

# Unbegrenzte Möglichkeiten

## XML: Anwendungen, Potentiale und Standards

Kai-Oliver Detken

Seit den Anfängen von XML im Jahr 1997 wurden der Begriff und die möglichen Einsatzgebiete unterschiedlich interpretiert. Anfangs war man der Meinung, daß die Metasprache XML das WWW grundsätzlich verändern und HTML ablösen wird. Andere wiederum sahen die Unterschiede zwischen XML und HTML hauptsächlich darin, daß XML hinsichtlich der Elementmenge unbegrenzt ist und ein strukturiertes Auszeichnen von Dokumenten ermöglicht. Die Grundidee von XML ist aber die Lösung von Kompatibilitätsproblemen beim Datenaustausch, so daß die Sprache für die Auszeichnung von Dokumenten und als Beschreibung von Metadaten bis hin zum Datenaustausch im Internet eingesetzt werden kann.

Der XML-Standard wurde vom WWW Consortium (W3C) 1997 entwickelt. Die Zielsetzung war dabei, ein universelles Format für strukturierte Dokumente und Daten im Web zu erzeugen. Die erste Spezifikation wurde im Februar 1998 in der Version 1.0 verabschiedet.

Vor XML gab es mit der Generalized Markup Language (SGML) bereits einen Standard für strukturierte Dokumente und Daten im Web. SGML konnte sich jedoch nicht durchsetzen, da er zu umfassend und aufwendig für den Einsatz in Standard-Web-Applikationen ist. Außerdem fehlte die Entwicklung von Tools zur Erleichterung des Einsatzes. Insofern ist die XML-Spezifikation kein neuer Ansatz, sondern verwendet eine Teilmenge der SGML-Syntax, wobei die Komponenten von SGML weggelassen wurden, da sie für die meisten Anwendungsfälle zu komplex und nicht unbedingt erforderlich sind. Um für XML mehr Akzeptanz zu erreichen, wurden diverse Tools und Zusatzstandardisierungen entwickelt.

Neben dem Webpublishing gewinnt XML im Datenaustausch und für das E-Business als zentraler Anwendung immer mehr an Bedeutung. In der aktuellen M-Commerce-Diskussion sehen viele Entwickler ebenfalls eine rosigere Zukunft für XML als kommende Applikation von Handheld und PDA.

### Unterschiede zu HTML

Anfangs ging man davon aus, daß XML den Standard HTML ablösen wird. Dabei ist HTML im Gegensatz zu XML keine Teilmenge von SGML, sondern eine Anwendung, d.h., HTML ist ein mit SGML definiertes Markup. Webbrowser erkennen die Struktur des HTML-Markup und parsen die darzustellenden HTML-Dokumente. Anhand der geparschten Information erzeugen sie die gewünschte Ausgabe.

Die verarbeitende Anwendung ist dabei der HTML-Browser.

XML ist eine Teilmenge von SGML, mit deren Hilfe Markup-Sprachen wie z.B. HTML, eXtended Hyper-Text Markup Language (XHTML), Synchronized Multimedia Language (SMIL) und Resource Description Framework (RDF) definiert werden können. Bei ihnen spricht man von XML-Anwendungen.

Für die Arbeit mit XML werden heute Tools zum Erfassen, Publizieren und Archivieren angeboten. Unter der Oberaufsicht des W3C werden ergänzende Standards wie XLink, XPointer, XPath und XSL entwickelt. Daneben bietet Sun Microsystems Schnittstellen und Klassen für die Programmiersprache Java an, mit denen XML-Anwendungen leicht in Java zu realisieren sind. Denkbar für die Entwicklung einer XML-Anwendung sind aber auch andere Programmiersprachen wie C++ oder Smalltalk. Wesentliche Bausteine für die Implementierung eigener XML-Anwendungen stehen von unterschiedlichen Anbietern teilweise als Open Source frei zur Verfügung.

### Funktionsweise

Ein XML-Dokument enthält Daten und Informationen darüber, in wel-

#### Das Thema in Kürze

Einer der möglichen Standards zur Entwicklung von E-Commerce-Systemen ist die eXtensible Markup Language (XML). Sie kann die Anpassung verschiedener Formate aufeinander vereinfachen, bislang eingesetzte Web-Techniken erweitern bzw. ablösen oder die Softwareentwicklung effizienter gestalten. Dieser Artikel zeigt auf, wozu XML in der Lage ist, und welche Ansätze es in einem aktuellen Forschungsprojekt der EU gibt.

