

Die Anwendung von XML in E- und M-Commerce

Seminararbeit
im Fach Rechnerarchitektur

der FH Regensburg,
Fachbereich Informatik
bei Prof. Dr. Bernhard Kulla

Gerald Able
Studiengruppe I5IW
Matrikelnummer 2077243

Präsentation am 12. Dezember 2001

1. Inhaltsverzeichnis

1. INHALTSVERZEICHNIS	2
2. EINFÜHRUNG IN DAS THEMA UND KLÄRUNG DER BEGRIFFE	3
2. 1. E-Commerce	3
2. 2. M-Commerce	4
2. 3. XML	5
2. 3. 1. Beispiel und Grundkonzepte	5
2. 3. 2. Geschichte der Markup-Sprachen	6
2. 3. 3. Einordnung von XML und begleitenden Standards	7
2. 3. 4. XML syntaktisch betrachtet	7
2. 3. 5. DTD für die Festlegung einer Grammatik	9
2. 3. 6. Stylesheets und Transformationen	10
2. 3. 7. Namensräume	11
2. 3. 8. DOM als standardisiertes Datenzugriffsmodell	11
2. 4. XML als optimales Werkzeug für die Datenbeschreibung	12
3. XML/EDI	13
3. 1. Geschichte von EDI	13
3. 2. XML als Ausweg	13
3. 3. konkretes Anwendungsbeispiel	14
3. 4. weitere Möglichkeiten und Anbieter	15
3. 5. Vergleich klassisches EDI vs. XML/EDI	16
4. SOAP	18
4. 1. Geschichte und benutzte Protokolle	18
4. 2. Codebeispiel	19
4. 3. begleitende Standards	20
4. 4. Fallbeispiel	21
5. FAZIT	23
6. LITERATURVERZEICHNIS	24

2. Einführung in das Thema und Klärung der Begriffe

2. 1. E-Commerce

Electronic Commerce, kurz E-Commerce ist ein häufig zu hörender Ausdruck in Nachrichten und Werbung. Es finden sich dementsprechend auch viele Versuche, dieses Marketingwort mit Bedeutung zu füllen. Laut Definition bei /Webagency01a/ versteht man unter E-Commerce das Abbilden realer Geschäftsprozesse und das effiziente Management der gesamten Geschäftsbeziehung unter Zuhilfenahme elektronischer Mittel. Dies wird meistens durch das Internet und dessen Protokolle realisiert.

Um die geforderte Effizienz zu erreichen, wird versucht Medienbrüche bei der Kommunikation mit den Geschäftspartnern zu vermeiden. Innerhalb eines Unternehmens liegen die Informationen meist elektronisch in einem definierten Format vor. An den Organisationsgrenzen zeigen sich Probleme beim Informationsübergang. So ist der vollautomatische elektronische Informationsaustausch zwischen Unternehmen und zwischen Unternehmen und Verbraucher nicht die Regel, sondern eher die Ausnahme. Eine vollständige Implementierung des E-Commerce kann hier die Geschäftsabläufe wesentlich vereinfachen und beschleunigen.

Ein weiterer Aspekt des E-Commerce ist das Management der Geschäftsbeziehungen. Dem Kunden sollen alle notwendigen Informationen im Pre-Sales, während der Vertragsabwicklung und bei der Kundenbetreuung im After-Sales-Bereich bereitgestellt werden. Durch die optimale Erfüllung der Informationsbedürfnisse erreicht der Anbieter hohe Kundenbindung.

Als Beispiel für eine E-Commerce-Anwendung sei folgendes Szenario angeführt: eine Person sucht im Internet Informationen über einen Musiker und entdeckt dabei einen E-Commerce-Anbieter. Dieser könnte ihm ausführliche Informationen über den Musiker, wie seine Biographie oder bereits veröffentlichte CDs präsentieren. Die CDs bietet der Anbieter in seinem Webshop an. Um Transportkosten zu sparen besteht auch die Möglichkeit, sich die Musik direkt über das Internet auf den heimischen Rechner senden zu lassen. Die Bezahlung erfolgt per Lastschrift oder ein E-Payment-System. Auf diese Weise ist es möglich, alle Einzelprozesse über das Internet zu realisieren. Besucht der Käufer die Seite ein weiteres Mal, wird er womöglich mit seinem Namen begrüßt und ihm werden Bands präsentiert, die zum gekauften Produkt in Beziehung stehen. Damit erschließt sich die Bedeutung des E-Commerce als Marketinginstrument.

Dies war ein Beispiel des sog. Business-to-Consumer (B2C-) Bereichs. Dabei befinden sich privater Konsument und einer kommerzielle Firma in einer Geschäftsbeziehung. Im Gegensatz dazu steht der Business-to-Business- (B2B-) Bereich. Dieser beschreibt Geschäfte zwischen Firmen. Darüber hinaus existieren Modelle wie z. B. eBay, die Transaktionen unter den Privatpersonen koordinieren. Dies

lässt sich wohl am besten durch den Ausdruck Consumer-to-Consumer (C2C) beschreiben. Elektronische Kommunikationsformen hauptsächlich aus dem B2B-Umfeld stehen im Blickpunkt der vorliegenden Arbeit.

Es werden schnell die Vorteile des E-Commerce für Firmen und Kunden klar: die Firma kann bei Vertragsverhandlung, Verkaufsabwicklung und Betreuung nach dem Kauf Personal und damit Kosten einsparen. Für den Kunden ergeben sich Vorteile bei der Informationsbeschaffung bereits vor dem Kauf durch Preis- und Qualitätsvergleiche. Als nachteilig könnte sich die vollständige Abwesenheit menschlicher Berater erweisen, die einige Konsumenten schätzen. Für Firmen ist eine umfassende Analyse der und Neustrukturierung der Abläufe notwendig. Dadurch ergibt sich bei der Einführung von E-Commerce-Techniken ein teilweise erheblicher Realisationsaufwand, der sich erst mittel- bis langfristig rentiert.

2. 2. M-Commerce

Eine andere Ausprägung des E-Commerce stellt der Mobile Commerce, kurz M-Commerce dar. /Webagency01b/ definiert ihn als „Konzept zur Nutzung von Informations- und Kommunikationsmittel zur mobilen Integration und Verzahnung unterschiedlicher Wertschöpfungsketten oder unternehmensübergreifender Geschäftsprozesse und zum Management von Geschäftsbeziehungen“. Allgemeinverständlich ist M-Commerce also die Weiterführung des E-Commerce auf mobiler Basis. Dabei sind v. a. Anwendungen im B2C-Bereich zu sehen, also Bestell- und Bezahlssysteme für private Kunden. Auch innerhalb von Firmen sind z. B. in der Lagerhaltung Lösungen denkbar.

Dem M-Commerce wurde ein enormes Marktpotential bescheinigt. Das renommierte Marktforschungsinstitut Durlacher Research prognostizierte ein Marktvolumen von 50 Milliarden US-Dollar auf dem europäischen Markt. Es ist allerdings fraglich, ob die Versprechungen eingehalten werden. Zahlreiche Hindernisse stellen sich der neuen Möglichkeiten in den Weg: neben der anfangs schlechten Verfügbarkeit M-Commerce-fähiger Mobiltelefone stellt die erzielbare Bandbreite über das derzeit gebräuchliche GSM-Mobilfunknetz (Global System for Mobile Communication) einen Wehrmutstropfen dar. Einen Lichtblick könnte das UMTS-Netz (Universal Mobile Telecommunications System) darstellen, welches beim stationären Betrieb des Endgeräts maximal 2 MBit/s übertragen kann. Mit entsprechenden Realisierungen ist frühestens 2003 zu rechnen.

