

Datenaustausch mit XML

Von Jörg Willig und Thorsten Planeth

Ein erster Überblick...

- 1. XML – Die Grundidee
- 2. Dokumenttypen und ihre Definition
- 3. Verarbeitung von XML
- 4. Anwendung von XML
- 5. Ein Beispiel
- 6. Fazit

1. XML – Die Grundidee

- 1.1 Was ist XML?
- 1.2 Visual Markup
- 1.3 Generic Markup
- 1.4 Ursprünge des Generic Markup
- 1.5 Ein XML-Dokument
- 1.6 Struktur eines XML-Dokumentes

1.1 Was ist XML?

- eXtensible Markup Language
- Spezifikation vom World Wide Web Consortium (W3C)
- XML ist eine Meta-Sprache zur Definition eigener Markup-Sprachen, wie z.B.:
 - MathML (Mathematical Markup Language)
 - CML (Chemical Markup Language)
 - SMIL (Synchronized Multimedia Integration Language)
 - XHTML
 - (SVG) Scalable Vektor Graphics
- Markup: Meta-Informationen zu einem Text

1.2 Visual Markup

- Unter ‚Markup‘ sind Auszeichnungen zu verstehen, mit denen Texte versehen werden
- ‚Visual markup‘ bezeichnet Auszeichnungen, die der Formatierung dienen
- Dies sind Attribute wie etwa ‚fett‘ oder ‚kursiv‘

1.3 Generic Markup

- ‚Generic Markup‘ meint die Verwendung von Attributen, die einem Textelement eine Bedeutung zuschreiben
- Ein solches Attribut kann z.B. innerhalb eines Dokumentes der ‚Titel‘ sein
- Diese Bezeichnung sagt nichts über die Darstellung des Textes aus
- Formatierungsanweisungen werden in stylesheets festgelegt (definiert in XSL)

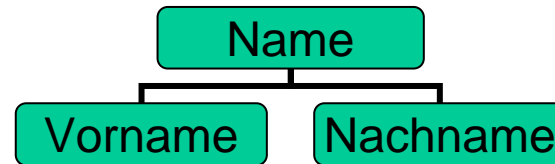
1.4 Ursprünge des Generic Markup

- Erste Ideen in den 60er Jahren
- Ein Meilenstein war die Entwicklung von SGML in den 80er Jahren
- Einsatz von SGML zur Speicherung und zum Austausch von Dokumenten durch die internationale Luftfahrtindustrie und das US Militär
- SGML ist sehr umfangreich
- XML ist eine kompakte Teilmenge von SGML die deren wesentlichen Elemente enthält

1.5 Ein XML Dokument

- `<?xml version „1.0“>`
`<Name>`
 `<Vorname>`
 Inge
 `</Vorname>`
 `<Nachname>`
 Meysel
 `</Nachname>`
`</Name>`
- Prolog im XML-Dokument spezifiziert XML-Version

1.6 Struktur eines Dokuments



Durch verschachtelte Elemente besitzt ein XML-Dokument eine Baumstruktur

2. Dokumenttypen

- 2.1 Definition von Dokumenttypen
- 2.2 XML Schema
- 2.3 Was bietet Schema?
- 2.4 Ein Beispiel
- 2.5 Namespaces
- 2.6 Deklaration von Namespaces
- 2.7 Auswirkungen von XML Schema

2.1 Definition von Dokumenttypen

- Zentrales Konzept von XML
- Ein Dokumenttyp beschreibt eine Klasse von Dokumenten, mit gleichem strukturellen Aufbau
- Beispiel für einen Dokumenttyp: HTML-Dokumente
- Dokumenttypdefinition
 - mit Document Type Definition (DTD)
 - oder XML Schema

2.2 XML Schema...

- ...ist eine Sprache zur Definition von Dokumenttypen
- Ein XML Schema beschreibt die Elemente, Attribute und die Bedingungen für deren Auftreten
- XML Schema bietet mehr Möglichkeiten als DTD und ist selbst in XML definiert
- Schema bietet
 - die bisherige textorientierte Beschreibung
 - eine datenorientierte Beschreibung und damit die Möglichkeit beliebige Datenstrukturen abzubilden

