

Inhalt

Vorwort	21
1 Einführung	23
1.1 Kleines Einstiegsprojekt zum Kennenlernen	23
1.1.1 Ein erstes XML-Dokument	23
1.1.2 Standardausgabe im Webbrowser	24
1.1.3 Wohlgeformtheit ist ein Muss	25
1.1.4 Gültige Dokumente per DTD oder Schema	25
1.1.5 Formatierte Datenausgabe	28
1.2 XML – universale Metasprache und Datenaustauschformat	29
1.2.1 Unabhängigkeit von Anwendungen und Plattformen	29
1.2.2 SGML → HTML → XML	30
1.2.3 Lob des Einfachen	31
1.2.4 Inhaltsbeschreibungssprache	31
1.2.5 Trennung von Inhalt und Form	31
1.2.6 Vom Dokumentformat zum allgemeinen Datenformat	32
1.2.7 Globale Sprache für den Datenaustausch	32
1.3 Übersicht über die Sprachfamilie XML	33
1.3.1 Kernspezifikationen	34
1.3.2 Ergänzende Spezifikationen	35
1.3.3 Programmierschnittstellen	35
1.3.4 XML-Anwendungen	36
1.4 XML-Editoren und Entwicklungsumgebungen	36
1.4.1 Editoren für XML	37
1.4.2 Schema- und Stylesheet-Designer	39
1.4.3 Entwicklungsumgebungen mit XML-Unterstützung	40
1.4.4 XML-Dokumente über Standardanwendungen	40
1.4.5 Parser und andere Prozessoren	41
1.5 Anwendungsbereiche	42
1.5.1 XML-Vokabulare	42
1.5.2 Datenaustausch zwischen Anwendungen	45
1.5.3 Verteilte Anwendungen und Webdienste	46

2.1	Aufbau eines XML-Dokuments	47
2.1.1	Entitäten und Informationseinheiten	47
2.1.2	Parsed und unparsed	48
2.1.3	Die logische Sicht auf die Daten	49
2.1.4	Der Prolog	51
2.1.5	Zeichencodierung	51
2.1.6	Standalone or not	52
2.1.7	XML-Daten – der Baum der Elemente	53
2.1.8	Start-Tags und End-Tags	54
2.1.9	Elementtypen und ihre Namen	55
2.1.10	Regeln für die Namensgebung	56
2.1.11	Elementinhalt	57
2.1.12	Korrekte Schachtelung	57
2.1.13	Attribute	58
2.2	Die Regeln der Wohlgeformtheit	60
2.3	Elemente oder Attribute?	60
2.4	Reservierte Attribute	61
2.4.1	Sprachidentifikation	61
2.4.2	Leerraumbehandlung	62
2.5	Entitäten und Verweise darauf	62
2.5.1	Eingebaute und eigene Entitäten	62
2.5.2	Zeichenentitäten	63
2.6	CDATA-Sections	64
2.7	Kommentare	64
2.8	Verarbeitungsanweisungen	65
2.9	Namensräume	66
2.9.1	Das Problem der Mehrdeutigkeit	66
2.9.2	Eindeutigkeit durch URIs	66
2.9.3	Namensraumname und Präfix	67
2.9.4	Namensraumdeklaration und QNamen	67
2.9.5	Einsatz mehrerer Namensräume	68
2.10	XML-Version 1.1	70

3	Dokumenttypen und Validierung	71
3.1	Metasprache und Markup-Vokabulare	71
3.1.1	Datenmodelle	71
3.1.2	Selbstbeschreibende Daten und Lesbarkeit	72
3.1.3	Dokumenttyp-Definition – DTD	72
3.1.4	Alternativen zu DTD	73
3.1.5	Vokabulare	73
3.2	Regeln der Gültigkeit	74
3.3	DTD oder Schema?	75
3.4	Definition eines Dokumentmodells	75
3.4.1	Interne DTD	76
3.4.2	Externe DTD	77
3.5	Deklarationen für gültige Komponenten	78
3.5.1	Vokabular und Grammatik der Informationseinheiten	78
3.5.2	Syntax der Dokumenttyp-Deklaration	79
3.5.3	Syntax der Elementtyp-Deklaration	79
3.5.4	Beispiel einer DTD für ein Kursprogramm	80
3.5.5	Inhaltsalternativen	82
3.5.6	Uneingeschränkte Inhaltsmodelle	83
3.5.7	Gemischter Inhalt	84
3.5.8	Inhaltsmodell und Reihenfolge	84
3.5.9	Kommentare	85
3.5.10	Die Hierarchie der Elemente	85
3.6	Dokumentinstanz	86
3.7	Attributlisten-Deklaration	87
3.7.1	Aufbau einer Attributliste	88
3.7.2	Attributtypen	89
3.7.3	Verwendung der Attributlisten	90
3.8	Verweis auf andere Elemente	91
3.9	Verwendung von Entitäten	91
3.9.1	Interne Entitäten	92
3.9.2	Externe Entitäten	93
3.9.3	Notationen und ungeparste Entitäten	94
3.9.4	Verwendung von Parameterentitäten	95
3.9.5	Interne Parameterentitäten	95
3.9.6	Externe Parameterentitäten	96