Kai-Oliver Detken ist Senior IT-Consultant der Decoit sowie freier Autor und Berater in Gröden/Bremen

```

<component>
  <data>
    <title>Titel</title>
    <text>Text</text>
    <price>
      <value>144,00</value>
      <currency>DM</currency>
    </price>
  </data>
</component>

```

Metadaten innerhalb eines XML-Dokuments

cher Struktur diese Daten abgelegt sind. Die Informationen, wie diese Daten dargestellt werden sollen, werden allerdings an anderer Stelle definiert. Ein XML-Dokument kann dadurch als Quelle für unterschiedliche Ausgabemedien dienen sowie für den Datenaustausch verwendet werden. Daten in XML-Dokumenten mit identischer Struktur können ohne weitere Bearbeitung quasi als Datencontainer hin- und hergetauscht werden. Falls erforderlich, kann das Extrahieren von Teildaten und das Überführen in andere Strukturen entweder mit XML-Anwendungen oder über XSL erfolgen. Auf bereits einmal in XML-Dokumenten aufbereitete Daten kann so immer wieder zugegriffen werden.

Die XML-Spezifikation ist eine Datenbeschreibungssprache, die als Metasprache bezeichnet wird. Mit der XML-Syntax kann für jede vorliegende Datenstruktur eine Markup-Sprache definiert werden. Die Struktur des XML-Dokuments kann dabei an die Komplexität der Informationen angepasst werden. Die einzelnen Dateneinheiten werden in den für sie definierten Markup-Tags im XML-Dokument gespeichert.

Die Bezeichnung der definierten Markup-Tags gibt Aufschluß über den Informationsgehalt der Daten. Eine so definierte Struktur beinhaltet dadurch neben den eigentlichen auch Metadaten. So können Informationen über eine Komponente gespeichert werden, die einen title-, text- oder price-Tag in einem data-Tag enthalten.

Die XML-Spezifikation definiert neben den Regeln für die Spezifikation eigener Markup-Sprachen auch das Verhalten von Anwendungen wie z.B. XML-Parser, die diese Dokumente verarbeiten sollen.

Ein XML-Parser ist immer Teil einer anderen XML-Anwendung und dient dazu, XML-Dokumente einzulesen, und den Zugriff auf die enthaltenen Daten und deren Struktur zu ermöglichen. Nach dem Lesen reicht er die enthaltenen Daten in Form einer Baumstruktur, die der Datenstruktur entspricht, an die XML-Anwendung weiter. Während des Parsens kann er prüfen, ob ein Dokument gültig (valid) und wohlgeformt (well formed) ist. Wertet der Parser während dieses Vorgangs eine Document Type Definition (DTD) aus, um die Korrektheit bzw. Gültigkeit eines XML-Dokuments zu prüfen, nennt man ihn validierenden Parser. Das geprüfte Dokument bezeichnet man dann als gültig und wohlgeformt. Prüft der Parser das Dokument nur auf die Einhaltung der allgemeinen XML-Syntax ohne die Grammatik einer DTD zu berücksichtigen, ist das Dokument nur wohlgeformt. Ist ein XML-Dokument nicht well formed, kann es nicht geparkt werden. Um auf die vom Parser in einer Baumstruktur aufbereiteten Daten zugreifen zu können, benötigen XML-Anwendungen eine Schnittstelle:

- *Document Object Model (DOM)* ist eine baumorientierte Schnittstelle, die den Zugriff auf die gesamte Baumstruktur ermöglicht. Nachteilig ist, daß XML-Dokumente vollständig geparkt sein müssen, bevor auf ihren Inhalt zugegriffen werden kann.
- *Simple API for XML (SAX)* wurde entwickelt, um durch eine ereignisorientierte Schnittstelle bereits während des Parsens über Ereignisse wie z.B. das Erreichen eines Anfangs- bzw. Ende-Markups infor-

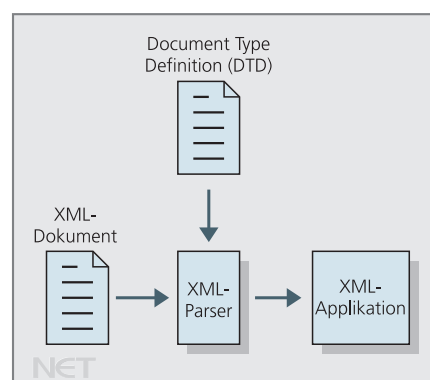
miert zu werden. Ein SAX-Parser arbeitet ein XML-Dokument immer sequentiell ab, wobei je vorgefundenem Element unterschiedliche Events ausgelöst werden.

