

Dieses Buch wurde von uns mit L^AT_EX₂e gesetzt. Die Bilder wurden mit xfig erstellt. Fast alle Beispiele wurden auch an realen Systemen getestet. Bei XML-Quelltexten war XML Spy eine gute Hilfe. Robert Tolksdorf und Denny Priebe halfen uns, die Layoutvorgaben des Verlages mit L^AT_EX umzusetzen, auch dafür herzlichen Dank.

Ganz besonders möchten wir uns bei unseren Familien bedanken, insbesondere Meike bei Ronny, Laura und Helena sowie Holger bei Anja, Antonia, Johannes, Katharina und Konstantin.

Aktuelle Informationen zu diesem Buch

*» Wir operieren natürlich immer auf der
auswechselbaren Grundlage der Tatsachen.«*

Dieser Verzweiflungsruf von Holger drückt die Probleme aus, die sich beim Schreiben über ein Thema wie XML ergeben. XML und die zahlreichen Entwicklungen im Umfeld sind in der Tat noch sehr dynamisch. Ursache dafür ist der fortschreitende Prozess beim W3C, der dazu führen wird, dass bestimmte Anteile, über die gegenwärtig noch diskutiert wird, erst nach dem Erscheinen des Buchs als offizielle Empfehlung verabschiedet werden. In der Zwischenzeit können sich natürlich noch Änderungen ergeben. Auch kommerzielle Systeme und frei verfügbare Tools unterliegen ständigen Änderungen, wahrscheinlich werden auch zahlreiche neue Werkzeuge und frei verfügbare Implementierungen hinzukommen.

Das führte dazu, dass wir während des Schreibens einzelne Abschnitte wiederholt überarbeiten mussten, weil sich die »Tatsachen« mal wieder geändert hatten. Veränderungen, neue Empfehlungen des W3C und neue Versionen der verfügbaren Tools werden aber auch nach Abgabe des Manuskripts an der Tagesordnung sein. Aktuelle Informationen, die dieses Buch ergänzen, werden wir deshalb im WWW bereitstellen.

Ein weiteres Problem ist, dass viele Informationen zum Thema XML nicht in Form von Büchern, Konferenzbeiträgen etc. vorliegen, die auch nach Jahren noch verfügbar sind. Man findet gerade in diesem Bereich vieles nur im Internet. Die Autoren der jeweiligen Webseiten sind hier berechtigt, Informationen zu verschieben, zu verändern oder auch ganz zu entfernen. Trotzdem wollen wir oft auf diese Hinweise nicht verzichten, da es sich um sehr aktuelle und frei verfügbare Informationen handelt. Deshalb finden sich in den Literaturhinweisen dieses Buchs viele URLs. Unmittelbar vor dem Druck des Buches wurden alle URLs noch einmal überprüft.

Den Veränderungen im Web kann man mit einem (statischen) Buch jedoch nicht folgen, deshalb stellen wir zusätzlich eine Webseite bereit,

auf der wir uns bekannte Änderungen fortschreiben. Sie hat die einprägsame Adresse:

www.xml-und-datenbanken.de

Darüber hinaus würden wir uns über Hinweise und Kommentare sehr freuen. Wir sind unter den folgenden Mail-Adressen zu erreichen:

meike@informatik.uni-rostock.de bzw.
hme@informatik.uni-rostock.de

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufbau des Buchs	3
1.2	Beispiel, Konventionen	5
2	Anwendungsszenarien	7
2.1	Generierung von XML-Dokumenten aus Datenbanken	7
2.2	Generierung von XML-Dokumenten für ein festgelegtes Schema	8
2.3	Visualisieren von Datenbankinhalten	10
2.4	Speicherung von datenzentrierten XML-Dokumenten	11
2.5	Speicherung von dokumentzentrierten XML-Dokumenten ..	12
2.6	Speicherung von XML-Dokumenten in bestehende Datenbanken	13
2.7	Austausch von Datenbankinhalten	14
2.8	Electronic Data Interchange (EDI)	15
2.9	Verbindung verschiedener Softwarekomponenten	17
2.10	Weitere Anwendungen	17
3	XML-Grundlagen	19
3.1	Was ist XML?	19
3.1.1	Das W3C	20
3.1.2	Entstehung von XML	21
3.1.3	Entwurfsziele von XML	21
3.1.4	Weitere Empfehlungen	22
3.2	XML-Syntax	23
3.2.1	Elemente	24
3.2.2	Attribute	28
3.2.3	Entwurfsentscheidung: Verwendung von Elementen oder Attributen	30
3.2.4	Entities	32
3.2.5	Processing Instructions	36
3.2.6	Kommentare	37
3.2.7	Aufbau eines Dokuments	37
3.3	Eigenschaften von XML	38
3.4	Zusammenfassung	39

3.5	Literatur	39
3.6	Verfügbare Software	40
4	XML-Prozessoren	43
4.1	SAX — Simple API for XML	44
4.2	DOM (Document Object Model)	46
4.3	Einsatz von SAX und DOM	52
4.4	Literatur	55
4.5	Verfügbare Implementierungen	55
5	Datenrepräsentation in XML	57
5.1	Repräsentation von relationalen Datenbanken in XML-Dokumenten	57
5.1.1	Ziele der Datenrepräsentation in XML	58
5.1.2	Generelle Probleme	59
5.1.3	Vollständige Abbildung der Datenbankinhalte	60
5.1.4	Ableitung von Element-Hierarchien aus Schlüssel- und Fremdschlüsselbeziehungen	75
5.1.5	Ableitung von XML-Dokumenten aus Ausschnitten einer Datenbank	76
5.1.6	Einsatz von individuellen Transformationsregeln	78
5.1.7	Mapping-Verfahren	84
5.1.8	Einsatz der verschiedenen Verfahren	86
5.2	Repräsentation von objektrelationalen Datenbanken	87
5.3	Repräsentation von anderen Daten in XML-Dokumenten	90
5.4	Literatur	92
5.5	Umsetzung in frei verfügbaren Tools	93
5.6	Kommerzielle Systeme	94
6	XML-Datenbanken — Architekturmodelle	99
6.1	Datenzentrierte, dokumentzentrierte und semistrukturierte XML-Dokumente	100
6.2	Die konzeptuelle, logische und physische Ebene von XML-Datenbanken	102
6.3	Logische Ebene	105
6.3.1	Anforderungen	105
6.3.2	Konkrete Umsetzungen	105
6.4	Physische Ebene	106
6.4.1	Anforderungen	107
6.4.2	Konkrete Realisierungen	107
6.5	Modellierung von XML-Dokumenten	108
6.5.1	Anforderungen	109
6.5.2	Konkrete Umsetzung	109
6.6	Zusammenfassung und weiterführende Betrachtungen	110