3.10 Formen der DTD-Deklaration	96
3.10.1 Öffentliche und private DTDs	97
3.10.2 Kombination von externen und internen DTDs	97
3.10.3 Bedingte Abschnitte in externen DTDs	98
3.11 Zwei DTDs in der Praxis	99
3.11.1 Die Auszeichnungssprache DocBook	99
3.11.2 Das grafische Format SVG	102

4 Inhaltsmodelle mit XML Schema 107

4.1 XML Schema – der XML-basierte Standard	107
4.1.1 Defizite von DTDs	107
4.1.2 Anforderungen an XML Schema	108
4.1.3 Die Spezifikation des W3C für XML Schema	108
4.2 Erster Entwurf eines Schemas	109
4.2.1 Verknüpfung von Schema und Dokument	112
4.2.2 Der Baum der Schema-Elemente	113
4.2.3 Elemente und Datentypen	114
4.2.4 Komplexe Typen mit und ohne Namen	114
4.2.5 Sequenzen	115
4.2.6 Vorgegebene und abgeleitete Datentypen	116
4.2.7 Wie viel wovon?	116
4.3 Genereller Aufbau eines XML-Schemas	116
4.3.1 Das Vokabular	116
4.3.2 Die Komponenten eines XML-Schemas	117
4.4 Datentypen	117
4.4.1 Komplexe Datentypen	118
4.4.2 Inhaltsmodelle und Partikel	118
4.4.3 Erweiterbarkeit durch Wildcards	120
4.4.4 Einfache Typen	120
4.4.5 Benannte oder anonyme Typen	121
4.4.6 Vorgegebene und benutzerdefinierte Datentypen	122
4.4.7 XML Schema 1.0 – Datentypen – Kurzreferenz	122
4.4.8 Werteraum, lexikalischer Raum und Facetten	126
4.4.9 Ableitung durch Einschränkung	127
4.4.10 Muster und reguläre Ausdrücke	128
4.4.11 Grenzwerte	130
4.4.12 Listen und Vereinigungen	130
4.4.13 Facetten der verschiedenen Datentypen	131

4.5	Definition der Struktur des Dokuments	133
4.5.1	Deklaration von Elementen	133
4.5.2	Attribute	135
4.5.3	Elementvarianten	136
4.5.4	Namensräume in XML Schema	136
4.5.5	Zielnamensraum	137
4.5.6	Umgang mit lokalen Elementen und Attributen	139
4.6	Häufigkeitsbestimmungen	142
4.7	Default-Werte für Elemente und Attribute	143
4.8	Kompositoren	144
4.8.1	<xsd:sequence>	144
4.8.2	<xsd:all>	145
4.8.3	<xsd:choice>	145
4.8.4	Verschachtelte Gruppen	146
4.9	Arbeit mit benannten Modellgruppen	146
4.10	Definition von Attributgruppen	148
4.11	Schlüsselemente und Bezüge darauf	148
4.11.1	Eindeutigkeit	149
4.11.2	Bezüge auf Schlüsselemente	150
4.12	Kommentare	152
4.13	Ableitung komplexer Datentypen	153
4.13.1	Erweiterungen komplexer Elemente	153
4.13.2	Einschränkung komplexer Elemente	154
4.13.3	Steuerung der Ableitung von Datentypen	155
4.13.4	Abstraktionen	156
4.13.5	Gemischtwaren	157
4.13.6	Leere oder Nichts	158
4.13.7	Wiederverwendbarkeit	159
4.14	Designvarianten	160
4.14.1	Babuschka-Modelle	160
4.14.2	Stufenmodelle	162
4.15	Übernahme von Schema-Definitionen	163
4.15.1	Schemas inkludieren	164
4.15.2	Schemas importieren	165
4.15.3	Zuordnung von Schemas in XML-Dokumenten	170
4.16	XML Schema 1.0 – Kurzreferenz	171

