

XML-Based
B2B Frameworks

国際的マーケットプレイスの実現に、
独占的な技術は要らない

ebXMLイニシアチブ議長 クラウス・ノーヨック氏 インタビュー



XML Magazine 編集部

「電子的にやりとりされるビジネス情報を、全ての団体が相互に、安全で、マナーに沿った方法で利用するために、XMLベースのオープンなインフラストラクチャーを提供する」それがebXMLの掲げる目標である。

去る10月に行なわれた「B2B Big Bang」のために、ebXMLの議長クラウス・ディーター・ノーヨック氏が来日した。「日本では皇居や芝公園を散歩するのが好きだ」というノーヨック氏に、ebXMLの活動について、また標準化の道のりについて聞いた。

ビジネスの垂直、 水平方向に適用可能な技術

まず、ebXMLの具体的な活動について教えてください。

普通、企業は別の企業と取引を行なうために、さまざまな書式を持っています。e-ビジネスモデルの取引を行なう際には、電子的にデータの交換やメッセージのやり取りを行ないます。ebXMLでは、このような電子商取引やデータ交換のために書式や仕組みを作成します。

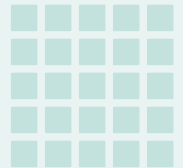
これらの仕組みは、現実のビジネスプロセスに密接に絡んでいるものですから、データを交換するという立場からではなく、ビジネスプロセスをどのように規定するかという立場から考慮されています。

このビジネスプロセスは、UMLで設

計しXMLで記述することになります。アプリケーションは、XMLで記述されたビジネスプロセスのモデルをダウンロードし、そこからどのような情報がいつ、誰に送信される必要があるのかについて判断し、動的に構築されるようにします。

メッセージを受け取る側である、別のビジネスプロセスのモデルを持った別の組織では、どんな情報が必要なのか、あらかじめ分かれますので、この仕組みによって、複数の企業を繋いでいくことが次のステップになります(図2)。

この時、問題のないデータだけを扱うのであればいいのですが、情報に例外や誤りのある場合があります。ビジネスプロセス上、データに誤りがあったとき、単にエラーとして扱うのではなく、何かの処理を行なって、その情報を受け取ることができるような仕組みが必要とされます。



この仕組みにより、メッセージ構造によって表現された、ビジネスオブジェクトを構成するデータの相互運用性を保ちます。このオブジェクトは、ビジネスの垂直、水平の両方向に適用できるものです。“レゴブロック”をご存知ですか？あいったものを使って、ビジネスプロセスを構築できるようになるイメージです。

いまお伺いしたようなebXMLの活動は、企業の中のどのような層を対象とするものですか？

ebXMLが成功するためには、ふたつの層の人々に働きかけを行なう必要があります。

ひとつは、企業内のさまざまなモデル、ビジネスモデルや情報のモデルを提供する側の人々です。これらの人々は、企業をデータではなく、ビジネスモデルの立場から診断します。つまりIT（情報技術）の見地ではなく、人々が何を行なうのか、

というビジネスプロセスの見地から物事を見るのです。これらの人々にインタビューをして、どのようなことをしたら良いのかについてのモデルを収集します。我々は、たくさんのモデルを収集し、それを交換するという仕事を行ないます。

もうひとつの層は、情報システムを構築するベンダーの人々です。こちらは今までお話ししてきたビジネスモデルではなく、ebXMLのフレームワークに関する働きかけです。彼らはアプリケーションを構築します。そのアプリケーションにebXMLの収集したビジネスモデルを読み込んで、それを情報の流通モデルとして、実際にメッセージを交換できるようにするのです。

このアプリケーションには、さらにふたつのもがあると思います。ひとつは大企業向けで、ebXMLの仕様による企業システムのようなアプリケーションです。もうひとつは、ソフトウェアベンダーによる、小さな企業向けのアプリケーションで、例えばWindowsのような汎

用OS上で動作するものです。

大企業と中小企業では使うソフトウェアが全く異なるのですが、これらのソフトウェアによって企業の規模に関わりなく同じebXMLが利用できることとなります。企業間で情報を共有して扱うことができるようになるでしょう。

