

# 1 Electronic Business XML (ebXML): Basics und Nutzen

XML schaffte es in nur kurzer Zeit zum Standard für Datenaustauschformate im Internet zu werden. Daher ist XML derzeit eines der populärsten Themengebiete im Bereich des Electronic Commerce. Der XML-Hype tritt nun auch in den ‚klassischen‘ elektronischen Datenaustausch oder kurz EDI ein, ein Bereich der in den letzten Jahrzehnten von traditionellen EDI Standards wie UN/EDIFACT oder ANSI X12 geprägt war. Diese traditionellen EDI Standards werden von vielen Großunternehmen zum Datenaustausch mit ihren Geschäftspartnern eingesetzt. Jedoch fanden UN/EDIFACT und ANSI X12 kaum Akzeptanz bei Klein- und Mittelunternehmen (KMU).

*XML für EDI?*

XML soll nun diese Lücke schließen. XML-basierte EDI-Lösungen werden in manchen Branchen bereits erfolgreich eingesetzt. Diese ersten Erfolgsgeschichten untermauern die Stärken von XML für EDI. Der Enthusiasmus über XML führte jedoch zu vielen unrealistischen Aussagen über das Potential von XML, die derzeitigen Probleme im EDI zu überwinden. Trotz seiner Stärken hinsichtlich allgemeiner Anwendbarkeit und Flexibilität, muss bedacht werden, dass XML selbst keine Lösung von irgend einem Modellierungs- oder Implementierungsproblem darstellt [Cove1998a]. Dies wird vor allem durch die Inkompatibilität existierender Branchenlösungen ersichtlich.

*Erste Erfolgsgeschichten*

Die EDI-Gesellschaft hat jahrzehntelange Erfahrung in der Analyse von organisationsübergreifenden Geschäftstransaktionen und in der Schaffung kompatibler Systemschnittstellen. Daher haben die beiden Schlüsselorganisationen UN/CEFACT und OASIS eine gemeinsame Initiative zur Schaffung eines globalen elektronischen Marktes auf der Basis von XML gestartet: ebXML – electronic business XML. Dieses Kapitel soll die Grundlagen und den Nutzen der ebXML-Initiative erläutern.

*UN/CEFACT und OASIS  
starten globale Initiative  
ebXML*

## 1.1 Die Bedeutung von EDI

*Internet – vom Verkaufsmedium zur Plattform effizienter zwischenbetrieblicher Kommunikation*

Als Mitte der 90iger Jahre „Electronic Commerce“ zum Schlagwort wurde, konzentrierte sich das meiste Interesse auf die Verwendung des Internet als neues Marketinginstrument und Verkaufsplattform. Im virtuellen wie auch im realen Geschäftsleben wurde zuerst der Verkauf als erster Erfolgsfaktor erachtet bevor die Bedeutung von effizienten Geschäftsprozessen erkannt wurde. Mittlerweile haben jedoch bereits Unternehmen und Organisationen die Mächtigkeit des Internets zur Unterstützung von effizienten Kommunikationsprozessen zwischen Partnern über die gesamte Wertschöpfungskette hinweg erkannt.

*B2B ist nicht gleich B2B*

Dementsprechend werden dem Business-to-Business (B2B) Bereich im Electronic Commerce (EC) stärkere Wachstumsraten als dem Business-to-Consumer (B2C) Bereich vorhergesagt. Doch stellt sich die Frage: Wodurch unterscheidet sich B2B von B2C? Sicherlich sind die strategischen Geschäftsmodelle für Konsumenten und Geschäftspartner unterschiedlich. Doch wenn wir den Kommunikationsprozess selbst betrachten, ist die Einteilung von EC nach Zugehörigkeit des Kommunikationspartners zu den Gruppen Konsument, Unternehmen und eventuell noch Verwaltung (Government) nicht ausreichend. Denn die eingesetzten Technologien innerhalb der Kommunikation zwischen Unternehmen hängen stark von der Größe und dem technologischen Reifegrad der Unternehmen ab. Für Großunternehmen stellt die Anwendungsintegration der auszutauschenden Information einen entscheidenden Faktor dar. Speziell sehr kleine Unternehmen (Mikrounternehmen) sind zwar in der Lage, das Internet als Kommunikationsplattform zu nutzen, setzen jedoch kaum interne Anwendungen zur Unterstützung der Geschäftstransaktion ein oder können es sich nicht leisten, automatische Schnittstellen zu Anwendungen zu schaffen.

*Wer erzeugt bzw. verarbeitet Information?*

Daher stellt die Art der Schnittstelle – sei es ein Mensch oder eine Maschine (Anwendung) – die Information beim Sender generiert bzw. beim Empfänger interpretiert, ein weiteres Einteilungskriterium für unterschiedliche Formen des EC dar. Dementsprechend können folgende 4 Arten des EC differenziert werden:

- ❑ **Mensch-zu-Mensch:** Ein Mensch erzeugt die zu sendende Information beim Sender und ein anderer Mensch empfängt und verarbeitet die Information beim Empfänger. Typisches Beispiel sind e-mails.
- ❑ **Mensch-zu-Maschine:** Beim Sender interagiert ein Mensch mit einer Anwendung, um die Information in einem strukturierten und maschineninterpretierbarem Format zu erzeugen. Diese Information wird zum Empfänger übermittelt, wo sie ohne Eingreifen ei-

nes Menschen verarbeitet werden kann. Diese Methode wird im WWW eingesetzt.

- ❑ **Maschine-zu-Mensch:** Information wird automatisch aus dem Informationssystem des Senders erzeugt, aber ein Mensch muss diese beim Empfänger interpretieren. Diese Form funktioniert als Einzige nur in eine Richtung, d.h. eine Rückmeldung kann üblicherweise nur auf Mesch-zu-Mensch Wege erfolgen.
- ❑ **Maschine-zu-Maschine:** Die Information wird automatisch aus der Anwendung des Senders generiert und vom empfangenden Informationssystem automatisch weiterverarbeitet. Für diese Methode steht der Term Elektronischer Datenaustausch (EDI).

Idealerweise wird Information nur einmal erfasst und dann über Unternehmensgrenzen hinweg die gesamte Wertschöpfungskette entlang genutzt. Menschliches Eingreifen, um bereits erfasste Information erneut einzutippen, unterbricht natürlich den Informationsfluss. Daher können optimale Erfolge in einer Geschäftsumgebung, wo alle beteiligten Partner Informationssysteme einsetzen, nur erzielt werden, wenn Maschine-zu-Maschine Schnittstellen genutzt werden.

