

B2B-Integration mit ebXML

Handel mit Format

Michael Merz

Für den elektronischen Austausch von Geschäftsdaten hat sich neben weiteren Standards das XML-basierte Rahmenwerk ebXML etabliert. Wie praxistauglich das Kommunikationsmodell ist, zeigt das papiNet-Projekt, ein Netzwerk aus über 80 Handelspartnern.



PapiNet ist ein internationales Kommunikationsnetzwerk, das innerhalb weniger Jahre auf über 80 Teilnehmer anwachsen konnte. Eine wesentliche Voraussetzung dafür war die konsequente Nutzung von XML-Standards und die Entwicklung einer einheitlichen, preiswerten Software zur Kommunikation zwischen den Teilnehmern.

Das papiNet-Projekt dient der Integration von Papierherstellern und ihren Kunden. Auf der Anbieterseite finden sich über zwanzig meist skandinavische und amerikanische Unternehmen wie UPM Kymmene, Stora Enso oder International Paper, während auf der Kundenseite Verlage und Druckereien vertreten sind: Der Axel Springer Verlag, Burda, Gruner & Jahr, News International, Time Warner und andere. Hinzu kommen Verarbeitungsbetriebe für Ver-

packungsmaterial, Aufkleber, Lamine sowie eine Vielzahl von Unternehmen im Bereich der Papierlogistik.

Wie viele andere Branchen litt die Papierindustrie zusammen mit ihren Kunden in den 90er-Jahren unter einer nicht-standardisierten und daher kostspieligen Kommunikation beim Austausch von Geschäftsdaten: Auf Basis des Edifact-Standards waren Dokumententypen wie Bestellungen, Lieferpapiere oder Rechnungen nur sehr vage spezifiziert, ganz zu schweigen von Prozessdefinitionen oder einheitlichen Übertragungsprotokollen über das Internet. Stattdessen versuchte man die B2B-Integration über X.400 mit Hilfe kostspieliger VAN-Betreiber (Value Added Network) voranzubringen. Eine Integration zwischen zwei Geschäftspartnern war damit aufwen-

dig, teuer und mangels Standards jedes Mal ein individuelles Projekt. Abgesehen davon kommunizierten die meisten Geschäftspartner nach wie vor per Fax oder E-Mail, sodass Übertragungsfehler an der Tagesordnung waren.

Erst in den 90er-Jahren führten Entwicklungen wie Open EDI, XML sowie die preiswerte Nutzbarkeit des Internet zu einem allgemeinen Umschwenken auf eine umfassende Betrachtungsweise des Integrationsproblems. Insbesondere das Standardisierungsprojekt Open EDI, das nicht zuletzt auf die Modellierung von Geschäftsprozessen sowie auf eine einheitliche Beschreibung von Schnittstellen für den Datenaustausch abzielte, floss seit dem Jahr 2000 in die Entwicklung des ebXML-Standards (Electronic Business XML) ein, den die Papierindustrie schließlich als Grundlage für ihre Datenkommunikation auswählte.

Peer-to-Peer ohne Papier

Zunächst ist der Ansatz des dezentralen, direkten Austausches von XML-Dokumenten zwischen Marktteilnehmern zu unterscheiden von der zentralisierten, Hub- beziehungsweise marktplatzbasierten Integration. Im ersten Fall existiert keine dritte Partei in der Mitte, weder technisch noch organisatorisch, während im zweiten Fall zumindest eine zentrale Weiterleitung und Konvertierung von Nachrichten erfolgt (Messaging-Hub). Bei einem Marktplatz kommen zentrale Applikationen hinzu wie Kataloge, Rechnungspräsentation, Statistiken et cetera. Dem gegenüber sehen Anwender einige Vorteile des dezentralen Messaging, für das man sich daher bei papiNet entschied:

- die Vermeidung potenzieller Fehlerquellen durch zusätzliche Kommunikationsverbindungen oder Konverter,
- Dritten wird der Zugang zu teilweise geheimen Daten verwehrt (viele Marktplätze scheiterten aufgrund des Misstrauens ihrer Teilnehmer),
- Ende-zu-Ende-Authentisierung von Dokumenten: Hier kann der Empfänger die Authentizität des vom Sender signierten Dokuments prüfen. Beim Austausch über einen Marktplatz wird diese Signatur aufgrund der Konvertiervorgänge automatisch ungültig.