その場合、企業間におけるビジネスモデルの違いをどのように吸収してゆくのですか？

大企業では、ひとつのを行なうために、いくつものステップを踏む必要があります。一方、小さな企業では、ひとつかふたつのステップで十分です。ですから、もし大企業で働くとするれば多くの作業が必要ですし、小さな企業では少ない作業で片付きます。

ebXMLの仕組みでは、大企業側で何が必要なのかということ、また、できることが限られる小さな企業側では、何ができるのかということ、それぞれシ

図1：ebXMLのWebページ

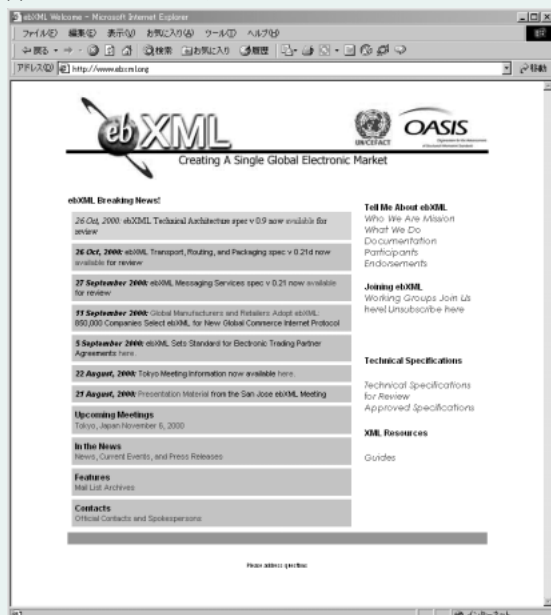
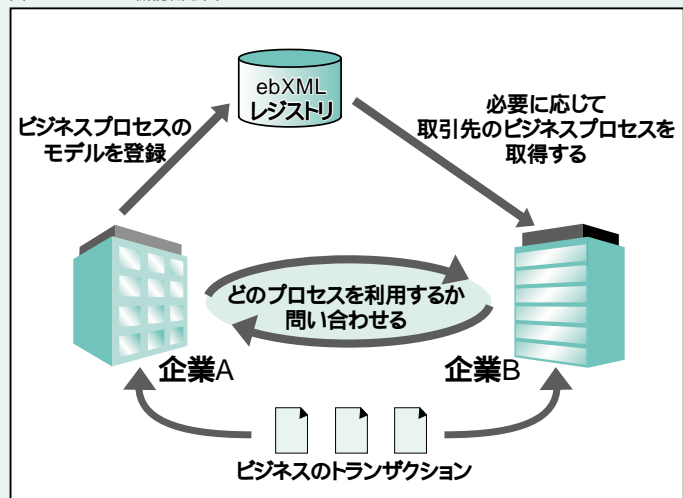


図2：ebXMLの機能概念図



システムが判別し、この両者のマッチングを行なうことがシステム全体の役割だと考えていただければいいでしょう。

大企業ではたくさんの方ができます。しかし小さな企業では限られたことしかできません。もし、あなたが大企業で働いているとして、何がしかの明確なビジネスモデルをすでに持っているとしたら、取引先となる小さな企業がそれに沿って取引を行なうには、ビジネスモデルが複雑過ぎることもあるでしょう。このような場合、ビジネスモデルの見直し、あるいは変更を行なう必要があると思われる。



XMLによってダイアグラムを統一的に保存可能に

もともとXMLが初めにあったのですか？ それともこのようなアイデアを実現するために、XMLがたまたま選ばれたということでしょうか？

あとの方ですね。もともとアイデア自体があって、XMLが登場したことによってそれを実現できるようになったのです。具体的にはXMI（注）という、XMLによってビジネスモデルを記述する統一された形式があります。XMLを採用したもっとも大きな理由ですが、XMLは機械によって判別可能だということです。これは重要なことです。

