

LEHRSTUHL FÜR
ALLG. BWL UND WIRTSCHAFTSINFORMATIK
UNIV.-PROF. DR. HERBERT KARGL

Franke, Thomas S.; Sulzbach, Simon

**Implikationen des Einsatzes der
eXtensible Markup Language –
Teil 1: XML-Grundlagen**

ARBEITSPAPIERE WI
Nr. 02/2000

Schriftleitung:
PD Dr. habil. Axel C. Schwickert

Information

Reihe: Arbeitspapiere WI

Herausgeber: Univ.-Prof. Dr. Axel C. Schwickert
Professur für BWL und Wirtschaftsinformatik
Justus-Liebig-Universität Gießen
Fachbereich Wirtschaftswissenschaften
Licher Straße 70
D – 35394 Gießen
Telefon (0 64 1) 99-22611
Telefax (0 64 1) 99-22619
eMail: Axel.Schwickert@wirtschaft.uni-giessen.de
<http://wi.uni-giessen.de>

Bis Ende des Jahres 2000 lag die Herausgeberschaft bei:

Lehrstuhl für Allg. BWL und Wirtschaftsinformatik
Johannes Gutenberg-Universität Mainz
Fachbereich Rechts- und Wirtschaftswissenschaften
Welderweg 9
D - 55099 Mainz

Ziele: Die Arbeitspapiere dieser Reihe sollen konsistente Überblicke zu den Grundlagen der Wirtschaftsinformatik geben und sich mit speziellen Themenbereichen tiefergehend befassen. Ziel ist die verständliche Vermittlung theoretischer Grundlagen und deren Transfer in praxisorientiertes Wissen.

Zielgruppen: Als Zielgruppen sehen wir Forschende, Lehrende und Lernende in der Disziplin Wirtschaftsinformatik sowie das IuK-Management und Praktiker in Unternehmen.

Quellen: Die Arbeitspapiere entstanden aus Forschungsarbeiten, Diplom-, Studien- und Projektarbeiten sowie Begleitmaterialien zu Lehr- und Vortragsveranstaltungen des Lehrstuhls für Allg. Betriebswirtschaftslehre und Wirtschaftsinformatik Univ. Prof. Dr. Herbert Kargl an der Johannes Gutenberg-Universität Mainz.

Hinweise: Wir nehmen Ihre Anregungen und Kritik zu den Arbeitspapieren aufmerksam zur Kenntnis und werden uns auf Wunsch mit Ihnen in Verbindung setzen.
Falls Sie selbst ein Arbeitspapier in der Reihe veröffentlichen möchten, nehmen Sie bitte mit dem Herausgeber (Gießen) unter obiger Adresse Kontakt auf.
Informationen über die bisher erschienenen Arbeitspapiere dieser Reihe und deren Bezug erhalten Sie auf dem Schlußblatt eines jeden Arbeitspapiers und auf der Web Site des Lehrstuhls unter der Adresse <http://wi.uni-giessen.de>

Arbeitspapiere WI Nr. 2/2000

- Autoren:** Franke, Thomas S.; Sulzbach, Simon
- Titel:** Implikationen des Einsatzes der eXtensible Markup Language – Teil 1: XML-Grundlagen
- Zitation:** Franke, Thomas S.; Sulzbach, Simon: Implikationen des Einsatzes der eXtensible Markup Language – Teil 1: XML-Grundlagen, in: Arbeitspapiere WI, Nr. 2/2000, Hrsg.: Lehrstuhl für Allg. BWL und Wirtschaftsinformatik, Johannes Gutenberg-Universität: Mainz 2000.
- Kurzfassung:** Die eXtensible Markup Language (XML) ist eine von der Generalized Markup Language (SGML) abgeleitete standardisierte Sprache zur Notation der Syntax von Auszeichnungssprachen (Metasprache). Die Kernidee von XML besteht in der strikten Trennung von Inhalt, Struktur und Darstellung. Mit XML werden aktuell bereits eine Reihe von Auszeichnungssprachen (sog. XML-Anwendungen), z. B. für Graphiken, eCommerce und die Darstellung chemischer Formeln, definiert. Der Einsatzbereich von XML ist nicht auf die Definition von Auszeichnungssprachen für das World Wide Web beschränkt, vielmehr kann XML als Basistechnologie für offene Datenformate und einfachen, plattformübergreifenden Datenaustausch eingesetzt werden. Die vorliegenden Arbeitspapiere WI Nr. 2/2000 und Nr. 3/2000 untersuchen systematisch die Implikationen, die von der Etablierung von XML ausgehen. Hierzu stellt das Arbeitspapier WI Nr. 2/2000 zunächst Konzept, Entwicklung, Funktionsweise und Peripherie von XML dar. Der zweite Teil in Arbeitspapier WI Nr. 3/2000 behandelt Einsatzgebiete und betriebswirtschaftliche Implikationen des Einsatzes von XML.
- Schlüsselwörter:** eXtensible Markup Language (XML), Document Type Definition (DTD), eXtensible Style Sheet Language (XSL), XML Linking, XML Pointer (XPointer), eBusiness, eCommerce, eIntegration, eWorkflow, EDI, XML/EDI, betriebliche Datenhaltung, zwischenbetrieblicher Datenaustausch, Internet, WWW, HTML, XHTML

Inhaltsverzeichnis

Vorbemerkung zu „Teil 1: XML-Grundlagen“	3
1 Das Konzept von XML	4
2 Die Entwicklung von XML	8
2.1 Das World Wide Web Consortium (W3C).....	8
2.2 Zeittafel der Entwicklung von XML.....	9
3 Die Funktionsweise von XML.....	10
3.1 Allgemeine Funktionsweise.....	10
3.2 Aufbau eines XML-Dokuments.....	15
3.3 Dokumenttyp-Definition.....	18
3.4 Wohlgeformte und gültige XML-Dokumente	23
4 Die Peripherie von XML.....	23
4.1 Funktionale Erweiterung von XML durch periphere Module	23
4.2 Stylesprachen	24
4.3 Verweistechiken.....	28
4.4 Namensräume.....	29
5 XML-Dokumente mit dem Document Objekt Model und XML-QL....	31
Literaturverzeichnis	33

Vorbemerkung zu „Teil 1: XML-Grundlagen“

Die Hypertext Markup Language (HTML) mit ihren zahlreichen Erweiterungen wird den Anforderungen des heutigen Internet nicht mehr gerecht. Das World Wide Web Consortium (W3C) hat daher am 10. Februar 1998 die erste offizielle Spezifikation einer neuen Sprache für das Internet, der eXtensible Markup Language (XML), veröffentlicht.¹ XML ist nicht, wie der Name suggeriert, eine erweiterbare Auszeichnungssprache – sie ist vielmehr eine von der Generalized Markup Language (SGML) abgeleitete standardisierte Sprache zur Notation der Syntax von Auszeichnungssprachen (Metasprache).² Mit XML werden aktuell bereits eine Reihe von Auszeichnungssprachen (sog. XML-Anwendungen), z. B. für Formeln, Multimediapräsentationen, EDI, eCommerce und Vektorgraphiken, definiert.³ Dies zeigt bereits, daß der Einsatzbereich von XML nicht auf die Definition von Auszeichnungssprachen für die Darstellung von Dokumenten im World Wide Web (WWW) beschränkt ist. Vielmehr kann XML als Basistechnologie für offene Datenformate und einfachen, plattform- und applikationsübergreifenden Datenaustausch eingesetzt werden – ähnlich wie einst der ASCII-Zeichensatz den rudimentären Datenaustausch zwischen Rechnerwelten ermöglichte.⁴

Kaum einer anderen Programmier- und Beschreibungssprache ist es gelungen, in so kurzer Zeit eine solche Bedeutung zu erlangen wie XML⁵ – und das obwohl XML keine technologischen oder konzeptuellen Neuigkeiten bietet. Der Erfolg läßt sich daher eher über die große Akzeptanz von XML als Standard für eine Sprache mit sehr einfacher Mächtigkeit erklären, dessen Entwicklung vergleichbar ist mit der von HTML. Beinahe alle namhaften Software-Anbieter waren an der Entwicklung des XML-Standards beteiligt und viele von ihnen haben für zukünftige Versionen ihrer Produkte XML-Unterstützung angekündigt bzw. bereits veröffentlicht.

Die vorliegenden Arbeitspapiere WI Nr. 2/2000 und Nr. 3/2000 untersuchen systematisch die Auswirkungen der Etablierung von XML auf eBusiness-Aktivitäten, insbesondere den zwischenbetrieblichen Datenaustausch und die innerbetriebliche Datenhaltung. Hierzu stellt das Arbeitspapier WI Nr. 2/2000 zunächst Konzept, Entwicklung, Funktionsweise und Peripherie von XML dar. Arbeitspapier WI Nr. 3/2000 behandelt Einsatzgebiete und betriebswirtschaftliche Implikationen des Einsatzes von XML.

-
- 1 Vgl. Bray, Tim; Paoli, Jean; Sperberg-McQueen, C.M.: Extensible Markup Language (XML) W3C Recommendation, Online im Internet: <http://www.w3.org/TR/1998/REC-xml-19980210>, 10.02.1998.
 - 2 Vgl. Tolksdorf, Robert: XML und darauf basierende Standards: Die neuen Auszeichnungssprachen des Web, in: Informatik Spektrum, Dez. 1999, S. 408.
 - 3 Vgl. Tolksdorf, Robert: XML und darauf basierende Standards: Die neuen Auszeichnungssprachen des Web, a. a. O., S. 415 ff. Vgl. auch Reinbold, Holger: XML basierte Graphiken fürs Web, in: it FOKUS, 2/2000.
 - 4 Vgl. Jung, Frank: Universelles Datenaustauschformat, in: it FOKUS, 2/2000, S. 9; Vgl. O'Donnell, Bob: Will XML become the universal document format?, Online im Internet: <http://www.info-world.com/cgi-bin/diplayNew.pl?odonnell/-971215od.htm>, 15.12.1997.
 - 5 Vgl. Farsi, Reza: XML, in: Informatik Spektrum, Dez. 1999, S. 436; Vgl. Sonntag, Ralph: Neue Offenheit für E-Business, in: Diebold Management Report, 11/99, S. 27-29; Vgl. Reinbold, Holger: XML in der Praxis, in: PC Professionell, 4/2000, S. 253.

1 Das Konzept von XML

Das World Wide Web Consortium (W3C) veröffentlichte am 10. Februar 1998 die erste offizielle Spezifikation der XML-Syntax in Form einer Recommendation. XML bekam damit die Bedeutung eines global gültigen Standards. Die Ziele, die bei der Entwicklung von XML verfolgt wurden, beschreibt das W3C in 10 Punkten:⁶

1. XML soll einfach im Internet nutzbar sein: Wie HTML muß auch XML leicht erlernbar und handhabbar sein, um es einer Vielzahl von Benutzern zu erlauben, auf einfache Art Web-Dokumente zu erstellen. Darüber hinaus muß es möglich sein, XML-Daten mit frei verfügbaren und weit verbreiteten Programmen darzustellen und zu verarbeiten.
2. XML muß eine Vielzahl von Anwendungsmöglichkeiten unterstützen: Das Einsatzgebiet von XML darf sich nicht auf das Erstellen von Websites beschränken. Es muß auch die Möglichkeit bieten, Daten zu verwalten, diese zwischen verschiedenen Systemen oder Applikationen auszutauschen, zu verarbeiten und über einen langen Zeitraum aufzubewahren, so daß sie auch zukünftigen Anwendungen zur Verfügung stehen können.
3. XML muß zu SGML kompatibel sein: Vorhandene SGML-Programme sollen XML-Daten interpretieren und verarbeiten können.
4. XML muß von einfachen Programmen verarbeitet werden können: Dieser Aspekt ist von großer Bedeutung für die Verwendung von XML als Standard für Websites und den Datenaustausch. Einfache, frei erhältliche Browser zur Darstellung von XML-Dokumenten können als Voraussetzung für die Verbreitung von XML als Standard gesehen werden. XML-verarbeitende Programme, sogenannte Parser, müssen einfach und in kurzer Zeit zu erstellen sein, damit die Verarbeitung von XML-Daten ohne hohen Aufwand möglich ist und eine möglichst große Zahl von Unternehmen XML als Format zur Haltung und zum Austausch von Daten verwenden kann. XML sollte so einfach sein, daß ein geübter Programmierer innerhalb einer Woche einen Parser codieren kann.⁷
5. Die Anzahl optionaler Merkmale von XML ist gering zu halten: Anders als bei SGML soll die XML-Syntax möglichst ganz auf optionale Spracheigenschaften verzichten. Solche Optionen würden zu Inkompatibilität zwischen einzelnen XML-Dokumenttypen führen, wenn in einem Dokument beispielsweise zwischen Groß- und Kleinschrift unterschieden würde, in einem anderen jedoch nicht. XML muß auf einheitlichen, nicht optionalen Auszeichnungsvorschriften beruhen, damit XML-Dokumente von Universalprogrammen interpretiert werden können
6. XML-Dokumente sollen für Menschen lesbar und hinreichend verständlich sein: Das Verständnis von XML-Daten und ihrer Zusammenhänge in einem Dokument darf

6 Vgl. Bray, Tim; Paoli, Jean; Sperberg-McQueen, C.M.: Extensible Markup Language (XML) W3C Recommendation, a. a. O.

7 Vgl. Bray, Tim: Beyond HTML: XML and Automated Web Processing, Online im Internet: http://developer.netscape.com/viewsource/bray_xml.html, 10.09.1999.

nicht von der visuellen Darstellung der Daten durch eine Software abhängen, sondern muß auch bei der Betrachtung des Quellcodes gegeben sein, so daß der Empfänger eines XML-Dokuments gegebenenfalls auch ohne XML-Browser imstande ist, Inhalt und Struktur des Dokuments zu verstehen.

7. Ergebnisse der Entwicklung von XML sollen schnellstmöglich veröffentlicht werden: Die Notwendigkeit einer neuen Auszeichnungssprache erforderte die schnelle Ausarbeitung und Veröffentlichung einer standardisierten Spezifikation.
8. Die XML-Spezifikation muß formell und kurzgefaßt sein: Durch eine vorwiegend formelle Beschreibung der Syntax soll die Spezifikation im Umfang begrenzt werden, um einfacher und schneller erlernbar zu sein.
9. XML-Dokumente müssen einfach erstellt werden können: Die Auszeichnung von Daten in XML-Format muß auch ohne spezielle Editoren möglich sein. Einfache Texteditoren sollten als Werkzeug genügen. XML-Daten müssen auch von einfachen Programmen, beispielsweise anhand von Script-Befehlen, automatisch erzeugt werden können.
10. Knappheit in der Auszeichnung von XML-Daten ist von minimaler Bedeutung: XML-Dateien sind im Vergleich zu Trennzeichen-separierten Datendateien weitaus umfangreicher, da für jedes Datum in einem XML-Dokument das XML-Tag gespeichert wird, auch wenn sich im Dokument n Daten der selben Datenstruktur befinden. Die durch umfangreiche selbstbeschreibende Markup-Elemente begründete Vergrößerung des Dateivolumens wird bewußt in Kauf genommen, da man sich davon eine bessere (manuelle) Lesbarkeit der Dokumente erhofft.

XML selbst ist eine im Umfang stark reduzierte Teilmenge von SGML, das sich aufgrund seiner Komplexität nicht für den Einsatz in weit verbreiteten Anwendungen mit vielfach ungeübten Benutzern, wie z. B. dem WWW, eignet. Bei der Reduktion von SGML auf XML stand die einfache Erstellbarkeit von XML-Anwendungen im Vordergrund.⁸ Dies zeigt sich im Umfang der Spezifikationen von XML und SGML: So umfaßt die XML-Spezifikation lediglich 26 Seiten, während die Spezifikation von SGML über 500 Seiten stark ist.⁹ Bestimmte Sprachmerkmale, die in SGML optional für jede Anwendung bestimmt werden können, wurden für XML einmalig in der Spezifikation festgeschrieben, so daß sie XML-weit einheitlich gelten und ihnen bei der Definition eines XML-basierten Dokumenttyps nicht explizit ein Wert zugewiesen werden muß:¹⁰

- Unterscheidung von Groß- und Kleinbuchstaben: Mit der SGML-Option NAME-CASE GENERAL wird festgelegt, ob eine SGML-basierte Sprache zwischen Groß-

8 Vgl. Bosak, Jon: XML, Java, and the future of the Web, Online im Internet: <http://www.metalab.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm>, 03.10.1997.