4.17 Exkurs zu XML Schema 1.1	181
4.17.1 Versicherungen	181
4.17.2 Lockerungen der Regeln für Inhaltsmodelle	182
4.17.3 Offene Modelle	183
4.17.4 Schemaweite Attribute	184
4.17.5 Anpassen von Schemas	184
4.17.6 Neue Datentypen	185
4.17.7 Einsatz bedingter Datentypen	185

5 Navigation und Verknüpfung 187

5.1 Datenauswahl mit XPath	187
5.1.1 Baummodell und XPath-Ausdrücke	188
5.1.2 Vom Dokument zum Knotenbaum	188
5.1.3 Dokumentreihenfolge	190
5.1.4 Knotentypen	191
5.1.5 Lokalisierungspfade	192
5.1.6 Ausführliche Schreibweise	194
5.1.7 Lokalisierungsstufen und Achsen	194
5.1.8 Knotentest	199
5.1.9 Filtern mit Prädikaten	200
5.1.10 Test von XPath-Ausdrücken	200
5.1.11 XPath 1.0-Funktionen	202
5.2 XPath 2.0	206
5.2.1 Erweitertes Datenmodell	207
5.2.2 Neue Konstrukte für Ausdrücke	207
5.2.3 Neue Datentypen	208
5.2.4 Neue Operatoren	209
5.2.5 Die erweiterte Funktionenbibliothek	209
5.3 XPath 3.0 und XPath 3.1	221
5.4 Verknüpfungen mit XLink	228
5.4.1 Mehr als Anker in HTML	228
5.4.2 Beziehungen zwischen Ressourcen	229
5.4.3 Link-Typen und andere Attribute	230
5.4.4 Beispiel für einen einfachen Link	232
5.4.5 Beispiel für einen Link vom Typ »extended«	233
5.4.6 XLink-Anwendungen	234
5.5 XBase	235

5.6	Über XPath hinaus: XPointer	236
5.6.1	URIs und Fragmentbezeichner	236
5.6.2	XPointer-Syntax	237
5.6.3	Das Schema »element()«	237
5.6.4	Das Schema »xmlns()«	238

6 Datenausgabe mit CSS 241

6.1	Cascading Stylesheets für XML	243
6.2	Arbeitsweise eines Stylesheets	243
6.3	Anlegen von Stylesheets	245
6.4	Vererben und Überschreiben	247
6.5	Selektortypen	248
6.6	Attributselektoren	249
6.7	Kontext- und Pseudoselektoren	249
6.8	Schriftauswahl und Textformatierung	250
6.8.1	Absolute Maßeinheiten	250
6.8.2	Relative Maßeinheiten	251
6.8.3	Prozentangaben	251
6.8.4	Maßangaben über Schlüsselwörter	251
6.9	Farbauswahl	252
6.10	Blöcke, Ränder, Rahmen, Füllung und Inhalt	252
6.11	Stylesheet-Kaskaden	254
6.12	Auflösung von Regelkonflikten	254
6.13	Zuordnung zu XML-Dokumenten	255
6.14	Schwächen von CSS	256

7 Umwandlungen mit XSLT 257

7.1	Sprache für Transformationen	257
7.1.1	Bedarf an Transformationen	257
7.1.2	Grundlegende Merkmale von XSLT	259
7.1.3	XSLT-Prozessoren	260