Auf der Basis dieser Voraussetzungen sowie der sich abzeichnenden Entwicklungen im XML-Umfeld entschied sich das papiNet-Konsortium, einen Dreiklang aus Standards zu schaffen (das „papiNet-Prinzip“):



- Die B2B-Integration einer großen Anzahl Handelspartner lässt sich auf der Basis des Business-Standards ebXML effektiv und kostengünstig realisieren.
- Mit zunehmender Nutzerzahl verringert sich der Aufwand sichtlich: In einem Praxisbeispiel konnte der Integrationsaufwand von 7 Monaten beim ersten Partner auf eine Woche beim vierten reduziert werden.
- Trotz technischer Hürden kann man mit einem verstärkten Datenaustausch via ebXML in Zukunft rechnen – Voraussetzung dafür ist allerdings die einfache Installation, Pflege und Bedienung der eingesetzten Messaging Software.

1. Definition einheitlicher Dokumentschemata, die für die Industrieteilnehmer verbindlich sind.
2. Definition organisationsübergreifender Geschäftsprozesse, welche die Gepflogenheiten der Papierbranche und ihrer verschiedenen Subbranchen sowie Geschäftsbereiche reflektieren.
3. Entwicklung einer ebXML-konformen Kommunikationssoftware, die quasi als „XML-Faxgerät“ von allen beteiligten Unternehmen einfach zu installieren ist und eine einfache Integration mit den Geschäftspartnern erlaubt.

Zwar war der ebXML-Standard dokumentiert, es mangelte jedoch an einsetzbaren Implementierungen. Der Einsatz kommerzieller Software hätte andererseits hohe Lizenz- und Integrationskosten für jeden einzelnen papiNet-Teilnehmer verursacht. Aus diesem Grunde fiel die Entscheidung, eine Kommunikationssoftware zu entwickeln, die auf Open-Source-Komponenten aufsetzt und damit für den Anwender keine Lizenzkosten verursacht. Gleichzeitig sponserten Unternehmen der Papierindustrie die Entwicklung der Kommunikationssoftware, die nun allen Geschäftspartnern kostenlos zur Verfügung steht.

Der ebXML-Envelope gibt Aufschluss über die kommunizierenden Partner, die Art der Nachricht, den Dokumententyp und weitere für den Informationsaustausch wichtige Details.

Auf diese Weise schlug man zwei Fliegen mit einer Klappe: Erstens konnte man sicherstellen, dass alle Teilnehmer ebXML als Basis der B2B-Integration einsetzen und zweitens verbreitete sich das Integrations-Know-how durch die Verwendung des Standards erheblich schneller als bei einer unkoordinierten Herangehensweise. Alle papiNet-Teilnehmer setzen daher – zum Teil seit über zwei Jahren – die Software ein.

ebXML als Basis

Der einheitliche Einsatz standardkonformer Software bewirkte vor allem auch, dass über 90 Prozent der Teilnehmer XML-Dokumente signiert und verschlüsselt übertragen. Normalerweise stellt es eine besondere Herausforderung dar, wenn die Zertifizierungsautoritäten der jeweiligen Partner untereinander nicht interoperabel sind. Im Falle von papiNet wird die Zertifizierung sowie die Verteilung der Zertifikate ebenfalls einheitlich betrieben, um eine rasche Verbreitung nicht zu behindern. Im Folgenden sollen nun einige techni-

sche Aspekte sowie Erfahrungen bei der B2B-Integration mit ebXML näher beleuchtet werden.

ebXML ist eine Sammlung aus aufeinander abgestimmten XML-Schemata zur B2B-Integration, die verschiedene Aspekte der Message-basierten Kommunikation abdecken (siehe [1]). Der in der Praxis gängigste Standard ist der ebXML Message Service (ebMS). Zum einen legt er fest, wie die Kommunikationsteilnehmer die Dokumentenübertragung und deren Quittierung abwickeln. Zum anderen definiert er einheitliche XML-Strukturen für den Transportumschlag (Envelope). So enthält ein ebXML-Envelope beispielsweise URIs (Uniform Resource Identifiers) und Identifikatoren der beteiligten Kommunikationspartner, eine Identifikation der Nachricht und des Transaktionstyps, ein Inhaltsverzeichnis (Manifest) der angehängten Geschäftsdokumente und gegebenenfalls eine elektronische Signatur auf Basis des XML Signature Standard. Als Inhalt (Payload) lassen sich beliebige XML-Dokumente, EDI-Nachrichten oder binäre Attachments in beliebiger Zahl an den