9 Vgl. Weitzel, Tim; Buxmann, Peter; Ladner, Frank; König, Wolfgang: Konzept und Anwendung der Extensible Markup Language, Online im Internet: <http://caladan.wiwi.uni-frankfurt.de/IWI/project-b3/deu/publikat/xml/index.htm>, 13.08.1999.

10 Vgl. Clark, James: Comparison of SGML and XML, Online im Internet: <http://www.w3.org/TR/NOTE-sgml-xml-971215>, 15.12.1997.

und Kleinschrift unterscheidet. XML unterscheidet grundsätzlich zwischen Groß- und Kleinschrift.

- Setzen von Start- und Endtags: In SGML kann optional festgelegt werden, ob Start- und Endtags immer notwendig sind oder ob sie in bestimmten Fällen weggelassen werden können (SGML- Option OMITTAG). In XML muß jedes Element mit einen Starttag <TAG> eröffnet und mit einen Endtag </TAG> geschlossen werden.

XML ist aufwärtskompatibel, das heißt, daß eine SGML-Anwendung XML-Dokumente interpretieren und verarbeiten kann, sofern das XML-Dokument durch eine Dokumenttyp-Definition (DTD) beschrieben wird und die in SGML-Dokumenten normalerweise optionalen Funktionen (Groß- und Kleinschreibung/Endtags) entsprechend der XML-Vorgaben deklariert wurden.

Die eXtensible Markup Language ist eine Metasprache zur Definition von textbasierten Auszeichnungssprachen – eine Art „Metagrammatik für kontextfreie Grammatiken“¹¹. Der Namensbestandteil „eXtensible“ verwirrt häufig, denn entgegen manchmal anzutreffenden Darstellungen bedeutet er nicht, daß es sich bei XML selbst um eine erweiterbare Sprache handelt, sondern vielmehr, daß es sich bei XML um eine Metasprache handelt, mit der eigene Sprachen (XML-Applikationen) definiert werden können.¹² Das bedeutet, daß der Autor eines XML-Dokuments alle verwendeten Elemente an die individuellen Ansprüche der auszuzeichnenden Daten anpassen kann, indem er ihnen selbst gewählte Elementnamen gibt und spezifische Attribute zuweist.

Die mit XML definierten Auszeichnungssprachen („Grammatiken“) werden in sogenannten Document Type Definitions (DTD) gespeichert und als XML-Anwendungen bezeichnet. Die XML-Anwendungen, z. B. XHTML oder XML/EDI, erlauben dann Beschreibung, Darstellung, Austausch und Manipulation von Daten anhand der definierten DTD. Die auf Basis der XML-Anwendung erstellten Dokumente nennt man Instanzen dieser XML-Anwendung. Ein Beispiel soll die Zusammenhänge verdeutlichen:

Die bisher mit SGML definierte Hypertext Markup Language (HTML) Version 4.0 wurde mit XML neu definiert. Die entstandene XML-Anwendung für HTML, das sogenannte XHTML 1.0, liegt nun in Form einer DTD vor.¹³ Diese DTD definiert, welche Sprachelemente und Optionen die Auszeichnungssprache HTML umfaßt. Auf Basis der DTD für XHTML können nun Dokumente (Instanzen) verfaßt werden. Diese müssen sich in ihrer Struktur nach der XHTML-DTD richten damit sie in Sinne der XML-Anwendung XHTML gültige Dokumente darstellen.

11 Tolksdorf, Robert: XML und darauf basierende Standards: Die neuen Auszeichnungssprachen des Web, a. a. O., S. 408.

12 Vgl. Weitzel, Tim; Buxmann, Peter; Ladner, Frank; König, Wolfgang: Konzept und Anwendung der Extensible Markup Language, a. a. O.

13 Vgl. Pemberton, Steven; Altheim, Murray; Austin, Daniel; Boumphrey, Frank; Burger, John; Donoho, Andrew W.; Dooley, Sam; Hofrichter, Klaus; Hoschka, Philipp; Ishikawa, Masayasu; ten Kate, Warner; King, Peter; Klante, Paula; Matsui, Shin'ichi; McCarron, Shane; Navarro, Ann; Nies, Zach; Raggett, Dave; Schmitz, Patrick; Schnitzenbaumer, Sebastian; Wilson, Chris; Wugofski, Ted; Zigmond, Dan: XHTML 1.0: The Extensible HyperText Markup Language, Online im Internet: <http://www.w3.org/TR/1999/PR-xhtml1-19990824>, 24.08.1999.

Damit die in einer Instanz gespeicherten Nutzdaten (content) sowohl für Menschen lesbar wie auch maschinell verarbeitbar sind, erfolgt die semantisch relevante Markierung der Nutzdaten, genau wie bei HTML-Dokumenten, durch Einklammerung von Dokumentbereichen mit Auszeichnungselementen (Tags)¹⁴. Während die SGML-Anwendung HTML jedoch eine vordefinierte Menge von Auszeichnungselemente und Eigenschaften besitzt (die sich zum größten Teil rein auf die Darstellung von Daten beschränken¹⁵), die durch den Anwender nicht erweitert werden kann, ist ein Anwender mit XML in der Lage, sich seine persönliche Auszeichnungssprache selbst zu definieren. Dabei kann er gezielt strukturelle Zusammenhänge (Datenstrukturen) zwischen einzelnen Elementen beliebiger Komplexität abbilden sowie Vorschriften über Inhalt, Struktur und Darstellung von unterschiedlichen Dokumenttypen machen. Dem Anwender bietet XML damit Möglichkeiten, die weit über die von HTML hinausgehen.¹⁶ Abb. 1 zeigt den Zusammenhang zwischen SGML, XML und HTML.

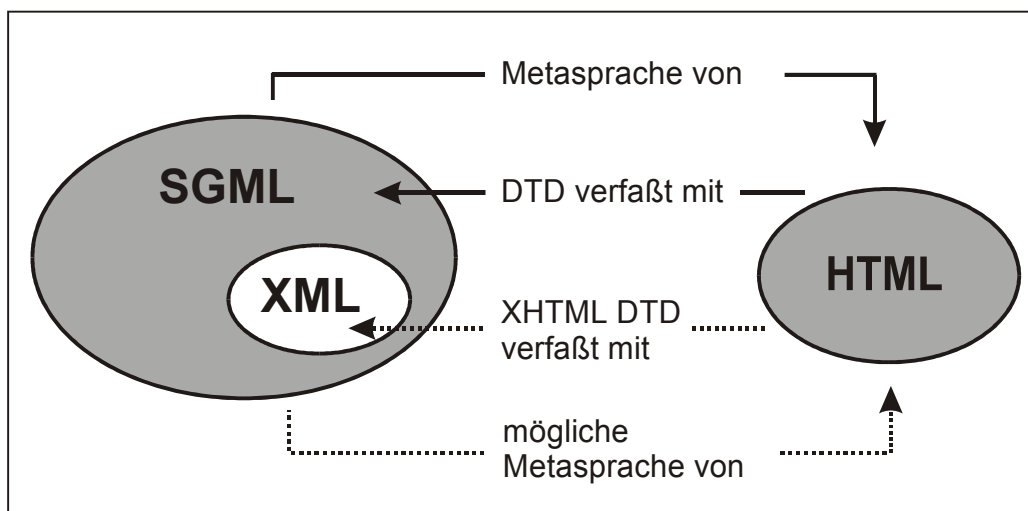


Abb. 1: SGML, XML und HTML¹⁷

Bestehende HTML-Dokumente können durch Hinzufügen der für XML notwendigen Deklarationen in XML-Dokumente umgewandelt werden. Allerdings müssen die HTML-Dokumente den Markup-Konventionen von XML genügen: Während in HTML Endtags zum Teil weggelassen werden können und zwischen Groß- und Kleinbuchstaben nicht unterschieden wird, muß es nach der Konvertierung zu jedem Start-Tag ein schließendes Endtag geben und beide Tag-Teile dürfen sich in Groß-/Kleinschreibung nicht unterscheiden.¹⁸

14 Vgl. John, Volker: XML - Weltsprache für das Internet, in: Objekt Spektrum, 5/99, S. 525. Siehe dazu auch Kap. 3.1.

15 Vgl. Harbarth, Jürgen: Einen Schritt weiter, in: N&C, 1/1999, S. 80.

16 Vgl. Bathe, Heike; Beer, Frank Michael: Warum kompliziert, wenn's auch einfach geht?, in: LANline 11/99, S. 88.

17 Vgl. Behme, Henning; Mintert, Stefan: XML in der Praxis, Bonn: Addison Wesley Longman Verlag 1998, S. 17.

18 Vgl. Behme, Henning; Mintert, Stefan: XML in der Praxis, a. a. O., S. 308 f.

2 Die Entwicklung von XML

2.1 Das World Wide Web Consortium (W3C)

XML ist eine Entwicklung des World Wide Web Consortiums (W3C), einem internationalen Gremium, dem vor allem IT-Hersteller und -Dienstleister sowie zahlreiche Forschungsorganisationen angehören.¹⁹ Das W3C wurde im Oktober 1994 am Europäischen Zentrum für Nuklearforschung (CERN) bei Genf gegründet. Seine Zielsetzung besteht unter anderem darin, Entwicklern und Anwendern eine Sammlung von Informationen über das WWW zur Verfügung zu stellen und die Entwicklung wie Etablierung von neuen Web-Technologien zu unterstützen. Derzeit ist das W3C in drei internationalen Forschungseinrichtungen untergebracht, am Laboratory for Computer Science des Massachusetts Institute of Technology (MIT/LCS) in den Vereinigten Staaten, am Institut National de Recherche en Informatique et en Automatique (INRIA) in Frankreich und an der Keio University in Japan. Geleitet wird das W3C von Tim Berners-Lee (Direktor) und Jean-François Abramatic (Vorsitzender). Die Mitgliedschaft im W3C steht jeder Organisation offen, beinahe alle wichtigen Computerhersteller, Software-Konzerne und IT-Dienstleister zählen heute zu seinen Mitgliedern. Dadurch wird das W3C zu einem neutralen Forum, in dem die Mitglieder gemeinsam und gleichermaßen stimmberechtigt an Web-Technologien arbeiten, die für ihre Organisation von kommerziellem Interesse sind.²⁰ Derzeit sind dem W3C 368 Organisationen angeschlossen, darunter Apple, AT&T, Compaq, Hewlett Packard, IBM, Intel, Microsoft, Motorola, Oracle und SAP.²¹ Obwohl das W3C Sprachen für das WWW entwickelt und veröffentlicht, die als universelle Standards gelten, ist es nicht gleichbedeutend mit Organisationen wie ISO oder ANSI. Die unterschiedliche Arbeitsweise wird zum Beispiel beim Vergleich von SGML und XML deutlich. Der ISO-Standard SGML (ISO-8879) ist eine sehr mächtige, komplizierte und detailliert beschriebene Standardsprache. Die Entwicklung von SGML begann bereits in den frühen sechziger Jahren bei IBM, bevor 26 Jahre später der Standard ISO-8879 veröffentlicht wurde.²² Die vom W3C verfolgte Entwicklungsstrategie unterscheidet sich hiervon in einigen Punkten grundlegend. So werden die vom W3C entwickelten Spezifikationen nicht als Standard, sondern als Empfehlung (Recommendation) bezeichnet.²³ Bei den veröffentlichten Empfehlungen, das zeigen

19 Vgl. o. V.: World Wide Web Consortium (W3C) Members, Online im Internet: <http://www.w3.org/Consortium/Member/List.html>, 11.07.1999.

20 Vgl. Berners-Lee, Tim: About the World Wide Web Consortium, Online im Internet: <http://www.w3.org/Consortium/>, 09.08.1997.

21 Vgl. o. V.: World Wide Web Consortium (W3C) Members, a. a. O.

22 Vgl. o. V.: The SGML Guide, Online im Internet: <http://www.oasis-open.org/cover/ileafgd.html>, 21.07.1999.

23 Das W3C unterscheidet Notes, Working Drafts (in development/last call), Candidate Recommendations, Proposed Recommendations und Recommendations. Die Reihenfolge der Darstellung beschreibt auch den Stand der Entwicklung der Spezifikation. Während eine Note lediglich eine Idee wiedergibt, die noch nicht einmal vom W3C verfolgt werden muß, stellt eine Candidate Recommendation bereits ein Dokument dar, daß von den Mitgliedern der entsprechenden W3C-Working-Group erarbeitet wurde. Allerdings besitzt erst eine Recommendation die Billigung des Vorsitzenden des W3C und damit den Charakter einer offiziellen Empfehlung. Näheres zu der Arbeit des W3C findet

auch einige der 10 Entwicklungsziele von XML²⁴, wird Wert auf einen minimalen zeitlichen Entwicklungsaufwand, eine im Umfang möglichst kurz gehaltene Beschreibung der gesamten Syntax und die einfache Handhabung der vorgeschlagenen sprachlichen Möglichkeiten gelegt.

2.2 Zeittafel der Entwicklung von XML

Die Entwicklung von XML begann im Juni 1996,²⁵ initiiert von einer Arbeitsgruppe des W3C, dem sogenannten SGML Editorial Review Board (ERB), das später in XML Work Group (WG) umbenannt wurde. Zu den Mitgliedern dieser Arbeitsgruppe zählen unter anderem der Vorsitzende der Working Group Jon Bosak von Sun Microsystems, Tim Bray von Textuality und Netscape, Jean Paoli von Microsoft, C. M. Sperberg-McQueen von der University of Illinois und Dan Connolly vom W3C.²⁶ Das W3C unterscheidet zeitlich drei Entwicklungsphasen von XML:²⁷

- Juni 1996: Gründung der XML Working Group (WG) im W3C
- Phase 1 (ab Juni 1996): Entwurf und Spezifikation von XML (Kern-XML)
- Phase 2 (ab März 1998): Gründung weiterer Arbeitsgruppen, die parallel zur XML-Entwicklung an peripheren Komponenten von XML arbeiten. Dazu zählen Stylesheets, Verweistechiken und Namensräume (Methoden zur Verwendung standardisierter Element- und Attributnamen). Erste XML-Anwendungen, wie die Mathematical Markup Language (MathML), die Chemical Markup Language (CML) und der spezielle CML-Browser JUMBO, werden entwickelt und der Entwurf einer XML-Anwendung von HTML 4.0 unter der Bezeichnung eXtensible Hypertext Markup Language (XHTML) wird veröffentlicht.²⁸
- Phase 3 (ab September 1999): Geplante Fortsetzung nicht abgeschlossener Aktivitäten, insbesondere der Entwicklung einer Abfragesprache für XML-Daten (XML-QL), standardisierter Verweistechiken (XLink und XPointer) und erweiterter Beschreibungsmöglichkeiten für validierbare Dokumenttypen (XML Schemas).

Abb. 2 zeigt die Entwicklung von XML mit einigen wichtigen Meilensteinen.

man bei o. V.: World Wide Web Consortium Process Document, Online im Internet: <http://www.w3.org/Consortium/Process/Process-19991111/process.pdf>, 01.08.2000.