*Unternehmensgrenzen sollen keinen Medienbruch erzwingen*

Die Verbindung der beteiligten Informationssysteme lässt die Unternehmensgrenzen verschwimmen und resultiert in interorganisationellen Informationssystemen [SwSw1992]. Das Ziel der interorganisationellen Informationssysteme ist es, die Geschwindigkeit und Genauigkeit zu erhöhen und gleichzeitig die Kosten von Geschäftstransaktionen zu verringern. Dies kann nur erreicht werden, wenn Daten direkt vor der Anwendung in einem Unternehmen zur Anwendung im anderem Unternehmen ohne menschliche Interaktion übertragen werden. Dies wird durch die Konzepte von EDI erreicht. Dementsprechend wird EDI als der elektronische Transfer von kommerziellen oder administrativen Daten zwischen Anwendungen unabhängig von Transportmedium, Hardware und Software definiert (vgl. [Berg1991; Emme1990; HiFe1989]). Heute assoziieren bereits viele den Term EDI mit einer veralteten Technologie und verwenden modernere Begriffe wie „Collaborative Systems“ oder „Silent Commerce“. Jedoch ist die Problematik, Daten zwischen heterogenen Anwendungen auszutauschen, noch immer dieselbe.

*Interorganisationelle Informationssysteme*

## 1.2 Die EDI Problematik

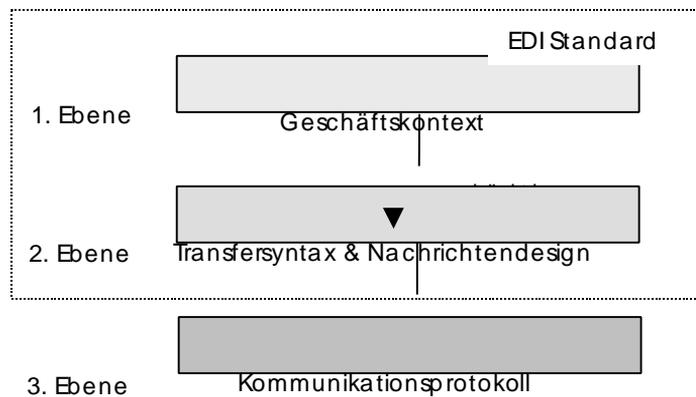
Wenn Information zwischen Menschen ausgetauscht wird, so liegt der Fokus nicht auf der Strukturierung der Daten sondern auf der Übermittlung der Daten vom Sender zum Empfänger. Zu diesem Zweck sind bereits eine Reihe von Standards wie SMTP, FTP oder X.400 in Verwen-

*EDI bedarf eines Handschlags auf 3 Ebenen*

dung. Im Falle des WWW, wo ein Mensch mit einer entfernten (remote) Anwendung interagiert, besteht der Bedarf, die Information zu strukturieren. Auf der einen Seite wird eine Struktur für die Präsentation im Browser (HTML) benötigt und auf der anderen Seite dafür, um die Daten, die in Formularfelder eingegeben wurden, (z.B. mit CGI) zu identifizieren. Der Geschäftskontext der auszutauschenden Daten wird immer durch die Formularfelder und damit durch die Anwendung am Server definiert. Im Falle von EDI kann festgestellt werden, dass der Geschäftskontext generell weder von der sendenden noch von der empfangenden Applikation bestimmt wird. Damit heterogene Applikationen miteinander kommunizieren können, müssen Vereinbarungen auf drei Ebenen, wie in Abb. 1-1 dargestellt, getroffen werden.

Abb. 1-1

*Ebenen der EDI  
Standardisierung*



*Geschäftskontext*

Auf oberster Abstraktionsebene muss eine Vereinbarung bezüglich des Geschäftskontext getroffen werden. Der Informationsfluss mit allen möglichen Szenarien und Daten, die diesen Informationsfluss unterstützen, muss definiert werden. Es sei darauf hingewiesen, dass die Semantik einer Geschäftstransaktion unabhängig davon ist, welche Transfersyntax bzw. welche Übermittlungsmethode eingesetzt wird. Da heterogene Applikationen üblicherweise auch unterschiedlich sind hinsichtlich der Daten, die sie verarbeiten müssen und auch können, ist eine Vereinbarung auf dieser Ebene unabdingbar.

*Transfersyntax &  
Nachrichtendesign*

Wenn klar ist worum es in der Geschäftstransaktion gehen soll, muss der Geschäftskontext in einem entsprechenden Format, welches für die Übertragung genutzt wird, abgebildet werden. Folglich umfasst die zweite Ebene zwei Aspekte: Eine Transfersyntax, die angibt wie die Daten zu strukturieren sind, ermöglicht die Erkennung von Datenwerten. Zusätzlich muss ein Nachrichtendesign, das dem Geschäftskontext folgt und auf der Transfersyntax basiert, sicherstellen, dass die Datenwerte die entsprechende Semantik erhalten.

Das Kommunikationsprotokoll auf unterster Ebene ist verantwortlich für die Übertragung der Daten vom Sender zum Empfänger. Existierende EDI Standards sind unabhängig vom Kommunikationsprotokoll. Sie basieren nur auf einer strukturierten Zeichenkette. Daher ist prinzipiell jedes Kommunikationsprotokoll mit dem eine Zeichenkette übertragen werden kann zur Übertragung von EDI Nachrichten geeignet. Trotzdem müssen Partner vor einem Datenaustausch vereinbaren auf welchem Weg mit welchem Kommunikationsprotokoll die Daten ausgetauscht werden.

*Kommunikations-  
protokoll*

### 1.3 XML als Basis für EDI

XML entwickelt sich zum Datenaustauschformat für Internetapplikationen [BoBr 1999]. Auf Grund der Tatsache, dass XML Dateien – wenn bestimmte Voraussetzungen erfüllt sind – von Maschinen gelesen werden können, wird XML als Alternative für den Austausch von Anwendung-zu-Anwendung vorgeschlagen [PeWe 1997].

*XML/EDI*

XML ist eine Sprache strukturierte Daten so zu beschreiben, dass die darin enthaltene Information selbsterklärend wird. Dabei definiert XML das Regelwerk zur Spezifikation von Markup-Sprachen für die Codierung bestimmter Dokumenten- bzw. Nachrichtentypen. Somit ist XML mehr als eine Sprache (wie z.B. HTML), nämlich eine Metasprache [BoBr 1999].