2.3 Was bietet Schema?

- 41 vordefinierte Datentypen (z.B. ‚String‘)
- Durch Erweiterung oder Einschränkung dieser Basisdatentypen bietet Schema die Möglichkeit beliebige Datenstrukturen zu beschreiben
- Durch ‚namespaces‘ können beliebige Elemente die gleichen Namen aber unterschiedliche Inhalte besitzen
- Beliebige Elemente können als eindeutige Elemente oder als Schlüssel definiert werden

2.4 Ein Beispiel

```
<xsd:element name="Rechnungsadresse"
  minOccurs="0" maxOccurs="unbounded"/>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string"/>
      <xsd:element name="strasse" type="xsd:string"/>
      <xsd:element name="plz" type="xsd:decimal"/>
      <xsd:element name="stadt" type="xsd:string"/>
      <xsd:element name="land" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

2.5 Namespaces

- Bei der Definition eigener Sprachen in XML kann es bei Elementnamen zu Konflikten kommen
- Ein Namensraum ist ein frei wählbarer Name, mit dem die Zugehörigkeit eines Elements angegeben werden kann
- Der Namensraum sagt aus, um welches Element es sich wirklich handelt

2.6 Deklaration von Namespaces

- In XML Schema erfolgt die Deklaration der Namensräume im ersten einleitenden schema-tag
- `<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema">`
- ‚xmlns‘ ist ein reservierter Bezeichner, der URI sein Wert
- In diesem Fall also <http://www.w3.org/2001/XMLSchema>

2.7 Auswirkungen von XML Schema

- Sowohl die Eingabe als auch der Austausch beliebiger Datenstrukturen kann standardisiert werden
- Verdrängung proprietärer Datenformate
- Verwendung von standardisierten Werkzeugen zur Verarbeitung dieser Daten

3. Verarbeitung

- 3.1 Parser
- 3.2 Verarbeitung - Baum-orientiert
- 3.3 Verarbeitung - Event-orientiert
- 3.4 Indirekt
- 3.5 Unterstützung von XML durch...

3.1 Parser

- Interpretation eines XML-Dokuments mittels Parser
- So erfolgt eine Analyse der Struktur und des Inhalts der Datei
- Die Dokumentenstruktur kann bei der Verwendung der Daten dann gezielt genutzt werden
- Ist ein Dokument geparst, so stehen dessen Daten für jede Anwendung bereit, die diesen Parser verwendet
- Validierende Parser überprüfen das XML-Dokument zuerst auf seine Gültigkeit – kontrollieren also, ob die Struktur der Daten mit den Vorgaben einer zugehörigen DTD oder eines Schemas übereinstimmt

3.2 Verarbeitung - Baum-orientiert

- DOM: Document Object Model → Standard API
- Dieses Modell beschreibt sprachneutral Klassen und Methoden für den Zugriff auf ein im Speicher befindliches XML-Dokument
- Der Zugriff erfolgt dabei so, als ob das Dokument in einer Datenstruktur abgelegt wäre, die einem Baumgraphen entspricht
- Jedes Element entspricht dabei einem Knoten
- Spezifikation vom W3C

3.3 Verarbeitung - Event-Orientiert

- SAX: Simple API für XML processing
- Basiert auf Interfaces und Klassen, deren Methoden als callbacks aufgerufen werden, während der Parser beim Einlesen den Dokumentbaum durchläuft
- Keine Baumstruktur sondern ein Elementstrom
- Die verwendeten Methoden stehen beispielsweise für bestimmte Start- und End-Tags, die dann ein Ereignis auslösen

3.4 Indirekt

- Durch Data Binding
- Verwendung von XML Data Islands
- Das ist XML
 - entweder direkt in eine Seite eingebettet
 - oder über eine Quellenangabe mit der Seite verknüpft
- Beispiel: Datenquellenangabe