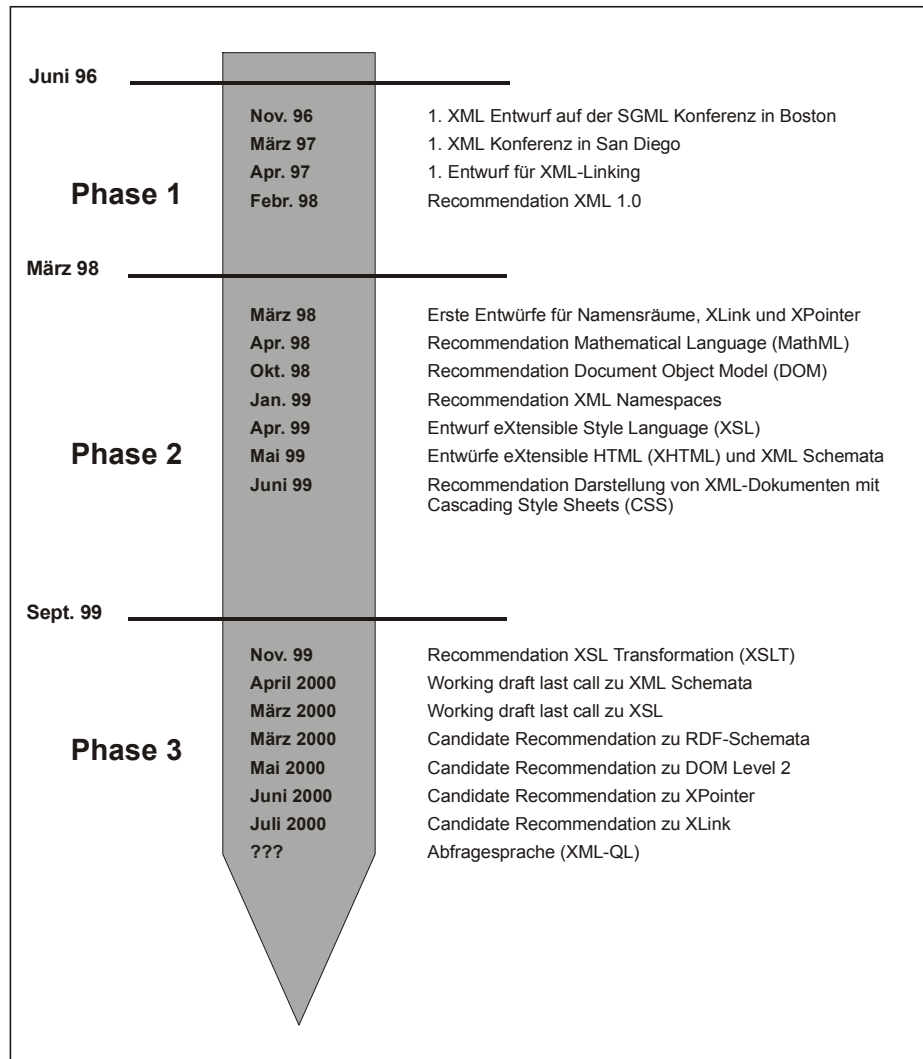
24 Vgl. Bray, Tim; Paoli, Jean; Sperberg-McQueen, C.M.: Extensible Markup Language (XML) W3C Recommendation, a. a. O.

25 Vgl. Connolly, Dan; Bray, Tim: Extensible Markup Language (XML) Activity, Online im Internet: <http://www.w3.org/XML/Activity.html>, 27.10.1999.

26 Vgl. Sperberg-McQueen, C. M.: Reports From the W3C SGML ERB to the SGML WG And from the W3C XML ERB to the XML SIG, Online im Internet: <http://www.w3.org/XML/9712-reports.html>, 04.12.1997 und Bray, Tim; Paoli, Jean; Sperberg-McQueen, C.M.: Extensible Markup Language (XML) W3C Recommendation, a. a. O.

27 Vgl. Connolly, Dan; Bray, Tim: Extensible Markup Language (XML) Activity, a. a. O.

28 Vgl. Connolly, Dan: Extensible Markup Language (XML), Online im Internet: <http://www.w3.org/XML/>, 10.11.1999.

Abb. 2: XML Zeittafel²⁹

3 Die Funktionsweise von XML

3.1 Allgemeine Funktionsweise

XML ist eine textbasierte Sprache zur Beschreibung von Daten. XML Dokumente (Instanzen) bestehen aus Auszeichnungselementen (Markups/Tags) und dem eigentlichen Inhalt (Nutzdaten). Wie bei SGML und HTML geschieht die semantisch relevante Markierung des Inhaltes durch die Umklammerung der Zeichen, Wörter, Ausdrücke oder Textabschnitte durch jeweils einen Start- und einem Endtag. Abb. 3 verdeutlicht dies an einem Beispiel:

²⁹ Vgl. Berners-Lee, Tim: W3C Technical Reports and Publications, Online im Internet: <http://www.w3.org/TR/>, 01.08.2000; Connolly, Dan; Bray, Tim: Extensible Markup Language (XML) Activity, a. a. O.; Quint, Vincent: W3C User Interface Domain, Online im Internet: <http://www.w3.org/UI/>, 18.08.1999 und Connolly, Dan: Extensible Markup Language (XML), a. a. O.

```
<Kapitel>3.3 Die Funktionsweise von XML</Kapitel>
```

Abb. 3: XML-Daten

Die im Beispiel beschriebene Information wird durch den Starttag `<Kapitel>` eingeleitet und durch den Endtag `</Kapitel>` abgeschlossen. Start- und Endtag gemeinsam bilden das Element „Kapitel“. Der Elementname muß in beiden Tags identisch sein, der Endtag wird dadurch gekennzeichnet, daß dem Elementnamen ein Schrägstrich vorangestellt wird. Üblicherweise bestehen XML-Elemente immer aus Start- und Endtag, die Ausnahme bilden sog. leere Tags, das heißt solche, die keine Daten beschreiben. Der Zeilenumbruch-Tag `
` in HTML ist ein Beispiel für einen leeren Tag. Im Gegensatz zu HTML ist die alleinige Verwendung des Starttags (ohne Endtag) in XML nicht zulässig. Für leere Elemente gelten in XML alternativ die beiden Notationen aus Abbildung 4.

```
<Elementname />
<Elementname></Elementname>
```

Abb. 4: Leere Elemente in XML

Neben der Bezeichnung können Elemente auch Eigenschaften, sogenannte Attribute enthalten, denen im Starttag Attributwerte zugewiesen werden können. Ein Element „Kapitel“ kann zum Beispiel die Eigenschaften Kapitelnummer (KNr) und Gliederungsstufe (GSt) besitzen. So könnte eine Kapitelüberschrift, wie in Abb. 5 abgebildet, ausgezeichnet werden.

```
<Kapitel KNr="3.3" GSt="2">Dokumenttyp-Definition</Kapitel>
```

Abb. 5: XML-Element mit Attributwerten

Die in den XML-Dokumenten verwendeten Auszeichnungselemente („Kapitel“, „Elementname“ etc.) werden mit der eXtensible Markup Language definiert und in einer DTD festgeschrieben.³⁰ Aufgrund der Gestaltungsfreiheit des Autors hinsichtlich der von ihm verwendeten Auszeichnungselemente stellt XML somit in doppelter Hinsicht eine Erweiterung zu HTML dar:

1. Der Autor ist nicht auf eine feste Anzahl vordefinierter Elemente beschränkt, sondern kann nach eigenem Ermessen Auszeichnungselemente definieren.
2. Durch die Möglichkeit der freien Wahl von Elementnamen und Attributen entstehen semantische Tags, die Metainformationen über die ausgezeichneten Daten enthalten. Das wird an dem in Abb. 5 dargestellten Quellcode-Beispiel deutlich: Die ausgezeichnete Textinformation wird durch den Elementnamen „Kapitel“ und die Attribute „KNr“ und „GSt“ erklärt. Die aus Textinformation, Elementname und Attributen zusammengesetzte Information läßt sich folgendermaßen interpretieren: „Dokumenttyp-Definition“ ist eine Kapitelüberschrift, sie trägt die Kapitelnummer „3.3“ und gehört zur Gliederungsstufe „2“.

³⁰ Vgl. Kapitel 1 „Das Konzept von XML“.

Als Grundkonzept der eXtensible Markup Language bezeichnen Weitzel et al.³¹ die von SGML abgeleitete, systematische Trennung von Inhalt, Struktur und Präsentation der ausgezeichneten Daten (in Abb. 6 graphisch dargestellt).

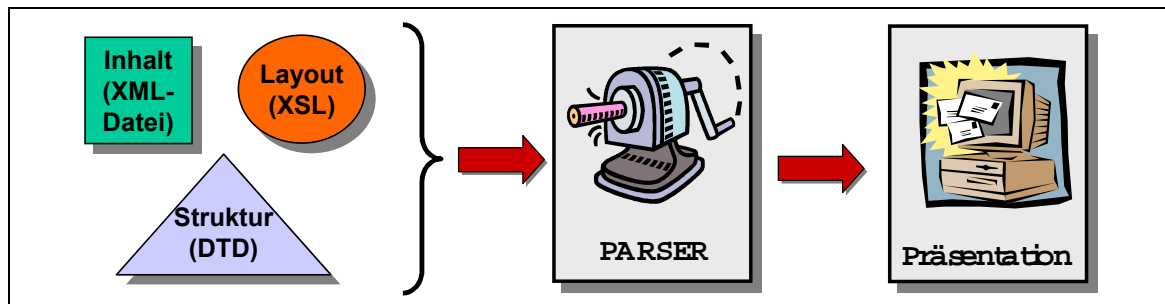


Abb. 6: Das Grundkonzept von XML

Der im XML-Umfeld häufig verwendete Begriff „Inhalt/Content“ umfaßt alle Nutzdaten, die in einem XML-Dokument beschrieben werden. Dabei handelt es sich überwiegend um Zeichendaten, aber auch um Tabellen, eingebundene Bilder u. a. Neben den Nutzdaten können XML-Dokumente auch Anweisungen zur Weiterverarbeitung der Daten, sogenannte processing instructions (PI), enthalten. Die übrigen Bestandteile eines XML-Dokuments, also Tags, Deklarationen und Kommentare, können unter dem Begriff Markup zusammengefaßt werden.

In der XML-Spezifikation wird zwischen der logischen und der physischen Struktur von XML-Dokumenten unterschieden. Die logische Struktur eines XML-Dokuments spiegelt die Zusammenhänge zwischen den ausgezeichneten Daten wider (Datenstruktur). Die logische Struktur von Daten kann entweder durch die Verwendung semantischer Elementnamen und Attribute oder durch hierarchisches Verschachteln von Elementen ausgedrückt werden.³²

Abb. 7 zeigt, wie die grobe Struktur des vorliegenden Arbeitspapiers anhand von semantischen XML-Tags, das heißt durch Elementnamen und Attributwerte, beschrieben werden kann. In diesem Beispiel werden die Attribute „Kapitelnummer“ (KNr) und „Gliederungsstufe“ (GSt) aus Abb. 5 verwendet. Ein Nachteil dieser Art der Strukturbeschreibung ist, daß sie auf die Assoziation des Lesers angewiesen ist, der weiß, daß ein Arbeitspapier in der Regel aus Kapiteln unterschiedlicher Gliederungsstufe besteht, die durch eine Kapitelüberschrift eingeleitet werden und Text oder weitere Unterkapitel enthalten können. Mit anderen Worten: Durch semantische Tags kann die Struktur eines XML-Dokuments zum Teil zwar für Menschen verständlich ausgedrückt werden, eine maschinenverständliche Beschreibung von Über- und Unterordnungsverhältnissen einzelner Elemente ist allerdings nicht automatisch erreicht. So ist für eine Applikation nicht automatisch erkennbar, ob die im Abb. 7 verwendete Gliederungsstufe 1 der Gliederungsstufe 2 über- oder untergeordnet ist. Weiterhin können auf diese Weise nur

31 Vgl. Weitzel, Tim; Buxmann, Peter; Ladner, Frank; König, Wolfgang: Konzept und Anwendung der Extensible Markup Language, a. a. O.

32 Vgl. Bray, Tim; Paoli, Jean; Sperberg-McQueen, C.M.: Extensible Markup Language (XML) W3C Recommendation, a. a. O.

strukturelle Zusammenhänge abgebildet werden, die relativ einfachen Mustern folgen. Die Beschreibung von multiplen Beziehungen zwischen Daten, wie sie in relationalen Datenbanken abgebildet werden können, ist beispielsweise in XML z. Zt. noch nicht möglich.

```

<Arbeitspapier>
<Überschrift KNr="1" GSt="1">Vorbemerkung zu ...</Überschrift>
<Text KNr="1">"Die Hypertext Markup Language (HTML) mit ...</Text>
<Überschrift KNr="3" GSt="1">Die Funktionsweise von XML</Überschrift>
<Überschrift KNr="3.1" GSt="2">Allgemeine Funktionsweise</Überschrift>
<Text KNr="3.1">XML ist eine textbasierte Sprache zur...</Text>
</Arbeitspapier>

```

Abb. 7: Auszeichnung des Arbeitspapiers mit XML

Als Beispiel für eine etwas komplexere Dokumentstruktur dient in Abb. 8 ein in XML ausgezeichnete elektronischer Rechnungsbeleg. Durch die Verschachtelung von Auszeichnungselementen können hierarchische Zusammenhänge beliebiger Komplexität abgebildet werden, die nicht nur für Menschen, sondern auch für Programme verständlich sind, mit denen die Daten weiterverarbeitet werden sollen.

XML erlaubt es, Daten, die in separat gespeicherten XML-Dokumenten abgelegt sind, vollständig oder ausschnittsweise in weitere XML-Dokument einzubinden. Dies ermöglicht eine zeitgleiche Bearbeitung von Dokument-Teilen durch verschiedene Autoren. Auch Bilder, Klang- oder Videodateien lassen sich auf diese Weise als externe Entities in ein XML-Dokument einbinden. Dadurch können in einem Netzwerk verteilte Daten zu XML-Dokumenten zusammengefügt werden, indem im Dokument auf sie verwiesen wird.³³ Diese Möglichkeit gilt unter der Prämisse, daß alle in einem Entity geöffneten Tags auch im selben Entity wieder geschlossen werden, also daß zu jedem Element sowohl ein Start- als auch ein Endtag vorhanden ist.³⁴

In den Abbildungen 7 und 8 wurden Beispiele für das strukturierte Auszeichnen unterschiedlicher Daten mit XML gegeben. Durch semantische Tags und die hierarchische Verschachtelung von Elementen wurde, neben den Daten selbst, auch die (logische) Struktur der Daten abgebildet. Die verwendeten Tags enthalten, im Gegensatz zu HTML-Tags, keine darstellungsbezogenen Informationen, sie beschreiben inhaltsorientiert (generic markup).

Vorschriften zur Präsentation (Darstellung auf dem Bildschirm oder beim Ausdrucken) der enthaltenen Nutzdaten sind optionaler Bestandteil von XML-Dokumenten. Kapitel 4.2 gibt einen Einblick in die existierenden Stylesprachen.

33 Mit HTML ist eine ausschnittsweise Einbindung externer, HTML-codierter textlicher Inhalte in eine HTML-Seite nicht möglich. Es besteht lediglich die Möglichkeit, auf externe HTML-Seiten zu verweisen (Link). (Externe) multimediale Inhalte können jedoch in HTML-Seiten integriert werden.

34 Vgl. Weitzel, Tim; Buxmann, Peter; Ladner, Frank; König, Wolfgang: Konzept und Anwendung der Extensible Markup Language, a. a. O. und Bray, Tim; Paoli, Jean; Sperberg-McQueen, C.M.: Extensible Markup Language (XML) W3C Recommendation, a. a. O.

```

<Rechnung>
<Rechnungskopf>
  <Rechnungsdatum>19.11.1999</Rechnungsdatum>
  <Rechnungsnummer>FAKT_191199</Rechnungsnummer>
  <Vertragsart>Dienstvertrag Zeit/Material</Vertragsart>
  <Kundennummer>12005667-01</Kundennummer>
  <Bestellnummer>981214-238-A</Bestellnummer>
  <Ust-ID>12XX00-ABC1</Ust-ID>
  <Rechnungsanschrift>
    <Abteilung>Zentrale Einkaufsabteilung</Abteilung>
    <Firma>Beispiel GmbH Deutschland</Firma>
    <Straße>Beispielstraße 12</Straße>
    <PLZ>D-55022</PLZ>
    <Ort>Mainz</Ort>
  </Rechnungsanschrift>
</Rechnungskopf>
<Rechnungspositionen>
  <Position Bezeichnung="Projektarbeit">
    <Leistung>Projektstätigkeit vor Ort</Leistung>
    <Menge Einheit="Manntage">19,75</Menge>
    <Einzelpreis Währung="DEM">2150,00</Einzelpreis>
    <Endwert Währung="DEM">42462,50</Endwert>
  </Position>
  <Position Bezeichnung ="Auftragsfahrt">
    <Leistung>Auftragsfahrt</Leistung>
    <Menge Einheit="Manntage">3,25</Menge>
    <Einzelpreis Währung="DEM">375,00</Einzelpreis>
    <Endwert Währung="DEM">1218,75</Endwert>
  </Position>
  <Summe_Positionen>
    <Preis Währung="DEM">43681,25</Preis>
    <MWSt Satz="16%" Währung="DEM">6989,00</MWSt>
    <Endbetrag Währung="DEM">50670,25</Endbetrag>
  </Summe_Positionen>
</Rechnungspositionen>
<Zahlungsbedingung>ohne Abzug sofort zahlbar</Zahlungsbedingung>
</Rechnung>

```

Abb. 8: Elektronische Rechnung in XML

Neben Inhalt und Präsentation sind auch die Struktureigenschaften vom eigentlichen XML-Dokument separiert. Diese formale Beschreibung des Dokumenttypen enthält alle im Sprachumfang der spezifischen Auszeichnungssprache verfügbaren Elemente, Attribute und Attributwerte sowie Vorgaben zu deren Struktur. Sie wird in der Dokumenttyp-Definition (DTD) festgelegt, die daher auch als formale Grammatik XML-basierter Sprachen (XML-Anwendungen) bezeichnet wird.³⁵ Eine solche Grammatik ist vor allem beim Austausch von XML-Dokumenten von Bedeutung, da sie es ermöglicht, die strukturelle und inhaltliche Validität der Dokumente zu prüfen. Dokumenttyp-Definitionen werden in Kapitel 3.3 behandelt.