Als konkretes Beispiel zum M-Commerce sei eine bereits existierende Lösung genannt, die auch ohne solche Breitbandfunktionen möglich ist. In Frankfurt/Main kann man seine Taxifahrt per Mobiltelefon zu bezahlen. Der Taxifahrer ruft die Firma Paybox an und hinterlässt die Rufnummer seines Kunden und den zu entrichtenden Betrag. Paybox ruft den Kunden an und informiert ihn über Zahlungsempfänger und Betrag. Ist der Kunde einverstanden bestätigt er die Zahlung durch Eingabe einer PIN. Paybox zieht das Geld per Lastschrift vom Kunden ein und leitet es an das Taxiunternehmen weiter.

2. 3. XML

2. 3. 1. Grundkonzepte

Als effizientes Mittel zur Umsetzung der E- und M-Commerce - Ideen hat sich XML etabliert. Die eXtensible Markup Language dient als Grundlage zur Beschreibung von Daten. Als Einführung in die Sprache hier ein Ausschnitt aus einer XML-Datei, wie sie in Kapitel 2.1. beim Einkauf einer CD zum Einsatz gekommen sein könnte:

```
<CD>
  <Interpret>Macy Gray</Interpret>
  <Titel>The ID</Titel>
  <Veroeffentlichung>17. September 2001</Veroeffentlichung>
  <Preis Waehrung="EUR">12,77</Preis>
</CD>
```

Hier wird eine CD mit den Attributen Interpret, Titel, Veröffentlichungsdatum und Verkaufspreis beschrieben. Man hätte diese Aufgabe auch mit der bekannten Auszeichnungssprache HTML, der Hypertext Markup Language lösen können:

```
<p>
  <b>Macy Gray</b><br>
  The ID<br>
  17. September 2001<br>
  12,77 EUR
</p>
```

Jetzt ist klar, dass Macy Gray fett zu schreiben ist. Die physische Auszeichnung `` für bold legt das fest. Auch sollen die Daten jeweils in einer eigenen Zeile erscheinen. Andere wesentliche Informationen gehen jedoch verloren. Ohne nähere Erläuterung ist die Bedeutung des Datums, das im ersten Listing noch eindeutig als Erscheinungsdatum identifiziert werden konnte, nun nicht mehr ersichtlich. `<Veroeffentlichung>` fungiert hier als sogenanntes semantisches Tag, welches den Inhalt beschreibt. Darüber hinaus existieren logische Tags, die die Struktur eines Dokuments beschreiben. Ein Beispiel dafür ist `<head>`. Es definiert den Ort für Verwaltungsinformationen am Anfang eines HTML-Dokuments.

XML hat sich auf die Fahnen geschrieben Informationen einfach inhaltlich auszuwerten. Dazu benötigt es semantische und logische Tags. Um die Darstellung der Daten kümmern sich andere Techniken wie CSS oder XSLT. Beide werden später kurz erläutert. Grundkonzept von XML ist die Trennung von Inhalt, Struktur und Formatierung eines Dokuments.

2. 3. 2. Geschichte der Markup-Sprachen

Diese Idee stand im Vordergrund als 1950 Ted Nelson erste Überlegungen zu einem Hypertext-System anstellte. Menschenlesbare Informationen sollten miteinander in zwangloser Reihenfolge verknüpft werden. Doch lange Zeit wurde dieser revolutionärer Ansatz nur unzureichend genutzt. Erst mit dem beispiellosen Boom des Internet und dessen Lingua Franca HTML wurde Nelsons Konzept 1989 von einem jungen britischen Informatiker wiederbelebt. Tim Berners-Lee arbeitete im



Abbildung 1 - Ted Nelson

Kernforschungszentrum CERN in Genf. Er versuchte das bereits existierende Internet für seine Zwecke zu nutzen. Ergebnis war HTML als einheitliche Definition für die Übertragung von Text über das Internet. Anfangs war an den Einsatz von Grafik oder gar den heute nicht mehr wegzudenkenden Multimedia-Elementen nicht zu denken. Erst 1993 entwickelte Marc Andersseen beim National Center for Supercomputing Applications in Illinois den grafischen Webbrowser Mosaic, Vorläufer des Netscape Navigators. HTML liegt seit 1997 in der vom W3C, dem World Wide Web consortium zertifizierten Version 4.0 vor.

Doch mit fortschreitenden Versionen von HTML wurden immer mehr physische Auszeichnungen eingeführt. Auch konnten sich mit zunehmenden kommerziellen Erfolg der Auszeichnungssprache die Firmen nicht mehr auf einheitliche Standards einigen. Unvollständige Implementierungen von HTML in den einzelnen Browsern waren die Folge.

Eine Lösung des Problems schien SGML zu bieten. Es entwickelte sich aus der 1969 von Charles Goldfarb, einem Mitarbeiter von IBM, erarbeiteten Generalized Markup Language (GML). Die Standard Generalized Markup Language wurde 1986 von der International Standardisation Organisation (ISO) in ihrer Norm 8879 standardisiert. SGML konnte sich nie durchsetzen, weil es zu komplex für den alltäglichen Einsatz ist. Die Definition der Sprache hatte einen Umfang von 500 Seiten.

Diese entscheidende Schwäche führte zu einem „SGML Editorial Review Board and Working Group“ des W3C. Unter dem Vorsitz von Jon Bosak (Sun Microsystems) entwickelten Tim Bray, James Clark, Dave Hollander (Hewlett-Packard), Tom Magliery (NCSA), Jean Paoli (Microsoft) und viele andere XML als Teilmenge von SGML. Die Spezifikation bringt es gerade noch auf 33 Seiten. Dadurch gilt es als wesentlich leichter erlern- und einsetzbar. Die W3C Recommendation wurde am 10. Februar 1998 herausgegeben.

Zum einen wurde eine flexible Datenstruktur gefordert: Die Sprache muss erweiterbar sein und sich so den wechselnden Gegebenheiten und Anforderungen anpassen. Konvertierungen müssen weitgehend problemlos durchführbar sein. Damit könnte XML eine allumfassende Basis für die gesamte Daten- und Informationsverarbeitung innerhalb und zwischen Firmen sein. Zum andern war

die Plattformunabhängigkeit bei der Konstruktion der Sprache ein wichtiger Aspekt: Der Datenaustausch zwischen Rechnern unterschiedlicher Architektur muss problemlos möglich sein. Die Kommunikation findet deshalb mit 7 bit US-ASCII statt. Landesspezifische Sonderzeichen werden maskiert. Um eine Vielzahl von Anwendungen zu ermöglichen, wurde die Spezifikation ohne Lizenzkosten offengelegt.