```
<XML SRC=„Datenquelle.xml“>
```

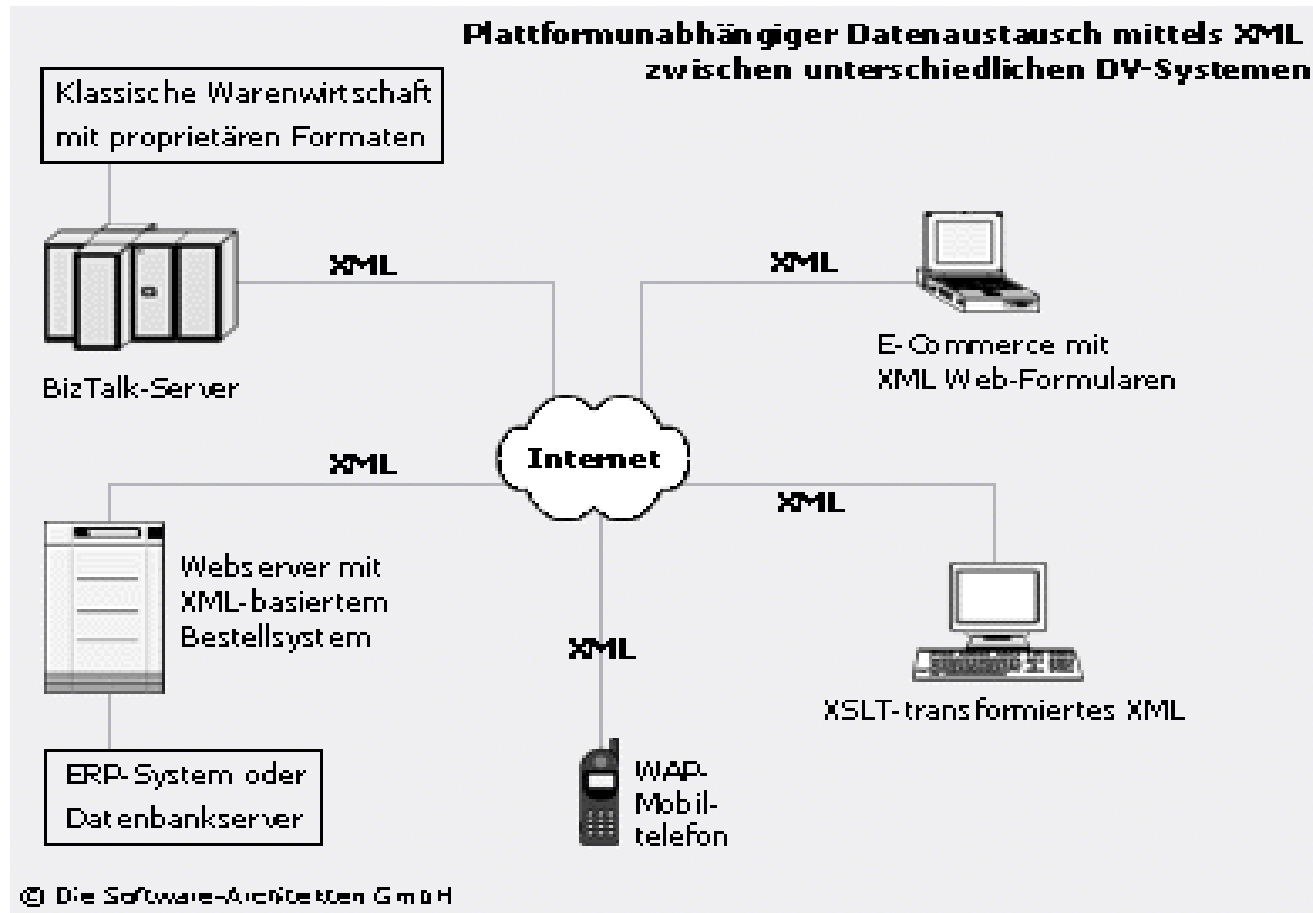
3.5 Unterstützung von XML durch...

- Java
- C++
- C
- Perl
- Python
- VB
- Cobol
- Fortran

4. Anwendung von XML

- 4.1 Datenaustausch
- 4.2 Verschiedene Formate
- 4.3 BizTalk

4.1 Datenaustausch



4.2 Verschiedene Formate

- Unternehmen, die am e-Commerce teilnehmen, legen öffentlich XML-Schemata ab, die Dokumente definieren, z.B. ein Schema ‚Bestellung‘
- Aber: Verschiedene Firmen können verschiedene Dokumentdefinitionen für diesen Typ haben, sogenannte offene Formate
- Notwendig ist beim Datenaustausch also eine Konvertierung in das jeweils andere Format
- Beispiel: BizTalk ermöglicht verschiedene Konvertierungsmöglichkeiten

4.3 BizTalk

- Der BizTalk-Server (MS) ermöglicht:
 - 1:1 Mapping
 - Element A der Firma A entspricht dem Element B im Format der Firma B
 - komplexeres Mapping
 - das Element A (z.B. Preis in DM) der Fa. A entspricht dem Element B der Fa. B, wenn es durch 1,96 geteilt wird (Preis in Euro)
 - Zudem können auch strukturierte Nicht-XML-Dateien wie etwa EDIFACT in XML überführt werden und umgekehrt

5. Ein Beispiel

- Datenaustausch per XML

6. Fazit

- 6.1 Argumente für XML
- 6.2 Fazit

6.1 Argumente für XML (1/4)

- Offener Standard mit hoher Akzeptanz
 - Selbst Microsoft unterstützt XML!
- XML Dokumente liegen als Textdateien vor
 - Beliebige Werkzeuge → Plattform-Unabhängigkeit