*XML ist eine Meta-  
sprache*

Da XML die Definition von Datenaustauschformaten erlaubt und es prinzipiell möglich ist DTDs zu erzeugen, die semantisch äquivalent zu UN/EDIFACT oder X12 Nachrichtentypen sind, liegt es auf der Hand, dass XML-basierte Sprachen für EDI genutzt werden können. Doch muss kritisch analysiert werden, welche Vorteile XML tatsächlich für EDI bringt und wo auch die Grenzen von XML in diesem Bereich liegen. Eine derartige Untersuchung wurde von der Techniken- und Methodenarbeitsgruppe (TMWG) des Zentrums für Handelserleichterung und Electronic Business der UNO (UN/CEFACT) durchgeführt [TMWG1999].

*XML ist prinzipiell für EDI  
geeignet*

Da nun XML eine Metasprache ist, kann jeder eine neue Sprache in Form einer DTD bzw. zukünftig als XML Schema erzeugen. Die Einfachheit des XML-Regelwerks erlaubt es auch Nicht-Technik-Experten innerhalb kürzester Zeit ihre Nachrichtenformate (DTDs oder Schema) zu entwickeln. Diese Flexibilität garantiert natürlich rasche Lösungen um Daten auszutauschen im Vergleich zum langwierigen Nachrichtenstandardisierungsprozess von UN/EDIFACT. Außerdem können die DTDs sehr einfach über das WWW zugänglich gemacht werden. Jedoch sollte nicht vergessen werden, dass jedem erlaubt ist, seine eigenen Tags zum Kennzeichnen der Daten zu gestalten. Nur dieser proprietäre Ansatz unter

*Flexibilität erlaubt  
rasche Lösungen*

Verzicht einer übergreifenden Abstimmung wie bei UN/EDIFACT ermöglicht schnelle Entwicklungszeiten. Daher zählt dieses Argument am stärksten bzw. fast ausschließlich für XML-Lösungen in einem geschlossenen Anwenderkreis, wo die Bedeutung der Tags und der Struktur jedem Teilnehmer bekannt ist. In diesem Fall liegt es nahe, von einem XML/EDI-Extranet zu sprechen.

*Einbindung von  
Multimediatdaten*

Aber neben der Flexibilität gibt es noch weitere Vorteile im Vergleich zu UN/EDIFACT und X12. Diese traditionellen EDI Standards wurden vor allem für die Übermittlung von textbasierten Geschäftsdaten entwickelt. Die Übertragung von nicht rein textbasierten Daten, wie z.B. Röntgenbilder oder CAD-Daten, stellt ein Problem dar. Durch das Konzept von ungeparsten Entities und Notations erlaubt XML die einfache Einbindung von Multimediatdaten zur Unterstützung der Geschäftstransaktion.

*Internet als  
Übertragungsmedium*

Weiters öffnet XML die Türe um das Internet anstatt proprietärer Value Added Networks (VANs) als Übertragungsmedium für EDI zu nutzen. Dies sollte zu stark reduzierten Übertragungskosten bei EDI-Transaktionen führen. So könnte auch der Unicode Standard genutzt werden, der mittels seines Zeichencodierungsverfahrens die Vermischung von Textelementen aus verschiedenen Sprachen dieser Welt unterstützt. Doch auch in diesem Fall ist anzumerken, dass traditionelle EDI Standards – da sie unabhängig vom Übertragungsprotokoll sind – über das Internet übertragen werden können.

*Write once,  
publish everywhere*

XML konzentriert sich auf die Definition wie die auszutauschenden Daten zu strukturieren sind und kümmert sich nicht um irgendwelche Programmierdetails. Diese strikte Trennung von Dokumentstruktur und -verarbeitung entspricht auch der Philosophie „Write once, publish everywhere“ der Mutter SGML. Da wir im Fall von EDI die Daten eher weiterverarbeiten als publizieren wollen, könnten wir das Prinzip auch als „Transmit once, process everywhere“ bezeichnen.

*XML wird durch Browser  
unterstützt –  
Traditionelle EDI  
Standards nicht*

Jedenfalls ist eine XML-Datei für sich selbst wertlos, wenn sie nicht mit einem Programm verknüpft wird, das sie verarbeitet. Das Gleiche bezüglich Trennung von Dokumentstruktur und -verarbeitung bzw. Notwendigkeit der Übergabe an eine Anwendung gilt auch für UN/EDIFACT- oder X12-Dokumente. Jedoch besteht ein sehr wesentlicher Unterschied: Die traditionellen EDI Standards wurden speziell für den Maschine-zu-Maschine Austausch entwickelt. Im Gegensatz dazu wurde XML in erster Linie für die Anwendung im WWW entwickelt. Daher unterstützen auch beispielsweise Internet Explorer 5.0 und Netscape 6 die Anwendung von XML. Mittels der eXtensible Style Sheet Language (XSL), eines Standards aus der erweiterten XML-Familie, können XML-Dateien in eine geeignete Präsentationsform im Browser gebracht werden. Natürlich wieder „Write once, publish everywhere“

folgend, kann mittels XSL die XML-Datei auch in ein anderes beliebiges Format konvertiert werden. Doch für die von uns betrachteten Aspekte des EC stellt die Browserunterstützung den wesentlichsten Punkt dar, da der Browser die gängigste Schnittstelle für Menschen im EC verkörpert.

Es ist möglich XML-Dokumente aus der eigenen Anwendung automatisch zu generieren und sie am eigenen Server gemeinsam mit einer XSL-Datei für den entsprechenden Dokumententyp abzulegen. Somit kann ein Kunde mit entsprechenden Zugriffsrechten jederzeit das Dokument in seinem Browser betrachten. Somit kann mit Hilfe von XML eine effiziente Maschinen-zu-Mensch Schnittstelle realisiert werden. Aber auch die Mensch-zu-Maschinen Kommunikation ist realisierbar. Eingaben, die ein Geschäftspartner manuell im Browser tätigt, könnten bei entsprechender Konfiguration als XML-Datei am eigenen Server eingehen und automatisch weiterverarbeitet werden. Daraus folgt, dass XML Maschinen-zu-Maschinen, Mensch-zu-Maschinen und Maschinen-zu-Mensch Kommunikation erlaubt.