Das XML-Dokument besitzt sowohl eine logische als auch eine physikalische Struktur. Aus logischer Sicht besteht es aus Deklarationen, Elementen, Kommentaren, Zeichenreferenzen und Verarbeitungsanweisungen (Processing Instruction), die innerhalb des Dokuments durch explizites Markup ausgezeichnet sind. Physikalisch besteht es aus einer Reihe von Dateneinheiten bzw. Knoten. Ein Knoten kann dabei auf andere Knoten verweisen, um diese in das Dokument einzubinden. Jedes Dokument beginnt mit einem Wurzel- oder Dokument-Knoten. XML-Dokumente müssen nicht, sollten aber immer mit einer XML-Deklaration beginnen, die die verwendete XML-Version spezifiziert. Der Verweis auf eine DTD bzw. die Definition von Elementen im Dokument ist nur dann zwingend erforderlich, wenn es auf Gültigkeit geprüft werden soll. Eine DTD ist ein Dokument, mit dem eine neue Metasprache definiert werden kann. Die DTD-Syntax ist nicht durch eine separate Spezifikation definiert, sondern Teil der XML-Spezifikation. Dabei ist die Deklaration von DTDs in Dokumenten bereits durch HTML bekannt.

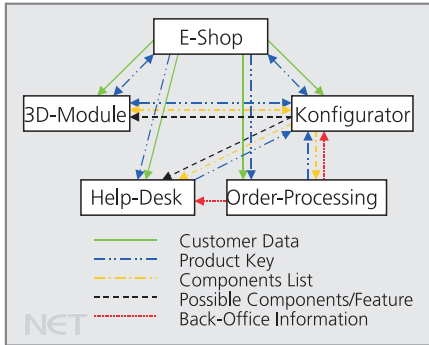
Eine DTD definiert die Struktur, wie sich Informationen in einem XML-Dokument zusammensetzen, welche Tags es in der neuen Metasprache geben soll, welche Werte für Attribute vorgegeben bzw. möglich sind, und wie die Tags ineinander geschachtelt sein können. Sie sollten so restriktiv wie möglich definiert werden, damit XML-Parser schneller die Gültigkeit eines XML-Dokuments prüfen können.

## Das Projekt INTELLECT

Das Forschungsprojekt INTELLECT ([www.ist-intellect.com](http://www.ist-intellect.com)) startete im Januar 2000 mit einer Laufzeit von zwei Jahren im Programm IST (Information Society Technologies) der Europäischen Union (EU). Sein Hauptziel ist das Zusammenführen vorhandener E-Commerce-Systeme und das Ent-



XML-Anwendungsarchitektur



Kommunikation zwischen den unterschiedlichen Modulen

wickeln einer Oberfläche, die die Produkte so realistisch wie möglich mit dazugehöriger Konfiguration abbildet. Die Projektteilnehmer kommen aus Frankreich, Deutschland, Österreich und Griechenland. Die Ergebnisse sollen sich dabei nicht nur auf eine bestimmte Produktart beschränken, sondern individuell anpaßbar sein, um auch für andere Industriesektoren relevant zu sein. Benutzerfreundlichkeit, einfacher Zugriff auf die Services, eigene Konfiguration der Produkte mit

integriertem Help-Desk-System, Sicherheit bezüglich Authentifizierung, Identifikation und Privatsphäre sowie eine hohe Qualität des Informationsmanagement werden dabei umgesetzt. Internationale Standards werden evaluiert, getestet und eingesetzt sowie neue Verfahren entwickelt.

