

# XML und Data Warehousing

Peter Gerstbach

Technische Universität Wien  
e0125397  
peter@gerstbach.at

**Zusammenfassung** In recent years the use of XML has become widespread as a standard for data exchange over the web and to represent semi-structured data. Thus, a large amount of data relevant for decision-making is stored in XML. On the one hand this poses a challenge for data integration in information systems including data warehouses. On the other hand XML offers new ways to work with data warehouses. For example, XML interfaces may be used for data exchange between warehouses or query engines may be used to combine warehouses and XML sources.

This term paper surveys the use of XML in the context of data warehouses. Different applications of the problem and their solutions are described and summarized to provide a coherent description of the state of the art.

## 1 Einführung

Data Warehousing ist ein wertvolles Werkzeug um aus vorhandenen Daten neue Informationen für die Entscheidungsunterstützung zu generieren. Bisher wurden hauptsächlich Daten aus relationalen Datenbanken für diese Analyse herangezogen. Da sich XML als Auszeichnungssprache in vielen Anwendungsbereichen etabliert hat, wird die Analyse dieser Daten im Rahmen eines Data Warehouses immer wichtiger. Frühe Unterstützung von XML wurde dadurch erreicht, dass XML Daten mittels Konverter in relationale Systeme eingespeist wurden. Das Vorhandensein massiver Datenbestände in XML macht es aber immer interessanter Ansätze zu entwickeln, die die Daten direkt verwenden und analysieren. In vielen Bereiche können auch durch hybride Modelle die Vorteile von XML und dem traditionellen Data Warehousing miteinander verbunden werden.

Ziel dieser Arbeit ist es, einen Überblick zu geben, wie XML-Techniken derzeit im Data Warehousing Bereich eingesetzt werden können. Dabei werden einerseits Modelle vorgestellt, die auf konzeptioneller Ebene das Thema Data Warehousing und XML behandeln, andererseits wird dargestellt wie existierende XML-Werkzeuge für Zwecke des Data Warehousing und Data Mining verwendet werden können.

## 2 Related Work

Eine große Anzahl von Arbeiten beschäftigen sich mit dem Themenfeld Data Warehousing und XML. Einige von diesen werden in den folgenden Abschnitten behandelt. Einen guten Überblick über derzeitige Möglichkeiten zeigen Borda-wekar und Lang in [3]. Die Autoren unterscheiden dabei drei unterschiedliche Möglichkeiten, XML Daten in einem Analyse-System zu verwenden:

- Verwendung von XML lediglich um OLAP-Ergebnisse zu repräsentieren,
- Zusätzliche Verwendung von XML bei den Eingabedaten und die
- Benutzung von XML sowohl für die Repräsentation als auch für die Verarbeitung.

Diese Arbeit konzentriert sich hauptsächlich auf die dritte Anwendung, da die Repräsentations-Problematik mit XML auch in Bereichen außerhalb von Data Warehousing bereits ausführlich behandelt wurde.

## 3 XML-Anwendung im Datawarehousing

Dem Data-Warehousing, also dem Betrieb eines Data Warehouses, werden unter anderem folgende Bereiche zugeordnet [17]: Beschaffung und Weiterverarbeitung der Daten im ETL-Prozess, die langfristige Datenhaltung im Data-Warehouse, und die Datenauswertung und -analyse.

Vor der Inbetriebnahme ist im Besonderen das Design des Data Warehouses von Bedeutung, im laufenden Betrieb auch der Austausch von Daten und Metadaten. In allen genannten Bereichen werden bereits heute XML Techniken angewandt, die in den folgenden Abschnitten beschrieben werden.

### 3.1 Data Warehouse Design

Üblicherweise wird das Design eines Data Warehouses, bestehend aus einer Faktentabelle und mehreren Dimensionstabellen, manuell oder teilweise automatisiert erstellt. Da die Daten zumeist aus einer relationalen Datenbank ausgelesen werden, können die Informationen über die Datenstruktur aus dem Datenbankschema verwendet werden um das Design des Data Warehouses zu erstellen.