2. 3. 3. Einordnung von XML und begleitender Standards

Der Themenkreis XML wird leichter verständlich, wenn man versucht die diversen Sprachen und Erweiterungen zu klassifizieren und die Beziehungen untereinander darzustellen. Folgende Grafik stellt die bisher gefallenen Abkürzungen zusammen:

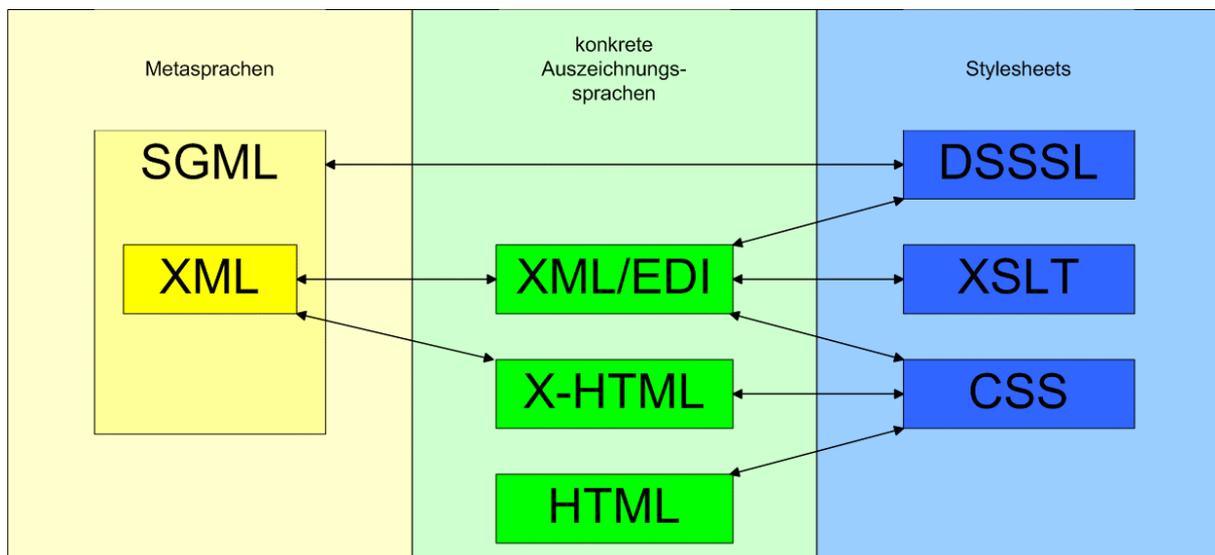


Abbildung 2 - Einordnung von XML und begleitenden Standards

Man erkennt links die Sprachen SGML und deren Teilmenge XML. Beide sind sogenannte Metasprachen, d. h. sie dienen als Konstrukt für die Definition von konkreten Sprachen. Als Beispiel hierfür sind in der Mitte XML/EDI, eine Sprache zur Kommunikation unter Firmen und X-HTML, die XML-konforme Ausgabe von HTML zu sehen. Um die Informationen zu visualisieren dienen Erweiterungen der Auszeichnungssprachen, die Stylesheets. SGML-Dokumente können mit DSSSL, der Document Style Semantics and Specification Language formatiert werden. Dokumente, die mit einer aus XML entwickelten Sprache geschrieben wurden, sind am flexibelsten zu gestalten. Sie können mit DSSSL, XSLT (Extensible Stylesheet Language for Transformations) oder CSS (Cascading Stylesheets) gestaltet werden. Webmaster werden CSS auch als Gestaltungsgrundlage für HTML-Seiten kennen.

2. 3. 4. XML syntaktisch betrachtet

Doch was macht eine Datei nun zu einem gültigen XML-Dokument? Im folgenden werden die syntaktischen Vorgaben für XML betrachtet:

1.gültiger Zeichenvorrat

XML-Dokumente werden im reinen Textformat erstellt. Die Zeichen werden in ISO/IEC 10646 definiert. Zeichen, die nicht vorkommen dürfen, aber trotzdem benötigt werden können mit Hilfe der Entity-Umschreibung benutzt werden. Dabei wird ein „Kaufmanns-Und“ (&) und eine abgekürzte Umschreibung des Zeichens angegeben. Alternativ ist die Angabe des ASCII-Codes erlaubt. Ein Semikolon beendet die Umschreibung. Ein Beispiel: der Umlaut „Ä“ entspricht folgenden Zeichenketten: „Ä“ oder etwas weniger intuitiv „Ä“.

2.Case-sensitivity

XML unterscheidet zwischen inhaltsgleichen Zeichenketten, je nachdem ob sie groß oder klein geschrieben sind. D. h. dass ein Tag <test> mit </test> und nicht etwa mit </Test> zu schließen ist. Dieses Verhalten stellt eine große Fehlerquelle dar. Man sollte deshalb auf durchgehende Schreibweise achten, um Fehler zu vermeiden.

3.nicht zu parsende Abschnitte

Wenn man Passagen des Dokuments von der Auswertung durch ein XML verarbeitendes Programm, genauer gesagt den XML-Parser, ausschließen will, so geschieht dies durch die Auszeichnung als CDATA-Abschnitt. Im nachfolgenden Code muss der Parser das Tag <fachbegriff> nicht kennen:

```
<![CDATA[
Was hier steht, interessiert den <fachbegriff>Parser</fachbegriff> nicht.
]]>
```

Den gleichen Effekt hätten übrigens folgende Codebeispiele, die den Abschnitt in einen Kommentar verstecken oder die Delimitierungszeichen umschreiben:

```
<!-- Was hier steht, interessiert den <fachbegriff>Parser</fachbegriff>
nicht. -->
```

```
Was hier steht, interessiert den
&lt;fachbegriff&gt;Parser&lt;/fachbegriff&gt; nicht.
```

4.Attribute

Zur genaueren Bestimmung von Tags ist es möglich zusätzlich Attribute anzugeben. Diese müssen grundsätzlich in Anführungszeichen gesetzt werden. Ob doppelte oder einfache ist dabei egal, solange öffnendes und schließendes identisch sind. Das nicht als Delimitierungszeichen eingesetzte Anführungszeichen ist als gültiges Zeichen im Attributwert erlaubt und muss nicht umschrieben werden.

```
<Price Currency="EUR">123,00</Price>
```

5. Grundgerüst der XML-Datei

Ein typisches XML-Dokument ist dreigeteilt: ein einleitender Prolog legt Version und Zeichensatz fest:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Wird ein Tag mit `<? . . . ?>` eingeleitet und geschlossen, so ist er eine Processing Instruction, eine Verarbeitungsanweisung. Processing Instructions die mit `<?xml` beginnen sind für die XML-Standarddefinition reserviert und sollten nicht für eigene Festlegungen benutzt werden. Es folgt die DTD, die Document Type Definition.

```
<!DOCTYPE dokument [  
<!ELEMENT daten (#PCDATA)>  

```

Die eigentlichen Nutzinformationen, die sich nach einigen Regeln halten müssen, um den Grundsätzen der Wohlgeformtheit und Gültigkeit zu entsprechen, schließen sich an. Wohlgeformtheit, oft ist auch der englische Ausdruck *wellformed* zu lesen, bedeutet, dass sich das Dokument vollständig an die offiziellen Regeln des W3C hält. Das Konsortium verlangt, dass es Prolog und mindestens ein Element, das sogenannte Wurzelement enthält. Dieses muss auch alle anderen Elemente umschließen. Alle unbedingt erforderlichen Attribute müssen angegeben werden und dabei dem definierten Typ und Wertebereich entsprechen. Eine sich überschneidende Anordnung der Tags ist nicht erlaubt.

Wenn eine zugehörige DTD existiert und verfügbar ist und das XML-Dokument in Bezug auf die darin enthaltenen Festlegungen korrekt formuliert ist, so wird es als „gültig“ bewertet.