 - für Menschen lesbar
- Unterstützung internationaler Zeichensätze
 - Unicode (UTF-8, UTF-16)

6.1 Argumente für XML (2/4)

- Strukturierter Aufbau (Baumstruktur)
 - Darstellung von Beziehungen möglich (1:n)
 - Direkter Zugriff auf Elemente möglich
 - Erleichtert automatische Verarbeitung

6.1 Argumente für XML (3/4)

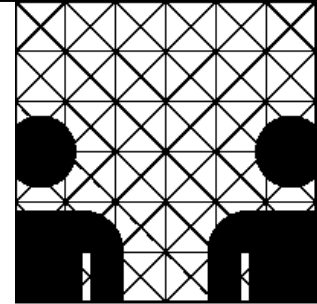
- Flexibel
 - Eigene Dokumenttypen definierbar
 - Darstellung beliebiger Datenstrukturen
- Über DTD oder XML Schema ist die Prüfung von Dokumenten möglich
- Transformation in andere Datenformate einfach
 - z.B. Erzeugen eines HTML Dokumentes aus einem XML Dokument

6.1 Argumente für XML (4/4)

- Festlegung der Darstellung über XSL möglich
 - Medienspezifische Darstellung
- Automatische Erzeugung des Programmcodes zum Erzeugen und Lesen von XML-Dokumenten möglich
 - Erlaubt Fokussierung auf Business Logik
- Niedrige Lernkurve

6.2 Fazit

- Ein Einwand gegen XML ist die Tatsache, dass XML-Dokumente durch reichhaltige Auszeichnung um ein Vielfaches grösser sind als der blosse Text
- Allerdings ist die durch diesen Overhead erzeugte Dokumentenstruktur von extrem hohem Nutzen
- XML ermöglicht die Definition allgemeiner Datenformate, die über das Internet problemlos austauschbar und mit denen beliebige Datenstrukturen darstellbar sind
- Es ermöglicht Unabhängigkeit von proprietären Formaten



Datenaustausch mit XML

Von Jörg Willig und Thorsten Planeth