*XML sowohl für Maschine-Maschine, Mensch-Maschine, Maschine-Mensch*

Wenn nun ein Unternehmen eine entsprechende Anwendung einsetzt und folglich seinerseits eine Maschinen-Schnittstelle einrichtet, muss es sich nicht kümmern, ob der Partner die Daten manuell oder maschinell erzeugt bzw. weiterverarbeitet. Der Input vom Partner bzw. der Output an den Partner könnte immer der gleiche XML-Dokumententyp sein. Somit können dort wo heute noch unterschiedliche Datenformate genutzt werden, alle Schnittstellen in Richtung XML kanalisiert werden. Siehe das Beispiel einer Bank: Mit ihren Großkunden führt sie Zahlungsabwicklungen über UN/EDIFACT durch. Privatpersonen und Kleinunternehmen verwenden die Internet-Banking-Lösung auf Basis von elektronischen Formularen über CGI. Ein weiterer Teil von Unternehmen verwendet ein von der Bank zur Verfügung gestelltes Softwarepaket, das die Daten in proprietärem Format an die Bank übermittelt. Für die Bank wäre es natürlich optimal wenn alle drei Varianten über das gleiche Datenformat ablaufen würden, was den Programmier- und Wartungsaufwand erheblich verringern würde.

*XML für manuelle als auch maschinelle Verarbeitung geeignet*

Das löst zwar nicht das eigentliche EDI-Problem der Integration heterogener Anwendungen, stellt aber auch für viele KMUs eine signifikante Verbesserung dar. Denn in vielen traditionellen EDI-Partnerschaften werden KMUs von ihren mächtigeren Partnern mit EDI-Nachrichten beglückt, ohne dass sie diese in eine Anwendung integrieren können. Daher stellt eine Browseroberfläche zur Unterstützung der Geschäftstransaktion für diese KMUs eine wesentliche Verbesserung zu den für sie „kryptischen“ EDI-Nachrichten dar (vgl. [WeBW2000]).

*KMUs ohne Anwendungsintegration profitieren von Browserlösung*

Einen weiteren entscheidenden Vorteil genießt XML im Vergleich zu traditionellen EDI Standards vor allem auf Grund der Popularität des Internet. Dadurch wird XML in Kürze einer breiten Öffentlichkeit an

*In Kürze eine Vielzahl an XML-Programmierern*

technisch Interessierten vertraut sein. Dieser Trend wird unterstützt einerseits durch das einfache Regelwerk von XML und die Aufnahme von XML in Lehrpläne andererseits, sodass in Kürze eine Reihe von XML-Programmierern vorhanden sein wird. Im Vergleich dazu ist nach über 10 Jahren der Kreis an UN/EDIFACT-Experten als sehr klein einzustufen. Damit werden viele Programmierer eine XML-Lösung gegenüber einer UN/EDIFACT-basierten Lösung vorziehen.

*Kostengünstige  
XML-Tools*

Diese Popularität von XML wird auch zu einer Reihe von kostengünstigen Werkzeugen für den EC-Einsatz und auch im Speziellen für den B2B Bereich führen. Schon heute wird vielfach behauptet, dass mittels XML Dokumente für KMUs kostengünstig zu integrieren sind, da sie relativ einfach mit einem XML-Parser bearbeitet werden können. Schließlich sind XML-Parser auch weit kostengünstiger, wenn nicht überhaupt gratis, im Vergleich zur kostspieligen EDI-Konvertersoftware. Doch ist dieses Argument mit ein bisschen Vorsicht zu genießen, da das Parsen – also die Identifikation von Datenwerten – nie die wirklich hohen Kosten einer EDI-Partnerschaft ausgemacht. Das wirklich kostspielige Problem ist immer zu garantieren, dass die auszutauschenden Dateien alle die für die beteiligten Anwendungen notwendigen Informationen enthalten und die Spezifikation und Implementierung der Übernahme dieser Informationen in die Anwendungen.

## 1.4 Probleme bei XML für EDI

*Anwender soll XML-  
Dateien nicht sehen*

Wie im vorigen Unterkapitel angeführt bietet XML eine Reihe von Vorteilen für den Einsatz im EDI-Bereich. Aber was ist mit dem so oft genannten Vorteil „XML ist für KMUs einfach anzuwenden, weil XML-Dateien vom Menschen gelesen werden können“? Dazu ist vorweg anzumerken, dass ein Anwender in einem KMU niemals die XML-Datei zu Gesicht bekommen soll. Wenn er sie ansieht, dann bereits aufbereitet in seinem Browser. Somit sind XML-Dateien nur für die Programmierer von Interesse. Die Lesbarkeit von Tags kann nur einen ersten Hinweis für den Programmierer liefern.

*Zusätzliche semantische  
Definition von Tags ist  
notwendig*

Jedoch verwechseln viele, die einen Ausspruch wie obigen tätigen, Lesbarkeit für den Menschen mit semantisch vollständiger Definition. Maschinen selbst besitzen keine Intelligenz. Ihnen muss (mittels eines Programms) exakt erklärt werden, welche Bedeutung hinter einem Tag steht, damit sie ihn verarbeiten können. Bei XML handelt es sich jedoch um ein rein syntaktisches Regelwerk. Die semantisch vollständige Definition kann wohl kaum in einem Wort erfolgen. Damit muss die Bedeutung eines Tags anderswo definiert werden. Am besten eignet sich hier

immer noch Prosa (vgl. Definition eines Terms in einem Glossar). Erst wenn die Bedeutung der Tags festgelegt ist, kann die Dokumententypdefinition eine Struktur definieren, sodass die Daten in XML-Dateien selbstbeschreibend werden. Somit kann die Gestaltung der Tags zur menschlichen Lesbarkeit – was wie die Variablenbenennung in Programmen keine Pflicht ist – immer nur eine Hilfestellung sein. Lesbarkeit für den Menschen ist daher wohl nur ein zweitrangiges Bewertungskriterium. Denn es wäre ja auch zweifelhaft zu behaupten, dass COBOL eine bessere Programmiersprache sei als C, nur weil COBOL-Programme in der Regel für den Menschen einfacher zu lesen sind als C-Programme.

Die Notwendigkeit einer semantisch vollständigen Definition außerhalb der Definition des Dokumententyps sei noch an einem Beispiel verdeutlicht. Wie soll eine Maschine wissen, welche Bedeutung sich hinter dem Tag `<Lieferdatum>` verbirgt? Ein Programmierer mag eine Idee haben, was mit Lieferdatum gemeint ist. Nehmen wir an, dass es sich tatsächlich um das Datum handelt, an dem die Waren geliefert werden sollen. Doch könnten die Programmierer des Lieferanten und des Kunden in einer EDI-Transaktion noch immer unterschiedliche Auffassungen vertreten. So könnte der eine annehmen, dass es jenes Datum sei, an dem er die Waren versendet, und der andere, dass er die Waren an diesem Datum erhält. Weiters könnten unterschiedliche Erwartungen auftreten, ob es sich um das späteste oder das exakte Lieferdatum handelt. Wirtschaftliche Folgen einer verfrühten Lieferung könnten auch fatal sein, wenn der Kunde noch keinen Lagerraum für die Waren zur Verfügung hat.