Diese Definitionen genügen um abschließend das kleinstmögliche wohlgeformte und gültige XML-Dokument anzugeben:

```
<?xml version="1.0"?>  
<!DOCTYPE dokument [  
<!ELEMENT daten (#PCDATA)>  
<daten></daten>
```

2. 3. 5. DTD für die Festlegung einer Grammatik

Die DTD ist ein grundlegendes Element von XML. In ihr werden die erlaubten Elemente und Attribute der Datei festgelegt. Mit der DTD wird eine Validierung des Dokuments mit einem XML-Parser erst

möglich. Dabei kann die DTD nicht nur im XML-Dokument stehen, wie gerade demonstriert, sondern auch extern gehalten werden und von mehreren XML-Dateien referenziert werden. So lassen sich viele Dokumente mit identischen Regeln definieren. Als Attributwert von `standalone` im Prolog muss dann „no“ gesetzt werden.

```
<!-- Musik.xml -->
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE dokument SYSTEM "Musik.dtd">
<CD>
  <Interpret>Macy Gray</Interpret>
  <Titel>The ID</Titel>
  <Veroeffentlichung>17. September 2001</Veroeffentlichung>
  <Preis Waehrung="EUR">12,77</Preis>
</CD>

<!-- Musik.dtd -->
<!ELEMENT CD(Interpret, Titel, Veroeffentlichung, Preis)>
  <!ELEMENT Interpret      #PCDATA>
  <!ELEMENT Titel          #PCDATA>
  <!ELEMENT Veroeffentlichung #PCDATA>
  <!ELEMENT Preis          #PCDATA>
  <!ATTLIST Preis Waehrung (EUR|DM)>
```

Hier ist eine komplette Definition des Einführungsbeispiels angegeben. Die Datei Musik.dtd stellt hierbei die Regeln für Musik.xml auf. In der DTD werden die Elemente und deren erlaubter Inhalt definiert. #PCDATA (Parsed Character Data) steht für beliebige Zeichenfolgen. Der Preis kann aufgrund der Zeile `<!ATTLIST Preis Waehrung (EUR|DM)>` entweder in € oder DM als Attribut im Tag `<Preis>` angegeben werden. Eine Prüfung, ob im Preis nur Zahlen angegeben wurden ist nicht möglich!

2. 3. 6. Stylesheets und Transformationen

Wenn XML-Dokumente nur wohlgeformt sind, aber über keine zugehörige DTD verfügen, ist es möglich, mit einem bereitliegenden Stylesheet die Formatierungsanweisungen für den Webbrowser anzugeben. Für einfach strukturierte Dateien reicht diese Darstellung mit CSS bereits aus, wobei natürlich keine gültigen XML-Texte entstehen. Eine Formatierung mit XSLT oder DSSSL ist clientseitig noch nicht möglich, zumindest für normale Webbrowser. Jedoch kann dies serverseitig geschehen. Der Browser empfängt dann normales HTML. Dabei werden aber bei jedem Abruf einer Dokuments die Quelltexte mit dem Stylesheet verarbeitet, gespeichert und zum Browser geschickt; ein Prozedere, dass v. a. bei umfangreichen Texten Zeit und Systemressourcen beansprucht. Mit fortschreitender

Browserentwicklung ist daher damit zu rechnen, dass diese Arbeit auf dem Client verrichtet werden wird.

Die Möglichkeit aufgrund von XSLT-Anweisungen XML-konforme Dokumente in andere umzuwandeln ist essentiell für die Integration fremder Datenformate. Sie erleichtert die Migration auf neue Systeme und die Kommunikation mit noch nicht auf einen einheitlichen Standard umgestellten.

2. 3. 7. Namensräume

Es können im XML-Dokument mehrere DTDs referenziert werden. Kommt es dabei zu doppelt definierten Elementen ist der Einsatz von Namensräumen (auch Namespaces genannt) angebracht. Namespaces sind nicht Bestandteil der XML-Definition, basieren jedoch darauf.

2. 3. 8. Document Object Model und Schemata

Auf die DTDs in allen Einzelheiten einzugehen würde den Umfang dieser Arbeit sprengen. Abschließend sei jedoch noch angesprochen, dass DTDs keine XML-Dokumente sind. Deshalb kann das Document Object Model (DOM) auf sie nicht angewendet werden. Das DOM ermöglicht den standardisierten Zugriff auf einzelne Elemente des Dokuments. Bei XML schon vielfach angewendet wäre bei der Verarbeitung von DTDs eine entscheidende Erleichterung. Aus diesem Grund wurden die sogenannten Schemata entwickelt. Sie sind XML-konform. Weiterhin bieten sie einige Neuheiten, wie die Unterstützung erweiterter Datentypen, ein konsistentes Vererbungskonzept und erweiterte Strukturspezifikationen (z. B. minimales oder maximales Auftreten eines Elements). Die Möglichkeit äquivalente Felddefinitionen (z. B. <Price> = <Preis>) oder offene und/oder unvollständige Definitionen zu erstellen sind jedoch möglicherweise kontraproduktiv, da sie die Formalismen von XML teilweise aushebeln.

2. 4. XML als optimales Werkzeug für die Datenbeschreibung

Jon Bosak, einer der Urväter von XML erkannte schon 1997 die Möglichkeiten seiner Sprache im Bezug auf die Organisation von Unternehmensvorgängen. In /Bosak97/ führt er an, dass sich XML gut zur variablen Darstellung von Informationen eignet, da streng auf die Trennung von Struktur, Inhalt und Layout des Dokuments geachtet wird. So können nur die für das jeweilige System sinnvollen Daten angezeigt werden (z. B. eine hochaufgelöste Version eines Produktkatalogs für eine Druckerei und eine minimalistische Version für WAP. Beide werden aus dem gleichen Datenbestand gespeist, der als XML-Datei vorliegt).



Abbildung 3 - Jon Bosak

Außerdem ist es leicht möglich, diesen Datenbestand zu durchsuchen. Softwareagenten sollen zukünftig selbstständig die Bedeutung von XML-Daten verstehen können und dem Anwender die Ergebnisse von Abfragen oder Vergleichen leichtverständlich visualisieren können. Ein Ansatz zur Durchführung dieses Szenarios wird im vierten Kapitel beschrieben.

Bosak schlägt weiterhin vor, XML als standardisiertes Datenaustauschformat zu nutzen. Dokumente, wie z. B. Rechnungen, Bestellungen oder Produktbeschreibungen sollen auf der Basis von XML beschrieben werden. Dies führt zu sogenannten XML/EDI Lösungen, die im nächsten Kapitel näher betrachtet werden.

3. XML/EDI

3. 1. Geschichte von EDI

(vgl. /Buxmann01/)

In den sechziger Jahren forschten einige Unternehmen an einer Möglichkeit der Inter-Enterprise-Kommunikation. Die Anstrengungen führten zum EDI-Standard (Electronic Data Interchange), der den Austausch von elektronischen Geschäftsdaten in standardisierter Form ermöglicht. Dabei geschieht die komplette Kommunikation ohne den Einsatz von Papier, sondern vollständig digital. Umfangreiche Vorteile, darunter Kosteneinsparungen, beschleunigte Geschäftsprozesse oder Just-in-time-Produktion sind theoretisch die Folge. Bereits früh wurde jedoch die Idee von einer Vielzahl von unterschiedlichen Implementierungen geplagt, da jedes Land, jede Branche und jedes Unternehmen andere Bedürfnisse und Vorstellungen von dem zu benutzenden Vokabular hatte. So entwickelten sich z. B. VDA für die Automobilindustrie, SWIFT für Banken, DAKOSY für die Transportbranche oder ANSI ASC X12 in den USA. Um das Problem der Kompatibilität zu lösen reifte unter Aufsicht der Vereinten Nationen und der ISO ein internationaler und branchenneutraler Standard namens UN/EDIFACT (EDI for Administration, Commerce and Transport) heran. Unter dem Mantel dieser ISO Norm 9735 entstanden branchenspezifische Subsets. Doch auch diese Lösung ist nicht besonders verbreitet. Nur etwa fünf Prozent der Zielgruppe für die ein Einsatz sinnvoll wäre nutzen EDI. Das hängt mit teilweise erheblichen Kosten für Einrichtung und Betrieb des Systems zusammen. Viele inkompatible Standards führten zudem zu großer Verunsicherung. So waren trotz UN/EDIFACT noch immer spezielle Implementierungsabsprachen zwischen den teilnehmenden Unternehmen nötig. Die damit geschaffenen Abhängigkeiten waren nicht erwünscht und verhinderten Flexibilität. Darüber hinaus zieht EDI die Einrichtung eines VANs (Value Added Networks) hinter sich, um die EDI-Nachrichten zu übermitteln. Billiger wäre es, die Kommunikation über das Internet abzuwickeln. Als Datenformat für den strukturierten Dokumentenaustausch über das Internet bietet sich XML als Ausweg geradezu an.