Bei Data Warehouse-Systemen, die die Daten aus XML Dokumenten beziehen, existiert jedoch kein relationales Modell, um diesen Prozess automatisieren zu können. XML Dokumente können jedoch gegen Dokumentstruktur-Definitionen, wie DTD oder XML Schema, validiert werden. In [5] beschreiben die Autoren die Möglichkeit mit einem halbautomatischen Prozess die Struktur eines Data Warehouses aus DTD bzw. XML Schema-Definitionen zu generieren. Dabei wird in einem ersten Schritt die DTD vereinfacht und anschließend ein Graph daraus gebildet. Dann erfolgt die manuelle Auswahl der Fakten, denen dann wieder automatisiert die Dimensionen anhand der Attribute aus der DTD zugeordnet werden können. Bei der Bestimmung der Kardinalität von Elementen in DTD ist es jedoch in manchen Fällen notwendig die XML-Daten direkt

abzufragen, da in DTDs nur zwischen den Kardinalitäten 1, ? und \* unterschieden werden kann. Das Überprüfen der Kardinalitäten kann jedoch automatisiert durch XPath-Abfragen durchgeführt werden.

### 3.2 Datenbeschaffung und -weiterverarbeitung

Der ETL-Prozess ist ein wesentlicher Verarbeitungsschritt, der bei der Übernahme von Daten aus operativen Systemen in ein Data Warehouse anfällt. Er besteht aus den drei Phasen *Extraktion*, *Transformation* und *Laden*. Im ersten Schritt werden relevante Daten ausgewählt und aus dem Quellsystem extrahiert, im zweiten Schritt werden diese Daten umgeformt, um dann im dritten Schritt in das Data Warehouse geladen zu werden [8]. Üblicherweise ist ein solches Quellsystem eine relationale Datenbank. Aufgrund der weiten Verbreitung von XML unterstützen aber die meisten ETL-Werkzeuge auch XML-Quelldaten.

In [11] wird ein System behandelt, das eine XML Sprache verwendet um aus heterogene Datenquellen einen OLAP-Würfel zu erstellen. Die Motivation für dieses System besteht darin, für Anwendungen die nur einen Teil der gesamten Daten benötigen, den OLAP-Würfel erst dann zu erstellen wenn er benötigt wird. Diese Aufgabe übernimmt ein eigener Server, der die externen Datenbanken abfragt. Da es sich um heterogene Datenquellen handelt, bietet sich in diesem Fall eine XML Sprache an. Einerseits besteht diese Sprache aus einem Schema das den OLAP-Würfel beschreibt und andererseits aus einem Schema, das Angaben über die Verteilung der Daten enthält. Beim Erzeugen des OLAP-Würfels werden dann entsprechend der angegebenen Verteilung die Daten aus den einzelnen Datenbanken angefordert, wobei die Ergebnisse der SQL Anfragen von einer Software in XML umgewandelt werden. Diese Daten können dann mit XML-Werkzeugen wie XSLT aggregiert und transformiert werden. Anschließend kann die API eines OLAP-Servers verwendet werden um die Daten in das System zu laden.

### 3.3 Langfristige Datenhaltung

Für die langfristige Datenhaltungen eines Data Warehouses kann im wesentlichen zwischen zwei Arten der Datenhaltung unterschieden werden: Bei der *physischen, mehrdimensionalen Datenhaltung* (MOLAP) werden proprietäre, mehrdimensionale Datenbanksysteme eingesetzt (MDDBMS) und mit einem OLAP-Server verbunden. Bei der virtuellen, mehrdimensionalen Datenhaltung (ROLAP) können hingegen relationale Datenbanksysteme (RDBMS) in Verbindung mit entsprechenden Modellierungstechniken und einer OLAP-Engine verwendet werden.

Seit der Einführung in 1996 wurde XML der defacto-Standard für die Repräsentation und den Austausch semistrukturierter Daten im Web. Da ein traditionelles Data Warehouse keine semistrukturierten Daten wie XML unterstützt, müssen andere Möglichkeiten gefunden werden diese Daten zu speichern. Im folgenden werden zwei unterschiedliche Ansätze näher erläutert: einerseits der

Einsatz von speziellen XML Warehouses andererseits der Verbund von traditionellen Data Warehouses und XML Repositories.

**XML Warehouses** Um die Verarbeitung und Speicherung von XML Daten zu verbessern, wurde schon früh begonnen native XML-Datenbanken einzusetzen. Auch relationale Datenbanken unterstützen bereits XML-Datentypen und -Abfragen. Für besonders große Mengen an XML Daten, besonders, wenn diese für Analysezwecke abgelegt werden, bietet sich ein sogenanntes XML Warehouse an, das auf die Verwaltung und Abfrage großer Datenmengen spezialisiert ist. Eine sehr frühe Entwicklung einer solchen Implementierung ist das Xyleme Projekt[18] das auch zu einem kommerziellen Produkt geformt wurde. Eine Übersicht über ähnliche Projekte und ein verteiltes XML Warehouse ist in [14] beschrieben.