*Was bedeutet  
<Lieferdatum>?*

Aber die Namensgebung der Tags ist nicht die einzige Entscheidung die der Designer eines Nachrichtentyps zu treffen hat. Die Struktur einer DTD wird in erster Linie von den existierenden Abhängigkeiten zwischen den in ihr enthaltenen Elementen bestimmt sein. Doch bleibt immer noch ein Gestaltungsfreiraum. Eine der wichtigsten Fragen – auf die es keine generelle Antwort geben kann – ist, welche Information in einem Element und welche in einem Attribut abgebildet wird. Beispielsweise kann ein Designer es vorziehen, die Währung in einem eigenen Element, das dem Element des Betrages folgt, auszudrücken. Ein anderer Designer zieht es vor, die Währung als Attribut zum Element des Betrages aufzunehmen.

*Element oder Attribut?*

Eine weitere Designentscheidung ist auf Grund der hierarchischen Struktur von DTDs zu treffen. Zwar mag vieles in unserem Leben einen hierarchischen Aufbau haben, aber es gibt auch noch weitere Datenstrukturen. Wie sind unter anderem n:m-Beziehungen zu hierarchisieren? Beispielsweise könnte es eine n:m Beziehung „bestellt“ zwischen Artikel und Standort in einer Bestellung geben. Sollen nun innerhalb der Bestellung zuerst die Artikel angeführt werden und für jeden Artikel für welche Standorte dieser bestellt wird? Oder ist es umgekehrt besser, die Standorte

*n:m-Beziehungen*

anzuführen und innerhalb der Standorte die Artikel zu spezifizieren? Wenn nun Artikel wie auch Standorte komplexe Objekte, d.h. aus weiteren Elementen aufgebaut sind, so würde bei obigen Lösungen Overhead durch Mehrfachangabe einzelner Standorte bzw. einzelner Artikel entstehen. Daher könnte es sinnvoll sein, die Definitionen von Artikeln und Standorten nicht zu schachteln, sondern in die Schachtelung nur Referenzen, eventuell mittels XLink, aufzunehmen.

*Problem erst bei  
Teilnahme an mehreren  
Extranets*

Jetzt stellt sich natürlich die Frage, wo das Problem liegt, wenn für alle obigen Punkte Entscheidungen und Definitionen getroffen werden. Für spezielle Einzelfälle, wo Dokumente in einem geschlossenen Anwenderkreis ausgetauscht werden und allen die Semantik der Dokumententypen bewusst ist, gibt es auch keine Probleme. Wenn jedoch Unternehmen mit Partnern außerhalb dieses Extranets Dokumente austauschen wollen, werden Probleme von XML im Bereich des EDI deutlich. Denn dann ist es höchst wahrscheinlich, dass die Definitionen für Nachrichten, obwohl für den gleichen Typ von Geschäftstransaktion bestimmt, für verschiedene Extranets völlig unterschiedlich aussehen. Auch wenn ein günstiger bzw. kostenloser Parser zur Verfügung steht, bedeutet dies, dass mit Unterstützung dieses Tools für jedes Extranet eine spezielle Schnittstelle zur internen Anwendung programmiert werden muss. Ein Aufwand der vielfach unterschätzt wird.

*XML-Dialekte*

Wenn nun jedes Unternehmen seine eigenen XM-Tags für Produktkataloge, Preisanfragen, Bestellungen, Rechnungen etc. einführen würde, so würde dies in einer Unzahl von DTDs enden. Die resultierende Inkompatibilität zwischen den einzelnen Lösungen würde dem elektronischen Datenaustausch nicht dienen. Jedoch haben bereits manche unternehmensübergreifende Organisationen begonnen, XML-basierte Spezifikationen für die Information, die in ihrer Branche am häufigsten ausgetauscht wird, zu entwerfen. Einige Beispiele hierfür wären das Internet Open Trading Protocol for Payment, RosettaNet, Open Buying on the Internet (OBI) oder Open Travel Alliance (OTA). Aber auch Frameworks, die sich über Industriezweige spannen, haben ihre eigenen XML-Dialekte entworfen. Beispiele hierfür wären das auch von Ariba unterstützte Commerce XML (cXML), Commerce One's XML Common Business Library (xCBL) oder die Lösungen der Open Applications Group (OAG). Doch sind diese branchenübergreifenden Lösungen stark vom Betreiber der jeweiligen Lösung abhängig und stehen zu einander in Konkurrenz.

*XML-Dialekte sind inkompatibel*

Jede dieser XML-basierten Spezifikationen stellt für den entsprechenden Industriezweig bzw. die entsprechende Anwendergruppe einen Schritt vorwärts dar. Denn wenn mehr Teilnehmer in einem Extranet die selbe Sprache sprechen fördert dies natürlich die Akzeptanz von XML-basiertem Datenaustausch. Doch besteht eine erhebliche Menge solcher

XML-Vokabulare. Alan Kotok von der Data Interchange Standards Association (DISA) erstellte einen Survey über 124 verschiedene XML Dialekte, die im Jänner 2000 im Einsatz waren [Koto2000]. Problematisch an der Vielzahl von Vokabularen ist, dass jede neue Spezifikation bestehende missachtet, von Neuem startet und entsprechende Konzepte neu erfindet. Jedoch sind gewisse Kernelemente in mehreren oder allen Branchen sinnvoll. Einige semantische Konzepte, die in einer vertikalen DTD verwendet werden, sind auch in einer anderen relevant, jedoch werden dafür unterschiedliche XML-Elemente verwendet.

Dieser Umstand erinnert an jene Situation, in der sich die EDI-Standardisierung in den 70er und 80er Jahren befunden hat. Jedoch sollten im Falle von XML bereits frühzeitig jene Fehler, die aus der EDI-Geschichte bekannt sind, vermieden werden. Die wahren Probleme im Entwurf und der Standardisierung von Nachrichten, die den Geschäftsbedürfnissen entsprechen, sind sicher unabhängig von der Syntax zu sehen. Die EDI-Gemeinde hat jahrzehntelange Erfahrung in der Abbildung von Geschäftsprozessen, während viele in der XML-Gemeinde erst begonnen haben diese Problematik zu erkennen.