3. 2. XML als Ausweg

XML definiert keine Inhalte, sondern nur Sprache zur Beschreibung von Dokumenten, deshalb ist die Implementierung aller EDI-Standards innerhalb XML möglich. Eine Vielzahl an Standardisierungsinitiativen hat sich gebildet, um die mittlerweile ca. 250 verschiedenen E-Business-Vokabulare zu einen. Ein Vertreter der Integration von Anwendungssystemen entlang unternehmensübergreifender Wertketten ist das XML/EDI-Framework. Es existiert bereits seit 1997 und versucht traditionelles EDI mit vollständiger Rückwärtskompatibilität in die XML-Welt zu überführen.

3. 3. konkrete Anwendung

Kunden sollen ihre Rechnungsdaten von einem Webserver manuell oder automatisch herunterladen können. Die Rechnungen sind dabei in XML codiert. Kundenspezifische XML-Formate sind möglich, d. h. jeder Kunde kann sein individuelles Format und spezifische Stylesheets nutzen. Voraussetzung dafür ist auf Seite des Kunden, der Client-Seite mindestens ein XML-fähiger Webbrowser, z. B. der Internet Explorer von Microsoft ab der Version 5.0. Damit kann er die Daten manuell abrufen. Für den automatischen Rechnungsabruf benötigt man zusätzlich eine Java-2-Unterstützung, die Java virtual machine. In diesem Fall sind hervorragende Möglichkeiten zur Integration gegeben. Rechnungsdaten können per Java-Applikation heruntergeladen und sofort im eigenen Inhouse-System weiterverarbeitet werden. Menschliche Interaktion ist theoretisch nicht mehr erforderlich.

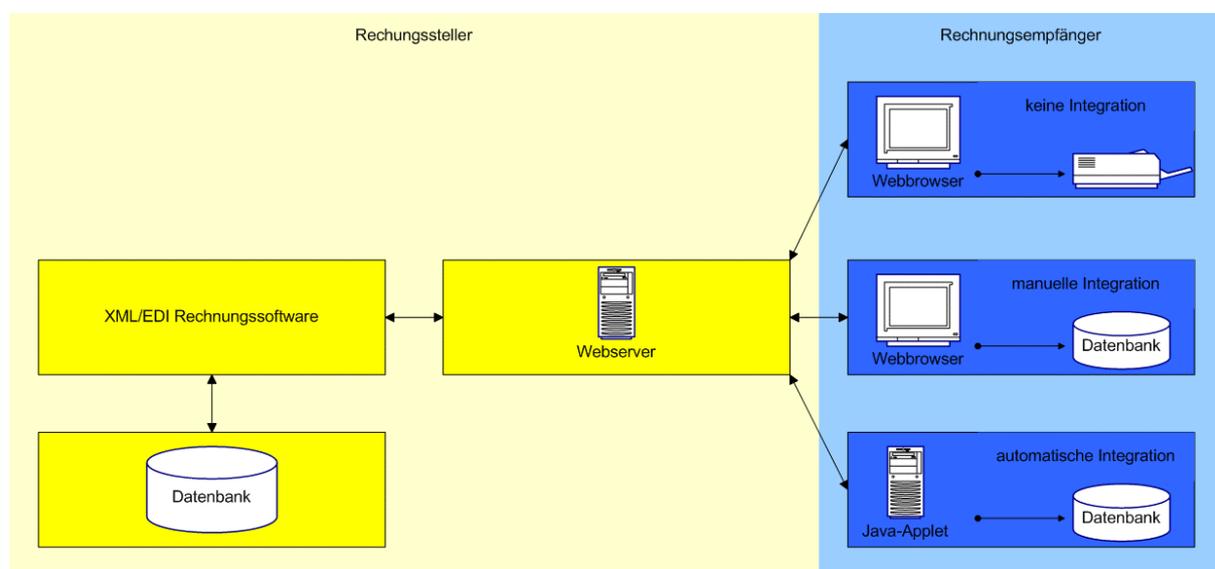


Abbildung 4 - Abläufe und Integrationsmöglichkeiten

Auf der Seite des Rechnungsstellers, der Server-Seite ist ein Webserver mit Java Servlet API erforderlich. Eine relationale Datenbank dient als Datengrundlage für die XML/EDI Rechnungssoftware. Diese besteht aus folgenden Komponenten: Ein Konfigurationseditor stellt Möglichkeiten zur Definition von Elementen, Struktur und Inhalten der Rechnungen bereit. Die integrierte Benutzerverwaltung ermöglicht eine Zuordnung der individuellen Konfiguration zum betreffenden Kunden. Eine Protokollfunktion hält alle Zugriffe des Empfängers fest.

Auf Client-Seite ist zwischen den möglichen Fällen zu unterscheiden: in den ersten beiden Möglichkeiten wird über einen üblichen Webbrowser zugegriffen. Die Authentifizierung erfolgt über eine eindeutige User-ID und Passwort Kombination. Der Anwender kann nun zwischen den bereitgestellten Rechnungen wählen und diese unter Einbeziehung des Stylesheets herunterladen. Jetzt kann sie gedruckt oder in das Inhouse-System eingegeben werden. Im Falle des automatischen Zugriffs kümmert sich um das Herunterladen eine Java-Applikation. Die gewonnenen Daten werden ohne weiteres Zutun in die Datenbank des Kunden eingespeist.

Technisch wird diese Lösung über eine Basisklassenbibliothek, ein Administrationsmodul und ein Webservermodul realisiert, die auf dem Server laufen. Die Basisklassenbibliothek enthält wichtige Methoden zur Datenbankanbindung und die Geschäftslogik. Das Administrationsmodul nutzt die bereitgestellten Klassen und wird als Java-Applikation realisiert. Die Kommunikation läuft hierbei mittels Methodenaufrufe der Objekte ab. Client und Server kommunizieren über SSL-verschlüsselte HTTP-Verbindungen. Auf Client-Seite ist zusätzlich die Java-Applikation für die automatische Integration erforderlich, falls gewünscht. Durch eine Konfigurationsdatei muss ihr die Struktur der Kunden-Datenbank mitgeteilt werden. Mit Hilfe der DOM-API liest die Applikation das erhaltene XML-Dokument aus und trägt es in die Datenbank ein.

Die eben vorgestellte Lösung ist unter dem Namen Seals InvoiceXchange bereits käuflich erhältlich. Lufthansa AirPlus, eine Tochter der Deutschen Lufthansa AG benutzt das Programm folgendermaßen: Anfangs liegen die Rechnungssätze im proprietären Format LARS vor. Diese werden nach XML gewandelt und in einer Datenbank gespeichert. Dabei kommt der Invoice-Standard von xCBL zur Anwendung. Jetzt ist die Distribution nach dem oben vorgestellten Modell möglich, jedoch wird die Authentifizierung per Smartcard und digitalem Zertifikat realisiert. Zusätzlich bietet InvoiceXchange weitere Exportmöglichkeiten: EDIFACT, SAP IDOCs (für R/3), CSV und serverseitiges HTML sind implementiert.