In traditionellen Datenbanken werden Views eingesetzt, um den Zugriff auf das Datenbankschema zu vereinfachen. Diese Anforderung kann auch an XML Warehouses gestellt werden. In Xyleme ist es beispielsweise möglich Views mit einer XQuery-ähnlichen Sprache zu definieren und abzufragen. Die Implementierung ist zudem verteilt und dadurch skalierbar [1].

**Data Warehouse im Verbund mit XML** Eine weitere Möglichkeit, XML Daten zu nutzen, ohne sie in ein Data Warehouse-taugliches Format zu konvertieren besteht darin, eine Kombination aus Data Warehouse und XML Repository einzusetzen. Beispielsweise beschreiben Pérez et al in [13] ein sogenanntes *Contextualized Warehouse* das die Verkaufsdaten aus einem betrieblichen Data Warehouse mit Wirtschaftsnachrichten im XML-Format aus einem Document Warehouse verbindet. Durch das Zusammenführen von Verkaufsdaten des Unternehmens mit den Wirtschaftsdaten einer gesamten Branche oder Region ist es möglich zusätzliche Zusammenhänge zu erklären, wie beispielsweise den Rückgang von Produktverkaufszahlen aufgrund einer Finanzkrise.

Zwei weitere Möglichkeiten für den Verbund von Data Warehouses und XML (*OLAP-XML Federations*) werden von Yin et al[19] und Jensen et al[10] behandelt. Unterschiedliche Optimierungswege für Abfragen an OLAP-XML Federations werden in [12] vorgeschlagen. Eine Architektur für ein solches System ist in Abbildung 1 zu sehen. Dabei übernimmt ein *Federation Manager* die Aufgabe die beiden Datenbanken zusammenzuführen und in den jeweiligen Sprachen die Abfragen zu stellen.

### 3.4 Datenauswertung und -analyse

Dieser Abschnitt beschäftigt sich mit den Möglichkeiten, wie XML-Daten analysiert werden können bzw wie XML Techniken die Analyse von herkömmlichen, relationalen Daten vereinfachen können.

Um in Data Warehouses unbekannte Informationen zu entdecken und zu extrahieren werden Data Mining Techniken angewendet. Die meisten Implementierungen erwarten dabei, dass die Daten in relationaler Form vorliegen. Um XML

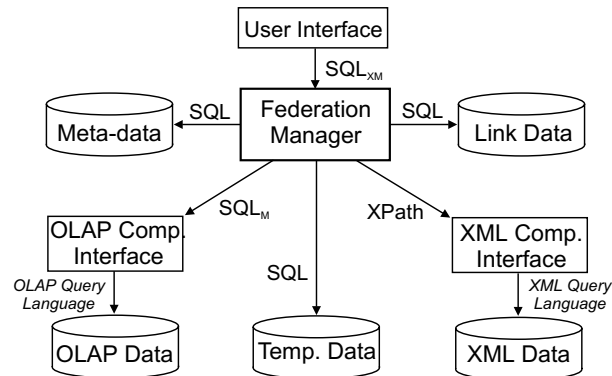


Abbildung 1. Architecture of a Federation System [12]

Daten analysieren zu können werden diese deswegen meist durch vorgelagerte Verarbeitungsschritte zuerst in relationale Form gebracht. Anschließend kann dann die Analyse durch normale OLAP- und Data Mining-Werkzeuge durchgeführt werden. Durch Techniken wie XPath und XQuery ist es jedoch auch möglich *XML Mining* direkt, ohne die vorherige Umwandlung, durchzuführen.

**Assoziationsanalyse** In [4] führen die Autoren einen eigenen Operator für XML Mining ein, der auf XPath basiert und an die Syntax von XQuery angelehnt ist. Listing 1.1 zeigt solch eine Abfrage, die mittels Assoziationsanalyse Regeln über Co-Autoren einer wissenschaftlichen XML-Datenbank ermittelt. Listing 1.2 zeigt einen Ausschnitt der Ergebnismenge.