*XML-Weg erinnert an EDI-Geschichte*

## 1.5 Voraussetzungen für erfolgreiches XML/EDI

Zum effektiven Einsatz von XML ist eine semantische Transparenz notwendig. Dies bedeutet, dass sowohl Menschen als auch Maschinen Information eindeutig und semantisch korrekt interpretieren können [Cove1998b]. Der Mangel an einem zu Grunde liegenden Standardgeschäftsmo-  
dell erschwert natürlich die weltweite Interoperabilität. Die Verwendung der in den UN/EDIFACT-Standardnachrichten enthaltenen Semantik eignet sich als Basis für Geschäftsmodelle und würde garantieren, dass existierende EDI-Anwendungen mit neuen, auf XML basierenden Anwendungen kompatibel wären.

*Semantische Transparenz ist Voraussetzung*

Jedoch ist das Überleiten der Semantik der UN/EDIFACT-Nachrichten in eine XML-Darstellung nicht unbedingt eindeutig definiert. Heute stellen zwar einige Organisationen eine Lösung für diese Überleitung vor. Vielmehr wahr ist jedoch, dass sie mittels XML eine eigene Markupsprache entworfen haben – also eine DTD – und für diese eine Überleitung definiert haben. Genauso unterschiedlich wie die entworfenen DTDs sind, so unterschiedlich sind natürlich auch die Überleitungen. Daher sind diese Lösungen zueinander nicht kompatibel.

*Mapping UN/EDIFACT – XML ist nicht eindeutig*

Auch erscheint es nicht sinnvoll, mittels eines Algorithmus eine automatische Überleitung durchzuführen. Dies liegt vor allem am hierarchischen Aufbau von UN/EDIFACT, wo Komponenten immer gleich aufge-

baut und voll nach unten durchlässig sind. Das heißt, ein gültiger Code für ein Datenelement in einem Segment ist in jeder Nachricht in der das Segment vorkommt gültig, wenn auch nicht semantisch sinnvoll. Beispielsweise ist der Code für ein Schlachtdatum für den Datenaustausch zwischen Schlachthaus und Fleischereien notwendig. Die Anwendung in einer Nachricht zur Patientenüberweisung erscheint jedoch zweifelhaft. Außerdem erhöht eine derartige Vorgangsweise die Länge der Übertragungsdatei ohne zusätzlichen Nutzen. Eine geeignete Basis für die Überleitung bilden eher die Message Implementation Guidelines (MIGs), da in ihnen die Semantik der Geschäftsprozesse definiert ist.

*Erfassung des Geschäftskontexts vor der Überleitung nach XML*

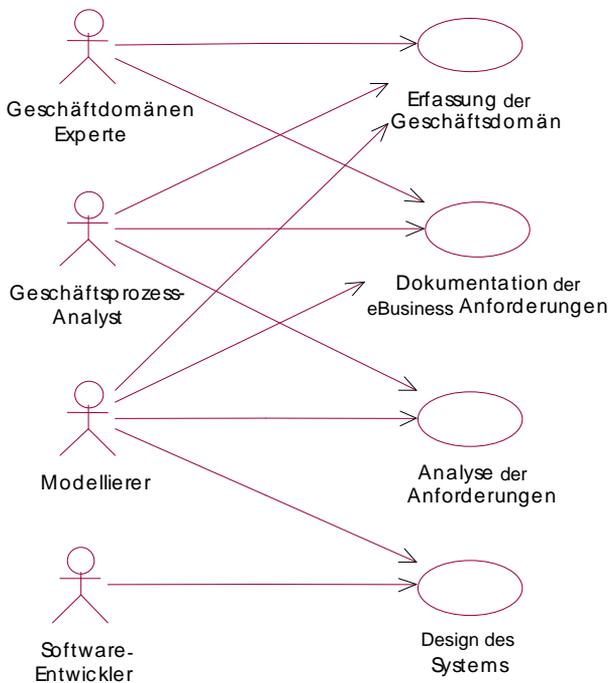
Da es auch noch eine Zukunft nach XML gibt, ist es auch sinnvoll eine Überleitung über eine syntaxneutrale Darstellung durchzuführen. Somit kann das Wissen über interorganisationelle Geschäftsprozesse auch über die Lebensdauer einer Transfersyntax gerettet werden. Schließlich ist für Geschäftsleute, die ihre Geschäftstransaktionen und die dafür benötigten Daten beschreiben, das für sie am wohl wenigsten Bedeutsame die Transfersyntax. Sie verstehen in der Regel ihr Geschäft und wissen daher, welche Daten sie von ihren Partnern erwarten, um eine bestimmte Geschäftstransaktion durchführen zu können bzw. welche Daten sie einem Partner für die Durchführung einer Geschäftstransaktion liefern können. Daher sollte das Wissen der EDI-Gemeinde, die sich hauptsächlich aus Anwendern zusammensetzt, genutzt werden um diesen – wie auch in 1.2 gefordert – Geschäftskontext aufzuzeichnen, bevor er nach XML transkribiert wird.

*Open-edi konform*

Eine derartige Vorgangsweise würde auch dem Open-edi Standard der ISO [ISO1996], welcher zwischen dem Business Oriented View (BOV) und dem Functional Service View (FSV) unterscheidet, entsprechen. Der Geschäftskontext würde dann im BOV standardisiert. Für jede Transfersyntax müsste dann eine für die Transfersyntax speziell entworfene Methode angewandt werden, um den Geschäftskontext eben in dieser Transfersyntax niederzuschreiben. So könnten bestehende UN/EDIFACT-Nachrichten (FSV) über eine neutrale Darstellung im BOV in eine einheitlich XML-Darstellung (FSV) übergeleitet werden. Natürlich könnte diese Vorgangsweise auch genutzt werden um bereits existierende XML-Dialekte in einen später allgemein anerkannten Dialekt zu übersetzen. Denn prinzipiell stellt jeder Dialekt nichts anderes als eine neue Implementierung des FSV dar und zwischen zwei FSVs kann immer über den BOV übersetzt werden. Voraussetzung ist neben der eindeutigen Methodologie für jede BOV-FSV Relation auch, dass beide Dialekte den selben Geschäftskontext unterstützen.

Abb. 1-2

UMM Workflows



Die BOV-Ebene nimmt in diesem Ansatz die zentrale Rolle ein, mit Hilfe derer eine gemeinsame Verständigung erreicht werden soll. Auch wenn der BOV nicht von der Transfersyntax beeinflusst wird, so bedarf es eines allgemein anerkannten Regelwerks, damit ein gemeinsames Verständnis erreicht wird. Die TMWG von UN/CEFACT entwickelt dazu den OO-edi Standard [TMWG1998], der dem Open-edi Framework folgt. Dabei definiert TMWG die sogenannte Unified Modeling Methodology (UMM) zur Definition von BOV standards [Huem1999, TMWG2000]. Diese Methodologie basiert auf der Unified Modeling Language (UML) [BoJR1998; Fowl1997] und ist eine angepasste Version von Rational's Unified Process (RUP) [Kruc1998]. Dies bedeutet jedoch nicht, dass alle Domänenexperten in Modellierer verwandelt werden müssen. Im Gegenteil, UMM definiert klar – wie in Abb. 1-2 überblicksweise dargestellt – welche Fähigkeiten ein Experte mitbringen muss und in welchen Schritten der Standarddefinition er beteiligt ist.