3. 4. weitere Möglichkeiten und Anbieter

Das Fallbeispiel InvoiceXchange führt die Vorteile von XML/EDI leicht verständlich vor. Doch nicht nur die Rechnungsstellung ist möglich, es lassen sich sämtliche Prozesse innerhalb einer Supply-Chain modellieren. Hierfür ist das SIMPLEX-Projekt zu nennen. Damit lässt sich die gesamte Bedarfs-, Produktion-, Beschaffungs- und Distributionsplanung unterstützen und folgende Dokument-Arten austauschen: Vorhersagen, Kataloganfragen, Lageranfragen, Bestellungen (Kauf- und Verkaufsoperationen, Reklamationen), Bestellbestätigungen, Lieferpläne, Monitoring Transaktionen, Lieferbestätigungen sowie –benachrichtigungen und selbstverständlich auch Rechnungen.

Microsofts Beitrag zum Thema stellt BizTalk dar. Mit BizTalk wird ein Rahmenwerk beschrieben, das den Aufbau der gültigen XML-Dokumente festhält. Es ist möglich bei www.biztalk.org eigene Schemata einzureichen, über die ein Verzeichnisdienst Auskunft gibt. Der Fokus von BizTalk liegt aber in der Kommunikation über Microsoft-eigene Server, die sich um Dinge wie Formatmapping und Validierung der Dokumente kümmern.

RosettaNet ist eine Initiative von Unternehmen der IT-Branche (u. a. AMD, Cisco und Intel) /vgl. Frank01/. Ziel ist es das Supply Chain Management zu vereinfachen und eine gemeinsame Infrastruktur für die Entwicklung und den Vertrieb von Soft- und Hardwareprodukten zu erstellen. Das Modell schreibt einige Vorgehensweisen vor: die Analyse der betroffenen Geschäftsprozesse, die

Entwicklung zu gewünschten Sollprozessen und die Identifikation von Schnittstellen zwischen den Prozessen. Ergebnis sind PIPs (Partner Interface Processes), die mit UML (Unified Modelling Language) und OCL (Object Constraint Language) zu visualisieren sind. Diese PIPs dienen als Grundlage für die Inhalte der zu versendenden Nachrichten. Diese sollen mit XML-DTDs definiert werden. Auch der Einsatz von Schemata ist im Gespräch. Im Einsatz ist RosettaNet bisher nicht.

Auch für ebXML existiert bisher nur ein Anwendungsszenario, benutzt wird es noch nicht. Die Zielsetzung von ebXML geht noch weiter: die Schaffung eines einzigen globalen, elektronischen Markts ist angestrebt. Hierbei liegt der Schwerpunkt auf der Modellierung von B2B-Prozessen, wobei kleine und mittelständische Betriebe und Unternehmen in Entwicklungsländern auch beteiligt werden sollen. Neben diesen hehren Zielen existiert leider noch kein genaues Konzept. Sicher scheint lediglich, dass die Kommunikation mit Hilfe von zentralen Repositorien, die allen am Prozess beteiligten Instanzen den Zugriff auf identische und aktuellste Daten erlauben. Ob ebXML jedoch jemals Erfolg haben wird ist fraglich, da noch keine Unterstützung seitens der Industrie bekundet wurde.

Ein Beispiel für ein bereits im Produktionsbetrieb benutztes System ist BMEcat. Der Bundesverband Materialwirtschaft Einkauf und Logistik e. V. (mit klingenden Namen wie Alcatel, BMW, DaimlerChrysler, Siemens, der Deutschen Telekom oder dem Fraunhofer IAO) hat dem Ziel seines System enge Grenzen gesetzt. Sinn und Zweck von BMEcat ist die Erstellung von Produktkatalogen. Dabei wurde den Informationsbedürfnissen der Kunden großer Umfang zugestanden. Dokumenttypen, die Dinge wie Bestellungen, Auftragsbestätigung und Rechnungen beschreiben sind in Planung.

Die DTD `bmecat_base.dtd` regelt grundsätzliche Eigenschaften eines Katalogs. Mögliche Aktionen sind das Anlegen eines neuen Katalogs, das Durchführen einer Produktaktualisierung oder das Durchführen einer Preisänderung. Die Elemente `T_NEW_CATALOG`, `T_UPDATE_PRODUCTS` und `T_UPDATE_PRICES` geben Zweck eines Dokuments an. Im zwingend vorgeschriebenen Header werden Eigenschaften zum Katalog und Verkäufer, u. U. auch zum Käufer angegeben.

3. 5. Vergleich klassisches EDI vs. XML/EDI

Zum einen ist ein Einstieg in die Welt von XML/EDI relativ einfach möglich, auf teure Soft- und Hardware kann verzichtet werden (auf Clientseite ist nur ein kostenlos erhältlicher, XML-fähiger Browser notwendig). Die vielfachen Exportmöglichkeiten sind durch Nutzung des DOM und XSLT leicht zu realisieren. Analyseprogramme sind leicht implementierbar. Dokumente können flexibel an die Bedürfnisse von Rechnungssteller und -empfänger angepasst werden. Zudem ist XML ein Metastandard und ermöglicht Abbildung von verschiedenen EDI-Standards. Dadurch lässt sich die Einführung der neuen Technik übergangslos bewerkstelligen. Schließlich sorgt XML/EDI für stark verringerte Kommunikationskosten durch die Nutzung des Internets anstatt VANs. Doch diese

Aspekte dürfen nicht über einen Nachteil hinwegtäuschen: der geringe Erfahrungsschatz und die Tatsache, dass viele Standardisierungen noch nicht abgeschlossen sind führen zu gewisser Unsicherheit. Trotzdem wird XML/EDI sich gegenüber traditionellen EDI-Lösungen durchsetzen, vor allem aus finanziellen und praktischen Gründen.

4. SOAP

4. 1. Geschichte und benutzte Protokolle

Im Zusammenhang mit Online-Systemen fällt oft der Begriff der Web Services. Ein bekanntes Beispiel für diese Art von Anwendung ist Microsoft Hailstorm. Der Ansatz ist dabei die zentralisiertere Speicherung und Zugriffsmöglichkeit auf Benutzerdaten. Dadurch könnte sich die Benutzung des Internets für den Benutzer ohne Vorkenntnisse erheblich erleichtern. Leider erhält Microsoft im Gegenzug unvorstellbare Möglichkeiten an Daten seiner Kunden zu kommen, sie zu analysieren und für eigene Zwecke zu nutzen. Microsoft bietet u. a. folgende Module an, die von jedem Gerät, das an das Internet angeschlossen ist bedienbar sind: myAddress, myProfile und myLocation zur persönlichen Identifikation und Beschreibung, myInbox und myContacts zum Bereitstellen von E-Mail-Funktionalität, myCalendar und myNotification zur Terminplanung, myDocuments, myApplicationSettings und myFavoriteWebSites. Dabei führen myServices und myUsage Buch über die genutzten Dienste.

Durch Schnittstellen zu HailStorm-kompatiblen Systemen können anderen Firmen wie z. B. E-Commerce Anbietern ihre Benutzerverwaltung teilweise ausgliedern und damit Anmeldung mit Microsoft Passport und Abrechnung mit Microsoft Wallet zu bewerkstelligen. Dies ermöglicht dem E-Commerce-treibenden erhebliche Kosteneinsparungen und die Bereitstellung besserer Lösungen, die er selbst nur schwerlich erbringen könnte.

Web Services sind eine spezielle Form verteilter Systeme, welche über die eigenen Organisationsgrenzen reichen. Ein Vertreter verteilter Systeme sind Remote Procedure Calls (RPC). XML-RPC macht es möglich diese Aufrufe über das Internet zu verschicken. Mit Servernamen, Portnummer und Methodename wird eine Funktion adressiert und über HTTP angesprochen. Verwendete XML-Schemata sind jedoch nicht standardisiert, daher sind die darauf entwickelten Anwendungen schlecht wartbar. Weitere Nachteile sind geringe Effizienz und aufwendige Datencodierung mit umfangreichen Nachrichten.