7	Schemabeschreibung und konzeptueller Entwurf von Schemata	113
7.1	Vorteile einer Schemabeschreibung	113
7.2	Methoden zur Schemabeschreibung	114
7.2.1	DTDs	114
7.2.2	XML Schema	115
7.3	Ableitung von Schema-Informationen aus Dokumentkollektionen	126
7.4	Vorteile einer konzeptuellen Modellierung	129
7.5	Methoden zur konzeptuellen Modellierung	130
7.5.1	Einsatz des Entity-Relationship-Modells	130
7.5.2	Modellierung mit ORM	133
7.5.3	Verwendung von UML	134
7.5.4	XML-Editoren	136
7.5.5	Einsatz der Entwurfsmethoden	137
7.6	Metriken für XML-Dokumente	138
7.6.1	Produktmetriken für XML	139
7.6.2	Ressourcenmetriken	144
7.6.3	Verwendung der Metriken	145
7.6.4	Metriken zur Bestimmung des Dokumentcharakters	146
7.7	Literatur	148
7.8	Systeme	152
8	XML-Datenbanken — Speicherungstechniken	155
8.1	Klassifikation der Speicherungsmethoden	155
8.2	Speicherung von XML-Dokumenten als Ganzes	158
8.2.1	Verwendung eines Volltextindexes	159
8.2.2	Verwendung eines kombinierten Volltext- und Strukturindexes	164
8.3	Speicherung der Graphenstruktur von XML-Dokumenten ...	167
8.3.1	Abbildung der Graphenstruktur von XML-Dokumenten ...	167
8.3.2	Abbildung von DOM-Strukturen auf objektorientierte oder relationale Datenbanken	171
8.4	Abbildung der Dokument- auf Datenbankstrukturen	175
8.4.1	Automatische Speicherung der Dokumente	176
8.4.2	Einsatz von Mapping-Vorschriften	181
8.5	Hybride Speicherung	185
8.6	Redundante Speicherung	188
8.7	Zusammenfassung	189
8.8	Literatur	190
8.9	Verfügbare Implementierungen	192
8.10	Kommerzielle Systeme	192

9	Speicherorganisation und Indexstrukturen	197
9.1	Werteindex	198
9.1.1	Erweiterbares Hashing	199
9.1.2	B ⁺ -Bäume	202
9.1.3	Signaturbäume	205
9.2	Volltextindex	209
9.2.1	Invertierte Datei	212
9.2.2	Tries, Patricia- und Suffixbäume	216
9.2.3	Suffix Arrays	218
9.2.4	Unterstützung von Mustersuche	220
9.2.5	Vergleich Volltextindizierung	222
9.3	Dokumentspeicherung	223
9.3.1	Überblick	223
9.3.2	Persistentes DOM in Infonyte-DB	223
9.3.3	Hybride Speicherung in Natix	226
9.4	XML-Struktur- oder Pfadindizes	228
9.4.1	XML-sensitiver Volltextindex	231
9.4.2	Nummerierungsschemata	232
9.4.3	XASR	237
9.4.4	Index Fabric	238
9.4.5	Pfadindizierung in SphinX	242
9.5	Zusammenfassung	245
9.6	Literatur	246
10	XML-Anfragesprachen	249
10.1	Kriterien für Anfragesprachen	249
10.1.1	Allgemeine Anforderungen	250
10.1.2	XML-Anforderungen	251
10.1.3	Grundoperationen	253
10.2	XPath 1.0	256
10.2.1	Dokument- und Datenmodell	256
10.2.2	Sprachüberblick	258
10.2.3	Pfadausdrücke	258
10.2.4	Selektionsprädikate	262
10.2.5	Funktionsbibliothek	265
10.3	XQuery 1.0	266
10.3.1	Sprachüberblick	266
10.3.2	Das Datenmodell	267
10.3.3	Einfache Ausdrücke	269
10.3.4	FLWR-Ausdrücke	271
10.3.5	Elementkonstruktoren	277
10.3.6	Verbünde	279
10.3.7	Gruppierung	281
10.3.8	Sortierung mit SORT BY	282
10.3.9	Verwendung von Quantoren	283

10.3.10	Bedingte Ausdrücke	284
10.3.11	Typsystem	285
10.3.12	Vergleichsoperationen	289
10.3.13	Statischer und dynamischer Kontext eines Ausdrucks	292
10.3.14	Anfrageprolog	292
10.3.15	Nutzerdefinierte Funktionen	293
10.3.16	XQuery-Standardisierung	294
10.3.17	XQuery-Implementierungen	295
10.4	Übersicht zu weiteren Anfragesprachen	295
10.4.1	XML-QL	295
10.4.2	XQL	300
10.4.3	XML und Updates	304
10.4.4	XQuery-Updates mit SUNFLWR	305
10.5	Weiterführende Literatur	306
11	XML-Unterstützung in Datenbanksystemen	311
11.1	DB2 mit XML- und TextExtender	312
11.1.1	XML Extender — Überblick	312
11.1.2	Speicherungs- und Zugriffstechniken	314
11.1.3	Beschreibung der XML-Speicherung mit DAD	317
11.1.4	TextExtender und Volltextsuche	321
11.1.5	Weitere Komponenten	321
11.1.6	Einsatzmöglichkeiten	322
11.2	Oracle 9i	323
11.2.1	Export von Datenbankinhalten mit XML-Syntax	324
11.2.2	XML-Speicherung mit XMLType	326
11.2.3	Speicherung von XML-Dokumenten mit intermedia Text .	328
11.2.4	Strukturierte Speicherung von XML	329
11.2.5	Kombination der Speicherungsvarianten	329
11.2.6	Einsatz	329
11.3	Microsoft SQL Server 2000 und XML-Unterstützung	330
11.3.1	Darstellung von SQL-Anfrageergebnissen als XML	332
11.3.2	Relationale Sichten auf XML mit OpenXML	337
11.3.3	XML-Sichten auf relationale Daten	339
11.3.4	Zusammenfassung der Möglichkeiten	341
11.4	eXtensible Information Server von eXcelon	342
11.5	Tamino	345
11.6	Infonyte-DB	347
11.7	POET Object Server Suite	349
11.8	Content Management Suite — CMS	352
11.9	Leistungsbewertung von Systemen und Benchmarks	355
11.9.1	Xmark	356
11.9.2	XOO7	356
11.9.3	XMach-1	357
11.10	Zusammenfassung und Ausblick	358

A	Beispiel »Hotelinformationen«	361
A.1	Konzeptueller Entwurf im ER-Modell	361
A.2	Abbildung auf das relationale Datenbankmodell	363
A.3	Beispiel-Datenbank	365
A.4	DTD, XML Schema und XML-Dokumente	366
B	Beispielanwendung »Plattenkritiken«	377
B.1	Konzeptueller Entwurf im ER-Modell	377
B.2	DTD, XML Schema und XML-Dokumente	378
C	XPath 1.0-Funktionsbibliothek	383
C.1	Funktionen auf Knotenmengen	383
C.2	Funktionen auf Zeichenketten	384
C.3	Funktionen auf numerischen Werten	385
C.4	Funktionen auf booleschen Werten	386
	Literaturverzeichnis	387
	Tabellenverzeichnis	405
	Abbildungsverzeichnis	407
	Personenregister	411
	Stichwortverzeichnis	415

1 Einleitung

XML und Datenbanken. Der Titel des Buchs lautet »XML und Datenbanken«.

XML steht für Extensible Markup Language und bezeichnet ein Dokumentenformat zur Darstellung semistrukturierter Daten. XML genießt zweifellos gegenwärtig eine große Popularität. Die Anwendungen sind vielfältig und reichen vom Zwischenformat zur Datenrepräsentation oder zum Datenaustausch bis hin zur Markup-Sprache für Volltextdokumente. Datenbanken hingegen sind eine seit Jahrzehnten bewährte Technologie zur Speicherung und Anfrage von großen strukturierten Datenbeständen.

Auf den ersten Blick lassen sich die Unterschiede beider Gebiete erkennen. XML und Datenbanken sind zwei sehr verschiedene Konzepte, die in unterschiedlichen Bereichen der Informatik entwickelt und angewendet werden.