7.1.4	Die Elemente und Attribute von XSLT	261
7.1.5	Verknüpfung zwischen Stylesheet und Dokument	263
7.1.6	Das Element <stylesheet>	264
7.1.7	Top-Level-Elemente	264
7.1.8	Template-Regeln	266
7.1.9	Attributwert-Templates	267
7.1.10	Zugriff auf die Quelldaten	268
7.2	Ablauf der Transformation	269
7.2.1	Startpunkt Wurzelknoten	270
7.2.2	Anwendung von Templates	270
7.2.3	Rückgriff auf versteckte Templates	271
7.2.4	Auflösung von Template-Konflikten	271
7.3	Stylesheet mit nur einer Template-Regel	272
7.4	Eingebaute Template-Regeln	272
7.5	Designalternativen	274
7.6	Kontrolle der Knotenverarbeitung	276
7.6.1	Benannte Templates	277
7.6.2	Template-Auswahl mit XPath-Mustern	278
7.6.3	Kontext-Templates	280
7.6.4	Template-Modi	280
7.7	Datenübernahme aus der Quelldatei	282
7.8	Nummerierungen	283
7.8.1	Einfach	284
7.8.2	Mehrstufig	284
7.8.3	Zusammengesetzt	285
7.9	Verzweigungen und Wiederholungen	287
7.9.1	Bedingte Ausführung von Templates	287
7.9.2	Wahlmöglichkeiten	287
7.9.3	Schleifen	289
7.10	Sortieren und Gruppieren von Quelldaten	292
7.10.1	Sortierschlüssel	292
7.10.2	Sortierreihenfolge	293
7.11	Parameter und Variablen	294
7.11.1	Parameterübergabe	294
7.11.2	Globale Parameter	295
7.11.3	Lokale und globale Variablen	296
7.11.4	Eindeutige Namen	297

7.11.5	Typische Anwendungen von Variablen in XSLT	297
7.11.6	Rekursive Templates	302
7.12	Hinzufügen von Elementen und Attributen	304
7.12.1	Elemente aus vorhandenen Informationen erzeugen	304
7.12.2	Attributlisten	306
7.12.3	Texte und Leerräume	306
7.12.4	Kontrolle der Ausgabe	306
7.13	Zusätzliche XSLT-Funktionen	307
7.13.1	Zugriff auf mehrere Quelldokumente	307
7.13.2	Zahlenformatierung	309
7.13.3	Liste der zusätzlichen Funktionen in XSLT	310
7.14	Mehrfache Verwendung von Stylesheets	311
7.14.1	Stylesheets einfügen	312
7.14.2	Stylesheets importieren	312
7.15	Übersetzungen zwischen XML-Vokabularen	313
7.15.1	Diverse Schemas für gleiche Informationen	314
7.15.2	Angleichung durch Transformation	315
7.16	Umwandlung von XML in HTML und XHTML	316
7.16.1	Datenübernahme mit Ergänzungen	317
7.16.2	Tabellenkopf mit CSS	319
7.16.3	Aufbau einer Tabelle	319
7.16.4	Transformation in XHTML	320
7.16.5	XHTML-Module	321
7.16.6	Allgemeine Merkmale von XHTML	322
7.16.7	Aufbau eines XHTML-Dokuments	322
7.16.8	Automatische Übersetzung	323
7.17	XSLT-Editoren	324
7.18	Kurzreferenz zu XSLT 1.0	326
7.19	XSLT 2.0	334
7.19.1	Die wichtigsten Neuerungen	334
7.19.2	Neue Funktionen in XSLT 2.0	340
7.19.3	Neue Elemente in XSLT 2.0	341
7.20	XSLT 3.0	345
7.20.1	Streams	345
7.20.2	Packages	346
7.20.3	Umgang mit Maps	347
7.20.4	XML und JSON	348
7.20.5	Iterationen	349

7.20.6	Akkumulatoren	351
7.20.7	Unterstützung für XSLT 3.0?	352
7.20.8	Liste der zusätzlichen Funktionen in XSLT 3.0	353
7.20.9	Neue Elemente in XSLT 3.0	354

8 Formatierung mit XSL 363

8.1	Transformation und Formatierung	363
8.2	Formatierungsobjekte	364
8.3	Baum aus Bereichen – Areas	365
8.4	XSL-Bereichsmodell	365
8.4.1	Block-Bereiche und Inline-Bereiche	366
8.4.2	XSL und CSS	366
8.5	Testumgebung für XSL	367
8.6	Aufbau eines XSL-Stylesheets	369
8.6.1	Baum der Formatierungsobjekte	370
8.6.2	Seitenaufbau	370
8.6.3	Seitenfolgen	371
8.6.4	Einfügen von Fließtext	372
8.6.5	Blockobjekte	372
8.7	Verknüpfung mit dem Dokument und Ausgabe	374
8.8	Inline-Formatierungsobjekte	375
8.9	Ausgabe von Tabellen	376
8.9.1	Tabellenstruktur	376
8.9.2	Zellinhalte	377
8.10	Listen	379
8.11	Gesucht: visuelle Editoren	381
8.12	Übersicht über die Formatierungsobjekte von XSL	381
8.12.1	Übergeordnete Objekte	381
8.12.2	Blockformatierung	383
8.12.3	Inline-Formatierung	383
8.12.4	Tabellenformatierung	384
8.12.5	Listenformatierung	385
8.12.6	Formatierung für Verknüpfungen	386
8.12.7	Out-of-Line-Formatierung	386
8.12.8	Andere Objekte	387