INTELLECT entwickelt fünf Module, die autonom voneinander arbeiten können:

- E-Shop: Hauptbenutzerschnittstelle des Kunden, um mittels Enterprise Java Beans (EJB) und XML/XSL-Technik Ware konfigurieren und ansehen zu können;
- Virtual Reality (VR): 3D-Darstellung der Produkte mittels Virtual Reality Markup Language (VRML) mit Java-3D-Technik;
- Configurator: Intelligente Konfiguration der Produkte aus der Datenbank (Oracle) durch Hinzunahme neuer Komponenten auf Basis von Constraints;
- Order Processing: Automatisches Bestellwesen mit Anbindung an

Warenwirtschaftssysteme;

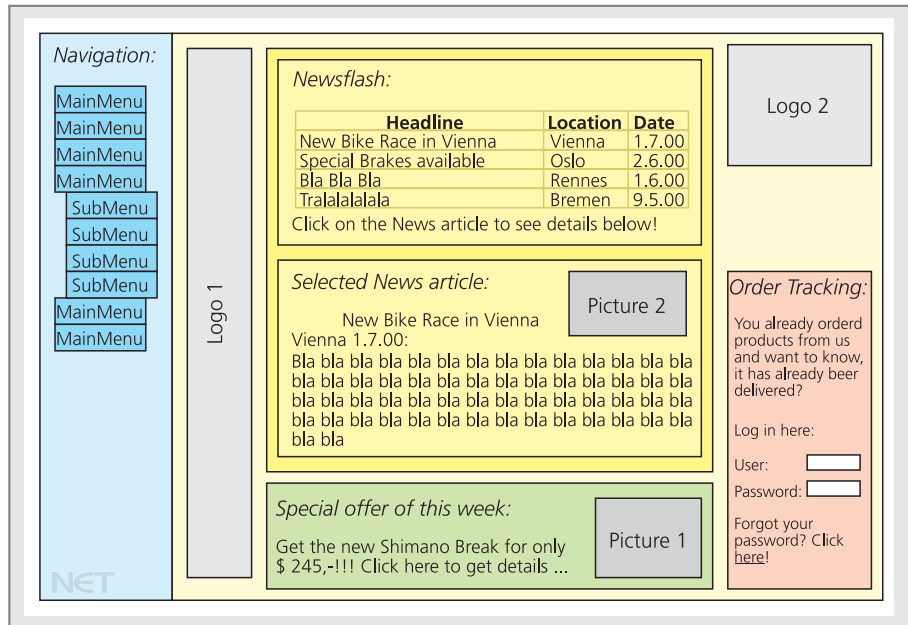
- Help-Desk: Direkte Unterstützung durch Mirroring-Mechanismen und Einbindung von Voice over IP (VoIP) mit Hilfe von Agenten.

Diese Module bilden das Gesamtsystem ab, können aber auch jeweils an andere Systeme angepaßt werden, wobei sie auf eine gemeinsame Produktdatenbank zurückgreifen. Dabei wird der Configurator als Front-end-System für die Datenbank genutzt, der E-Shop als Hauptschnittstelle für den Kunden und die Auftragsbearbeitung als Front-end-System für das Back-Office. Die Datenbank enthält alle Produkt- und Bestellinformationen, die auf verschiedene Gebiete und Module aufgeteilt werden. Die Informationen sind im Lieferantensystem verfügbar und werden kontinuierlich aktualisiert. Eine zweite Datenbank enthält Multimediadaten wie Sound, Video und 3D-Ansichten, um die Produkte realistisch darstellen zu können. Für die interaktive Konfiguration wird ein VR-Modul bereitgestellt.

Es bietet individuelle Konfiguration der Objekte, textbasierte und sprachanimierte Erläuterungen sowie Hinweise zu Auswahl, Design, Stil und Interoperabilität der einzelnen Objekte. XML wird in diesem Projekt an drei unterschiedlichen Stellen eingesetzt:

- **Softwaremodule:** Die Softwaremodule wurden unabhängig voneinander entwickelt. Die Kommunikationsschnittstellen zwischen ihnen wurden definiert, und der Datenaustausch zwischen ihnen findet über XML statt. Dadurch kann der Prototyp auf unterschiedliche Anforderungen schnell verändert werden. Die Entwicklung erfolgt parallel und somit effizienter.
- **Web-Publishing:** Die Oberfläche des Prototypen soll auf unterschiedlichste Produkte umsetzbar sein. Aus diesem Grund hat man die Web-Oberfläche in verschiedene Bereiche zerlegt, die alle als XML/XSP ausgeführt werden. Alle Informationen kommen aus der Datenbank und werden als XSL dargestellt.
- **Order Processing:** XML ermöglicht die Datenanpassung an unterschiedliche Medien. So können z.B. E-Mails in Excel-Tabellen umgewandelt werden und umgekehrt. Die Anbindung von weiteren Medien ist denkbar.