Listing 1: ebXML-Envelope

```
POST /pontonxp/SoapListener HTTP/1.1
Host: 192.168.200.63
SoapAction: "ebXML"
Content-Type: multipart/related; type="text/xml";
boundary="--ch35.001f"Troll"> <h$f$>--ch35.001f"Troll"> <h$f$>--_Part_63_1930164715.1067431429161"

--ch35.001f"Troll"> <h$f$>--ch35.001f"Troll"> <h$f$>--_Part_63_1930164715.1067431429161
Content-Type: text/xml
Content-Transfer-Encoding: 8bit
Content-Id: <ebXMLheader>

<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Header>
    <eb:MessageHeader xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
      soap-env:mustUnderstand="1" eb:version="2.0">
      <eb:From>
        <eb:PartyId eb:type="PontonCertificate">papitest</eb:PartyId>
        <eb:PartyIdhttp://papitest.ponton-consulting.de:8080/pontonxp/SoapListener</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="PontonCertificate">StoraEnso</eb:PartyId>
      </eb:To>
      <eb:CPAId>www.papiNet.org/data/CPABasicHTTP.xml</eb:CPAId>
      <eb:ConversationId>CID-1067431428973@ponton-consulting.de</eb:ConversationId>
      <eb:Service eb:type="papiNet">Production</eb:Service>
      <eb:Action>Invoice</eb:Action>
      <eb:MessageData>
        <eb:MessageId>MID-1067431428833@ponton-consulting.de</eb:MessageId>
        <eb:Timestamp>2003-10-29T12:43:48</eb:Timestamp>
        <eb:TimeToLive>2003-10-29T12:46:48</eb:TimeToLive>
      </eb:MessageData>
      <eb:DuplicateElimination />
    </eb:MessageHeader>
    <eb:SyncReply xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
      soap-env:mustUnderstand="1" eb:version="2.0" soap-env:actor="http://schemas.xmlsoap.org/soap/actor/next" />
    <eb:AckRequested xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
      soap-env:mustUnderstand="1" eb:version="2.0" eb:signed="true" />
  </soap-env:Header>
  <soap-env:Body>
    <eb:Manifest xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd" eb:version="2.0">
      <eb:Reference xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="cid:MID-1067431428833@ponton-consulting.de.xml" xlink:role="http://www.papiNet.org/ebXMLAttachment/Roles/BusinessDocument">
        <eb:Schema eb:location="http://www.papiNet.org/data/InvoiceV1R10.dtd" eb:version="V1R10" />
        <eb:Description xml:lang="en">papiNet Payload</eb:Description>
      </eb:Reference>
    </eb:Manifest>
  </soap-env:Body>
</soap-env:Envelope>
```

Envelope anhängen. Bestandteil von ebXML ist außerdem ein XML-Schema zur Spezifikation von Geschäftsprozessen (Business Process Specification Schema – BPSS). Auf Basis dieses Schemas lassen sich Rollen, Prozesse und Aktivitäten unterscheiden.

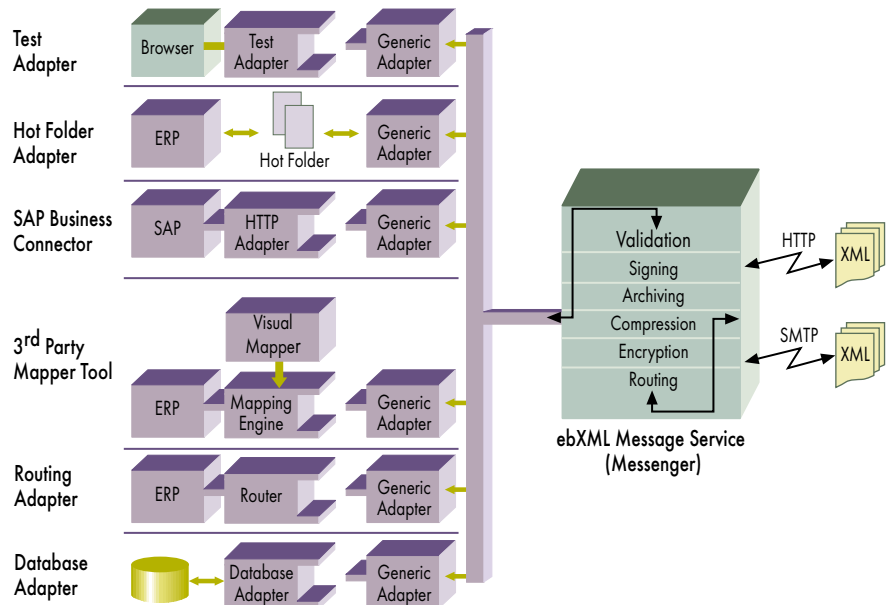
Des Weiteren ist CPPA (Collaboration Partner Profile and Agreement) ein Teilstandard von ebXML. Dokumente dieses Typs legen fest, welche Partner zusammenarbeiten und bei welchen Geschäftsprozessen sie Dokumente welchen Typs übertragen beziehungsweise Aktivitäten beim Partner auslösen. Die Verzahnung mit dem ebMS erfolgt mit der Verwendung einheitlicher URIs zur Identifikation von Partnern, Nachrichtentypen und Transaktionen. Der CPPA-Standard zerfällt in zwei Teilkomponenten: CPP und CPA (Collaboration Partner Profiles und Collaboration Partner Agreements).