OO-edi

## 1.6 ebXML – Die Initiative

*ebXML zur  
Standardisierung von  
XML Nachrichtentypen für  
Electronic Business*

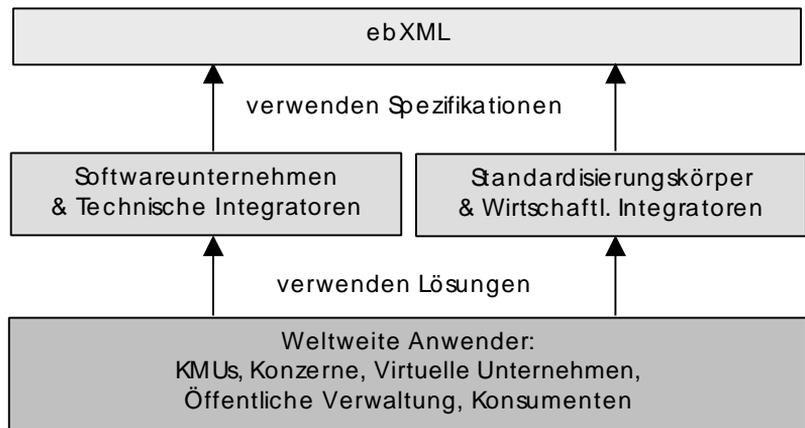
Die TMWG empfahl dem Steering Committee von UN/CEFACT eine Initiative zu starten, um die verschiedenen in Konkurrenz stehenden Projekte zur Definition von XML Formaten auf eine gemeinsame Plattform zu stellen, um so Interoperabilität für die Zukunft zu sichern. Um im Sinne von UN/CEFACT eine Kooperation der XML- und der traditionellen EDI-Welt von ausreichender Akzeptanz zu initiieren, ging man mit der am XML-Sektor bedeutendsten Dachorganisation OASIS eine Partnerschaft ein, um im November 1999 ein weltweites Projekt zur Standardisierung von XML Spezifikationen für Electronic Business zu starten: electronic business XML (ebXML). Diese Initiative ist für jeden offen und die Projektlaufzeit ist mit 18 Monaten veranschlagt.

*Offenes, technisches  
Framework*

Der Zweck von ebXML – wie in den Terms of Reference festgelegt [EBXM1999] - ist die Untersuchung und Entwicklung der technischen Grundlagen, auf denen die weltweite Implementierung von XML standardisiert werden kann. Das Ziel ist die Schaffung eines offenen, technischen Frameworks, um XML für den Austausch von Geschäftsdaten in Anwendung-zu-Anwendung, Anwendung-zu-Mensch und Mensch-zu-Anwendung in konsistenter und einheitlicher Weise zu ermöglichen. Mit Hilfe dieses Frameworks sollen die Eintrittsbarrieren in das Electronic Business, im Speziellen für KMUs, überwunden werden.

Abb. 1-3

*ebXML  
Anwender*



*Wer verwendet ebXML  
Spezifikationen?*

Die Ergebnisse des Projektes sollen die technische Spezifikation zur Unterstützung des nationalen und internationalen Austauschs von Geschäftsdaten umfassen. Wesentlich ist, dass die Entwicklung von Nachrichteninhalten nicht in den Rahmen von ebXML fällt. Demnach ist es das Ziel, wie in Abb. 1-3 dargestellt, Spezifikationen für Standardkomitees, die XML Datenaustauschformate bzw. Geschäftsprozessmodelle erzeugen,

und für Softwarefirmen, die darauf aufbauende Software entwickeln, zu liefern. Die eigentlichen Anwender verwenden dann die von den Softwarefirmen auf den Markt gebrachte Software bzw. verwenden die Definitionen der Standardkomitees, um interne Adaptionen durchzuführen. Natürlich wäre es wünschenswert, wenn parallel zur Initiative bereits ebXML-konforme Datenaustauschformate bzw. Geschäftsprozessmodelle von Organisationen entwickelt und an den aktuellen Stand angepasst würden.

Bei der Schaffung der technischen Spezifikation von ebXML wird folgende Vision verfolgt: Das Framework ermöglicht allen Industriegruppen oder Standardisierungskörpern ein einheitliches Vorgehen zur Schaffung ihrer Geschäftsmodelle. Diese Modelle beinhalten alle möglichen Aktivitäten – inklusive Datensicht – um einen bestimmten Geschäftszweck zu erreichen. Diese Modelle werden in einem globalen, virtuellen Repository registriert und gespeichert. Die Anwender – Unternehmen und Organisationen – müssen jeweils jene Szenarien der Modelle, die sie unterstützen, identifizieren und ebenfalls registrieren lassen. Damit können Geschäftspartner, die mit ihnen eine Geschäftstransaktion abwickeln wollen, feststellen, ob sie zumindest ein gemeinsames Szenario zur Durchführung unterstützen.

Um Interoperabilität zu erreichen muss eine gemeinsame Methode zur Schaffung der Geschäftsmodelle verwendet werden (z.B. UMM). Auch muss eine für alle einheitliche Vorgangsweise bei der Überleitung des Geschäftskontext nach XML DTDs gegeben sein. Um zu verhindern, dass einzelne Standardisierungsgruppen das Rad immer wieder von neuem erfinden und stattdessen ein einheitliches Konzept verwenden, werden innerhalb von XML auch die gängigsten Anwendungsobjekte (vgl. Business Objects [Casa1995, Wesk1999] definiert. Diese Anwendungsobjekte werden von den Modellierern bei der Dokumentation ihrer Geschäftsprozesse verwendet. Sie sollten daher einerseits in syntaxneutraler Version als Geschäftsmodellkomponente, als auch in XML DTD Komponente für die spätere Verwendung im Datenaustausch vorliegen.

Um die Kommunikation zwischen zwei Partner zu garantieren muss – wie in 1.2 definiert – auch eine Vereinbarung bezüglich Kommunikationsprotokoll getroffen werden. Dementsprechend definiert ebXML ein Vorgabe hinsichtlich Messaging Format, Packaging und Routing.

