
 - unqualified elements, 283
- borrowing constructs from the same namespace
 - unqualified elements, 278
 - using include and redefine, 279
- borrowing constructs from the same namespace or no namespace, 273
 - include mechanism, 273
 - redefine mechanism, 276
- choice element, 79
- comments, 27
- complex types, 14
- creating for existing database, 441
 - business problem, 441
 - determining documents scope, 443
 - migrating database to XML, 443
 - relational database schema, 442
 - technologies to be used, 441
- creating schemas from multiple namespace examples, 270
- declaring elements, 13
- deriving complex types, 101
- DTFDs, 10
- element declarations, 16
- elements
 - type definitions, 16
- Expansion Slots, 507
- grammar based, 494
- identity constraints, 100, 322
- instance document, 13
- multiple schemas
 - how many namespaces should be used, 284
- namespace support, 101
- namespaces, 183
 - flexibility of support, 183
 - reasons for use, 183
- need for, 10
- processing with XSLT, 512
- relative XPaths, 327
- representing relationships
 - aggregation, 433
 - association, 435
 - composition, 433
 - inheritance, 435
- representing schemas, 433
- representing type information, 444
- re-using information, 270
 - rules, 270
- schema components, 102
- sequence element, 79
- Simple Type Definition, 47
- simple types, 14
- symbol spaces for schema components, 366
- syntax example, 13
- traversal path
 - creating, 444
 - importance of, 444
 - using constructs from other schemas, 101
- XML Schema Datatypes Namespace, 193
- XML Schema Instance Namespace, 194
- XML Schema Namespace, 193
- XSLT, 357
- XML Schemas and XSLT**
 - creating a form, 375
 - accessing a form from instances, 375
 - creating documentation, 358
 - schema information, 360
 - summarizing components, 361
 - generating a form, 376
 - navigating model groups, 384
 - navigating schemas, 364
 - navigating between components, 365
- XML Schemas, creating, 445**
 - complex example, 450, 452
 - document scope, 452
 - logical schema attributes, 454
 - logical schema entities, 453
 - logical schema relationships, 454
 - orphans, 455
 - traversal path, 453
 - creating user-defined datatypes, 446
 - common user-defined type definitions, 447
 - lookup tables, 446
 - modeling logical schema attributes, 448
 - elements or attributes, 448
 - modeling logical schema entities, 448
 - modeling the logical schema relationship, 449
 - orphaned elements, 450
 - using key-keyrefs relationships, 449
 - using parent-child relationships, 449
- XML Spy 3.5 and 4.0 (Altova)**
 - XML Schema conformance, 665
- XML4C and XML4J (IBM)**
 - schema conformance, 660
- XML-Data, 44**
- XPath**
 - features and capability, 508
 - locating nodes, 325
- XPath 2.0**
 - and Schematron, 516
- XPath expressions**
 - required for defining identity constraints, 327
- XPath location paths, 327**
- xsi:nil attribute**
 - XML Schema Instance Namespace, 194
- xsi:noNamespaceSchemaLocation attribute**
 - XML Schema Instance Namespace, 194
- xsi:schemaLocation attribute**
 - XML Schema Instance Namespace, 194
- xsi:type attribute, 135**
 - XML Schema Instance Namespace, 194
- XSLT, 357**
 - capability provided, 357
 - convert XML Schema into HTML page, 360
 - creating a form, 375
 - creating documentation, 358
 - summarizing components, 361
 - generating a form, 376
 - navigating schemas, 364
 - summarizing components, 361
- XSLT 2.0**
 - and Schematron, 516

XSLT elements

- apply-templates element, 587
- copy element, 586
- for-each element, 363, 575, 595
- import element, 575
- key elements, 365, 368, 372
- param element, 574
- stylesheet element, 574
- transform element, 574, 578

XSLT functions

- document() function, 363, 376, 494
- generate-id() function, 365
- key() function, 366

XSLT namespace, 568

XSLT stylesheet for PNML, 583

- controlling the transformation, 585
 - copy element, 585
- copy element
 - places gaining tokens, 587
 - places losing tokens, 586
- identifying states which gain and lose tokens, 583
 - eight key variables, 583
- rich tokens, 587

XSV (XML Schema Validator)

- validation, 31