Leider unterstützen RPC keine moderne Techniken wie z. B. Objekte. Daher wurde auf Unix-Seite die Common Object Request Broker Architecture (CORBA) und auf Microsoft-Seite das Component Object Model (COM+) bzw. das Distributed Component Object Model (DCOM) entwickelt. Aber auch diese Lösungen führen zu Problemen. Ein direktes Kommunizieren beider Plattformen ist ohne spezielle Mittlersoftware nicht möglich.

Eine Lösung stellt SOAP, das Simple Object Access Protocol, als Protokollsuite mit offengelegten, weitverbreiteten Standards HTTP, XML, SSL u. a. dar. Es wurde am 8. Mai 2000 vom W3C in der Version 1.1 standardisiert. Die Reihe von Mitstreitern und Unterstützern ist beachtlich: Microsoft, IBM, die Apache Organisation, Sun Microsystems, ONE, Brazil, Borland, u. v. m. fördern SOAP. Es ist zu

erwarten, dass SOAP und die angrenzenden Protokolle einen weitgehenden Einfluss auf die Entwicklung des Internets haben. Anwendungen sollen in Zukunft nicht mehr auf PCs installiert, sondern von verschiedenen Servern aus dem Internet bezogen und zu Applikationen verschmolzen werden.

Das Protokoll wurde bewusst einfach gestaltet (vgl. /Develop01/): als Transportprotokoll wird HTTP benutzt, aber auch andere Protokolle wie SMTP sind möglich. Die Präsentationsschicht wird mit XML gestaltet. Daraus lassen sich folgende Vorteile erkennen: SOAP ist relativ einfach plattformunabhängig zu implementieren, da es auf offenen Standards basiert. Sicherheitsbedenken können durch den Einsatz von Secure Socket Layer (SSL) - gesicherten Übertragungen zerstreut werden. Firewalls werden durch den Einsatz von HTTP als Transportschicht auf TCP/IP Port 80 und dem Content-Type (text/xml-SOAP) unterstützt. Darüber hinaus ist das Verfahren gut skalierbar: werden die Anforderungen an einen einzelnen Webserver zu groß, ist es möglich Webfarmen und Load Balancing einzusetzen.

4. 2. Codebeispiel

Der Aufbau einer Kommunikation mittels SOAP wird bei /W3Schools01/ kurz beschrieben: Mit einem Frage und Antwortspiel durch XML-codierte SOAP-Messages werden die Nachrichten übertragen. Die einzelne Message ist durch einen Envelope umrahmt (Zeile 1-8, s. u.). Darin ist ein Header optional möglich. Er dient dazu Eigenschaften festzulegen ohne sie vorher mit den kommunizierenden Partnern auszuhandeln. Ein Body ist obligatorisch vorgeschrieben (Zeile 2-7). In einer SOAP Nachricht kann auch ein Weg definiert werden, der angibt welche Zwischenstationen eine Nachricht passieren muss, bevor sie zum Bestimmungsknoten gelangt. Eine SOAP Nachricht darf keine DTD und keine Information darüber, wie die Nachricht verarbeitet werden soll, enthalten. Statt der DTD werden die Namespaces envelope und encoding verwendet. Die genauen Festlegungen finden sich hier: <http://schemas.xmlsoap.org/soap/envelope/>

Hier ein einfachstes Code-Beispiel bei dem der Preis von IBM-Aktien abgefragt wird:

```
1: <SOAP-ENV:Envelope>
2:   <SOAP-ENV:Body>
3:     <xmlns:m="http://www.stock.org/stock" />
4:     <m:GetStockPrice>
5:       <StockName>IBM</StockName>
6:     </m:GetStockPrice>
7:   </SOAP-ENV:Body>
8: </SOAP-ENV:Envelope>
```

Wichtigstes Element im obenstehenden Listing ist der Methodenaufruf in Zeile 4-6. Eine Antwort des Servers könnte so aussehen:

```

1: <SOAP-ENV:Envelope>
2:   <SOAP-ENV:Body>
3:     <xmlns:m="http://www.stock.org/stock" />
4:     <m:GetStockPriceResponse>
5:       <Price>34.5</Price>
6:     </m:GetStockPriceResponse>
7:   </SOAP-ENV:Body>
8: </SOAP-ENV:Envelope>
    
```

In Zeile 4-6 liefert das Codebeispiel die Antwort auf die Anfrage. Im Fehlerfall würde der Server eine ähnlich aufgebaute Fehlermeldung zurückschicken, ein sogenanntes „Fault element“. In einer SOAP-Message kann nur ein Fault element stehen, und zwar im Body.

4. 3. begleitende Standards

Im Zusammenspiel der einzelnen Komponenten spielen noch weitere Protokolle eine Rolle. So müssen die abgebotenen Dienste auch von anderen Rechner aufgefunden werden. Dazu dient Universal Description, Discovery and Integration (UDDI) als Verzeichnisdienst, in seiner Funktion vergleichbar mit einem Domain Name Server (DNS) – System.

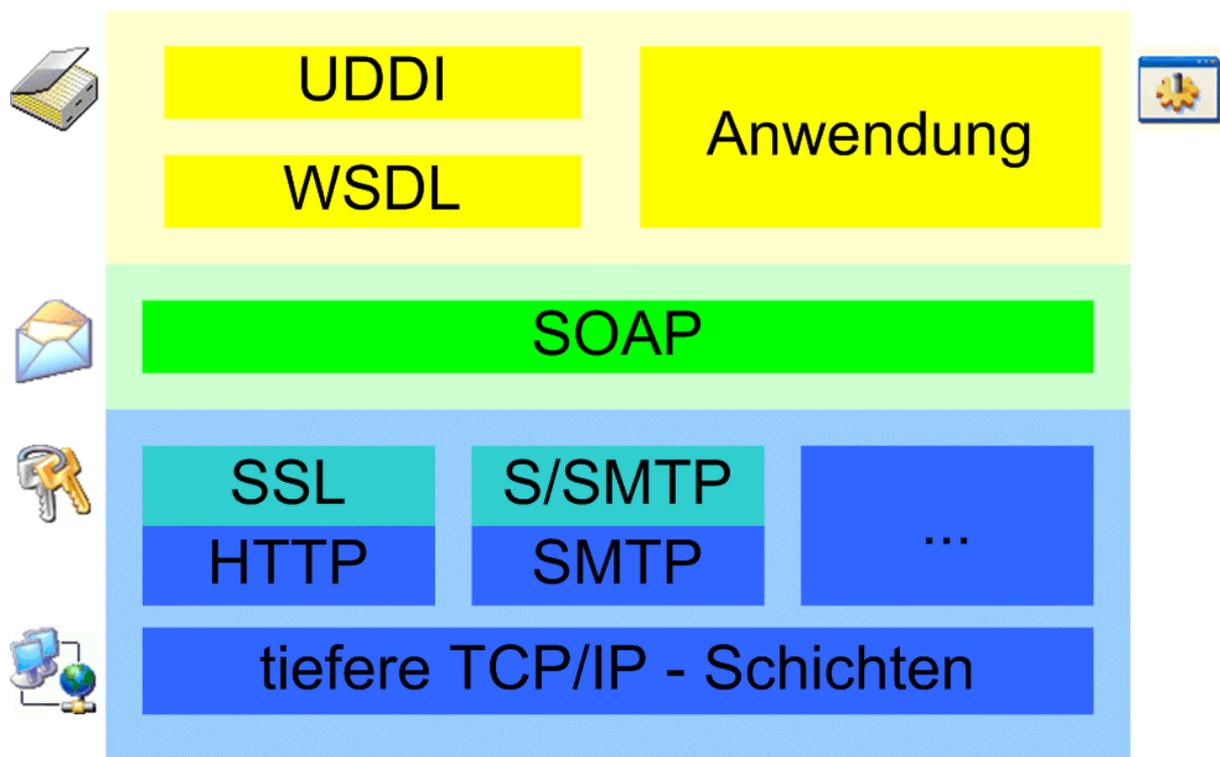


Abbildung 5 - die SOAP Protokollhierarchie

Softwarefirmen, Programmierer oder Standardisierungsgremien tragen in der UDDI Business Registry (UBR) verschiedene Typen von Web Services ein. Die bereitstellenden Firmen tragen in der Registry nun ihre angebotenen, konkreten Services ein, die von der UBR mit einem eindeutigen Schlüssel versehen wird. Nun können Suchmaschinen oder Anwendungsprogramme die UBR abfragen und erhalten die Spezifikationen für den gewünschten Dienst als Ergebnis. Nun können die Programme mit dem Dienst eine direkte Verbindung herstellen.