## 1.7 Ausblick

Die Projektlaufzeit von 18 Monaten, in der das endgültige ebXML Framework geschaffen sein soll, ist hinsichtlich ihrer Dauer, egal von wel-

*18 Monate – kurz oder lang?*

cher Seite betrachtet, als extrem einzustufen. Einerseits sind 18 Monate für die Komplexität und die Größenordnung des Projektes als extrem kurz zu bezeichnen. Andererseits sind 18 Monate bei der rasanten Geschwindigkeit, in der Electronic Commerce Lösungen produziert werden, extrem lang. Auf jeden Fall wird es sich bis Mai 2001 weisen, ob es gelingt, mit Hilfe von ebXML, einen einzelnen globalen elektronischen Markt zu schaffen, in dem Großunternehmen als auch KMUs problemlos miteinander kommunizieren können.

*XML ist eine Möglichkeit für EDI, keine Lösung*

Doch sollte festgehalten werden, dass XML eher als Möglichkeit für einen revolutionären Ansatz im EDI denn als Lösung gesehen wird. Obwohl XML eine Menge an Vorzügen für EDI aufweist, ist XML alleine noch keine Garantie für eine "Plug & Play"-Lösung für EDI [Huem2000]. Der Erfolg des WWW war auch in der einfachen Anwendbarkeit für jedermann und nicht im speziellen in der Sprache HTML begründet. Daher kann EDI auch nur einen durchschlagenden Erfolg bei KMUs landen, wenn es gelingt einen Datenaustausch ohne vorhergehende Absprachen zu ermöglichen. Denn KMUs werden von ebXML nur einen Nutzen ziehen, wenn sie „plug&play“-Software für ebXML erhalten können. Solange KMUs das Mapping von Datenaustauschformat und Strukturen der eigenen Anwendung selbst durchführen müssen, wird EDI immer nur Großunternehmen nützen.

## Literaturverzeichnis

- [Berg1991] *Berge, J.*: The EDIFACT Standards. NCC Blackwell Limited, Oxford 1998.
- [BoBr1999] *Bosak, J.; Bray, T.*: XML and the Second-Generation Web. Scientific American, <http://www.sciam.com/1999/0599issue/0599bosak.html>, 1999, Abruf am 2000-07-14.
- [BoJR1998] *Booch, G.; Jacobson, I.; Rumbaugh, J.*: The Unified Modeling Language User Guide. Addison Wesley, Reading 1998.
- [Casa1995] *Casanave, C.*: Business Object Architectures and Standards. In: *Sutherland, J.; Patel, D.; Casanave, C.; Hollowell, G.; Miller, J. (Hrsg.)*: Business Object Design and Implementation: OOPSLA'95 Workshop Proceedings. Springer, 1997.
- [Cove1998a] *Cover, R.*: The Essence and Quintessence of XML. Retrospects and Prospects. OASIS, [http://www.oasis-open.org/html/essence\\_of\\_xml.html](http://www.oasis-open.org/html/essence_of_xml.html), 1998, Abruf am 2000-07-14.
- [Cove1998b] *Cover, R.*: Managing Names and Ontologies: An XML Registry and Repository. [http://www.oasis-open.org/html/registry\\_and\\_repository.html](http://www.oasis-open.org/html/registry_and_repository.html), 1998, Abruf am 2000-07-14.
- [EBXM1999] *ebXML: ebXML Terms of Reference.* <http://www.ebXML.org>, 1999, Abruf am 2000-07-14.
- [Emme1990] *Emmelhainz, M.A.*: Electronic Data Interchange: A Total Management Guide. Van Nostrand Reinhold, New York 1990.
- [Fowl1997] *Fowler, M.; Scott K.*: UML Distilled. Addison Wesley, Reading 1997.
- [HiFe1989] *Hill, N.C.; Ferguson, D.M.*: Electronic Data Interchange: A Definition and Perspective. In: EDI Forum: The Journal of Electronic Data Interchange 1 (1989) 1, S. 5-12.
- [Huem1999] *Huemer, C.*: Modeling Inter-Organizational Systems with UML. In: *Klein, S.; Gricar, J.; Pucihar A. (Hrsg.)*: Proceeding of the 12<sup>th</sup> Bled Electronic Commerce Conference. Bled 2000, S. 566-580.
- [Huem2000] *Huemer, C.*: XML vs. UN/EDIFACT or Flexibility vs. Standardization. In: *Klein, S.; O'Keefe, B.; Gricar, J.; Podloga, M. (Hrsg.)*: Proceeding of the 13<sup>th</sup>

- Bled Electronic Commerce Conference. Bled 2000, S. 680-692.
- [Koto2000] *Kotok, A.:* Extensible and More. <http://www.xml.com/pub/2000/02/23/ebiz/index.html>, 2000-02-23, Abruf am 2000-07-14.
- [ISO1996] *ISO:* The Open-edi Reference Model. ISO/IEC JTC1/SC30 IS 14662, Geneva 1996
- [PeWe1997] *Peat, B.; Webber, D.:* Introducing XML/EDI – „the E-business framework“. <http://www.geocities.com/WallStreet/Floor/5815/start.htm>, 1997, Abruf am 2000-07-14.
- [SwSw1992] *Swatman, P.M.C.; Swatman P.A.:* EDI System Integration: A Definition and Literature Survey. In: The Information Society 8 (1990) 3.
- [TMWG1998] *TMWG:* Reference Guide – “The Next Generation of UN/EDIFACT” (Revision 12). CEFACT/TMWG/N010/R12, <http://tmwg.harbinger.com/documentlist.html>, 1998, Abruf am 2000-07-14.
- [TMWG1999] *TMWG:* TMWG Recommendations on XML. CEFACT/TMWG/N089/R5. <http://tmwg.harbinger.com/documentlist.html>, 1999, Abruf am 2000-07-14.
- [TMWG2000] *TMWG:* Unified Process for UN/CEFACT. CEFACT/TMWG/N090, <http://tmwg.harbinger.com/documentlist.html>, 2000, Abruf am 2000-07-14.
- [WeBW2000] *Weitzel, T.; Buxmann, P.; v. Westarp, F.:* A Communication Architecture for the Digital Economy – 21<sup>st</sup> century EDI. In: Proceedings of the 33<sup>rd</sup> Hawaii International Conference on System Sciences. Maui 2000.
- [Wesk1999] *Weske, M.:* Business-Objekte: Konzepte, Architekturen, Standards. In: Wirtschaftsinformatik (1999) 1.