Datenbanken werden eingesetzt, um große Datenmengen sicher zu speichern und effizient anfragen zu können. Sie dienen in erster Linie der Speicherung von strukturierten Daten. XML ermöglicht neben der Darstellung strukturierter Daten auch die Repräsentation von semistrukturierten Daten, also Daten, deren Struktur unregelmäßig ist, wechseln kann oder gar nicht explizit vorhanden ist. Auch Dokumente, die Volltextinformationen beinhalten, können mit XML-Syntax dargestellt werden.

Wie sieht nun der Zusammenhang zwischen beiden Gebieten aus? Die Forderung nach konsistenter und zuverlässiger Speicherung von XML-Dokumenten suggeriert den Einsatz von Datenbanken oder die Verwendung und Anpassung von Datenbanktechnologien. Wir werden im Verlauf der weiteren Kapitel betrachten, welche Technologien aus dem Gebiet Datenbanken für XML adaptiert wurden und inwiefern sie angepasst und erweitert wurden, um für XML geeignet zu sein. XML ist ein relativ neues Format zur Darstellung und Speicherung von Daten, während Datenbanken eine schon seit sehr langer Zeit bestehende und vielfältig eingesetzte Technologie sind. Es bietet sich bei der Entwicklung von Technologien für XML an, auch die verfügbaren Technologien im Bereich Datenbanken anzusehen und teilweise in angepasster Form zum Einsatz zu bringen. Dafür können bei der Speicherung und Indizierung von XML-Dokumenten Datenbanktechniken eingesetzt oder

zugrunde gelegt werden. Im Bereich Anforderungen an XML-Anfragesprachen und XML-Anfragesprachen selbst sind ebenfalls viele Parallelen zu Datenbank-Anfragesprachen sichtbar. Auch bei der Entwicklung von Updatesprachen sind die Updatefunktionen der Datenbank-Anfragesprachen eine nützliche Basis. Wir werden im weiteren Verlauf daher Fragen untersuchen, wie

- ❑ Welche Speicherungstechniken existieren für XML-Dokumente?
- ❑ Wie erfolgt eine geeignete Indizierung der gespeicherten XML-Dokumente?
- ❑ Welche Anfragesprachen und Updatemöglichkeiten für XML-Dokumente gibt es?
- ❑ Wie kann die Abbildung von XML-Dokumenten auf Datenbankinformationen erfolgen?

Datenbanken und XML. Drehen wir den Buchtitel doch einmal um, dann erhalten wir: »Datenbanken und XML« — und damit die zweite Sichtweise auf diese Problematik. Hierbei werden wir beleuchten, wie XML die klassische Datenbankwelt ergänzt.

Die Anforderungen an Datenbanksysteme steigen und es wird bedingt durch verschiedene Anwendungen notwendig, neue Arten von Informationen zu speichern. Über strukturierte Daten hinaus, die zum Beispiel für technische und ökonomische Anwendungen gespeichert werden, besteht der Wunsch, weitere Informationsarten, z.B. Multimediatypen (Bild, Audio, Video und Volltextdaten), innerhalb von Datenbanksystemen zu speichern. Mit der ExtenderTechnologie wird in verschiedenen Datenbanksystemen versucht, die Speicherung neuer Datentypen zu realisieren und Anfragen an die neuen Datentypen umzusetzen. Die Anfragen an Multimediatypen sind in den SQL99-Standard eingegangen. XML-Dokumente stellen ebenfalls einen neuen Datentyp dar, der innerhalb von Datenbanken gespeichert, verwaltet und angefragt werden soll.

XML ermöglicht die Speicherung von semistrukturierten Informationen. Durch das eingesetzte Markup enthalten XML-Dokumente Informationen über die Struktur, die in Anfragen einbezogen werden sollen. Man erreicht durch die Speicherung von XML-Dokumenten innerhalb von Datenbanksystemen, dass Datenbanken um neue, irreguläre Anteile erweitert werden. Daraus ergeben sich jetzt Fragen wie

- ❑ Welche neuen Anforderungen stellt XML an Datenbanksysteme?
- ❑ Welche Erweiterungen haben Datenbanksysteme, um XML-Dokumente zu speichern und zu verarbeiten?

Ein weiterer Aspekt ist, dass die Daten aus Datenbanken für andere Anwendungen verfügbar gemacht werden sollen. XML bietet sich dazu

als Zwischen- und Austauschformat an. Die Transformation von Daten aus Datenbanken in XML-Syntax ist also eine weitere relevante Aufgabe, die sich durch die folgende Frage ausdrücken lässt:

- Wie können XML-Dokumente aus den Inhalten von Datenbanken generiert werden?

Im Verlauf des Buchs werden wir uns diesen und weiteren Fragen widmen.

1.1 Aufbau des Buchs

Im Folgenden sollen die einzelnen Kapitel des Buchs kurz vorgestellt werden.

Kapitel 2: Anwendungsszenarien Kapitel 2 stellt einige typische Anwendungen vor, zeigt dabei bestehende Teilaufgaben und gibt an, in welchen nachfolgenden Kapiteln des Buchs eine genaue Darstellung von Lösungen dafür erfolgt.

Kapitel 3: Grundlagen — XML In Kapitel 3 wird eine kurze XML-Einführung gegeben. Dabei werden Begriffe wie Element, Attribut, Entity, usw. erklärt und anhand vieler Beispiele erläutert. DTDs (Document Type Definitions) werden hier ebenfalls eingeführt.

Kapitel 4: XML-Prozessoren Der Zugriff auf XML-Dokumente wird durch so genannte XML-Prozessoren unterstützt. Generell sind zwei verschiedene Varianten von Parsern verfügbar. Sie werden in diesem Kapitel vorgestellt.

Kapitel 5: Datenrepräsentation von XML In Kapitel 5 wird dargestellt, wie sich aus Datenbankinhalten XML-Dokumente generieren lassen. Dabei können aus vollständigen Datenbanken XML-Dokumente mit feststehender Syntax generiert werden. Aus Datenbankausschnitten, die durch Anfragen oder Sichten spezifiziert werden, lassen sich XML-Dokumente mit fester Syntax erzeugen, und es können variable Methoden eingesetzt werden, die aus Ausschnitten von Datenbanken spezielle XML-Dokumente mit einer vom Anwender festgelegten Syntax generieren. Alle diese Verfahren werden ausführlich dargestellt. Verfügbare Implementierungen in kommerziellen Datenbanksystemen und frei verfügbare Tools werden am Ende des Kapitels beschrieben.

Kapitel 6: XML-Datenbanken — Architekturmodelle In diesem Kapitel wird zunächst dargestellt, wie XML-Dokumente klassifiziert werden können. Auf der Klassifikation aufbauend werden verschiedene Verfahren für die konzeptuelle, die logische und die physische Ebene der Speicherung von XML-Dokumenten beschrieben. Die dabei eingesetzten Verfahren sind teilweise an Verfahren aus den Bereichen Datenbanken und Dokumentverarbeitung angelehnt, teilweise mit diesen auch identisch. Zusätzlich haben sich neue Verfahren etabliert, die speziell auf XML-Anwendungen zugeschnitten sind. Das Kapitel dient als Systematisierung der Ansätze, XML-Dokumente zu speichern. Die hier eingeführten Verfahren werden in den folgenden drei Kapiteln (7–11) ausführlicher dargestellt.