35 Vgl. Macherius, Ingo: Revolution der Experten, Online im Internet: <http://www.heise.de/ix/artikel/1997/06/106/artikel.html>, 14.05.1997.

Die konzeptionelle Trennung von Inhalt, Struktur und Darstellung durch die Verwendung von DTDs und Stylesheets impliziert nicht, daß diese zwangsläufig als separate Dateien vorliegen müssen. Sie können auch im XML-Dokument selbst enthalten sein. Eine physische Trennung von XML-Daten, DTD und Stylesheet erscheint jedoch insbesondere dann sinnvoll, wenn sich mehr als nur ein XML-Dokument auf eine bestimmte DTD oder ein bestimmtes Stylesheet bezieht. Durch die Auslagerung von DTD und Stylesheet, die allen Dokumenten zentral zur Verfügung gestellt werden, wird die Größe der einzelnen Dokumente reduziert. Dies spielt vor allem bei der Datenübertragung eine wichtige Rolle, wo nach einmaliger Übertragung von DTD und Stylesheet zum Empfänger nur noch die reinen XML-Daten übertragen werden müssen und somit der Übertragungsverkehr verringert wird.

3.2 Aufbau eines XML-Dokuments

XML-Dokumente (Instanzen) tragen i. d. R. die Dateierdung „.xml“. Ein XML-Dokument besteht typischerweise aus drei Teilen: dem Prolog, den Zeichendaten (Character Data) und Auszeichnungselementen (Markup).³⁶

Ein XML-Dokument beginnt mit dem sog. Prolog. Der Prolog enthält die XML Declaration, in der angezeigt wird, zu welcher XML-Version das Dokument konform ist. Der Prolog kann im Anschluß an die Definition des XML-Standards die Document Type Declaration (nicht zu verwechseln mit der Document Type *Definition*) enthalten, die angibt, ob sich das Dokument auf eine externe (separat gespeicherte) DTD bezieht bzw. ob eine DTD im XML-Dokument (interne DTD) enthalten ist. Bei einer externen DTD enthält sie einen Verweis auf die separat gespeicherte DTD-Datei. Dokumente, die sich nicht auf eine DTD beziehen, benötigen keine Dokumenttyp-Deklaration.³⁷ Abb. 9 zeigt den Prolog zum XML-Dokument „Rechnung“ aus Abb. 8, das sich auf eine externe DTD mit der Bezeichnung „rechnung.dtd“ bezieht. Die entsprechende DTD für das Rechnungsdokument zeigt Abb. 15 in Kapitel 3.3.

```
<? xml version="1.0" standalone="no" ?>  
<! DOCTYPE Rechnung SYSTEM "rechnung.dtd">
```

Abb. 9: Prolog mit XML Declaration und Document Type Declaration

Im Prolog kann durch eine spezielle processing instruction (PI) auch der Bezug des XML-Dokuments zu einem Stylesheet hergestellt werden.³⁸ Dem Prolog folgt direkt das Hauptdokument, in dem die eigentlichen Daten, wie in den vorherigen Ausführungen dieses Kapitels erläutert, durch Auszeichnungselemente beschrieben werden.

36 Vgl. Bray, Tim; Paoli, Jean; Sperberg-McQueen, C.M.: Extensible Markup Language (XML) W3C Recommendation, a. a. O.

37 Vgl. Bray, Tim; Paoli, Jean; Sperberg-McQueen, C.M.: Extensible Markup Language (XML) W3C Recommendation, a. a. O.

38 Vgl. dazu Abb. 17 „Stylesheet-Anweisung im XML-Prolog“ sowie Kapitel 4.2 „Stylesprachen“.

XML-Dokumente können vier unterschiedliche Datentypen enthalten: Analysierbare Daten, nicht analysierbare Daten, Verarbeitungsanweisungen und Kommentare. Alle für die Verarbeitung durch einen XML-Parser vorgesehenen Zeichendaten und Auszeichnungselemente eines XML-Dokuments werden als analysierbare Daten (parsed character data oder PCDATA) bezeichnet. Als analysierbar gelten alle Zeichen des ISO-10646-Zeichensatzes, der aus etwa 33.000 Zeichen besteht und neben den arabischen Ziffern und den Buchstaben der lateinischen Sprache auch z. B. griechische, kyrillische, hebräische und arabische Zeichen umfaßt.³⁹ Dabei interpretiert der XML-Parser Zeichen, die entsprechend der XML-Syntax in spitzen Klammern stehen, als Tags (Markup), die zwischen Start- und Endtag eingeschlossenen Zeichen als Nutzdaten (Content/PCDATA). Abb. 10 zeigt einen PCDATA-Abschnitt.

```
<Tag>Content</Tag>
```

Abb. 10: Analysierbare Daten in XML (PCDATA)

Daten, die von einem XML-Parser nicht interpretiert werden sollen oder können, werden als nicht analysierbare Daten (character data oder CDATA) ausgezeichnet. Daten innerhalb eines CDATA-Abschnitts können, mit Ausnahme der als Abschluß verwendeten Folge von Klammern „]]>“, beliebige Zeichen und Zeichenfolgen verwenden und müssen nicht konform zur XML-Syntax sein. Nicht analysierbare Daten werden wie in Abb. 11 dargestellt ausgezeichnet.

```
<![CDATA[nicht analysierbare Zeichenfolge]]>
```

Abb. 11: Nicht analysierbare Daten in XML

In XML-Dokumenten enthaltene Verarbeitungsanweisungen (processing instructions oder PI) werden zwar vom XML-Parser interpretiert, zählen jedoch nicht zu den analysierbaren Daten (PCDATA). Sie werden in der in Abb. 12 verwendeten Notation geschrieben und vom Parser als PI-Abschnitt erkannt. Beispiel für eine Verarbeitungsanweisung in einem XML-Dokument ist die in Abb. 9 gezeigte Deklaration der XML-Version.

```
<? Verarbeitungsanweisung ?>
```

Abb. 12: Verarbeitungsanweisung in XML

Kommentare werden in XML als eigenständiger Datentyp behandelt. Sie können, ebenso wie CDATA-Abschnitte, abgesehen von zwei aufeinander folgenden Trennstrichen, beliebige Zeichen und Zeichenfolgen enthalten und werden von einem XML-Parser nicht analysiert. Kommentare können überall im Dokument verwendet werden, also auch innerhalb des Prologs. Sie werden wie in Abb. 13 notiert.⁴⁰

39 Vgl. Johnson, Charles David: ISO-10646 Concept Dictionary, Online im Internet: http://cns-web.bu.edu/pub/djohnson/web_files/i18n/ISO-10646.html, 22.09.1997 und Everson, Michael: Universal Character Set Collections, Online im Internet: <http://www.indigo.ie/egt/standards/iso10646/ucs-collections.html>, 23.09.1999.

40 Vgl. Bray, Tim; Paoli, Jean; Sperberg-McQueen, C.M.: Extensible Markup Language (XML) W3C Recommendation, a. a. O.

```
<-- Kommentar -->
```

Abb. 13: Kommentar in XML

XML schreibt eine streng hierarchische Schachtelung der Elemente vor. Das heißt, daß jedes Element, das in einem anderen Element durch einen Starttag geöffnet wurde, im selben Element auch mit einem Endtag geschlossen werden muß. Die Ausnahme sind leere Elemente, die nur durch einen Tag ausgedrückt werden.⁴¹ Durch die hierarchische Ordnung der Elemente kann die logische Struktur eines XML-Dokuments in einem Baumdiagramm dargestellt werden. Abb. 14 zeigt die Struktur des Rechnungsdokuments aus Abb. 8 in einem Baumdiagramm.

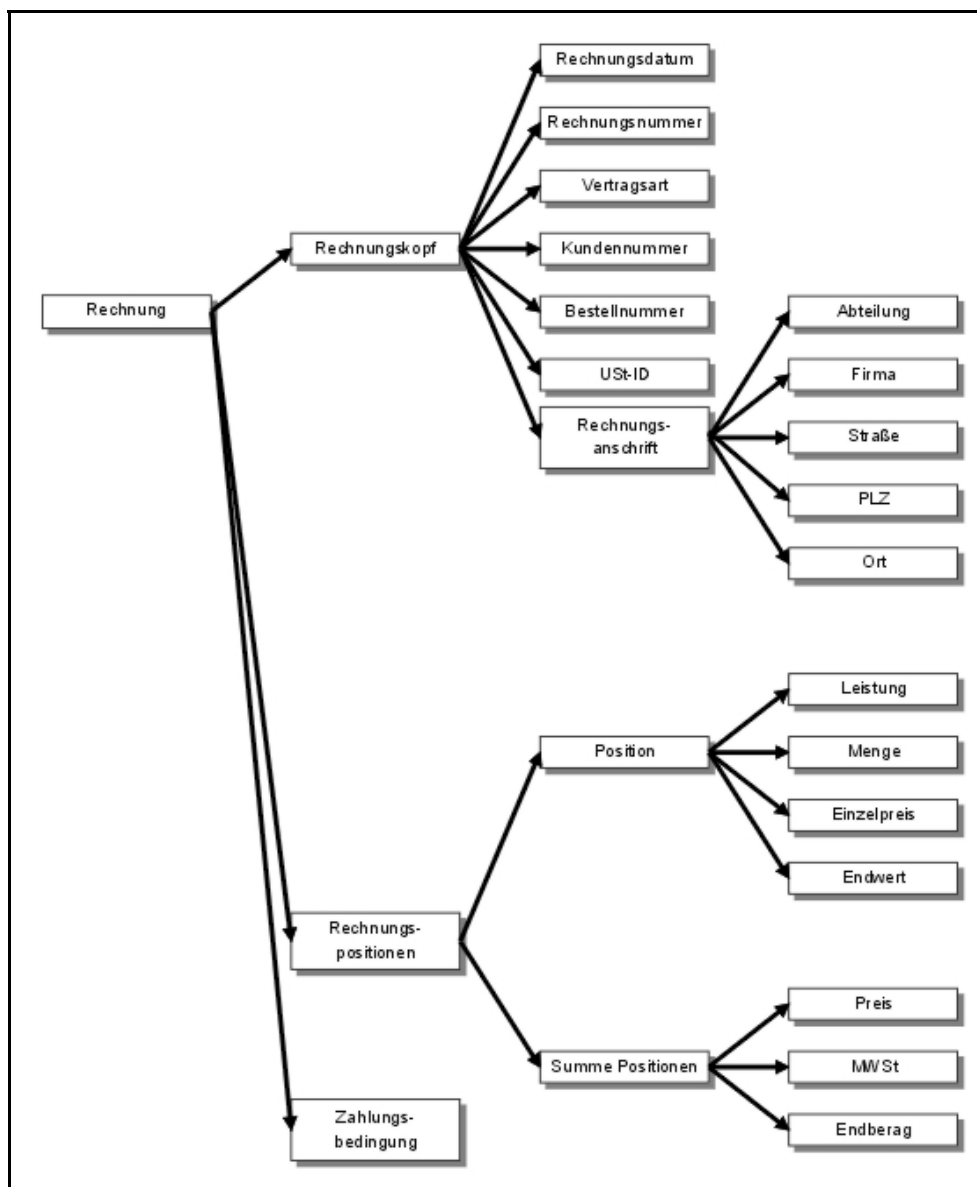


Abb. 14: Baumstruktur des elektronischen Rechnungsdokuments

41 Vgl. Abb. 4 „Leere Elemente in XML“.

Das oberste Element in einem Dokument wird als Wurzelement (root-element) bezeichnet.⁴² Das Wurzelement, im obigen Beispiel das Element Rechnung, schließt alle anderen Elemente ein. Die Darstellung der Daten und ihrer Zusammenhänge gleicht der in einem hierarchischen Datenbankmodell, in dem alle Daten in einer Baumstruktur abgebildet werden und die Beziehungen, die zwischen den Daten bestehen, durch Über- bzw. Unterordnungsverhältnisse beschrieben werden. Dabei kann jedem Element genau ein Element in der nächsthöheren Hierarchiestufe übergeordnet, und eine beliebige Zahl von Elementen untergeordnet sein. Als grundsätzliches Problem des hierarchischen Datenbankmodells wird die Ummöglichkeit der Darstellung von n:m-Beziehungen zwischen Elementen angesehen.⁴³ Die Darstellung von 1:n-Beziehungen im hierarchischen Datenbankmodell impliziert häufig die Wiederholung von Daten. Am Beispiel des Rechnungsdokuments aus Abb. 8 wird deutlich, daß alle Rechnungen, die an eine bestimmte Rechnungsanschrift gestellt werden, jeweils die Informationen Abteilung, Firma, Straße, PLZ und Ort der Rechnungsempfängers im Rechnungskopf enthalten müssen. Für den Versand der Rechnung an den Rechnungsempfänger oder für den Ausdruck eines Rechnungsbelegs kann es sinnvoll sein, daß alle Daten im Dokument enthalten sind, bei der Archivierung von Rechnungen kann es jedoch zu unnötiger Redundanz kommen, wenn viele Rechnungen die gleiche Rechnungsanschrift im Rechnungskopf tragen. XML bietet hier die Möglichkeit, das Element Rechnungsanschrift, mit allen enthaltenen Adreßinformationen eines bestimmten Rechnungsempfängers, als externes Entity,⁴⁴ also als separates XML-Dokument, zu speichern und im Rechnungskopf der jeweiligen Rechnungen darauf zu verweisen.

3.3 Dokumenttyp-Definition

Wie in Kapitel 3.1 erläutert, beziehen sich XML-Dokumente i. d. R. auf Dokumenttyp-Definitionen. In der DTD werden die zur Beschreibung der Daten definierten Elemente (Markups/Tags) und deren Attribute festgelegt und der formale Aufbau (Grammatik) der Dokument-Instanz beschrieben. Mintert bezeichnet den Dokumenttyp als Beschreibung einer Klasse von Dokumenten, die sich in ihrem strukturellen Aufbau gleichen.⁴⁵ Das gilt beispielsweise für wissenschaftliche Arbeiten, die nach bestimmten Richtlinien angefertigt werden müssen. In Tab. 1 sind Richtlinien des Lehrstuhls für Allg. BWL und Wirtschaftsinformatik der Johannes Gutenberg-Universität Mainz zum formalen Aufbau wissenschaftlicher Arbeiten abgebildet. In diesen Richtlinien werden die einzelnen Abschnitte genannt, aus denen sich wissenschaftliche Arbeiten zusammensetzen. Dabei sind einige der Abschnitte zwingend vorgeschrieben, andere, wie z. B. die Eidesstattliche Erklärung, werden beispielsweise nur in Diplomarbeiten verlangt. Die Dokumentstruktur wird durch die Reihenfolge der Elemente vorgegeben.