Listing 1.1. XMine Query [4]

```

XMINE RULE
IN document (" http://...")
FOR ROOT IN //People/*/Publications/*
LET BODY := ROOT/Author, HEAD := ROOT/Author
EXTRACTING RULES WITH
  SUPPORT = 0.1 AND CONFIDENCE = 0.2
RETURN
<RULE support={ SUPPORT } confidence={ CONFIDENCE }>
  <BODY> { FOR $item IN BODY
    RETURN <Item> { $Item } </Item> } </BODY>
  <HEAD> { FOR $item IN HEAD
    RETURN <Item> { $Item } </Item> } </HEAD>
</RULE>

```

Listing 1.2. XMine Query Result [4]

```

<RULE support="0.25" confidence="0.33">

```

```

<BODY><Item><Author>Holmes</Author></Item></BODY>
<HEAD><Item><Author>Stolzmann</Author></Item></HEAD>
</RULE>

```

Da XQuery Turing-vollständig ist, muss es auch möglich sein mit XQuery solch eine Assoziationsanalyse ohne proprietäre Erweiterungen durchzuführen. Eine Implementierung des Apriori-Algorithmus, der auch ohne jegliche Vor- oder Nachbearbeitung auskommt, wird in [16] vorgestellt. Performance-Messungen zeigen jedoch, dass bei großen Itemsets das mehrfache Einlesen der Datenbank einen großen Geschwindigkeitsnachteil bringen kann. Um eine vergleichbare Geschwindigkeit zu einer C++ Implementierung zu erreichen wäre es nötig, Update- und Insert-Erweiterungen für XQuery zu entwickeln, um Zwischenwerte zu speichern und das mehrfache Einlesen der Datenbank zu vermeiden.

Die Anwendung von XPath bzw. XQuery für Analyse-Zwecke offenbart in diesem Zusammenhang auch, dass in den derzeitigen Implementierungen noch viele Optimierungsmöglichkeiten stecken. Im Gegensatz zu XSLT ist es besonders bei XPath und XQuery-Abfragen möglich, Abfrage-Optimierungen ähnlich wie bei relationalen Abfragen durchzuführen (vergl. [6]).

**Klassifikation** Ein weiteres Teilgebiet des Data Mining ist die Klassifikation. Derzeit wenden viele Klassifikationsmethoden für XML Dokumente lediglich Methoden an, die auf Information Retrieval basieren. Dadurch werden die XML Daten nur als Ansammlung von Worten behandelt und eine Menge an Informationen, die in den Dokumenten versteckt sind, werden nicht für die Klassifikation herangezogen. In [20] präsentieren die Autoren eine Methode die auf alle semistrukturierten Daten anwendbar ist und neben dem Inhalt auch die Struktur von XML-Daten für die Klassifikation heranziehen und dadurch bessere Ergebnisse erhalten als bei reinen IR-basierten Methoden.

**XML for Analysis** Abseits der Analyse von XML Daten gibt es auch Bestrebungen die Vorteile von XML bei der Analyse von normalen Data Warehouses zu verwenden. *XML for Analysis* (XMLA) ist ein aktueller Ansatz eine standardisierte API für Online Analytical Processing and Business Intelligence (BI) zu formen. XMLA wurde ursprünglich von Microsoft entwickelt und wird derzeit auch von Hyperion, SAP und SAS unterstützt. Der Standard ist gedacht, um Clients eine API zur Verfügung zu stellen, um multidimensionale Datenbanken abzufragen. Die API basiert auf Web Services (HTTP, SOAP und XML) und benutzt die von Microsoft entwickelte Abfragesprache MDX für multidimensionale Daten.

Abbildung 2 zeigt die Architektur von XML for Analysis. Ein Benutzer kann beispielsweise über die GUI eines Web Service Clients eine Abfrage durchführen. Der Client sendet diese Abfrage via HTTP und SOAP an den Web Service, der dann die eigentliche Abfrage der Datenquellen durchführt und das Ergebnis zurückliefert. [15]

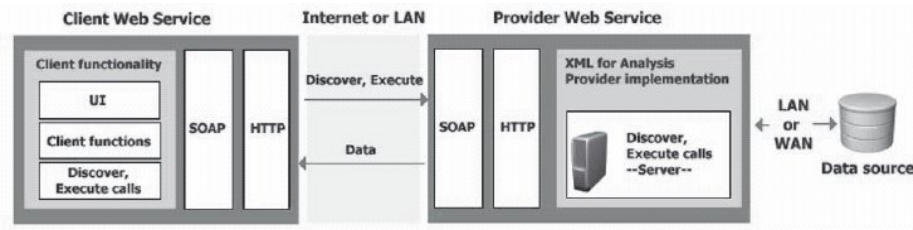


Abbildung 2. Architektur von XML for Analysis [15]