Kapitel 7: Schemabeschreibung und konzeptueller Entwurf von Schemata Dieses Kapitel ist zweigeteilt: Zum einen werden hier zwei Methoden vorgestellt, mit denen das Schema von XML-Dokumenten dargestellt werden kann — Document Type Definition und XML Schema. Im zweiten Teil des Kapitels wird gezeigt, wie eine konzeptuelle Modellierung der Schemabeschreibungen erfolgen kann. Es besteht die Möglichkeit, das Entity-Relationship-Modell, das Object Role Model oder die Unified Modeling Language (UML) zu adaptieren. Weiterhin kann die Struktur von XML-Dokumenten und Schemabeschreibung graphisch visualisiert werden.

Kapitel 8: XML-Datenbanken — Speicherungstechniken Die logische und physische Speicherung von XML-Dokumenten ist Inhalt dieses Kapitels. Dabei werden drei grundsätzliche Verfahren dargestellt, die die Abbildung auf interne Strukturen beschreiben. Die XML-Dokumente können unverändert gespeichert und indiziert werden. Es ist möglich, die Graphenstruktur der XML-Dokumente als Basis für die Speicherung zu verwenden. Die dritte Variante ist die strukturierte Speicherung von XML-Dokumenten in Datenbanken. Es wird ebenfalls vorgestellt, welche Verfahren für welche Arten von Dokumenten geeignet sind. Existierende Implementierungen in kommerziellen Systemen und frei verfügbare Tools werden zum Abschluss des Kapitels beschrieben.

Kapitel 9: Speicherorganisation und Indexstrukturen Nachdem im vorherigen Kapitel mögliche Speicherungs- und Indizierungsvarianten für XML-Dokumente dargestellt wurden, erfolgt in Kapitel 9 die Darstellung, wie die Speicherungsverfahren genau aussehen. Dazu werden allgemeine Indexstrukturen vorgestellt und Möglichkeiten zur Speicherung eines XML-Indexes angegeben.

Kapitel 10: XML-Anfragesprachen Ausgehend von generellen Anforderungen an Anfragesprachen werden diese für spezielle XML-Anfragesprachen dargelegt. Viele Vorschläge und Realisierungen stützen sich auf XPath 1.0, das als eine Anfragemöglichkeit an XML-Dokumente betrachtet wird. Mit XQuery wird ausführlich der absehbare Standard im Bereich Anfragesprachen beschrieben. Eine Übersicht zu weiteren Sprachen sowie verfügbaren Implementierungen beschließt das Kapitel.

Kapitel 11: Systeme In vielen Kapiteln des Buchs werden bestimmte Verfahren und Techniken vorgestellt und am Ende findet sich eine Aufzählung, in welchen frei verfügbaren Tools und kommerziellen Systemen diese Techniken umgesetzt sind. In diesem Kapitel jedoch wird die Betrachtung von der anderen Seite vorgenommen. Es werden verschiedene kommerzielle Systeme zum Entwurf, zur Verwaltung und zur Speicherung von XML-Dokumenten vorgestellt.

1.2 Beispiel, Konventionen

Im weiteren Verlauf des Buchs werden wir vieles anhand von Beispielen erläutern. Dabei werden wir überwiegend mit Anwendungen aus dem Tourismusbereich arbeiten. Diese erscheinen uns für das Thema geeignet, da Tourismusinformationen einerseits typische XML-Anwendungen für die Darstellung im World Wide Web sind, die auch vielfach multimediale Elemente enthalten. Andererseits existieren im Tourismusbereich Hoteldatenbanken für Buchungssysteme, in denen ähnliche Informationen stehen. Der Einsatz von XML- und Datenbank-Anwendungen ist also typisch für diese Branche. Transformationen zwischen XML-Dokumenten und Datenbankinhalten lassen sich anhand der Beispielanwendung ebenfalls gut beschreiben.

Ein weiterer Grund für die Auswahl des Anwendungsbeispiels ist, dass wir in Projekte involviert waren, die sich mit diesem Thema beschäftigten. Das ist in unserem Bundesland naheliegend, in dem der Tourismus der größte Wirtschaftszweig ist.

Alle beschriebenen Beispiele (Hotels/Museen/Veranstaltungen) kennen wir persönlich, die Auswahl wurde jedoch aus »XML-Sicht« getroffen und stellt somit keine Wertung dar. Wir profitieren auch nicht von der Verwendung der Beispiele. :-)

2 Anwendungsszenarien

In diesem Kapitel werden einige typische Anwendungen dargestellt und es wird gleichzeitig angegeben, in welchen Kapiteln des Buchs sich dazu relevante Informationen finden. Das kann als Unterstützung für die Leser dienen, die sich schnell über die Realisierung konkreter Aufgaben informieren wollen. Auch können Entwickler und Anwender, die bestimmte Aufgaben im Blick haben, das Buch so gezielter lesen. Die einzelnen Teilaufgaben der Anwendungen werden jeweils in Abbildungen veranschaulicht. Dabei wird dargestellt, aus welchen Teilaufgaben sich die Anwendungsszenarien zusammensetzen und in welchem Kapitel des Buchs sich Lösungen für diese Teilaufgaben befinden.

Das Interesse an XML in der Industrie und der Forschung ist sehr groß, die Anzahl der Anwendungen, die XML einsetzen, steigt sprunghaft an, die Anwendungsgebiete sind vielfältig. Wir wollen im Folgenden einige typische Anwendungsgebiete von XML angeben. Die Beispiele sind recht allgemein gehalten, damit die Anzahl der Anwendungsszenarien übersichtlich bleibt.

2.1 Generierung von XML-Dokumenten aus Datenbanken

Viele Informationen sind in Datenbanken gespeichert. Sollen diese Informationen für andere Anwendungen bereitgestellt werden, so kann das über das Generieren von XML-Dokumenten aus den Inhalten von Datenbanken erfolgen.

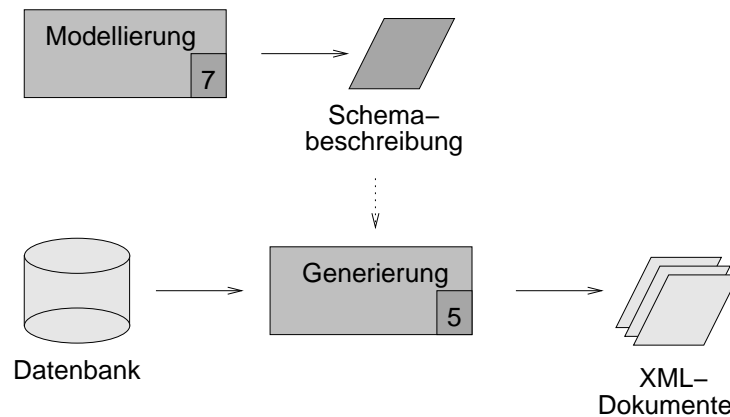
Kapitel 3,5,7,11

Abbildung 2.1 veranschaulicht die dabei erforderlichen Teilaufgaben.

Voraussetzung für die Aufgabe ist, dass man mit dem Dokumentenformat XML vertraut ist. *Kapitel 3* gibt dazu eine Einführung und beschreibt einige generelle Entwurfsprinzipien für XML-Dokumente.

Kapitel 5 beschreibt das eigentliche Problem der Abbildung von Datenbankinhalten in XML-Dokumente im Detail und erläutert mehrere Möglichkeiten dazu. Einige Methoden bilden die Daten der Datenbank oder Ausschnitte einer Datenbank als XML-Dokumente mit fester und nicht veränderbarer Struktur ab. Es gibt auch Methoden, die die gespeicherten Informationen auf XML-Dokumente mit einem

Abbildung 2.1
Generierung von XML-
Dokumenten aus
Datenbanken



gegebenen Schema abbilden. Alle dazu verfügbaren Methoden werden in Kapitel 5 beschrieben, einschließlich der dafür existierenden Tools (siehe Abschnitt 5.5).