UDDI besteht aus den Service Type Registrations (dem eindeutigen Schlüssel und einem Verweis auf eine Definition des anzuwendenden Namespaces) und drei Abstraktionsebenen. Die White pages stellen allgemeine Informationen über das anbietende Unternehmen bereit. Yellow pages katalogisieren und kategorisieren das Unternehmens und grenzen damit das Umfeld der Anwendung schon etwas genauer ein. Schließlich bieten die sogenannten Green pages eine technische Beschreibung der angebotenen Dienste. Als Standard hierfür wird die Web Service Description Language (WSDL) genutzt. Auch WSDL ist ein XML-Format und stellt mit seinen Schnittstellendefinitionen quasi den „Kit“ zwischen den einzelnen Bauteilen eines Web Angebots dar. Wiederverwendbarkeit der Definitionen ist durch den modularen Aufbau möglich. Momentan sind die WSDL-Methodenbeschreibungen meistens noch zu ungenau, Ziel ist es jedoch sie so exakt festzuschreiben, damit dynamisch und während der Laufzeit der Web Applikation integrierbare Module möglich werden.

SOAP bietet Schnittstellen zur Suche und Manipulation der UDDI-Records. Die Datensicherheit wird durch Authorisierungsschlüssel realisiert.

4. 4. eine Anwendung aus der Praxis

Zur besseren Verdeutlichung soll hier ein kleines Beispiel dienen: Angenommen, eine Person will eine Reise unternehmen. Ein Agent auf der Internetseite seines Reisebüros soll folgende Ressourcen beschaffen: Hotelzimmer, Mietwagen und Flug. Nach einer Abfrage von relevanten Eigenschaften wie Gebiet, Zeitraum, maximale Kosten, Ausstattung, Anzahl der Mitreisenden und anderen Attributen soll sich der Agent im Netz auf die Suche machen. Dabei wird er einen UDDI-Dienst anfragen und mit den zurückgelieferten Web Services Kontakt aufnehmen und sie unter Zuhilfenahme der WSDL-Schnittstellendefinitionen abfragen. Nach der Präsentation der gefundenen Möglichkeiten soll der User auswählen können. Zum Abschluss des Vorgangs soll der Agent die Einheiten reservieren und die Bezahlung per Kreditkarte abwickeln.

Leider ist ein solcher Agent noch nicht realisiert, da er mit einigen grundlegenden Schwierigkeiten zu kämpfen hat. Ein Hauptproblem ist das nicht einheitliche Interface zu Auskunfts- und Buchungssystemen. Hier müssen noch Nachbesserungen am WSDL-Standard vorgenommen werden, um eine bessere Interoperabilität zu gewährleisten. Noch müssen die Abfragen manuell und durch individuelle Programmierung angepasst werden, falls neue Datenquellen benutzt werden sollen. Dies nimmt der Lösung jene Flexibilität, die sie verspricht.

Weitere Probleme stellen sich in den Weg, falls eine der Aktionen nicht zum Erfolg führte (z. B. konnte kein passendes Zimmer gefunden werden), aber die anderen Aktionen schon durchgeführt wurden (also der Flug schon gebucht wurde). Es sind also momentan keine verteilten Transaktionen möglich. Auch hier müssen noch Nachbesserungen vorgenommen werden, um die Praxistauglichkeit zu erzielen.

5. Fazit

XML hat bereits in vielen Anwendungen der Datenverarbeitung Einzug gehalten. Auch diese Arbeit entstand mit XML. StarOffice von Sun Microsystems nutzt es zum Speichern der Texte und Präsentationen. Außerdem gibt es eine Vielzahl an Sprachen um Geschäftsprozesse zu beschreiben. XML/EDI wird auf absehbare Zeit die verschiedenen EDI-Dialekte einen und so für erhebliche Erleichterungen im Geschäftsablauf unter Firmen sorgen. Der Endverbraucher wird dies hoffentlich durch attraktive Preise und innovative Dienstleistungen bemerken. SOAP hingegen wird dem Endanwender direkt durch neue Informationsmöglichkeiten im World Wide Web verständlich werden. Die Entwicklung steht leider noch am Anfang, doch die allgemeine Aufbruchsstimmung wird bald zu noch mehr interessanten Produkten führen.

6. Literaturverzeichnis

/Bosak97/

Jon Bosak

XML, Java, and the future of the Web.

<<http://www.ibiblio.org/pub/sun-info/standards/xml/why/xmlapps.html>>

1997-10-03

Abruf 2001-11-28

/Buxmann01/

Peter Buxmann, Frank Ladner, Tim Weitzel

Anwendung der Extensible Markup Language (XML), Konzeption und Implementierung einer WebEDI-Lösung

Wirtschaftsinformatik

Verlag Vieweg, Wiesbaden

Heft 3, Juni 2001

S. 257 ff.

/Develop01/

DevelopMentor, o. V.

SOAP Frequently Asked Questions (3)

<<http://www.develop.com/soap/soapfaq.htm>>

Abruf 2001-11-28

/Frank01/

Ulrich Frank

Standardisierungsvorhaben zur Unterstützung des elektronischen Handels: Überblick über anwendungsnahe Ansätze

Wirtschaftsinformatik

Verlag Vieweg, Wiesbaden

Heft 3, Juni 2001

S. 283 ff.

/Kuschke01/

Michael Kuschke, Ludger Wölfel

Zu Diensten - Web Services als E-Business-Evolution

iX, Magazin für professionelle Informationstechnik

Verlag Heise, Hannover

Ausgabe 11/2001

Seite 149ff

ISSN: 0935-9680

/Pott00/

O. Pott, G. Wielage

XML - new technology . Praxis und Referenz

Verlag Markt+Technik, München

ISBN: 3-8272-5737-9

/W3Schools01/

W3Schools, o. V.

SOAP School

<<http://www.w3schools.com/soap/>>

Abruf 2001-11-28.

/Webagency01a/

WEBAGENCY E-Commerce AG, o. V.

Was ist e-Commerce? (Akademische Definition).

<<http://www.webagency.de/infopool/e-commerce-knowhow/ak981021.htm>>.

Was ist e-Commerce? (Praktische Definition).

<<http://www.webagency.de/infopool/e-commerce-knowhow/ak981021.htm>>.

Abruf 2001-11-28.

/Webagency01b/

WEBAGENCY E-Commerce AG, o. V.

Mobile Commerce – Das neue Zauberwort?!?

<<http://www.webagency.de/infopool/e-commerce-knowhow/mobile-commerce.htm>>.

Abruf 2001-11-28.