Ein Profil beschreibt dabei die Rolle, die ein Partner im Rahmen eines Prozesses einnehmen kann – zum Beispiel Lieferant oder Kunde – und schränkt diese gegebenenfalls im Hinblick auf bestimmte Transaktionen und Dokumenten-Subsets ein. Das Profil beschreibt auch, anhand welcher Protokolle (etwa HTTP(S) oder SMTP) ein Partner mit seinem Gegenüber in Verbindung treten kann. Im Idealfall lässt sich aus zwei solchen Profilen ein CPA automatisch „destillieren“, wenn es keine Konflikte bei den Prozessen und Protokollen gibt, andernfalls ist ein Einigungsprozess zwischen den Parteien nötig.

Schließlich ist der ebXML Registry Service als Teilstandard zu nennen, der einen Verzeichnisdienst definiert, in dem die Teilnehmer Ressourcen wie CPPs ablegen und publizieren können.

Die Idealvorstellung von ebXML ist, dass potenzielle Geschäftspartner (technisch) zueinander finden, indem im Rahmen branchenüblicher Standardprozesse Profile definiert, publiziert und zu CPAs destilliert werden und der ebMS anschließend einen Protokoll- und prozesskonformen Austausch von Geschäftsdokumenten sicherstellt. Die Parameter zur Übertragung einer XML-Datei enthält das CPA.

Dies ist zum heutigen Zeitpunkt leider noch eine Vision, da die meisten ebXML-Implementierungen zumeist nur den ebMS abdecken und beim Registry Service beziehungsweise der Prozessbeschreibungssprache Konkurrenz aus dem Web-Service-Lager droht. Eine Liste der verfügbaren kommerziellen beziehungsweise prototypischen



Der Messenger ist für die Kommunikation mit den Handelspartnern – etwa das Verschicken, Verschlüsseln oder Signieren der zu verschickenden Daten – verantwortlich. Die auf der linken Seite befindlichen Adaptern bieten verschiedene Alternativen für die Backend-Integration in die Firmenanwendungen. Sie bestehen aus einem allgemeinen (Generic Adapter) und einem spezifischen Teil (Test-Adapter, HTTP-Adapter et cetera). Wenn die mitgelieferten Adaptern nicht ausreichen, lassen sich individuelle programmieren.

Implementierungen der jeweiligen Standards findet sich auf der ebXML-Website (www.ebxml.org).

Dennoch sind Unternehmen schon gut bedient, wenn sie wenigstens die Vorteile des ebMS nutzen können, da sich aufgrund eines gemeinsamen formellen Vokabulars der Integrationsaufwand erheblich reduzieren lässt.

Einheitliche Sprache

So auch im papiNet-Projekt. Die Anforderung war hier, nach außen einheitlich ebXML zu „sprechen“ und nach innen auf unterschiedlichem Wege eine Integration zur jeweiligen ERP-Anwendung (Enterprise Resource Planning) des Partners vorzunehmen. Aus diesem Grunde wurden zwei Hauptkomponenten entwickelt: der Messenger und Adaptern (siehe Abb. 1).

Der Messenger führt die Validierung, Authentifizierung, Archivierung, Verschlüsselung sowie die Erzeugung und den Austausch von ebXML-Nachrichten durch. Diese Implementierung des ebMS sieht dazu zahlreiche Verarbeitungsfilter vor, die man beliebig erweitern kann.