Wählt man von den existierenden Möglichkeiten eine aus, bei der der Anwender die Syntax der resultierenden XML-Dokumente individuell festlegen kann, so muss man im Vorfeld definieren, wie die XML-Dokumente aussehen sollen. Man benötigt also vorher ein explizit dargestelltes Schema für die XML-Dokumente. Möglichkeiten zur Darstellung der Schemainformationen sind in Kapitel 7 dargestellt; Varianten, die zum konzeptuellen Entwurf des Schemas existieren, werden ebenfalls in Kapitel 7 vorgestellt.

Eine Übersicht über Datenbanksysteme, die auch beinhaltet, welche Varianten zur Ausgabe der Informationen im XML-Format bestehen, wird in Kapitel 11 gegeben.

2.2 Generierung von XML-Dokumenten für ein festgelegtes Schema

Kapitel 3,5,11 Dieses Szenario ist vielfach anzutreffen. Informationen sind bereits in Datenbanken gespeichert und sollen für andere Anwendungen im XML-Format verfügbar gemacht werden. Dabei ist vorgegeben, wie das Format der entstehenden Daten aussehen soll. Das prinzipielle Vorgehen wird in Abbildung 2.2 dargestellt.

Die Anwendung ähnelt der vorherigen, ist jedoch dadurch eingeschränkt, dass aus Datenbanken nicht XML-Dokumente mit beliebiger Syntax erzeugt werden sollen, sondern die Syntax der XML-Dokumente durch eine Schemabeschreibung gegeben ist (DTD oder XML Schema).

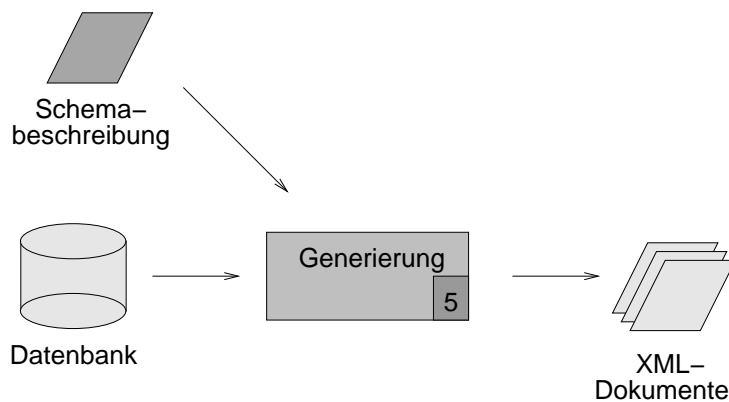


Abbildung 2.2
Generierung von XML-Dokumenten mit vorgegebenem Schema

Kapitel 5 beschreibt die Abbildung von Datenbankinhalten in XML-Dokumente und erläutert in Abschnitt 5.1.6 Möglichkeiten für eine anwenderdefinierte Abbildung.

Eine Übersicht über Datenbanksysteme, die Daten im XML-Format mit vorgegebenem Schema ausgeben können, wird in *Kapitel 11* gegeben.

Es ist alternativ dazu möglich, XML-Dokumente aus Datenbanken zu erzeugen und durch XSLT in das gewünschte Format zu transformieren. Die Vorgehensweise ist in *Abbildung 2.3* angegeben und gleicht der vorher dargestellten Methode.

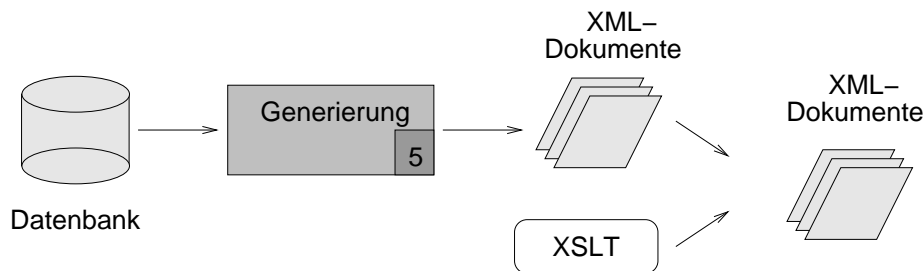


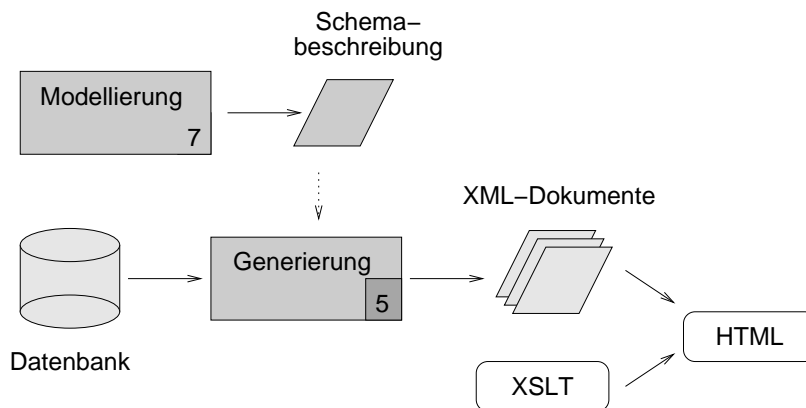
Abbildung 2.3
Zweistufige
Generierung von XML-Dokumenten

Das Verfahren unterscheidet sich dadurch, dass aus der Datenbank zunächst die XML-Dokumente mit einer feststehenden Syntax erzeugt werden, (siehe *Kapitel 5*, Abschnitt 5.1.3). Im zweiten Schritt erfolgt durch XSLT eine Anpassung der XML-Dokumente an die gewünschte Syntax. Zum Thema XSLT, das nicht in diesem Buch behandelt wird, ist in Abschnitt 3.5 eine Empfehlung für weiterführende Literatur gegeben.

2.3 Visualisieren von Datenbankinhalten

Kapitel 3,5,7 Datenbankinhalte müssen durch Applikationen aufbereitet werden, bevor sie einem Anwender präsentiert werden können. Da XML-Dokumente sich in Verbindung mit Style Sheets gut zur Präsentation eignen, bietet sich hier ein Anwendungsgebiet mit XML als Zwischenformat, um die Inhalte aus Datenbanken zu visualisieren und zum Beispiel auf Webseiten darzustellen. Für diese Anwendung muss man eine geeignete Repräsentation der Daten aus der Datenbank als XML-Dokumente finden. Abbildung 2.4 veranschaulicht die Anwendung.

Abbildung 2.4
Visualisieren von
Datenbankinhalten



In *Kapitel 3* kann man sich mit den Grundlagen von XML vertraut machen.

Im ersten Schritt zur Visualisierung von Datenbankinhalten kann eine konzeptuelle Modellierung des Schemas erfolgen. Möglichkeiten einer Schemabeschreibung und Verfahren zur konzeptuellen Modellierung von XML-Dokumenten werden in *Kapitel 7* vorgestellt.

Kapitel 5 beschreibt verschiedene Methoden für die Umwandlung der Daten aus den Datenbanken ins XML-Format. Diese Teilaufgabe ähnelt damit dem oben dargestellten *Generieren von XML-Dokumenten aus Datenbanken*.