9 Abfragen mit XQuery 389

9.1	Datenmodell und Verfahren	390
9.1.1	Zur Syntax	391
9.1.2	Instanzen des Datenmodells	392
9.1.3	W3C-Empfehlungen zu XQuery	394
9.2	Abfragepraxis	395
9.2.1	XQuery-Modul	395
9.2.2	Zugriff über das Web	397
9.3	FLWOR-Ausdrücke	397
9.3.1	Variablen in XQuery	401
9.3.2	Steuerung der Ausgabe	401
9.4	Fortgeschrittene Optionen	403
9.4.1	Auswertung zweier verbundener Dokumente	403
9.4.2	Kollektionen auswerten	404
9.4.3	Benutzerdefinierte Funktionen	405
9.5	Implementierungen	406
9.6	Neuerungen in XQuery 3.1	407

10 Programmierschnittstellen für XML 413

10.1	Abstrakte Schnittstellen: DOM und SAX	413
10.2	Document Object Model (DOM)	415
10.2.1	DOM Level	416
10.2.2	Objekte, Schnittstellen, Knoten und Knotentypen	417
10.2.3	Die allgemeine Node-Schnittstelle	417
10.2.4	Knotentypen und ihre Besonderheiten	419
10.2.5	Zusätzliche Schnittstellen	420
10.2.6	Zugriff über Namen	421
10.2.7	Verwandtschaften	421
10.2.8	Das Dokument als DOM-Baum	422
10.2.9	Document – die Mutter aller Knoten	424
10.2.10	Elementknoten	425
10.2.11	Textknoten	425
10.2.12	Attributknoten sind anders	426
10.2.13	Dokumentfragmente	426
10.2.14	Fehlerbehandlung	427

10.3	DOM-Implementierungen	427
10.4	Die MSXML-Implementierung von DOM	428
10.4.1	Schnittstellen in MSXML	429
10.4.2	Erweiterungen für Laden und Speichern	432
10.4.3	Erweiterungen der Node-Schnittstelle	432
10.5	Fingerübungen mit DOM	434
10.5.1	Daten eines XML-Dokuments abfragen	435
10.5.2	Zugriff über Elementnamen	441
10.5.3	Zugriff auf Attribute	442
10.5.4	Abfrage über einen Attributwert	443
10.5.5	Fehlerbehandlung	445
10.5.6	Neue Knoten einfügen	446
10.5.7	Neue Elementknoten	449
10.5.8	Neue Attributknoten	450
10.5.9	Unterelementknoten und Textknoten	450
10.5.10	Request und Response	451
10.6	Alternative zu DOM: Simple API for XML (SAX)	452
10.6.1	Vergesslicher Beobachter am Datenstrom	453
10.6.2	SAX2 unter Java	453
10.6.3	Der Kern der SAX-Schnittstellen	455
10.6.4	»ContentHandler«	456
10.6.5	Attribute	458
10.6.6	SAX2-Erweiterungen	459
10.6.7	Hilfsklassen	460
10.6.8	SAXParser und XMLReader	462
10.6.9	Konfigurieren des Parsers	463
10.6.10	Kleine Lagerauswertung mit SAX	465
10.6.11	Aufruf des Parsers	468
10.6.12	Fehlerbehandlung	469
10.6.13	SAX-Beispiel 1	471
10.6.14	SAX-Beispiel 2	474
10.6.15	SAX und DOM	476
10.7	Arbeit mit XML-Klassen in Visual Basic	477
10.7.1	XML-Architektur im .NET Framework	477
10.7.2	Lesen von XML-Daten	479
10.7.3	XML-Reader im Vergleich zum SAX-Reader	480
10.7.4	Arbeitsweise von »XmlReader«	480
10.7.5	XML-Dokument mit »XMLTextReader« auswerten	481
10.7.6	Lesen von XML-Fragmenten	484
10.7.7	Validierung anhand von XML-Schemas oder DTDs	486