### 3.5 Datenaustausch

In Abschnitt 3.2 wurden bereits Möglichkeiten für den Import von XML-Daten erörtert. Eine andere Möglichkeit der Datenbeschaffung ist der Austausch von kompletten Data Cubes aus einem externen Data Warehouse. Dies ermöglicht die Erstellung eines virtuellen Data Warehouses, das die Daten aus unterschiedlichen im Unternehmen oder auch unternehmensübergreifend existierenden Data Warehouses integriert. Eine wichtige Anforderung ist dabei ein herstellernabhängiges Austauschformat, wofür sich XML besonders anbietet. In [9] beschreiben die Autoren *XCube*, eine Sammlung von XML-basierten Sprachen, die eine Beschreibung der Datenschemata, Dimensionen, Fakten und auch Abfragen eines Data Warehouses ermöglicht. Durch die Kombination der Sprachen ist es möglich Daten über einen Webserver anzubieten, herunterzuladen oder auch nur Teilbereiche von Cubes zu verwenden.

Um nur die Metadaten (ohne die Daten selbst) eines Data Warehouses zu beschreiben existieren weitere Ansätze, beispielsweise die *Common Warehouse Metamodel Specification* der Object Management Group [7]. Sie basiert auf UML, MOF und XMI und hat das Ziel einen einfachen Austausch von Metadaten zwischen Data Warehouse-Werkzeugen zu ermöglichen. Ein weiterer Ansatz ist MetaCube, für den mit MetaCube-X auch eine XML-basierte Beschreibung existiert [2].

## 4 Zusammenfassung

Die vorliegende Arbeit fasst die Anwendungsmöglichkeiten von XML im Data Warehousing zusammen. Grob kann man dabei zwischen Ansätzen unterscheiden, die direkt mit XML-Daten arbeiten bzw. Ansätzen die XML-Techniken nutzen, um auf vorhandene Data Warehouses zuzugreifen oder interoperabel zu machen.

Zusammenfassend zeigt diese Arbeit, dass XML in vielen Bereichen des Data Warehousing bereits sinnvoll eingesetzt werden kann. Begonnen mit dem Design eines OLAP-Würfels, der Datenbeschaffung, der Datenhaltung und im besonderen der Auswertung und Analyse. Da die Verbreitung von XML in Zukunft weiter steigen wird, fallen auch immer mehr Daten im XML-Format an, die für

Analysen in Frage kommen, um Entscheidungen schneller und besser fallen zu können. Aus diesem Grund wird das Thema XML für Business Intelligence in Zukunft eine immer stärkere Rolle spielen.