Der zweite für diese Aufgabe benötigte Schritt ist nicht Inhalt dieses Buchs. Die als Zwischenformat erzeugten XML-Dokumente können mit Hilfe von XSLT transformiert und aus ihnen andere Formate wie zum Beispiel HTML generiert werden. Die HTML-Präsentation von Datenbankinhalten kann zur Visualisierung eingesetzt werden. XSLT selbst ist in diesem Buch nicht beschrieben. Hierzu empfehlen wir in *Abschnitt 3.5* weiterführende Literatur.

2.4 Speicherung von datenzentrierten XML-Dokumenten

Bei diesem Szenario gehen wir davon aus, dass XML-Dokumente vorliegen, die gespeichert und verwaltet werden sollen. Datenzentrierte XML-Dokumente bedeuten dabei, dass die Dokumente sehr regelmäßig strukturiert sind und in diesen überwiegend Daten (also keine umfangreichen Texte) enthalten sind. Abbildung 2.5 veranschaulicht die Teilschritte der Anwendung.

Kapitel 6,8,10,11

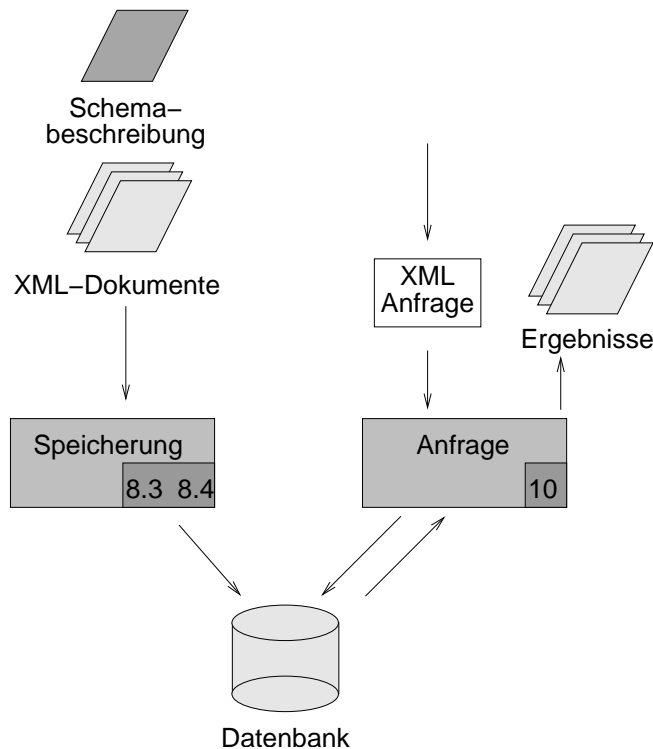


Abbildung 2.5
Speicherung von datenzentrierten XML-Dokumenten

Die Unterscheidung von XML-Dokumenten nach der Eigenschaft daten- oder dokumentenzentriert findet man in *Kapitel 6, Abschnitt 6.1*.

Für die Speicherung datenzentrierter XML-Dokumente gibt es mehrere geeignete Möglichkeiten. Einen ersten Überblick darüber gibt *Kapitel 6*.

Eine genauere Beschreibung der Speicherung von XML-Dokumenten erfolgt in *Kapitel 8*. Dabei kann die Graphenstruktur des Dokuments gespeichert werden (*Abschnitt 8.3*). Eine andere Möglichkeit ist die strukturierte Speicherung in einem Datenbanksystem (*Abschnitt 8.4*).

Um Informationen der gespeicherten XML-Dokumente nutzbar zu machen, müssen diese angefragt werden können. *Kapitel 10* gibt einen

umfassenden Überblick über XML-Anfragesprachen und stellt deren Funktionsumfang und Besonderheiten dar.

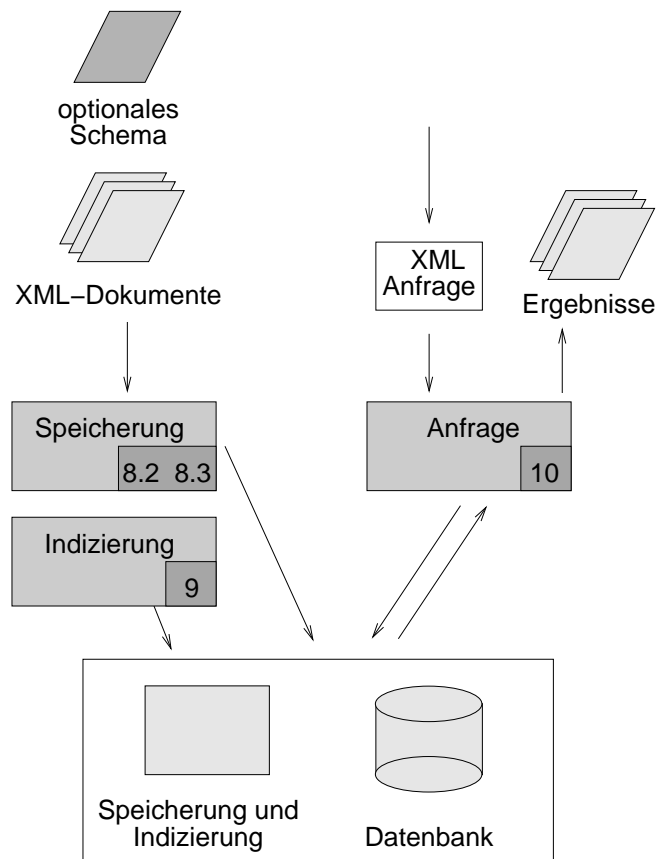
In *Kapitel 11* werden kommerzielle Produkte vorgestellt, die zur Speicherung von XML-Dokumenten eingesetzt werden.

2.5 Speicherung von dokumentenzentrierten XML-Dokumenten

Kapitel 6,8,9,10,11

Das Szenario ähnelt dem vorherigen, es sollen ebenfalls XML-Dokumente gespeichert werden. Die Dokumente sehen hier jedoch anders aus, weshalb es erforderlich ist, andere Methoden zur Speicherung heranzuziehen. Dokumentenzentrierte XML-Dokumente sind Dokumente, die unregelmäßig strukturiert sind. In diesen sind überwiegend Texte mit Markup enthalten. Es werden also keine reinen Daten dargestellt. Abbildung 2.6 zeigt die Teilschritte dieser Anwendung.

Abbildung 2.6
Speicherung von dokumentenzentrierten XML-Dokumenten



Die Unterscheidung der XML-Dokumente nach der Eigenschaft daten- oder dokumentenzentriert findet man in *Kapitel 6*, Abschnitt

6.1. Für die Speicherung dokumentenzentrierter XML-Dokumente gibt es mehrere Verfahren. Eine erste Einordnung stellt Kapitel 6 dar.

Eine genauere Beschreibung der Speichermethoden für XML-Dokumenten erfolgt in *Kapitel 8*. Dokumente können vollständig innerhalb oder außerhalb von Datenbanksystemen gespeichert und indiziert werden (siehe Abschnitt 8.2). Eine andere Variante ist die Speicherung der Graphenstruktur des XML-Dokuments (*Abschnitt 8.3*). Beide Varianten werden anhand verschiedener konkreter Methoden dargestellt.

Mögliche interne Realisierungen der Indizes, um auf Texte, Werte und XML-Informationen zuzugreifen, werden in Kapitel 9 dargestellt.

Um die Informationen in den auf diese Weise gespeicherten XML-Dokumenten zu nutzen, müssen sie angefragt werden können. *Kapitel 10* gibt einen umfassenden Überblick über existierende XML-Anfragesprachen.