42 Vgl. Bray, Tim; Paoli, Jean; Sperberg-McQueen, C.M.: Extensible Markup Language (XML) W3C Recommendation, a. a. O.

43 Vgl. Stahlknecht, Peter; Hasenkamp, Ulrich: Einführung in die Wirtschaftsinformatik, 8., vollständig überarb. und erweitert. Aufl., Berlin et al.: Springer-Verlag 1997, S. 109 ff.

44 Vgl. Kapitel 3.1 „Allgemeine Funktionsweise“.

45 Vgl. Mintert, Stefan: Einführung in die Extensible Markup Language (XML), Online im Internet: <http://computerphilologie.uni-muenchen.de/jg99/xml.html>, 08.03.1999.

Nr.	Bezeichnung	Notwendigkeit
1.	Titelblatt	zwingend
2.	Inhaltsverzeichnis	zwingend
3.	Abbildungsverzeichnis	nur wenn mehr als eine (1) Abbildung vorhanden
4.	Tabellenverzeichnis	nur wenn mehr als eine (1) Tabelle vorhanden
5.	Abkürzungsverzeichnis	nur wenn nichtgeläufige Abkürzungen verwendet
6.	Textkorpus der Arbeit	zwingend
7.	Anhang	nur wenn benötigt; maximal 5 Seiten
8.	Literaturverzeichnis	zwingend
9.	Eidesstattliche Erklärung	nur bei Diplomarbeiten

Tab. 1: Richtlinien für die Gestaltung von wissenschaftlichen Arbeiten⁴⁶

Mit Hilfe einer Dokumenttyp-Definition (DTD) lassen sich ähnliche Richtlinien zur Erstellung von XML-Dokumenten formulieren. Dabei werden in der DTD im wesentlichen die nachfolgende Angaben über den Aufbau von Dokumenten gemacht:

- Spezifizierung der verwendeten Auszeichnungselemente: Alle Elemente, die zur Erstellung von Dokumenten eines bestimmten Dokumenttyps zulässig sind, werden in der DTD definiert. Außerdem kann jeweils festgelegt werden, ob es sich um ein normales oder ein leeres Element handelt,⁴⁷ ob Elemente weggelassen werden dürfen oder ob sie mehrfach enthalten sein können.
- Inhalt einzelner Auszeichnungselemente: Elemente können sowohl Daten (PCDATA, CDATA oder PI) als auch andere Elemente enthalten. In der DTD kann der Inhalt eines Elements vorgeschrieben werden.
- Spezifizierung zulässiger Attribute: Für jedes Element kann eine Liste möglicher Attribute aufgestellt werden.
- Festlegung der möglichen Attributwerte und deren Eigenschaften: In der DTD kann jedem Attribut eine Menge von möglichen Attributwerten sowie ein Standardwert (default) zugeordnet werden. Zusätzlich läßt sich der Datentyp des Attributwertes festlegen sowie, ob ein Attributwert beim Auszeichnen zwingend genannt werden muß (required), ob er gegebenenfalls aus dem Kontext der ausgezeichneten Daten hervorgehen soll (implied) oder ob er nur den Standardwert annehmen darf (fixed).
- Logische Dokumentstruktur: In der DTD wird vorgeschrieben, in welcher Reihenfolge die definierten Elemente beim Auszeichnen ausschließlich verwendet werden dürfen, das heißt, wie sie zu verschachteln sind.⁴⁸

46 Vgl. Schwickert, Axel C.: Richtlinien für die Gestaltung von Wissenschaftlichen Arbeiten, 6., unveränderte Aufl., Mainz: 1998, S. 9.

47 Vgl. Kapitel 3.1 „Allgemeine Funktionsweise“.

48 Vgl. Weitzel, Tim; Buxmann, Peter; Ladner, Frank; König, Wolfgang: Konzept und Anwendung der Extensible Markup Language, a. a. O.; Behme, Henning; Mintert, Stefan: XML in der Praxis, a. a. O., S. 61 und Bray, Tim; Paoli, Jean; Sperberg-McQueen, C.M.: Extensible Markup Language (XML) W3C Recommendation, a. a. O.

Die DTD dient also der Typisierung von XML-Dokumenten oder Teilen von XML-Dokumenten, die in ihrem Aufbau gewissen Grundmustern folgen. Für den Einsatz von DTDs sprechen hauptsächlich zwei Gründe:⁴⁹

1. Möglichkeit der Definition XML-basierter Auszeichnungssprachen: Erst durch eine DTD wird XML zur Metasprache, in der spezielle XML-basierte Sprachen zur Auszeichnung von Dokumenten eines bestimmten Dokumenttyps spezifiziert werden können. Autoren spezieller XML-Dokumente, wie zum Beispiel elektronischer Rechnungen, eMails, Webseiten o. ä., können Auszeichnungsregeln vorgeschrieben werden, in denen alle zur Datenbeschreibung zulässigen Elemente und Attribute sowie die Dokumentstruktur festgelegt sind. Die DTD wird daher oft als Grammatik XML-basierter Auszeichnungssprachen bezeichnet.⁵⁰ Ein Beispiel für eine konkrete, XML-basierte Auszeichnungssprache ist das bereits erwähnte XHTML, die XML-DTD für HTML 4.0.⁵¹ Dokumente, die sich auf eine bestimmte DTD beziehen, können von einem XML-Parser auf strukturelle und inhaltliche Validität geprüft werden. Dazu analysiert der Parser zuerst die DTD und prüft dann die syntaktische Richtigkeit des Dokuments.⁵²
2. Schematisierung verarbeitungsrelevanter Bestandteile von Dokumenten: Für den Entwickler von Programmen zur Verarbeitung von XML-Dokumenten enthält die DTD allgemeine Informationen darüber, durch welche Elemente die zu verarbeitenden Daten in einem Dokument beschrieben werden. So ist es anhand einer DTD möglich, Programme zu schreiben, mit denen alle Dokumente verarbeitet werden können, die sich auf diese DTD beziehen. Das gilt auch für Stylesheets, die Darstellungsanweisungen für bestimmte XML-Elemente beinhalten. Ein Stylesheet, in dem die Darstellung aller darzustellenden Elemente einer bestimmten DTD beschrieben wurde, kann zur Formatierung aller XML-Dokumente verwendet werden, die in Bezug auf die verwendete DTD gültig sind.

Die DTD kann entweder Bestandteil der Dokumenttyp-Deklaration und somit des XML-Dokuments sein, das sie beschreibt, oder als separate Datei, mit der Dateiendung „.dtd“, gespeichert werden. In diesem Fall enthält die Dokumenttyp-Deklaration des XML-Dokuments die Verweisadresse der DTD.⁵³ Eine gemischte Verwendung von externen und internen DTDs ist zulässig.⁵⁴ In Kapitel 3.2 wurde auf die Möglichkeit hingewiesen, XML-Dokumente in kleinere Speichereinheiten (Entities) zu untergliedern. Jedes Entity kann auf eine andere DTD verweisen, so daß ein Dokument, das aus mehreren Entities besteht, durch verschiedene DTDs beschrieben werden kann.

49 Vgl. Dünhölter, Kuno: Das Web automatisieren mit XML, Online im Internet: <http://members.aol.com/xmldoku/>, 01.09.1998.

50 Vgl. Weitzel, Tim; Buxmann, Peter; Ladner, Frank; König, Wolfgang: Konzept und Anwendung der Extensible Markup Language, a. a. O. und Macherius, Ingo: Revolution der Experten, a. a. O.

51 Vgl. Pemberton, Steven; Altheim, Murray; Austin, Daniel; Boumphrey, Frank; Burger, John; Donoho, Andrew W.; Dooley, Sam; Hofrichter, Klaus; Hoschka, Philipp: XHTML 1.0: The Extensible HyperText Markup Language, a. a. O.

52 Siehe dazu auch Kapitel 3.4 „Wohlgeformte und gültige XML-Dokumente“.

53 Vgl. Abb. 9 „Prolog in XML-Dokument“.

54 Vgl. John, Volker: XML - Weltsprache für das Internet, a. a. O., S. 24.

Abb. 15 zeigt eine DTD für das in Abb. 8 dargestellte Rechnungsdokument. Die abgebildete DTD enthält eine Reihe sogenannter Elementtyp- und Attributlisten-Deklarationen. Elementtyp-Deklarationen beginnen mit „<!ELEMENT“, gefolgt vom Namen des deklarierten Elements und einer in Klammern geschriebenen Vorschrift über die enthaltenen Daten. Im Beispiel sind dies entweder andere Elemente oder Daten (PCDATA, CDATA, PI). Durch verschiedene Operatoren können Regeln über die Verwendung von Elementen beim Auszeichnen präzisiert werden.

```

<!ELEMENT Rechnung (Rechnungskopf, Rechnungspositionen,
    Zahlungsbedingung)>
<!ELEMENT Rechnungskopf (Rechnungsdatum, Rechnungsnummer, Vertragsart,
    Kundennummer, Bestellnummer, USt-ID, Rechnungsanschrift)>
  <!ELEMENT Rechnungsdatum(#PCDATA)>
  <!ELEMENT Rechnungsnummer(#PCDATA)>
  <!ELEMENT Vertragsart(#PCDATA)>
  <!ELEMENT Kundennummer(#PCDATA)>
  <!ELEMENT Bestellnummer(#PCDATA)>
  <!ELEMENT USt-ID(#PCDATA)>
  <!ELEMENT Rechnungsanschrift (Abteilung?, Firma, Straße, PLZ, Ort)>
    <!ELEMENT Abteilung(#PCDATA)>
    <!ELEMENT Firma(#PCDATA)>
    <!ELEMENT Straße(#PCDATA)>
    <!ELEMENT PLZ(#PCDATA)>
    <!ELEMENT Ort(#PCDATA)>
<!ELEMENT Rechnungspositionen(Position+, Summe Positionen)>
  <!ELEMENT Position(Leistung, Menge, Einzelpreis, Endwert)>
    <!ATTLIST Position Bezeichnung ID #Required)>
    <!ELEMENT Leistung(#PCDATA)>
    <!ELEMENT Menge(#PCDATA)>
      <!ATTLIST Menge
        Einheit(Manntage|Mannstunden|Stück)Manntage #IMPLIED>
    <!ELEMENT Einzelpreis(#PCDATA)>
      <!ATTLIST Einzelpreis
        Währung (DEM|EUR)DEM #IMPLIED>
    <!ELEMENT Endwert(#PCDATA)>
  <!ELEMENT Summe_Positionen(#PCDATA)>
    <!ELEMENT Preis(#PCDATA)>
      <!ATTLIST Preis
        Währung (DEM|EUR)DEM #IMPLIED>
    <!ELEMENT MWSt(#PCDATA)>
      <!ATTLIST MWSt
        Satz #CDATA #REQUIRED
        Währung (DEM|EUR)DEM #IMPLIED>
    <!ELEMENT Endbetrag(#PCDATA)>
      <!ATTLIST Endbetrag
        Währung (DEM|EUR)DEM #IMPLIED>
<!ELEMENT Zahlungsbedingung(#PCDATA)>

```

Abb. 15: DTD für elektronisches Rechnungsdokument

Hierbei bedeutet „?““, daß ein Element nicht zwingend verwendet werden muß und höchstens einmal innerhalb eines übergeordneten Elements vorkommen darf. Durch „+“ wird festgelegt, daß ein Element ein- oder mehrmals hintereinander folgen kann. Ein mit „*“ endendes Element kann null-, ein- oder mehrmals verwendet werden. Werden zwei oder mehr Elemente durch das Pipe-Symbol „|“ getrennt, so kann wahlweise eines der Elemente verwendet werden. Geschlossen wird die Deklaration mit „>“.

Die Attributlisten-Deklaration beginnt mit „<!ATTLIST“ und enthält den Namen des Elements für das die Attribute deklariert werden und eine Liste von einem oder mehreren Attributen, die neben dem Attributnamen, den Typ, eventuelle Vorgabewerte sowie einen Standardwert des jeweiligen Attributs enthalten können. Die Attributlisten-Deklaration muß ebenfalls mit „>“ geschlossen werden. In Abb. 15 werden beispielsweise für das Element MWSt zwei Attribute deklariert: Das Attribut „Satz“ muß (#REQUIRED) beim Auszeichnen einen Wert in Form einer Zeichenkette (CDATA) annehmen, dieser kann aber beliebig gewählt werden. Das Attribut „Währung“ kann wahlweise den Attributwert „DEM“ oder „EUR“ annehmen. Standardmäßig erhält das Attribut den Wert „DEM“, so daß er beim Auszeichnen einer XML-Rechnung nicht explizit benannt werden muß. Mit anderen Worten ist der Vorgabewert „DEM“ implizit im Attribut „Währung“ enthalten (#IMPLIED). Die syntaktischen Regeln zur Erstellung von DTDs sind Bestandteil der XML-Spezifikation.

Derzeit wird von der XML Schema Working Group des W3C ein weiteres Modul zur Formulierung von Auszeichnungsvorschriften entwickelt. Mit Hilfe sogenannter XML-Schemas sollen die Möglichkeiten zur Erstellung von Vorschriften über Struktur, Syntax und Daten einer Klasse von formal identischen XML-Dokumenten erweitert werden. Unter anderem werden folgende Ziele genannt:⁵⁵

- Bereitstellung einfacher, vordefinierter Datentypen, wie Boolean, String, Date, Binary.
- Möglichkeiten der Ableitung benutzerdefinierter Datentypen durch Modifikation der Eigenschaften vorgegebener Datentypen.
- Erweiterung der Einschränkungsmöglichkeiten für Elementinhalte und Attributwerte, wie minimale/maximale Anzahl von Zeichen, Wertober- und Untergrenze bei Zahlen und Datumswerten, Muster für mögliche Werte.

Durch die Möglichkeit, für XML-Elemente bestimmte inhaltliche Einschränkungen, wie zum Beispiel die maximale Anzahl von Zeichen zu formulieren oder benutzerdefinierte Datentypen festzulegen, wäre es beispielsweise möglich, Feldeigenschaften von Datenbanktabellen abzubilden und somit genauere Vorschriften über den Inhalt von XML-Elementen zu machen. Bestimmte Elementinhalte könnten von einem entsprechenden XML-Parser hinsichtlich ihrer Übereinstimmung mit einem vorgegebenen Datentyp oder der Einhaltung inhaltlicher Beschränkungen überprüft werden.

⁵⁵ Vgl. Malhotra, Ashok; Maloney, Murray: XML Schema Requirements, Online im Internet: <http://www.w3.org/TR/NOTE-xml-schema-req>, 15.02.1999; Thompson, Henry S.; Beech, David; Maloney, Murray; Mendelsohn, Noah: XML Schema Part 1: Structures, Online im Internet: <http://www.w3.org/TR/1999/WD-xmlschema-1-19991105/>, 05.11.1999 und Biron, Paul V.; Malhotra, Ashok: XML Schema Part 2: Datatypes, Online im Internet: <http://www.w3.org/TR/1999/WD-xmlschema-2-19991105/>, 05.11.1999.

3.4 Wohlgeformte und gültige XML-Dokumente

Im Gegensatz zu HTML verlangt XML eine strenge Einhaltung der Syntaxregeln. Grundsätzlich werden zwei Qualitätsarten bei XML-Dokumenten unterschieden: Wohlgeformte (well-formed) und gültige (valid) Dokumente. Als wohlgeformt gelten alle Dokumente, die den syntaktischen Regeln der XML-Spezifikation des W3C entsprechen:

- Zu jedem öffnenden Tag muß ein schließendes Gegenstück vorhanden sein, Ausnahmen sind die besonders zu kennzeichnenden leeren Tags.
- Öffnendes und schließendes Tag müssen die gleiche Groß- und Kleinschreibung benutzen.
- Tags müssen in der entgegengesetzten Reihenfolge der Öffnung wieder geschlossen werden; d. h., sie dürfen sich nicht überkreuzen. Abb. 16 verdeutlicht dies an einem Beispiel.