Mit der internen Anwendung kann der Messenger über einen oder mehrere

Adaptern verbunden sein, die auf unterschiedliche Weise den Datenaustausch mit dem Backend durchführen. Als Schnittstelle dient hier der Generic Adapter, ein Java-Paket zur Steuerung der Dokumentenübertragung. So wird zum Beispiel ein Database-Adapter eingesetzt, der XML-Dokumente direkt anhand von SQL-Abfragen aus der Datenbank erzeugt. Umgekehrt erlaubt er die Speicherung von XML-Dokumenten in der Datenbank. Der Hot-Folder-Adapter gestattet einen einfachen Austausch über das Dateisystem, während der HTTP-Adapter die direkte Verbindung zu anderen Anwendungen (zum Beispiel dem SAP Business Connector) ermöglicht. MS Office 2003 oder auch OpenOffice erlauben beispielsweise den Export von XML-Dokumenten direkt in das Ausgangsverzeichnis, sodass eine direkte Integration zwischen der Erstellung eines Bestellformulars und dem Message Service des Kunden möglich ist.

Einige papiNet-Teilnehmer haben über die mitgelieferten Generic Adapter hinaus Individualadaptern entwickelt, die direkt mit einem ERP-System verbunden sind. Hierzu steht ein Adaptor-Framework zur Verfügung, das die Integration via Programmierschnittstellen unterstützt.

Besonders interessant ist die Nut-

zung des noch relativ jungen, Anfang 2002 entwickelten Standards XML Signature: Dieser erlaubt, Signaturobjekte in XML zu repräsentieren und damit in die Struktur eines XML-Dokuments einzubetten. Das gilt auch für den ebXML-Envelope selbst. Der XML-Signature-Standard ermöglicht es, mehrere Datenobjekte „auf einen Streich“ zu signieren. Die Objekte müssen dabei nicht zusammenhängen, sondern man kann sie sogar entfernt per URI referenzieren. Gleiches gilt auch für ebXML: Hier werden Envelope und Payload-Attachments gemeinsam signiert. Der Vorgang kettet beide aneinander, sodass ein Teilnehmer weder den (geschäftlich relevanten) Inhalt des Payload-Dokuments noch die Übertragungsdaten des Envelope noch die Verkettung unbemerkt verändern kann.

Einsatz von Signaturen

Obwohl der Teilbaum der XML-Signatur in den ebXML Envelope eingebettet ist, lässt sich die Signatur immer noch verifizieren. Normalerweise müsste eine Veränderung des Originaldokuments (Envelope) die Signatur ungültig werden lassen. XML Signature sieht hier jedoch qua Standard die Möglichkeit der „Embedded Signature“ vor: Sie kann explizit in das signierte Objekt eingebettet werden. Ein standardkonformer Verifikationsalgorithmus kann das erkennen und bezieht die Signaturdaten nicht in die Prüfung ein.

Nach über zwei Jahren Laufzeit lässt sich festhalten, dass mit papiNet ein recht ausgewogenes Verhältnis aus Standardisieren und „offen lassen“ sowie aus Zentralisieren und Dezentralisieren erreicht wurde. Während der Dreiklang „Prozesse-Dokumente-Message Service“ streng einheitlich definiert wurde, reicht die Standardisierung jedoch nicht zu weit in die Privatsphäre der Teilnehmer hinein: Die Backend-Integration kann jeder auf seine Weise realisieren und bei der Kommunikation mit Partnern ist kein zentraler Hub involviert, den dank der weitgehenden Standardisierung auch nie jemand vermisste. Zentralisiert ist die Weiterentwicklung von XML-Schemata und der Messenger-Software, deren Support sowie die Veröffentlichung von Zertifikaten. Dezentral funktioniert die Peer-to-Peer-Kommunikation über das Internet.

Die zu diesem Zweck eingesetzte Messenger-Software ist vollständig in

Java entwickelt und kann auch von nicht papiNet-Teilnehmern als Trial-Version heruntergeladen und getestet werden (erhältlich unter www.pontonconsulting.de). Dabei sind zwei Kommunikationspartner zu simulieren, indem die Installation des Messenger beispielsweise auf zwei PCs erfolgt. Das Installationspaket enthält auch das JRE 1.4.1, sodass keine besonderen Voraussetzungen an die Systemumgebung bestehen.