例を示しましょう。まずUMLがある

とします。UMLはダイアグラムですが、この保管形式が今まではバラバラでした。そのため実際にファイルを開いてダイアグラムを見るまでは、それが必要なものかどうか分かりませんでした。しかしXMLによってダイアグラムを統一して保存できるようになったため、人間ではなくプログラムが判断を下せるようになったのです。

注：XMIはXML Metadata Interchangeの略。Object Management Groupで策定されており、UMLベースのモデリングツール間で、メタデータに関する情報を交換するための言語。

現状のXMLに不満はありますか？

ひとつ挙げるとすれば、さまざまな企業が独自にEDIをXMLに置き換えようとした結果、EDIのように統一されたフォーマット、これはebXMLのような標準のことで、これが無くなってしまったことです。XMLには、そもそも事前に定義されたタグというものが存在しないので、EDIをXMLに変換したそれぞれの実装が独自のタグを定義しており、これが互換性の妨げになっているということです。

そのためebXMLではビジネスモデル、ビジネスプロセス、ビジネスオブジェクトの3つで共通した辞書を用意し、これらを協調させることで互換性を保つようにはしようと思っています。鍵はビジネスオブジェクトにあって、たとえば各企業が同じレゴのブロックを使って別々の建物を作ることを目指しています。

ビジネスモデル、ビジネスプロセス、ビジネスオブジェクトに相互運用性を持たせることで、先ほど触れたようにプログラムによって自動処理するということも可能になるのです。

今後の活動については、どのようになっていますか？

全体のフレームワークが来年の5月ぐらいにでき上がる予定です。それから、ビジネスモデルやビジネスモデルなどのコンテンツを作る段階に移ります。今のモデルはスタティックなものなのですが、これをもっとダイナミックなものにしていきます。

今のebXMLでは、やり取りされているメッセージのモデルやプロセスはメッセージの外部にありますが、将来は必要なものがドキュメントそのものの内部にあるモデルとなります。

ドキュメント自身がビジネスモデルを理解していれば、アプリケーションやエージェントのようなものが自動的にビジネスプロセスを判断し、適切な処理を行なえるようになります。これが大体、今

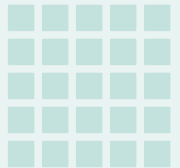
マイクロソフトとの関係、標準化の行く末

後5年くらいのプランになります。

マイクロソフトのBizTalkについてはどう思われますか？

我々はマイクロソフトにもebXMLに参加するよう呼びかけています。実際、ミーティングにも一度参加しております。しかしその時に彼らは、もうここに来るのはやめるという判断をしました。我々の活動についてはよく見ていただいて、アナウンスメントなどではebXMLに統合して行くといっています。ですが、マイクロソフトのことですからね...

ある標準の問題というのは、参加者全員がそれに従わなければいけません。独占的な技術がそこに混入すれば、全体が



立ち行かなくなります。マイクロソフトには継続して話をしているのですけれど...。我々は彼らがebXMLに参加することを期待しています。幸い、我々の活動にはXMLのメジャープレイヤーが参加していますので、そういったことが圧力になって、マイクロソフトを標準仕様に向かせることになるかと思っています。最終的にはユーザーが、どちらを使うのかを決めることになるでしょう。

ebXMLを利用する上で、もっとも大切なこととは何でしょうか？

ソフトウェアベンダーが、アプリケーションを作るのだと思います。

それは具体的にはどういうことですか？

小さな企業に対してという意味です。結局、EDIは中小企業までには行き渡りませんでした。なぜならEDIは非常に高価で、日本円にして700万円から1000万円ほどかかります。小さな企業に対しては、3万~4万円という値段で提供できなければなりません。

XMLのような新しいテクノロジーによるデータ交換が成功するためには、ソフトウェアベンダーがそのテクノロジーを利用して、いかに安価にソフトウェアを作れるか、にかかっていると思います。ebXMLは、実際にそれを行なおうとしているのです。