<p>Gültig: <Position><Leistung>Auftragsfahrt</Leistung></Position></p> <p>Ungültig: <Position><Leistung>Auftragsfahrt</Position></Leistung></p>

Abb. 16: Verschachtelung von Tags

Jedes XML-Dokument muß zumindest wohlgeformt sein. Gültige Dokumente müssen sich darüber hinaus auf eine DTD beziehen und mit deren formalen Regeln übereinstimmen. Das heißt, daß sie ausschließlich Elemente und Attribute enthalten dürfen, die in der DTD definiert werden und ihre logische Struktur mit der, durch die DTD vorgegebenen Struktur identisch sein muß. Die Validität eines Dokuments, das heißt, ob das Dokument zu den formalen Auszeichnungsvorschriften einer DTD konform ist, kann von einem XML-Parser geprüft werden.⁵⁶

4 Die Peripherie von XML

4.1 Funktionale Erweiterung von XML durch periphere Module

Im Umfeld von XML existieren zahlreiche periphere Module, die teilweise bereits vom W3C als Recommendation bezeichnet werden bzw. sich noch in der Entwicklung befinden. Diese Module enthalten Regeln zur funktionalen Erweiterung der Auszeichnungssprache XML (XML-Kern). Dazu gehören, neben den im vorherigen Kapitel bereits erläuterten Dokumenttyp-Definitionen: Stylesheets, Transformations- und Verarbeitungs-

⁵⁶ Vgl. Bray, Tim; Paoli, Jean; Sperberg-McQueen, C.M.: Extensible Markup Language (XML) W3C Recommendation, a. a. O.

regeln, Hyperlink-Techniken, Datenzugriffsmodelle u. v. a. Aufgrund der Vielzahl optional einsetzbarer Zusatzmodule bezeichnet Bos XML als „family of technologies“.⁵⁷ In Tab. 2 sind die wesentlichen Module aus dem Umfeld von XML aufgezählt, die im Folgenden erläutert werden.

Bezeichnung	Funktion
Schemas	Erweiterung und Verfeinerung der DTD (Inhalt & Datentypen)
XSL	Stylesheets für XML-Dokumente
CSS	Stylesheets für HTML- und XML-Dokumente
XSLT	Transformation von XML-Dokumenten
XLink	Hyperlink-Sprache für XML
XPointer	Direktverweise in XML-Dokumente
Namespaces	Identifikation bestimmter XML-Elemente mit einem URI
DOM	Datenzugriffsmodell für objektorientierte Programmierung
XML-QL	Abfrage-Sprache für XML-Daten

Tab. 2 : Periphere Module zu XML

Die Zusatzmodule werden/wurden alle separat voneinander entwickelt und spezifiziert, so daß XML-Autoren nur jeweils die Syntax der Module lernen müssen, die sie zur Erstellung ihrer jeweiligen XML-Dokumente oder -Anwendungen benötigen. Beispielsweise benötigt der Entwickler eines XML-basierten Datenaustauschformats, im Gegensatz zum Autor einer Website, keine Sprache zur Erstellung von Stylesheets oder Hyperlinks.

Aufgrund des sehr geringen Umfangs der XML-Spezifikation und dem daraus resultierenden, ebenfalls geringen Lernaufwand, eignet sich XML selbst für sehr einfache Aufgaben. Die funktionelle Erweiterbarkeit von XML, die durch den Einsatz peripherer Module erreicht wird, ermöglicht aber auch die Entwicklung komplexer Anwendungen. XML wird dadurch zu einer sehr mächtigen (Meta-) Sprache.⁵⁸

4.2 Stylesprachen

Wie bereits in Kapitel 3.1 erläutert, ist die Trennung von Inhalt, Struktur und Darstellung einer der zentralen Aspekte der eXtensible Markup Language. Da XML nicht auf eine feste Anzahl von vordefinierten Tags beschränkt ist, sondern die Verwendung einer beliebigen Anzahl frei definierbarer Auszeichnungselemente zuläßt, können Anweisungen zur Präsentation nicht, wie bei HTML, fest im Browser implementiert werden. An-

57 Bos, Bert: XML in 10 points, Online im Internet: <http://www.w3.org/XML/1999/XML-in-10-points>, 01.08.2000.

58 Vgl. Weitzel, Tim; Buxmann, Peter; Ladner, Frank; König, Wolfgang: Konzept und Anwendung der Extensible Markup Language, a. a. O.

weisungen zur spezifischen Darstellung von XML-Tags werden daher in sogenannten Stylesheets zusammengefaßt. Diese beinhalten hauptsächlich Regeln darüber, wie Daten, die mit einem bestimmten XML-Element ausgezeichnet sind, in verschiedenen Ausgabemedien dargestellt werden sollen. Stylesheets sind i. d. R. in eigenständigen Dateien untergebracht, es ist aber genauso gut möglich, die Stylesheet-Informationen direkt in das XML-Dokument einzubetten.

Die Verwendung von Stylesheets zur Darstellung von XML-Dokumenten bringt vier wesentliche Vorteile:

1. Beim Auszeichnen von Daten mit XML muß keine Rücksicht auf die spätere Form der Darstellung genommen werden. Die Tags zur Auszeichnung können, unabhängig von der späteren Präsentation, so gewählt werden, daß Bedeutung und Struktur der Daten deutlich werden und von Programmen interpretiert und verarbeitet werden können. Für das fertige XML-Dokument können nachträglich, in einem oder mehreren Stylesheets, Anweisungen für beliebige Darstellungsformen und Ausgabemedien getroffen werden.
2. Der Nutzer eines XML-Dokuments kann dessen Darstellungsform selbst bestimmen oder verändern, indem er ein eigenes Stylesheet erstellt oder Darstellungsanweisungen in einem vorhandenen Stylesheet entsprechend ändert.⁵⁹ So kann er beispielsweise bestimmte Inhalte eines Dokuments stärker hervorheben oder ausblenden.⁶⁰
3. Die Trennung von Inhalt und Präsentation führt bei Dokumenten, die nicht zur optischen Darstellung, sondern lediglich zur maschinellen Weiterverarbeitung bestimmt sind, zu schlankeren Dateien, da bei diesem Verwendungszweck Anweisungen zur Präsentation überflüssig sind. Stylesheets müssen beim Austausch von XML-Dokumenten nur dann übertragen werden, wenn die Dokumente dem Empfänger angezeigt werden sollen bzw. wenn dieser nicht bereits über ein entsprechendes Stylesheet zur Formatierung des Dokuments verfügt.

Der Bezug des XML-Dokuments zu einem Stylesheet wird durch eine processing instruction (PI) im Prolog des Dokuments hergestellt, welche die Verweisadresse und den Typ des Stylesheets enthält. Abb. 17 zeigt eine solche PI.

```
<? xml-stylesheet href="mystyle.css" type="text/css" ?>
```

Abb. 17: Stylesheet-Anweisung im XML-Prolog⁶¹

Im Umfeld von XML existieren unterschiedliche Konzepte zur Erstellung von Stylesheets; derzeit entwickelt das W3C eine eigene Stylesprache für XML-Daten. Diese Sprache besteht aus zwei Teilsprachen, der eXtensible Stylesheet Language (XSL), für

59 Vgl. Weitzel, Tim; Buxmann, Peter; Ladner, Frank; König, Wolfgang: Konzept und Anwendung der Extensible Markup Language, a. a. O.

60 Vgl. Winter, Dirk; Zitzmann, Georg; Draisbach, Marc: XML-Überblick, Online im Internet: <http://turing.informatik.fh-wiesbaden.de/projekt2/>, 28.04.1998.

61 Vgl. Clark, James: Associating Style Sheets with XML documents Version 1.0, Online im Internet: <http://www.w3.org/1999/06/REC-xml-stylesheet-19990629>, 29.06.1999.

die seit März 2000 der Working Draft in der letzten Entwicklungsphase ist („last call“),⁶² und XSL Transformations (XSLT), deren offizielle Spezifikation im November 1999 veröffentlicht wurde.⁶³ Das Konzept der XML-Stylesprache basiert zum Teil auf der Document Style Semantics and Specification Language (DSSSL), der Stylesprache für SGML, und zum Teil auf dem erweiterten Sprachumfang von Cascading Style Sheets (CSS2).⁶⁴

XSL wird der Teil der Stylesprache sein, in der Stilregeln für XML-Dokumente festgelegt werden. Dazu soll XSL einen erweiterbaren Katalog von Formatierungsobjekten (formatting-objects) und -eigenschaften (formatting-properties) bereitstellen, mit denen beschrieben werden kann, wie der Inhalt bestimmter Elemente eines Dokuments, bzw. wie das gesamte Dokument dargestellt werden soll. Über 90 v. H. der Formatierungsobjekte und -eigenschaften sollen aus der Syntax der Cascading Style Sheets Sprache (CSS2) übernommen werden. Für XSL sind dennoch reichere Darstellungsmöglichkeiten geplant, als sie momentan mit CSS2 zur Verfügung stehen. So soll XSL eine präzisere Kontrolle über das Seitenlayout bieten, wodurch es beispielsweise möglich wird, Seitenumbrüche, Zeilenabstände, Textrichtung und Laufweite in einem XML-Dokument zu steuern oder Text in mehreren Spalten darzustellen.⁶⁵

Ein weiteres, wesentliches Element der XML-Stylesprache ist XSL Transformations (XSLT), eine Sprache, in der Konstruktionsregeln für XML-Dokumente programmiert werden können. Mit Hilfe von XSLT läßt sich die Baumstruktur eines XML-Dokuments (source-tree)⁶⁶ in eine andere Baumstruktur (result-tree) transformieren, wenn für Präsentation oder Weiterverarbeitung des Dokuments eine andere Struktur geeignet erscheint.⁶⁷ So können beispielsweise entsprechend gekennzeichnete Informationen eines XML-Dokuments zur Darstellung alternativ in Textform belassen oder in Tabellenform umgewandelt werden. Ebenso kann eine Liste nach bestimmten Selektions- oder Sortierkriterien umgeformt werden, bevor sie dargestellt wird. Wie bei einer Textverarbeitungs-Software können für entsprechend gegliederte Dokumente Inhaltsverzeichnisse erstellt oder Seitenzahlen vergeben werden.⁶⁸

Die Formatierung von Dokumenten mit der XML-Stylesprache erfolgt, wie in Abb. 18 dargestellt, in zwei Schritten:

62 Vgl. Adler, Sharon; Berglund, Anders; Caruso, Jeff; Deach, Stephen; Grosso, Paul; Gutentag, Eduardo; Milowski, Alex; Parnell, Scott; Richman, Jeremy; Zilles, Steve: Extensible Stylesheet Language (XSL) Specification W3C Working Draft, Online im Internet: <http://www.w3.org/TR/2000/WD-xsl-20000327/xslspec.html>, 01.08.2000.

63 Vgl. Clark, James: XSL Transformations (XSLT) Version 1.0 W3C Recommendation, Online im Internet: <http://www.w3.org/TR/1999/REC-xslt-19991116>, 16.11.1999.

64 Vgl. Adler, Sharon; Berglund, Anders; Caruso, Jeff; Deach, Stephen; Grosso, Paul; Gutentag, Eduardo; Milowski, Alex; Parnell, Scott; Richman, Jeremy; Zilles, Steve: Extensible Stylesheet Language (XSL) Specification W3C Working Draft, a. a. O.

65 Vgl. Adler, Sharon; Berglund, Anders; Caruso, Jeff; Deach, Stephen; Grosso, Paul; Gutentag, Eduardo; Milowski, Alex; Parnell, Scott; Richman, Jeremy; Zilles, Steve: Extensible Stylesheet Language (XSL) Specification W3C Working Draft, a. a. O.

66 Vgl. Abb. 14 „Baumstruktur eines XML Dokuments“.

67 Vgl. Clark, James: XSL Transformations (XSLT) Version 1.0 W3C Recommendation, a. a. O.

68 Vgl. Dönhöler, Kuno: Das Web automatisieren mit XML, a. a. O.

1. Transformation der Struktur des XML-Dokuments (source-tree) mit XSLT in ein neues XML-Dokument (result-tree), das schon XSL-Darstellungsanweisungen enthält. Gegebenenfalls Erstellung eines Inhaltsverzeichnisses oder Manipulation der enthaltenen Daten mit Hilfe von Programmbefehlen (Bedingungen, Sortierbefehle u. a.).⁶⁹
2. Präsentation der transformierten Dokumentstruktur (result-tree) mit XSL-Darstellungsanweisungen durch ein Programm, das XSL-Anweisungen interpretiert und umsetzt.

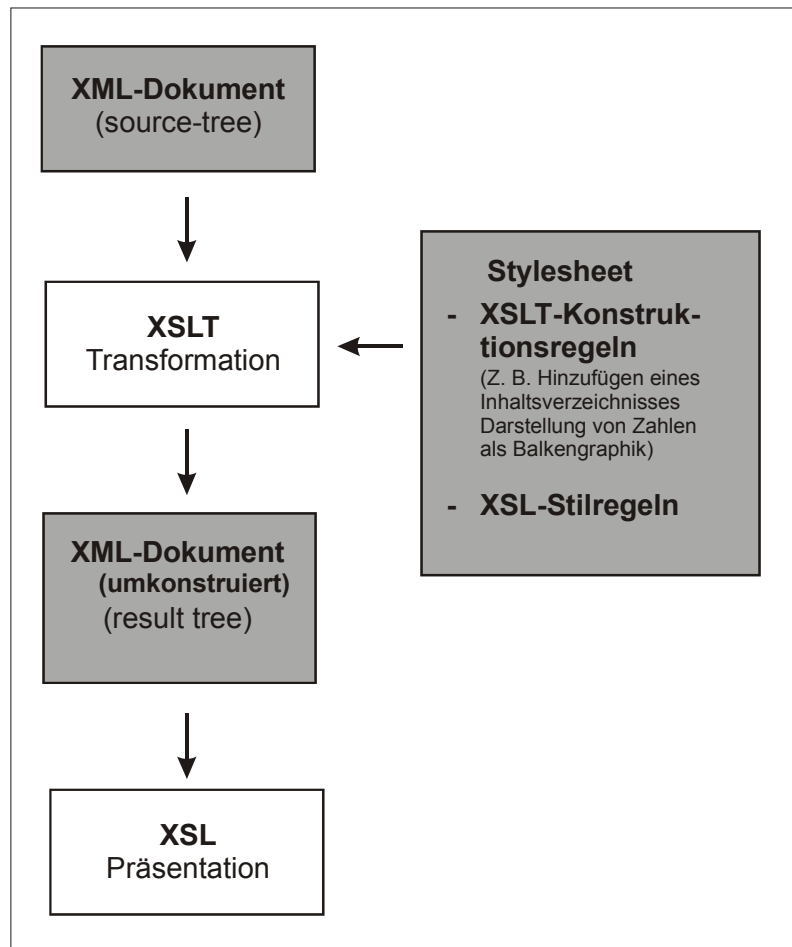


Abb. 18: Funktionsweise der XML-Stylesprache⁷⁰

Mit XSLT kann ein XML-Dokument beispielsweise auch in ein HTML- bzw. XHTML-Dokument transformiert werden, das dann von einem HTML-Browser dargestellt werden kann.⁷¹ In diesem Fall sind die Style-Anweisungen im Browser enthalten und können zusätzlich mit Hilfe von CSS ergänzt werden. Dieses Verfahren dürfte solange von

69 Vgl. Clark, James: XSL Transformations (XSLT) Version 1.0 W3C Recommendation, a. a. O.

70 Vgl. Bos, Bert; Lilley, Chris; Adler, Sharon; Zilles, Steve: Style Sheets Activity Statement, Online im Internet: <http://www.w3.org/Style/Activity>, 26.10.1999.

71 Vgl. Bos, Bert; Lilley, Chris; Adler, Sharon; Zilles, Steve: Style Sheets Activity Statement, a. a. O.

Interesse sein, wie die Entwicklung von XSL noch nicht abgeschlossen ist und die endgültige Version noch nicht in den gängigen Browsern implementiert ist.