10.7.8	Schreiben von XML-Daten	488
10.7.9	»XmlTextWriter«	492
10.7.10	XML-Serialisierung und -Deserialisierung	496
10.8	Zugriff auf XML-Daten mit »LINQ to XML«	502
10.8.1	LINQ to XML	502
10.8.2	X-Klassen	502
10.8.3	Functional Construction	503
10.8.4	XML-Literale	504
10.8.5	Schreiben und Laden von XML-Dateien	507

11 Kommunikation zwischen Anwendungen 509

11.1	XML-Webdienste	510
11.1.1	Gemeinsame Nutzung von Komponenten	510
11.1.2	Offengelegte Schnittstellen	510
11.1.3	Endpunkte	511
11.2	Beispiel für einen Webdienst	511
11.2.1	Webdienst mit ASP.NET	511
11.2.2	Einrichten eines Webdienstes	512
11.2.3	Webmethoden	514
11.2.4	Test des Webdienstes	515
11.2.5	Aufruf einer Methode	516
11.2.6	Nutzen des Webdienstes über eine Anwendung	517
11.2.7	Einfügen des Verweises auf den Webdienst	517
11.3	Nachrichten mit SOAP	519
11.3.1	Ein Rahmen für Nachrichten	519
11.3.2	Grundform einer SOAP-Nachricht	520
11.4	Dienstbeschreibung	523
11.4.1	Das WSDL-Vokabular	523
11.4.2	WSDL unter ASP.NET	524
11.5	Webdienste registrieren und finden	527
11.5.1	UDDI	527
11.5.2	Disco	527
11.5.3	Safety first!	528

12 XML in Office-Anwendungen	529
<hr/>	
12.1 XML in Microsoft Office	530
12.1.1 Der Standard Office Open XML	530
12.1.2 Open XML für Excel	532
12.1.3 Open XML für Word	535
12.2 Die Alternative OpenDocument	536
12.3 Einsatz benutzerdefinierter Schemas in Office 2016	540
12.3.1 Zuordnen eines Schemas	541
12.3.2 Optionen beim Öffnen von XML-Dokumenten	542
12.3.3 Daten als XML-Tabelle übernehmen	542
12.3.4 XML-Tabellenbereiche	545
12.3.5 XML-Zuordnungen	546
12.3.6 Datenaktualisierung	547
12.3.7 Öffnen als schreibgeschützte Arbeitsmappe	548
12.3.8 Verwenden von XSLT-Stylesheets	548
12.3.9 Datenquelle und Tabelle manuell verknüpfen	551
12.3.10 »XmlMap«-Objekte	554
12.3.11 Tabelle auf der Basis eines eigenen Schemas	554
12.3.12 Fehlererkennung	554
12.3.13 XML-Dokumente erzeugen	555
12.3.14 Schema-Einschränkungen	556
13 Mapping – von XML oder nach XML	557
<hr/>	
13.1 Codegenerierung für Transformationen	557
13.1.1 Oberfläche und Dateiformate	558
13.1.2 Funktionsbibliotheken	560
13.1.3 Von Schema zu Schema	561
13.2 Datenausgabe	563
13.3 Stylesheet-Generierung	564
13.4 Eigene Funktionen	565
13.5 Mapping von Datenbankdaten	566
13.6 Mapping für Excel-Tabellen	569
13.7 EDIFACT und ANSI X12	570

13.8 Mapping zwischen JSON und XML	571
13.9 Webdienste	575
14 Publizieren mit EPUB	577
<hr/>	
14.1 Electronic Publication	577
14.1.1 Content Documents	578
14.1.2 Paket-Format	581
14.1.3 Open Container Format	585
14.2 Tools für EPUB	587
14.3 Autorentools	591
15 HTML5 und XHTML	595
<hr/>	
15.1 Unerfüllte Erwartungen	595
15.2 Die Wiederbelebung von HTML	598
15.3 Fixer oder lebendiger Standard?	599
15.4 Was ist neu?	599
15.5 DOCTYPE und Ausführungsmodus	600
15.6 HTML vs. XHTML	601
15.7 Aussichten	602
15.8 XML-Inhalte im Browser	603
15.8.1 Die Wiederbelebung von SVG	603
15.8.2 Dynamische Illustrationen	604
15.9 Freiwillige gesucht	610