最後にユーザーへのメッセージをお願いします。

我々は、ひとつの国際マーケットプレイスを作成したいと思っています。ぜひ、日本のソフトウェアベンダーに、ebXMLに対応したソフトウェアを作成してもらえよう、要望をあげていただければと思っています。仕様については我々の方から提供していますので、ユーザーの方からベンダーに対して働きかけを行っていただければと思います。そこに独占

ebXML記者説明会開催

フレームワークの完成は予定よりも早く

クラウド ノーヨック氏へのインタビューを行なった10月の時点では、フレームワーク完成は来年5月の予定となっていたが、11月に東京、臨海副都心にあるタイム24ビルで行なわれたebXML記者説明会の席で、「来年3月には最終バージョン 1.0として完成する」とebXMLイニシアチブメンバーによる説明があった。

現バージョン 0.9でも電子商取引が可能であることは検証済みとのことだが、11月のebXML東京フォーラムでは、フレームワークをより洗練されたものにするための話し合いが行なわれた。

また、完成後のメンテナンスに関しては、来年2月に行なわれるバンクーバーフォーラムで議論する予定。ebXMLワーキンググループにより最終検証を行ない、来年5月のウィーンフォーラムで最終バージョン 1.0の発表というスケジュールになっており、ebXMLではフレームワークの完成に向けて着々と準備を進めている。



ebXMLのサイトではバージョン 0.9についてレビューしている



記者説明会では、バージョン 0.9のデモが行なわれた

XML-Based
B2B Frameworks

RosettaNet、BizTalkを利用した 企業間取引のサンプル

B2Bフレームワーク実装の ケーススタディ

XML Magazine 編集部

「XMLでB2B」という言葉ばかりが先行している感のある昨今であるが、実際のところXMLを使ってB2Bを行なうときに、どのようなかたちで実装されているのだろうか？

東芝アドバンストシステムによるリファレンス実装の例を見ながら、必要な知識、技術についてあらためて検証してみる。

システムの目的

今回ご紹介するシステムは架空のPCメーカーを想定し、パーツ調達を電子化するというシナリオの下に実装されている。社内の購買システムを、社外の取引先企業や、外部に構築されたマーケットプレイスに接続し、そこで得られた情報を基に取引を行なうものである。PCメーカーの社内システムには、web-Method B2Bサーバーを接続し、このサーバーを介して、社外との接続を行なう。

今回は、RosettaNetとBizTalkのプロトコルをサポートし、CPUとハードディスクドライブの調達をシミュレーションしている。完成されたデモの流れを見ながら、システムの各部の動きと、その中で何が行なわれているのか解説しよう。

システムの全体像

図1の画面は、このPCメーカーのイントラネットサイトで閲覧できるCPUのカタログである。これは、社外のCPUメーカーのWebサイトで公開されているオンラインのカタログから自動的に取得されており、随時、最新の製品名、価格や、納期などの情報が表示されている。発注者は、この中から必要な部材を選択する。

図1：PCメーカーで閲覧できるCPUのカタログ

モデル	品名	価格	納期	社名	備考
Intel Pentium 4	Intel Pentium 4 Processor 3.0GHz	18000	18000	社外CPUメーカー	
Intel Pentium 4	Intel Pentium 4 Processor 3.0GHz	18000	18000	社外CPUメーカー	
Intel Pentium 4	Intel Pentium 4 Processor 3.0GHz	18000	18000	社外CPUメーカー	
Intel Pentium 4	Intel Pentium 4 Processor 3.0GHz	18000	18000	社外CPUメーカー	
Intel Pentium 4	Intel Pentium 4 Processor 3.0GHz	18000	18000	社外CPUメーカー	
Intel Pentium 4	Intel Pentium 4 Processor 3.0GHz	18000	18000	社外CPUメーカー	
Intel Pentium 4	Intel Pentium 4 Processor 3.0GHz	18000	18000	社外CPUメーカー	
Intel Pentium 4	Intel Pentium 4 Processor 3.0GHz	18000	18000	社外CPUメーカー	
Intel Pentium 4	Intel Pentium 4 Processor 3.0GHz	18000	18000	社外CPUメーカー	
Intel Pentium 4	Intel Pentium 4 Processor 3.0GHz	18000	18000	社外CPUメーカー	