In *Kapitel 11* werden kommerzielle Produkte vorgestellt, die zur Speicherung von dokumentenzentrierten XML-Dokumenten geeignet sind.

2.6 Speicherung von XML-Dokumenten in bestehende Datenbanken

Ein häufiges Szenario ist das folgende: Für eine bestimmte Anwendung werden Datenbanken eingesetzt, in denen zahlreiche Informationen gespeichert werden. Zusätzlich sollen auch Informationen aus XML-Dokumenten in diese Datenbanken integriert werden. *Abbildung 2.7* fasst die einzelnen Teilaufgaben dabei zusammen.

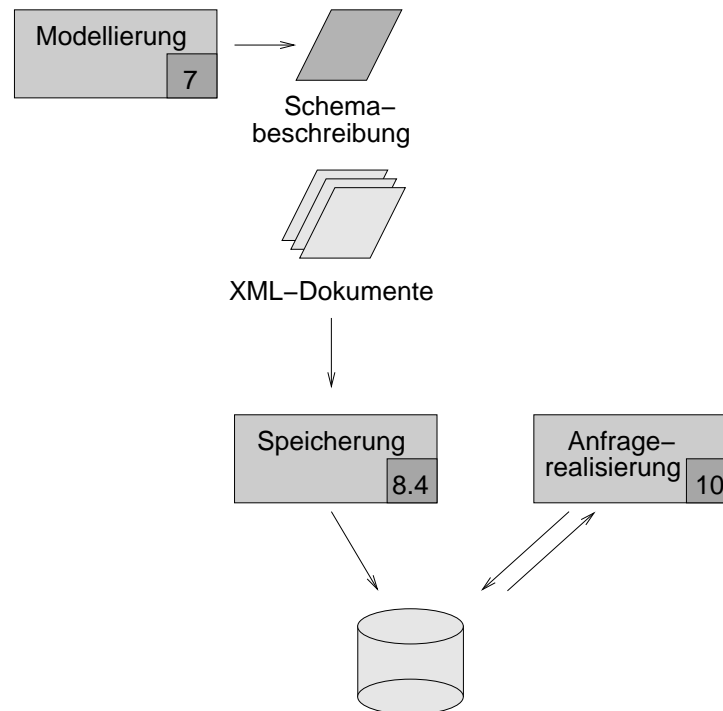
Kapitel 6,7,8,10,11

Man hat also mit der Speicherung von XML-Dokumenten den gleichen Fall wie im obigen Szenario. Allerdings ist die Methode zur Speicherung der XML-Dokumente dadurch festgelegt, dass ein konkretes Datenbanksystem vorgegeben ist und das Datenbankschema bereits existiert.

Auch in diesem Szenario ist es wichtig festzustellen, welche Daten sich in den XML-Dokumenten befinden und welche Arten von Anfragen sich auf den Dokumenten realisieren lassen sollen. Eine Klassifikation von XML-Dokumenten wird in *Kapitel 6* vorgestellt. Darauf aufbauend kann man dann feststellen, welche Art von Speichermethoden sich für die gegebene Dokumentkollektion eignet.

Die Speicherung von XML-Dokumenten kann entweder als CLOB innerhalb eines Datenbanksystems oder als strukturierte Speicherung in einem Datenbanksystem erfolgen. Im zweiten Fall benötigt man eine Schemabeschreibung der XML-Dokumente. Diese muss entweder gegeben sein oder aber sie kann mit Methoden, die in *Kapitel 7*, Abschnitt 7.3 vorgestellt werden, aus den Dokumenten abgeleitet werden. Beide

Abbildung 2.7
Speicherung von XML-Dokumenten in vorgegebene Datenbanken



Varianten zur Speicherung von XML-Dokumenten werden in *Kapitel 8* anhand vieler verschiedener konkreter Methoden erläutert.

Die Informationen der gespeicherten Dokumente müssen angefragt werden können. Das kann über Datenbank-Anfragesprachen oder XML-Anfragesprachen erfolgen. *Kapitel 10* gibt einen umfassenden Überblick über XML-Anfragesprachen und stellt deren Funktionsumfang und Besonderheiten dar.

In *Kapitel 11* werden Datenbanksysteme vorgestellt, die die Speicherung von daten- und dokumentenzentrierten XML-Dokumenten realisieren.

2.7 Austausch von Datenbankinhalten

Kapitel 3,5,7,8,11

Der Austausch von Informationen zwischen verschiedenen Datenbanksystemen stellt in föderierten heterogenen Umgebungen ein großes Problem dar. XML kann als Zwischenformat für den Austausch von Informationen zwischen verschiedenen Datenbanksystemen dienen. Dabei werden XML-Dokumente eingesetzt, um Daten und gleichzeitig die Strukturinformationen in leicht analysierbarer Form zu speichern.

Bei einem solchen Szenario ergeben sich die in *Abbildung 2.8* dargestellten Teilaufgaben.

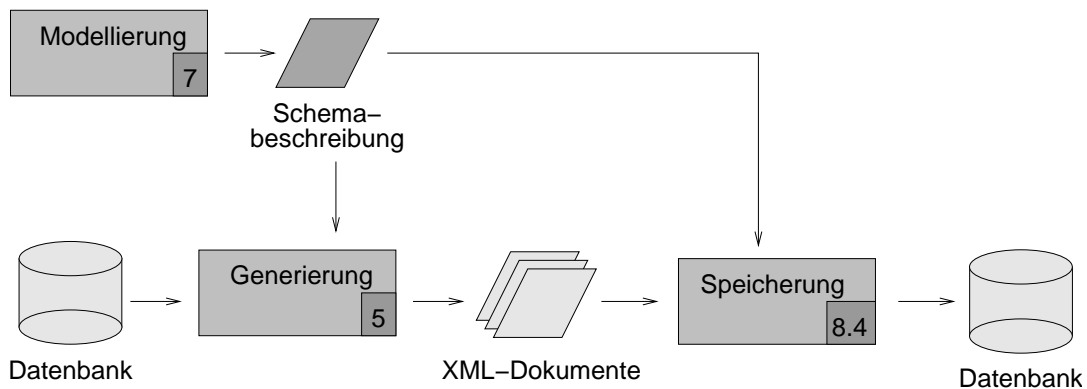


Abbildung 2.8
Austausch von
Datenbankinhalten

Für die Anwendung benötigt man die Prozesse der Generierung von XML-Dokumenten aus Datenbanken (*Kapitel 5*) und die Speicherung von XML-Dokumenten in Datenbanken mit Hilfe von Verfahren, die eine variable Abbildung erlauben (*Abschnitt 8.4*). Damit ist also diese Aufgabe eine Kombination der oben vorgestellten Anwendungen.

2.8 Electronic Data Interchange (EDI)

Der elektronische Datenaustausch ist gegenwärtig eines der Hauptanwendungsgebiete von XML.

Kapitel 3,4,5,6,7,8,11

Dort wird XML eingesetzt, um Daten auszutauschen, die meist strukturiert sind. Oft existiert ein festes Schema für diese Daten. Die Vorteile von XML, das hier frei definierte proprietäre Formate ablöst, liegen darin, dass die Daten selbstbeschreibend und dadurch für Applikationen leichter auswertbar sind. Durch ein definiertes Schema für XML-Dokumente existiert gleichzeitig eine Beschreibung für die Schnittstelle zwischen den einzelnen Anwendungen. XML-Dokumente sind Unicode-Files, die sich problemlos im Internet übertragen lassen.