## Literatur

1. AGUILERA, VINCENT, SOPHIE CLUET, TOVA MILO, PIERANGELO VELTRI und DAN VODISLAV: *Views in a large-scale XML Repository*. The VLDB Journal, 11(3):238–255, 2002.
2. BINH, NGUYEN THANH, A. MIN TJOA und OSCAR MANGISENGI: *MetaCube-X: An XML Metadata Foundation for Interoperability Search among Web Data Warehouses*. In: *Design and Management of Data Warehouses*, Seite 8, 2001.
3. BORDAWEKAR, RAJESH R. und CHRISTIAN A. LANG: *Analytical Processing of XML Documents: Opportunities and Challenges*. SIGMOD Rec., 34(2):27–32, 2005.
4. BRAGA, DANIELE, ALESSANDRO CAMPI, STEFANO CERI, MIKA KLEMETTINEN und PIERLUCA LANZI: *Discovering Interesting Information in XML Data with Association Rules*. In: *SAC '03: Proceedings of the 2003 ACM Symposium on Applied Computing*, Seiten 450–454, New York, NY, USA, 2003. ACM Press.
5. GOLFARELLI, MATTEO, STEFANO RIZZI und BORIS VRDOLJAK: *Data Warehouse Design from XML Sources*. In: *DOLAP '01: Proceedings of the 4th ACM International Workshop on Data Warehousing and OLAP*, Seiten 40–47, New York, NY, USA, 2001. ACM Press.
6. GOTTLÖB, G., CH. KOCH und R. PICHLER: *XPath Query Evaluation: Improving Time and Space Efficiency*. In: *Proc. 19th Int. Conf. on Data Engineering (ICDE 2003)*, Bangalore, India, 2003. IEEE Computer Society.
7. GROUP, OMG OBJECT MANAGEMENT: *Common Warehouse Metamodel (CWM) Specification*. <http://www.omg.org/docs/formal/03-03-02.pdf>, März 2003. Version 1.1, Volume 1.
8. HEILMANN, HEIDI, HANS-GEORG KEMPER und HENNING BAARS: *HMD-Glossar*. HMD - Praxis der Wirtschaftsinformatik, 43(247):117–118, Februar 2006.
9. HÜMMER, WOLFGANG, ANDREAS BAUER und GUNNAR HARDE: *XCube: XML for Data Warehouses*. In: *DOLAP '03: Proceedings of the 6th ACM International Workshop on Data Warehousing and OLAP*, Seiten 33–40, New York, NY, USA, 2003. ACM Press.
10. JENSEN, MIKAEL R., THOMAS H. MØLLER und TORBEN BACH PEDERSEN: *Specifying OLAP Cubes on XML Data*. Journal of Intelligent Information Systems, 17(2-3):255–280, 2001.
11. NIEMI, TAPIO, MARKO NIINIMÄKI, JYRKI NUMMENMAA und PETER THANISCH: *Constructing an OLAP Cube from Distributed XML Data*. In: *DOLAP '02: Proceedings of the 5th ACM International Workshop on Data Warehousing and OLAP*, Seiten 22–27, New York, NY, USA, 2002. ACM Press.
12. PEDERSEN, DENNIS, KARSTEN RIIS und TORBEN BACH PEDERSEN: *Query Optimization for OLAP-XML Federations*. In: *DOLAP '02: Proceedings of the 5th ACM International Workshop on Data Warehousing and OLAP*, Seiten 57–64, New York, NY, USA, 2002. ACM Press.
13. PÉREZ, JUAN MANUEL, RAFAEL BERLANGA, MARÍA JOSÉ ARAMBURU und TORBEN BACH PEDERSEN: *A Relevance-extended Multi-dimensional Model for a Data Warehouse Contextualized with Documents*. In: *DOLAP '05: Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP*, Seiten 19–28, New York, NY, USA, 2005. ACM Press.

14. RAJUGAN, R., ELIZABETH CHANG und THARAM S. DILLON: *Conceptual Design of an XML FACT Repository for Dispersed XML Document Warehouses and XML Marts*. In: *CIT '05: Proceedings of the The Fifth International Conference on Computer and Information Technology*, Seiten 141–149, Washington, DC, USA, 2005. IEEE Computer Society.
15. TANG, ZHAOHUI, JAMIE MACLENNAN und PETER PYUNGCHUL KIM: *Building Data Mining Solutions with OLE DB for DM and XML for Analysis*. *SIGMOD Rec.*, 34(2):80–85, 2005.
16. WAN, JACKY W. W. und GILLIAN DOBBIE: *Mining Association Rules from XML Data using XQuery*. In: *CRPIT '04: Proceedings of the Second Workshop on Australasian Information Security, Data Mining and Web Intelligence, and Software Internationalisation*, Seiten 169–174, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.
17. WIKIPEDIA: *Data Warehouse*. <http://de.wikipedia.org/wiki/Data-Warehouse>, Mai 2006.
18. XYLEME, LUCIE: *Xyleme: A Dynamic Warehouse for XML Data of the Web*. In: *IDEAS '01: Proceedings of the International Database Engineering & Applications Symposium*, Seiten 3–7, Washington, DC, USA, 2001. IEEE Computer Society.
19. YIN, XUEPENG und TORBEN BACH PEDERSEN: *Evaluating XML-extended OLAP Queries Based on a Physical Algebra*. In: *DOLAP '04: Proceedings of the 7th ACM International Workshop on Data Warehousing and OLAP*, Seiten 73–82, New York, NY, USA, 2004. ACM Press.
20. ZAKI, MOHAMMED J. und CHARU C. AGGARWAL: *XRules: an Effective Structural Classifier for XML Data*. In: *KDD '03: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seiten 316–325, New York, NY, USA, 2003. ACM Press.