Ein weiteres Konzept zur Darstellung von XML-Dokumenten besteht in der Verwendung von DSSSL-Stylesheets. Auch DSSSL erlaubt sowohl die Transformation von Dokumenten als auch die Definition von Darstellungsanweisungen. Somit können XML-Dokumente mit DSSSL in HTML transformiert werden, um von HTML-Browsern dargestellt zu werden.⁷²

Schließlich können XML-Daten auch mit Hilfe von Cascading Style Sheets (CSS) dargestellt werden. Entsprechende Anweisungen sind in der zweiten W3C-Recommendation (CSS2) spezifiziert.⁷³ CSS2 bieten allerdings nicht die Möglichkeit der Transformation, so daß Daten nur genau in der Reihenfolge dargestellt werden können, in der sie im XML-Dokument ausgezeichnet sind. Derzeit wird die dritte Version von CSS (CSS3) entwickelt, die einige Zusatzfunktionen beinhalten soll, wie z. B. Regeln zur Erstellung von Audio-Stylesheets, Darstellung von Text in mehreren Spalten, Definition von Seitenumbrüchen und unterschiedliche Textrichtungen.⁷⁴

4.3 Verweistechiken

Hyperlinks (oft als Verweis oder eng. Link bezeichnet) bestehen immer aus Ausgangs- und Endpunkt. Während bei HTML die Markup-Elemente für das Einrichten von Hyperlinks (der Tag `` für den Ausgangspunkt, der Tag `` für einen Zielpunkt) im HTML-Standard integriert sind, gehören sie bei XML nicht zum XML-Kernbestand. Sie sind im peripheren XML-Linking-Modul zu finden. Dieses Modul wird von der XML Linking Working Group gegenwärtig noch bearbeitet, seit Juni bzw. Juli 2000 existieren Candidate Recommendations für XPointer und XLink.⁷⁵ XML-Linking setzt sich aus zwei Teilen zusammen:⁷⁶

XLink

XLink enthält die Markup-Elemente für die Erstellung von Hyperlinks in XML-Dokumenten. Mit Hilfe von XLink können, wie in HTML, einfache, unidirektionale Links erzeugt werden, die auf eine bestimmte Ressource verweisen. Als Verweisadresse kann hier ein Uniform Resource Identifier (URI) verwendet werden. Darüber hinaus unterstützt XLink aber auch erweiterte Linkeigenschaften wie

72 Vgl. Behme, Henning: Faulheit siegt, in: iX, 11/1998.

73 Vgl. Bos, Bert; Wium Lie, Hakon; Lilley, Chris; Jacobs, Ian: Cascading Style Sheets, level 2 CSS2 Specification, Online im Internet: <http://www.w3.org/TR/1998/REC-CSS2-19980512>, 12.05.1999.

74 Vgl. Bos, Bert; Lilley, Chris; Adler, Sharon; Zilles, Steve: Style Sheets Activity Statement, a. a. O.

75 Vgl. Daniel, Ron Jr.; DeRose, Steve; Maler, Eve: XML Pointer Language (XPointer) Version 1.0 W3C Candidate Recommendation, Online im Internet: <http://www.w3.org/TR/xptr>, 23.07.2000 und DeRose, Steve; Maler, Eve; Orchard, David; Trafford, Ben: XML Linking Language (XLink) Version 1.0 W3C Candidate Recommendation, Online im Internet: <http://www.w3.org/TR/xlink/>, 01.08.2000.

76 Vgl. DeRose, Steven J.: XML Linking An Introduction by Steven J. DeRose, Online im Internet: <http://www.stg.brown.edu/~sjd/xlinkintro.html>, 27.09.1999.

z. B. Verweise eines Ausgangspunktes auf mehrere Zielressourcen, multidirektionale Verweise und das Einrichten von Verweisen auf Ressourcen, deren Dateityp die Definition eines genau bestimmten Zielpunktes nicht unterstützen (z. B. eine bestimmte Szene in einer Videoanimation). Im Unterschied zu HTML, bei dem ein Zielpunkt im Zieldokument mit einem sog. „Anker“ gekennzeichnet werden muß, benötigt man für XLinks kein Zielzeichen im Zieldokument, sondern verwendet XPointer. XLink-Verweise verfügen über zahlreiche Eigenschaften, die eine Steuerung des Verhaltens von Links zulassen, z. B. die Modifikation des Auslöseereignisses eines Verweises oder die Art der Anzeige des Verweisziels. Eine Anzeigeoption soll es erlauben, Zieldokumente eines Verweises (oder Teile davon), direkt in das Dokument einzubinden, das den Link enthält. Weiterhin können Links semantische Informationen enthalten, die das Ziel des Links näher beschreiben.

XPointer

XPointer dienen der genauen Definition des Ziels eines Verweises. Wie in HTML können auch in XML Verweise auf bestimmte Stellen innerhalb von Dokumenten gesetzt werden. Dazu dienen sogenannte Pointer, für die es, im Gegensatz zu HTML, nicht erforderlich ist, daß im Zieldokument zuvor ein Verweisziel („Anker“) definiert wurde. Es ist kein Eingriff im Zieldokument notwendig, es können somit auch Links auf beliebige Bereiche eines Fremddokuments gesetzt werden, wenn für das Fremddokument kein Schreibzugriff besteht. Ein Pointer kann auf eine bestimmte Stelle wie auch auf einen Bereich innerhalb eines XML-Dokuments verweisen, wobei er sich an der Dokumentstruktur orientiert. Verweisziel können Elementtypen, Attributwerte, Zeichen oder Zeichenfolgen in Elementen sowie relative Positionen im Dokument sein.

Mit XML besteht die Möglichkeit, Hyperlinks als separate Entities (eine Art Variable, die zentral in der DTD definiert wird)⁷⁷ auszuzeichnen, auf die in XML-Dokumenten referenziert werden kann. So können Verweisadressen, die in einer beliebigen Zahl von Web-Pages erscheinen sollen, zentral verwaltet werden. Um eine Verweisadresse zu aktualisieren muß nur noch einmalig das Entity in der DTD geändert werden. Da sich diese Änderung automatisch auf alle Dokumente auswirkt, die auf das entsprechende Entity verweisen, sinkt z. B. der Aufwand zur Behebung von sog. „Broken-Links“.

4.4 Namensräume

Eine unter dem Aspekt der globalen Verwendung von DTDs bedeutende Erweiterung der XML-Syntax sind Namensräume (Namespaces). Jede DTD definiert eine abgegrenzten Menge an Tags, wobei die Namen der Tags eindeutig sind, da Namensüberschneidungen nicht erlaubt sind. Werden allerdings mehrere DTDs in einem Dokument verwendet, kann es zu Überschneidungen in der Bezeichnung der Tags kommen.

⁷⁷ Vgl. John, Volker: XML - Weltsprache für das Internet, a. a. O., S. 24.

So beschreibt das Element „Position“ im Beispiel der elektronischen Rechnung aus Abb. 8 eine Rechnungsposition, in einem anderen Dokumenttyp könnte durch ein Element „Position“ die Rolle beschrieben werden, die eine Person innerhalb einer Organisation einnimmt oder eine bestimmte Stellung, in die ein Roboter-Werkzeugarm zu Beginn eines Arbeitsschrittes gebracht werden muß.

Um die Zweideutigkeit von Element- und Attributnamen zu vermeiden, können sich innerhalb eines XML-Dokuments verwendete Namen (local names) auf Namen eines Namensraums (universal names) beziehen, deren Bedeutung innerhalb dieses Namensraums eindeutig festgelegt ist. Eine Beibehaltung des „local name“ ist dabei möglich, so daß Element- und Attributnamen in einem Dokument nach wie vor von seinem Autor bestimmt werden können.⁷⁸

Sowohl Elemente als auch Attribute in XML-Dokumenten können Namen aus einem Namensraum verwenden. Ziel von Namensräumen ist, daß Element- und Attributnamen für XML-Dokumente definiert werden können, deren Bedeutung eindeutig und universell gültig ist.⁷⁹

In Abb. 19 wird die Verwendung eines Namensraums für das Element „Position“ aus der oben beschriebenen Rechnung dargestellt. Der verwendete Namensraum wird hier durch den URI „<http://www.namensraum.de/rechnungen>“ identifiziert und enthält einen Elementnamen „Rechnungsposition“, auf den sich das Element Position bezieht. In der ersten Zeile steht die sogenannte Namensraumdeklaration (namespace declaration), in der festgelegt wird, daß sich das Element Rechnungspositionen und alle enthaltenen Elemente und Attribute auf einen bestimmten Namensraum beziehen können, der die Bezeichnung „faktura“ und die Verweisadresse „<http://www.namensraum.de/rechnungen>“ hat. Die zweite Zeile enthält die Information, daß der Elementname „Position“ der Bezeichnung „fakt_position“ aus dem Namensraum „faktura“ gleichbedeutend ist.⁸⁰

```
<Rechnungspositionen
xmlns:faktura='http://www.namensraum.de/rechnungen'>
<Position faktura:fakt_position>Auftragsfahrt</Position>
</Rechnungspositionen>
```

Abb. 19: Namensraum-Referenz in XML

Der Vorteil des Einsatzes von Namensräumen ist, daß Element- und Attributnamen innerhalb von Dokumenten, unabhängig davon, in welcher Sprache sie geschrieben sind, durch den Verweis auf einen Namen in einem Namensraum eine eindeutige und universelle Bedeutung erhalten.

78 Vgl. Clark, James: XML Namespaces, Online im Internet: <http://www.jclark.com/xml/xmlns.htm>, 04.02.1999.

79 Vgl. Clark, James: XML Namespaces, a. a. O.

80 Vgl. Bray, Tim; Hollander, Dave; Layman, Andrew: Namespaces in XML, Online im Internet: <http://www.w3.org/TR/1999/REC-xml-names-19990114>, 14.01.1999.

5 XML-Dokumente mit dem Document Objekt Model und XML-QL

Einer der wesentlichen Vorteile von XML gegenüber HTML ist die Möglichkeit, strukturierte Daten beliebiger Komplexität mit individuell definierbaren, semantischen Auszeichnungselementen zu beschreiben, wodurch ihre Bedeutung und Struktur erhalten bleiben und eine maschinelle Verarbeitung ermöglicht wird. XML-Dokumente können von jeder Programmiersprache aus unterschiedlich organisierten Datenbeständen erzeugt werden.⁸¹ Die XML-Syntax erlaubt es, Programme und Script-Befehle als PI auszuzeichnen, so daß ein XML-Dokument neben den Daten auch Anweisungen zu ihrer Verarbeitung erhalten kann.

Die formale Festlegung von Struktur und Inhalt einer Klasse von Dokumenten in einer DTD, erlaubt es, Programme zur Verarbeitung einer unbegrenzten Zahl von Dokumenten eines bestimmten Dokumenttyps zu schreiben. Die Fähigkeit, komplexe Datenstrukturen plattformübergreifend zu beschreiben macht XML zu einem mächtigen, universell einsetzbaren Instrument zur Datenbeschreibung. In diesem Zusammenhang sprechen viele Autoren von einer besonderen Synergie beim Einsatz von XML als plattformübergreifendes Datenformat und Java als plattformübergreifende Programmiersprache.⁸² Behme und Mintert bezeichnen die Kombination von XML und Java als mögliche Schlüsseltechnologie der nächsten Jahre.⁸³

Zur Verarbeitung von XML-Dokumenten wurden vom W3C diverse Entwürfe und Spezifikationen von Zusatztechnologien veröffentlicht, darunter eine Schnittstelle, die den Zugriff von Programmiersprachen auf HTML- und XML-Dokumente regelt (DOM) sowie eine Abfragesprache für XML-Daten (XML-QL).

Bereits seit Oktober 1998 liegt eine offizielle W3C-Spezifikation des Document Object Model (DOM) vor. Das DOM ist eine plattform- und sprachenunabhängige Schnittstelle, ein sogenanntes Application Programming Interface (API), das beschreibt, wie Programmier- und Script-Sprachen dynamisch auf Inhalt, Struktur und Layout von XML- oder HTML-Dokumenten zugreifen und diese erzeugen bzw. verändern können.⁸⁴ Durch das DOM werden alle Elemente in einem Dokument als Objekte definiert und in einer Objekthierarchie dargestellt. Weiterhin werden Bedeutung und Eigenschaften der Objekte beschrieben und festgelegt, welche Methoden sie haben, also wie auf sie zugegriffen und wie sie verarbeitet werden können.⁸⁵

81 Vgl. Flynn, Peter; Allen, Terry; Borgman, Tom; Cover, Robin; DuCharme, Bob; Maden, Christopher; Quin, Liam; Sperberg-McQueen, Michael; Weber, Joel; Murata,.: Frequently Asked Questions about the Extensible Markup Language, Online im Internet: <http://www.ucc.ie/xml>, 02.05.2000.

82 Vgl. Morgenthal, JP: Portable Data/Portable Code: XML&Java Technologies, Online im Internet: <http://java.sun.com/xml/ncfocus.html>, 15.06.1999; Weitzel, Tim; Buxmann, Peter; Ladner, Frank; König, Wolfgang: Konzept und Anwendung der Extensible Markup Language, a. a. O. und Bosak, Jon: XML, Java, and the future of the Web, a. a. O.

83 Vgl. Behme, Henning; Mintert, Stefan: XML in der Praxis, a. a. O., S. 222.

84 Vgl. Wood, Lauren; Le Hégaré, Philippe: Document Object Model (DOM), Online im Internet: <http://www.w3.org/DOM/>, 11.12.1999.

85 Vgl. Robie, Jonathan: Document Object Model (DOM) Level 1 Specification Version 1.0, Online im Internet: <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001>, 01.10.1999.

Für das Jahr 2000 ist die Veröffentlichung der XML Query Language (XML-QL) geplant. Zu Beginn der dritten Entwicklungsphase von XML, im September 1999, wurde vom W3C eine entsprechende XML Query Working Group beauftragt, eine flexible Abfragesprache für XML-Dokumente auszuarbeiten.⁸⁶ Die XML-QL soll für XML-Dokumente Möglichkeiten der Datenmanipulation bieten, wie sie die Structured Query Language (SQL) für relationale oder die Object Query Language (OQL) für objektorientierte Datenbanken bereit stellen. Dabei sollen von der XML-QL vor allem die nachfolgend aufgezählten Aufgaben unterstützt werden:⁸⁷

- Extrahieren von Daten aus großen XML-Dokumenten,
- Datenkonvertierung zwischen relationalen oder objektorientierten Datenbanken und XML-Daten,
- Transformation von Daten eines Dokumenttyps in einen anderen,
- Zusammenführung von Daten aus mehreren unterschiedlichen Quellen.

Ziel ist weiterhin, daß die XML-QL-Abfragen in Form einer PI in XML-Dokumente eingebettet werden können.

86 Vgl. Connolly, Dan; Bray, Tim: Extensible Markup Language (XML) Activity, a. a. O.

87 Vgl. Deutsch, Alin; Fernadez, Mary; Florescu, Daniela; Levy, Alon; Suciu, Dan: A Query Language for XML, Online im Internet: <http://www.research.att.com/~mff/xmlql/doc/files/final.html>, 19.12.1999.