Für die Anwendung benötigt man wie im vorherigen Anwendungsfall Prozesse, die XML-Dokumente erzeugen und speichern. Damit ist auch hier die Anzahl der notwendigen Teilprozesse sehr groß. Dazu kommt, dass keine Einschränkungen existieren, die besagen, aus welchen Informationen die XML-Dokumente generiert und wie sie gespeichert werden sollen. Das Szenario EDI veranschaulicht *Abbildung 2.9*.

Für eine solche Anwendung kann *Kapitel 3* als Einführung in XML empfohlen werden, sofern diese Informationen dem Leser nicht bereits bekannt sind.

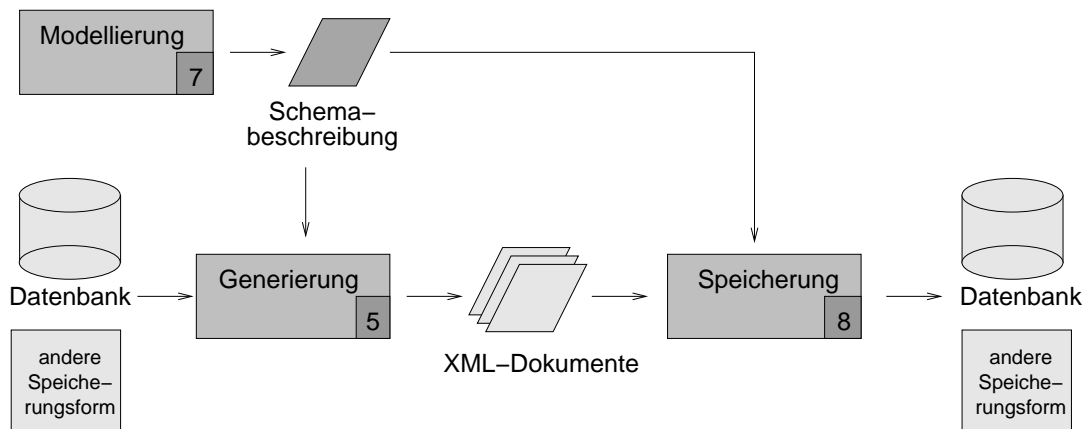


Abbildung 2.9
Electronic Data
Interchange (EDI)

Der elektronische Datenaustausch kann als Kombination der Generierung von XML-Dokumenten aus Datenbanken oder anderen Formaten sowie der Speicherung von XML-Dokumenten gesehen werden.

Deshalb muss man bei der Realisierung dieser Anwendung beide Prozesse (die Dokumentgenerierung und die Dokumentspeicherung) kennen.

Abschnitt 5.3 beinhaltet dazu allgemeine Prinzipien zum Generieren von XML-Dokumenten. Verschiedene Methoden, um XML-Dokumente aus Datenbankinhalten zu erstellen, werden im gesamten *Kapitel 5* angegeben. Dort sind auch allgemeine Prinzipien dargestellt, die bei der Erzeugung von XML-Dokumenten aus anderen Quellen zu beachten sind.

Dabei kann es sinnvoll sein, das Schema der XML-Dokumente vorzugeben. Methoden zur Schemadarstellung und zur konzeptuellen Modellierung von Schemata beschreibt *Kapitel 7*.

Kapitel 8 gibt einen Überblick über Speicherungsmöglichkeiten für XML-Dokumente und beschreibt in detaillierterer Form mehrere Varianten, um XML-Dokumente innerhalb von Datenbanken zu speichern.

Soll die Speicherung der Inhalte nicht durch eine XML-Datenbank oder durch Abbildung auf ein Datenbanksystem, sondern durch die Applikation selbst erfolgen, so stehen ereignisorientierte und baumorientierte XML-Prozessoren bereit, die eine Unterstützung beim Analysieren und Auslesen der Dokumentinformationen liefern. *Kapitel 4* beschreibt beide Möglichkeiten des Einsatzes von XML-Prozessoren.

2.9 Verbindung verschiedener Softwarekomponenten

Bei der Verbindung verschiedener Softwarekomponenten und dem Austausch von Daten zwischen ihnen wurden oder werden häufig proprietäre Formate eingesetzt. Der Einsatz von XML bietet hier den Vorteil, dass diese Formate lesbarer werden und besser beschrieben werden, sodass die Wartung und der Austausch von Software-Komponenten erleichtert wird. Abbildung 2.10 zeigt die Vorgehensweise und die dabei bestehenden Teilaufgaben.

Kapitel 3,4,5,7

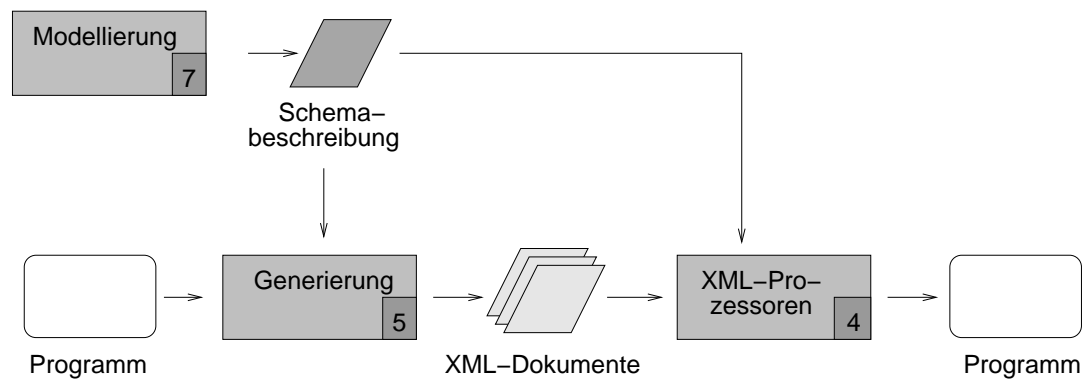


Abbildung 2.10
Kommunikation von
Software durch
Austausch von XML-
Dokumenten

Existiert ein explizites Schema für die eingesetzten XML-Dokumente, so hat es den Charakter einer Dokumentation des Austauschformates. Damit entstehen Vorteile bei der Wartung von Software, besonders im Vergleich zu proprietären Formaten, für die oft keine Dokumentation der Schnittstellen vorliegt.

Diese Anwendung benötigt keine dauerhafte Speicherung der Formate, hat also keinen Datenbankbezug und ist damit nicht im Fokus des Buchs.

Dennoch finden sich hier einige Teilaufgaben, die in den verschiedenen Kapiteln dargestellt werden. Die Erstellung von expliziten Schemata ist Inhalt des *Kapitels 7*. Die XML-Dokumente werden durch Programme erzeugt, allgemeine Kriterien dazu sind in *Abschnitt 5.3* angegeben. XML-Prozessoren können eingesetzt werden, um XML-Dokumente zu analysieren, in *Kapitel 4* findet sich eine Darstellung dazu.

2.10 Weitere Anwendungen

Natürlich können die hier dargestellten Anwendungsszenarien nur exemplarisch einige typische Aufgaben darstellen und nicht vollstän-

dig sein. Auch werden konkrete Anwendungen oft komplizierter sein als hier angegeben.

Die hier aufgezählten Fälle sind als ein erster Einstieg in die Thematik gedacht. In den folgenden Kapiteln werden einzelne Aufgaben und Teilaspekte wesentlich ausführlicher behandelt als hier angerissen.