Es versteht sich, dass Standards wie ebXML nur einen Rahmen liefern können, der erst durch detaillierte Konventionen innerhalb einer Community wie papiNet ausgefüllt werden muss. So beispielsweise bei der einheitlichen Benennung von Partnern, einem besonders komplexen Problem. Historisch betrachtet verfügt jeder Teilnehmer über mindestens eine Identifikation seiner Geschäftspartner – dies können Kunden- oder Lieferantennummern sein, aber auch Umsatzsteuer-IDs. Eindeutig sind diese IDs jedoch zumeist nur in der Domäne des jeweiligen Unternehmens, und selbst wenn sie global eindeutig sind (wie Umsatzsteuer-IDs, IANA-Nummern oder IP-Adressen), ist noch lange nicht sichergestellt, dass alle Teilnehmer den gleichen Nummernkreis verwenden. Aus diesem Grunde unterschied man bei papiNet zwischen der „Partner ID“, die bei einem Teilnehmer lokal zur Identifikation des Gegenübers dient und einer oder mehreren PartyIDs, wie sie ebXML im CPP, CPA und Envelope vorsieht. PartyIDs sind global eindeutig. Der Messenger hat die Aufgabe, die verschiedenen IDs zu verwalten beziehungsweise ineinander zu übersetzen.

Die Einigung auf Nummernkreise ist nur ein Beispiel für den Vereinbarungsprozess, der jenseits des ebXML-Standards noch erforderlich ist. Um daher den allgemeinen ebXML-Standard als Grundlage eines Handelsnetzes zu nutzen, entwickelte man im Rahmen des papiNet-Projekts Interoperabilitätsrichtlinien, die die offenen Aspekte von ebXML konkretisieren.

Weitere typische Integrationshürden stellen Firewalls, Proxies und die Verwendung lokaler IP-Adressen dar. Wenn zwei Messenger beispielsweise per HTTP kommunizieren, müssen sie einander auch über diese Hindernisse hinweg erreichen können. Die technische Lösung ist für einen Systemadministrator häufig trivial, jedoch kann ihre Umsetzung bei größeren Organisationen die Integration um Wochen

verzögern. Eine checklistenbasierte Koordination und die regelmäßige Einbeziehung aller relevanten Stellen der Geschäftspartner hilft, unnötige Verzögerungen zu minimieren.

Auf dem Weg zum „XML-Faxgerät“

Zusammenfassend bestätigt das Projekt papiNet einmal mehr Metcalfe's Gesetz: Ebenso wenig wie das erste Faxgerät jemandem nützt, profitiert auch der erste Nutzer eines ebXML Message Service. Wenn jedoch Teilnehmer ihre Kommunikationsverbindung vom ersten auf den zweiten, dritten, vierten Partner erweitern, können sie den Nutzen dieses Standards unmittelbar erleben. So benötigte ein großer Hamburger Verlag für die Integration mit dem zweiten Papierhersteller nur noch ein Drittel des Aufwands, der für den ersten erforderlich war – und für den vierten weniger als fünf Prozent.

Darüber hinaus haben sich auch andere Branchen vom papiNet-Prinzip inspirieren lassen: So erfolgt der Austausch administrativer Daten zwischen Anzeigenagenturen und Verlagen im Rahmen des Projekts „Open Booking System“ auf der Basis der gleichen Software. Lediglich die branchenspezifischen Schemata und Stylesheets waren für die neuen Teilnehmer zu konfigurieren. Hier starteten bis Anfang 2004 mehrere Verlage und Anzeigenagenturen mit der Integration.

Ähnlich wie sich Faxgeräte zunächst schleppend, dann jedoch rasch im Geschäftsverkehr durchsetzten, so entwickeln sich ebXML Message Services als „Faxgeräte“ für den XML-Datenaustausch. Das entscheidende Kriterium ist dabei immer die leichte Handhabbarkeit der Kommunikationssoftware. Hier ist für die Zukunft Positives zu erwarten, da sich der Gebrauch von XML als „Lingua Franca“ inzwischen auf Datenbanken über Office-Werkzeuge bis zum Datenaustausch erstreckt. (ur)

MICHAEL MERZ

ist Geschäftsführer der Ponton Consulting GmbH, die das papiNet-Projekt begleitet.

Literatur

- [1] Stefan Mintert; E-Commerce; Big Business; Grundlage für globalen Handel; iX 1/2002, S. 110 