Durch den Einsatz von XML konnten die Entwicklungszyklen des Projektes heruntergefahren werden. Machbar ist auch eine schnelle Anpassung an Kundenanforderungen, da man unabhängig von den darzustellenden Produkten einzelne Teilbereiche sofort austauschen kann. Dies betrifft Visualisierungen genauso wie einzelne Module. Allerdings gibt es auch Nachteile: Die Web-Seiten lassen sich nicht mehr einfach erneuern, sondern müssen mit Hilfe der Datenbank und des Web-Servers verwaltet werden. Ein anderes Problem besteht in der Darstellung von XSL in heutigen Browsern. Um eine einheitliche Darstellung zu gewährleisten, wird Cocoon eingesetzt. Es nimmt eine Übersetzung in HTML vor, so daß mit jedem Browser die Seiten betrachtet werden können. Das bedeutet allerdings einen weiteren Zwischenschritt und somit eine zusätzliche Fehlerquelle.



XML-Design des Front-ends von dem E-Shop-System

### Fazit

Mit Hilfe von XML können Markup-Sprachen definiert werden, die sowohl syntaktische als auch semantische Anforderungen für einen bestimmten Dokumententyp wie z.B. HTML beinhalten. Sie werden mit Hilfe sogenannter Document Type Definitions (DTD) bzw. Schemata formuliert. Die so definierte neue Markup-Sprache mit ihren Tags stellt die Metadaten dar. Die Datenrepräsentation innerhalb einer Web- bzw. Softwareanwendung erfolgt in Form eines oder mehrerer XML-Dokumente. Aus ihnen lassen sich mit XSL verschiedene, vom Ausgabemedium abhängige Formate erzeugen. Die DTDs bzw. Schemata, auf denen XML-Dokumente basieren, müssen allen beteiligten Anwendungen zugänglich gemacht werden. Bislang hatte sich für den Datenaustausch über einheitliche Schnittstellen EDI etabliert. Im E-Commerce-Bereich werden z.B. Kundendaten oder Artikelbewegungen über EDI aktualisiert. EDI ist jedoch eine kostspielige und komplexe Lösung, so daß immer mehr XML-Lösungen statt oder für EDI eingesetzt werden.

Eine reine XML-Umsetzung bringt somit einige Vorteile mit sich, unabhängig davon, ob man sie im Web-Publishing oder zur Datenanpassungen vornimmt. Die Informationen, die in

XML- bzw. XSL-Dokumenten abgebildet werden, müssen allerdings persistent oder dynamisch sein. Daraus ergeben sich folgende Anforderungen:

- Persistente XML-Dokumente müssen geeignet abgelegt und verwaltet werden können;
- XML-Daten müssen aktualisiert bzw. erzeugt werden. Dafür wird eine XML-Anwendung benötigt, die unter Einhaltung der definierten Regeln (DTD bzw. XML-Schema) gültige XML-Dokumente generiert;
- Vor der Transformation in die geforderte Datensicht müssen den XML-Dokumenten die richtigen XSL-Dokumente zugeordnet werden;
- Werden unterschiedliche XSL-Dokumente für verschiedene Ausgabemedien definiert, müssen diese bei Anfragen richtig zugewiesen werden.

Damit können Web-Design und -Entwicklung völlig unabhängig voneinander durchgeführt werden. Das gleiche gilt für die Entwicklung unterschiedlicher Softwaremodule. Es werden XSL-Dokumente für unterschiedliche Datenansichten erzeugt, während Anwendungen mit Hilfe von XML-Parsern entwickelt werden können, die auf die Datenstruktur und die Daten selbst eingehen und diese verarbeiten. Diese Vorteile werden XML zu einer breiten Anwendungsbasis verhelphen. (bk)