Literaturverzeichnis

- Adler, S. et al.: Extensible Stylesheet Language (XSL) Specification W3C Working Draft, Online im Internet: <http://www.w3.org/TR/2000/WD-xsl-20000327/xslspec.html>, 01.08.2000.
- Bathe, Heike; Beer, Frank Michael: Warum kompliziert, wenn's auch einfach geht?, in: LANline 11/99, S. 88.
- Behme, Henning: Faulheit siegt, in: iX, 11/1998, S. 186-190.
- Behme, H.; Mintert, S.: XML in der Praxis, Bonn: Addison Wesley Longman Verlag 1998.
- Berners-Lee, Tim: W3C Technical Reports and Publications, Online im Internet: <http://www.w3.org/TR/>, 01.08.2000.
- Berners-Lee, Tim: About the World Wide Web Consortium, Online im Internet: <http://www.w3.org/Consortium/>, 09.08.1997.
- Biron, Paul V.; Malhotra, Ashok: XML Schema Part 2: Datatypes, Online im Internet: <http://www.w3.org/TR/1999/WD-xmlschema-2-19991105/>, 05.11.1999.
- Bos, Bert: XML in 10 points, Online im Internet: <http://www.w3.org/XML/1999/XML-in-10-points>, 01.08.2000.
- Bos, Bert; Lilley, Chris; Adler, Sharon; Zilles, Steve: Style Sheets Activity Statement, Online im Internet: <http://www.w3.org/Style/Activity>, 26.10.1999.
- Bos, B.; Wium Lie, H.; Lilley, C.; Jacobs, I.: Cascading Style Sheets, level 2 CSS2 Specification, Online im Internet: <http://www.w3.org/TR/1998/REC-CSS2-19980512>, 12.05.1999.
- Bosak, Jon: XML, Java, and the future of the Web, Online im Internet: <http://www.metabolab.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm>, 03.10.1997.
- Bray, Tim: Beyond HTML: XML and Automated Web Processing, Online im Internet: http://developer.netscape.com/viewsource/bray_xml.html, 10.09.1999.
- Bray, Tim; Hollander, Dave; Layman, Andrew: Namespaces in XML, Online im Internet: <http://www.w3.org/TR/1999/REC-xml-names-19990114>, 14.01.1999.
- Bray, T.; Paoli, J. et al.: Extensible Markup Language (XML) W3C Recommendation, Online im Internet: <http://www.w3.org/TR/1998/REC-xml-19980210>, 10.02.1998.
- Clark, James: Associating Style Sheets with XML documents Version 1.0, Online im Internet: <http://www.w3.org/1999/06/REC-xml-stylesheet-19990629>, 29.06.1999.
- Clark, James: Comparison of SGML and XML, Online im Internet: <http://www.w3.org/TR/NOTE-sgml-xml-971215>, 15.12.1997.
- Clark, J.: XML Namespaces, Online im Internet: <http://www.jclark.com/xml/xmlns.htm>, 04.02.99.
- Clark, James: XSL Transformations (XSLT) Version 1.0 W3C Recommendation, Online im Internet: <http://www.w3.org/TR/1999/REC-xslt-19991116>, 16.11.1999.
- Connolly, Dan: Extensible Markup Language (XML), Online im Internet: <http://www.w3.org/XML/>, 10.11.1999.
- Connolly, Dan; Bray, Tim: Extensible Markup Language (XML) Activity, Online im Internet: <http://www.w3.org/XML/Activity.html>, 27.10.1999.
- Daniel, R. Jr.; DeRose, S.; Maler, E.: XML Pointer Language (XPointer) Version 1.0 W3C Candidate Recommendation, Online im Internet: <http://www.w3.org/TR/xptr>, 23.07.2000.
- DeRose, S.; Maler, E.; Orchard, D.; Trafford, B.: XML Linking Language Version 1.0 W3C Candidate Recommendation, Online im Internet: <http://www.w3.org/TR/xlink/>, 01.08.2000.
- DeRose, Steven J.: XML Linking An Introduction by Steven J. DeRose, Online im Internet: <http://www.stg.brown.edu/~sjd/xlinkintro.html>, 27.09.1999.
- Deutsch, A.; Fernandez, M.; Florescu, D.; Levy, A.: A Query Language for XML, Online im Internet: <http://www.research.att.com/~mff/xmlql/doc/files/final.html>, 19.12.1999.
- Dönhölder, Kuno: Das Web automatisieren mit XML, Online im Internet: <http://members.aol.com/xmldoku/>, 01.09.1998.

- Everson, Michael: Universal Character Set Collections, Online im Internet: <http://www.indigo.ie/egt/standards/iso10646/ucs-collections.html>, 23.09.1999.
- Farsi, Reza: XML, in: Informatik Spektrum, Dez. 1999, S. 436-438.
- Flynn, P.; Allen, T. et al.: Frequently Asked Questions about the Extensible Markup Language, Online im Internet: <http://www.ucc.ie/xml>, 02.05.2000.
- Harbarth, Jürgen: Einen Schritt weiter, in: N&C, 1/1999, S. 80-83.
- John, Volker: XML - Weltsprache für das Internet, in: Objekt Spektrum, 5/99, S. 22-28.
- Johnson, Charles David: ISO-10646 Concept Dictionary, Online im Internet: http://cns-web.bu.edu/pub/djohnson/web_files/i18n/ISO-10646.html, 22.09.1997.
- Jung, Frank: Universelles Datenaustauschformat, in: it FOKUS, 2/2000, S. 9-14.
- Macherius, Ingo: Revolution der Experten, Online im Internet: <http://www.heise.de/ix/artikel/1997/06/106/artikel.html>, 14.05.1997.
- Malhotra, Ashok; Maloney, Murray: XML Schema Requirements, Online im Internet: <http://www.w3.org/TR/NOTE-xml-schema-req>, 15.02.1999.
- Mintert, Stefan: Einführung in die Extensible Markup Language (XML), Online im Internet: <http://computerphilologie.uni-muenchen.de/jg99/xml.html>, 08.03.1999.
- Morgenthal, JP: Portable Data/Portable Code: XML&Java Technologies, Online im Internet: <http://java.sun.com/xml/ncfocus.html>, 15.06.1999.
- O'Donnell, B.: Will XML become the universal document format?, Online im Internet: <http://www.infoworld.com/cgi-bin/diplayNew.pl?/odonnell/-971215od.htm>, 15.12.1997.
- o. V.: The SGML Guide, Online im Internet: <http://www.oasis-open.org/cover/ileafgd.html>, 21.07.1999.
- o. V.: World Wide Web Consortium (W3C) Members, Online im Internet: <http://www.w3.org/Consortium/Member/List.html>, 11.07.1999.
- o. V.: World Wide Web Consortium Process Document, Online im Internet: <http://www.w3.org/Consortium/Process/Process-19991111/process.pdf>, 01.08.2000.
- Pemberton, Steven et al.: XHTML 1.0: The Extensible HyperText Markup Language, Online im Internet: <http://www.w3.org/TR/1999/PR-xhtml1-19990824>, 24.08.1999.
- Quint, V.: W3C User Interface Domain, Online im Internet: <http://www.w3.org/UI/>, 18.08.99.
- Reibold, Holger: XML in der Praxis, in: PC Professionell, 4/2000, S. 253-257.
- Reinbold, Holger: XML basierte Graphiken fürs Web, in: it FOKUS, 2/2000, S. 16-20.
- Robie, Jonathan: Document Object Model (DOM) Level 1 Specification Version 1.0, Online im Internet: <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001>, 01.10.1999.
- Schwickert, Axel C.: Richtlinien für die Gestaltung von Wissenschaftlichen Arbeiten, 6., unveränderte Aufl., Mainz: 1998.
- Sonntag, R.: Neue Offenheit für E-Business, in: Diebold Management Report, 11/99, S. 27-30.
- Sperberg-McQueen, C. M.: Reports From the W3C SGML ERB to the SGML WG And from the W3C XML ERB to the XML SIG, Online im Internet: <http://www.w3.org/XML/9712-reports.html>, 04.12.1997.
- Stahlknecht, Peter; Hasenkamp, Ulrich: Einführung in die Wirtschafts-informatik, 8., vollständig überarb. und erweit. Aufl., Berlin et al.: Springer-Verlag 1997.
- Thompson, H. S.; Beech, D.; Maloney, M.; Mendelsohn, N.: XML Schema Part 1: Structures, Online im Internet: <http://www.w3.org/TR/1999/WD-xmlschema-1-19991105/>, 05.11.1999.
- Tolksdorf, Robert: XML und darauf basierende Standards: Die neuen Auszeichnungssprachen des Web, in: Informatik Spektrum, Dez. 1999, S. 407-421.
- Weitzel, Tim; Buxmann, Peter; Ladner, Frank; König, Wolfgang: Konzept und Anwendung der Extensible Markup Language, Online im Internet: <http://caladan.wiwi.uni-frankfurt.de/IWI/projectb3/deu/publikat/xml/index.htm>, 13.08.1999.
- Winter, Dirk; Zitzmann, Georg; Draibach, Marc: XML-Überblick, Online im Internet: <http://turing.informatik.fh-wiesbaden.de/projekt2/>, 28.04.1998.
- Wood, Lauren; Le Hégarret, Philippe: Document Object Model (DOM), Online im Internet: <http://www.w3.org/DOM/>, 11.12.1999.

Bisher erschienen

Stand: Dezember 2000 – Den aktuellen Stand der Reihe erfahren
Sie über unsere Web Site unter <http://wi.uni-giessen.de>

Nr. 1/1996	Grundlagen des Client/Server-Konzepts.....	Schwicker/Grimbs
Nr. 2/1996	Wettbewerbs- und Organisationsrelevanz des Client/Server-Konzepts.....	Schwicker/Grimbs
Nr. 3/1996	Realisierungsaspekte des Client/Server-Konzepts	Schwicker/Grimbs
Nr. 4/1996	Der Geschäftsprozeß als formaler Prozeß - Definition, Eigenschaften, Arten	Schwicker/Fischer
Nr. 5/1996	Manuelle und elektronische Vorgangsteuerung.....	Schwicker/Rey
Nr. 6/1996	Das Internet im Unternehmen - Neue Chancen und Risiken	Schwicker/Ramp
Nr. 7/1996	HTML und Java im World Wide Web.....	Gröning/Schwicker
Nr. 8/1996	Electronic-Payment-Systeme im Internet.....	Schwicker/Franke
Nr. 9/1996	Von der Prozeßorientierung zum Workflow-Management - Teil 1: Grundgedanken, Kernelemente, Kritik	Maurer
Nr. 10/1996	Von der Prozeßorientierung zum Workflow- Management - Teil 2: Prozeßmanagement und Workflow	Maurer
Nr. 11/1996	Informationelle Unhygiene im Internet.....	Schwicker/Dietrich/Klein
Nr. 12/1996	Towards the theory of Virtual Organisations: A description of their formation and figure.....	Appel/Behr
Nr. 1/1997	Der Wandel von der DV-Abteilung zum IT-Profitcenter: Mehr als eine Umorganisation.....	Kargl
Nr. 2/1997	Der Online-Markt - Abgrenzung, Bestandteile, Kenngrößen	Schwicker/Pörtner
Nr. 3/1997	Netzwerkmanagement, OSI Framework und Internet SNMP	Klein/Schwicker
Nr. 4/1997	Künstliche Neuronale Netze - Einordnung, Klassifikation und Abgrenzung aus betriebswirtschaftlicher Sicht	Strecker/Schwicker
Nr. 5/1997	Sachzielintegration bei Prozeßgestaltungsmaßnahmen.....	Delnef
Nr. 6/1997	HTML, Java, ActiveX - Strukturen und Zusammenhänge.....	Schwicker/Dandl
Nr. 7/1997	Lotus Notes als Plattform für die Informationsversorgung von Beratungsunternehmen.....	Appel/Schwaab
Nr. 8/1997	Web Site Engineering - Modelltheoretische und methodische Erfahrungen aus der Praxis	Schwicker
Nr. 9/1997	Kritische Anmerkungen zur Prozeßorientierung	Maurer/Schwicker
Nr. 10/1997	Künstliche Neuronale Netze - Aufbau und Funktionsweise	Strecker
Nr. 11/1997	Workflow-Management-Systeme in virtuellen Unternehmen	Maurer/Schramke
Nr. 12/1997	CORBA-basierte Workflow-Architekturen - Die objektorientierte Kernanwendung der Bausparkasse Mainz AG	Maurer
Nr. 1/1998	Ökonomische Analyse Elektronischer Märkte.....	Steyer
Nr. 2/1998	Demokratiopolitische Potentiale des Internet in Deutschland	Muzic/Schwicker
Nr. 3/1998	Geschäftsprozeß- und Funktionsorientierung - Ein Vergleich (Teil 1)	Delnef
Nr. 4/1998	Geschäftsprozeß- und Funktionsorientierung - Ein Vergleich (Teil 2)	Delnef
Nr. 5/1998	Betriebswirtschaftlich-organisatorische Aspekte der Telearbeit	Polak
Nr. 6/1998	Das Controlling des Outsourcings von IV-Leistungen	Jäger-Goy
Nr. 7/1998	Eine kritische Beurteilung des Outsourcings von IV-Leistungen.....	Jäger-Goy
Nr. 8/1998	Online-Monitoring - Gewinnung und Verwertung von Online-Daten.....	Guba/Gebert
Nr. 9/1998	GUI - Graphical User Interface.....	Maul
Nr. 10/1998	Institutionenökonomische Grundlagen und Implikationen für Electronic Business.....	Schwicker
Nr. 11/1998	Zur Charakterisierung des Konstrukts "Web Site".....	Schwicker
Nr. 12/1998	Web Site Engineering - Ein Komponentenmodell.....	Schwicker
Nr. 1/1999	Requirements Engineering im Web Site Engineering – Einordnung und Grundlagen.....	Schwicker/Wild
Nr. 2/1999	Electronic Commerce auf lokalen Märkten	Schwicker/Lüders
Nr. 3/1999	Intranet-basiertes Workgroup Computing	Kunow/Schwicker
Nr. 4/1999	Web-Portale: Stand und Entwicklungstendenzen.....	Schumacher/Schwicker
Nr. 5/1999	Web Site Security.....	Schwicker/Häusler
Nr. 6/1999	Wissensmanagement - Grundlagen und IT-Instrumentarium.....	Gaßen
Nr. 7/1999	Web Site Controlling.....	Schwicker/Beiser
Nr. 8/1999	Web Site Promotion	Schwicker/Arnold
Nr. 9/1999	Dokumenten-Management-Systeme – Eine Einführung	Dandl
Nr. 10/1999	Sicherheit von eBusiness-Anwendungen – Eine Fallstudie	Harper/Schwicker
Nr. 11/1999	Innovative Führungsinstrumente für die Informationsverarbeitung	Jäger-Goy
Nr. 12/1999	Objektorientierte Prozeßmodellierung mit der UML und EPK	Dandl
Nr. 1/2000	Total Cost of Ownership (TCO) – Ein Überblick.....	Wild/Herges
Nr. 2/2000	Implikationen des Einsatzes der eXtensible Markup Language – Teil 1: XML-Grundlagen.....	Franke/Sulzbach
Nr. 3/2000	Implikationen des Einsatzes der eXtensible Markup Language – Teil 2: Der Einsatz im Unternehmen	Franke/Sulzbach
Nr. 4/2000	Web-Site-spezifisches Requirements Engineering – Ein Formalisierungsansatz	Wild/Schwicker
Nr. 5/2000	Elektronische Marktplätze – Formen, Beteiligte, Zutrittsbarrieren	Schwicker/Pfeiffer
Nr. 6/2000	Web Site Monitoring – Teil 1: Einordnung, Handlungsebenen, Adressaten.....	Schwicker/Wendt
Nr. 7/2000	Web Site Monitoring – Teil 2: Datenquellen, Web-Logfile-Analyse, Logfile-Analyzer	Schwicker/Wendt
Nr. 8/2000	Controlling-Kennzahlen für Web Sites.....	Schwicker/Wendt
Nr. 9/2000	eUniversity – Web-Site-Generierung und Content Management für Hochschuleinrichtungen.....	Schwicker/Ostheimer/Franke

Bestellung (bitte kopieren, ausfüllen, zusenden/zufaxen)

Adressat: Professur für BWL und Wirtschaftsinformatik
 Fachbereich Wirtschaftswissenschaften
 Licher Straße 70
 D – 35394 Gießen
 Telefax: (0 641) 99-22619

Hiermit bestelle ich gegen Rechnung die angegebenen Arbeitspapiere zu einem Kostenbeitrag von DM 10,- pro Exemplar (MwSt. entfällt) zzgl. DM 5,- Versandkosten pro Sendung.

Nr.	An
1/1996	
2/1996	
3/1996	
4/1996	
5/1996	
6/1996	
7/1996	
8/1996	
9/1996	
10/1996	
11/1996	
12/1996	

Nr.	An
1/1997	
2/1997	
3/1997	
4/1997	
5/1997	
6/1997	
7/1997	
8/1997	
9/1997	
10/1997	
11/1997	
12/1997	

Nr.	Anz
1/1998	
2/1998	
3/1998	
4/1998	
5/1998	
6/1998	
7/1998	
8/1998	
9/1998	
10/1998	
11/1998	
12/1998	

Nr.	Anz
1/1999	
2/1999	
3/1999	
4/1999	
5/1999	
6/1999	
7/1999	
8/1999	
9/1999	
10/1999	
11/1999	
12/1999	

Nr.	Anz
1/2000	
2/2000	
3/2000	
4/2000	
5/2000	
6/2000	
7/2000	
8/2000	
9/2000	

Absender:

Organisation _____

Abteilung _____

Nachname, Vorname _____

Straße _____

Plz/Ort _____

Telefon _____ Telefax _____ eMail _____

Ort, Datum _____ Unterschrift _____