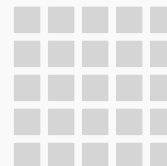


図2：注文書の閲覧画面

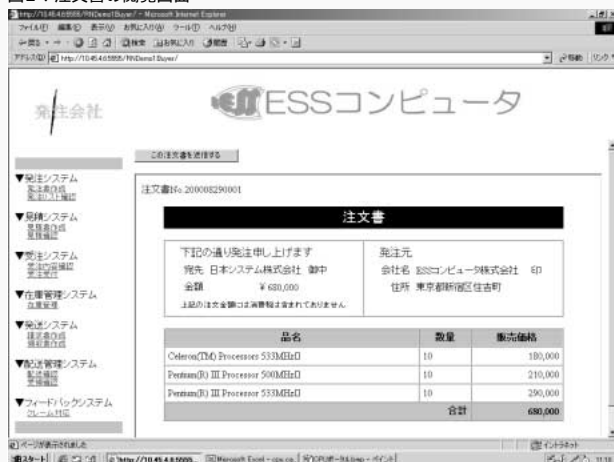
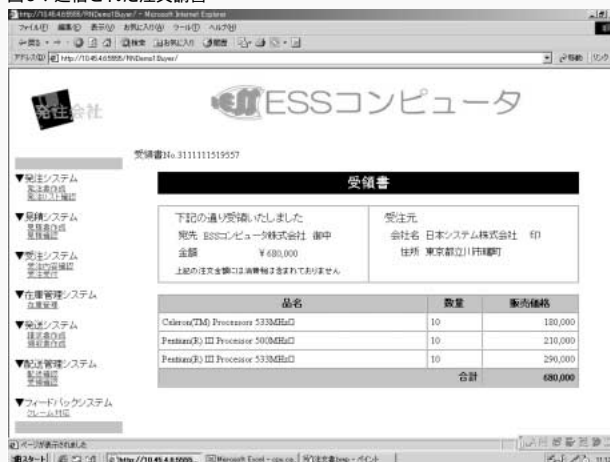


図3：返信された注文請書



PCメーカーではWeb上で選択した製品の情報を元に注文書を作成する(図2)、PCメーカーからCPUメーカーに送られた注文書はCPUメーカー側で確認され、注文請書が返送される(図3)。

手続きだけを見ればごく普通の取引業務であるが、この間にやり取りされる各書類にXMLが利用されている。リスト1、2をご覧ください。これは図2、3に示した発注書、発注請書の実体をなすXML文書である。実際に画面に表示させる場合には、それぞれの企業ごとの表示形式にあわせて、スタイルシートを適用している。

これらの文書をご覧いただければお分かりだと思うが、それぞれのルートタグが<Pip3A4 ~>という形式で記述されており、このXML文書はRosettaNetで定義されているPIP (Partner Interface Process) によるものであることが見て取れる。

<PurchaseOrder>タグ内の<Global ProductIdentifier>で指定された製品番号から、この両者が同じ製品を扱ったものであることが分かるだろう。一方、異なる点に注目してみると、ルートタグの

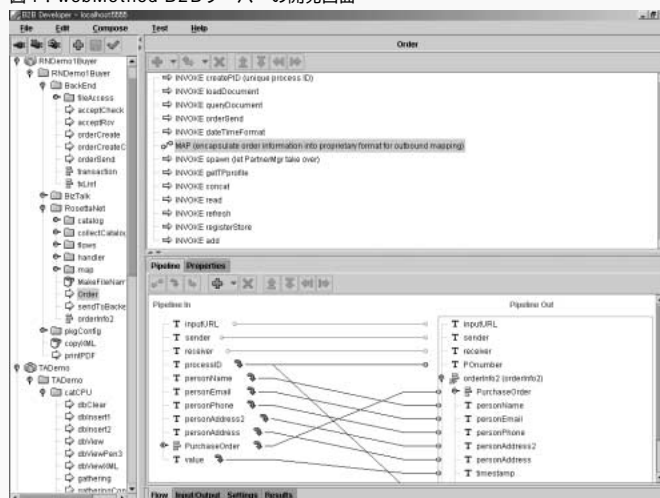
名称が<Pip3A4PurchaseOrderRequest>、<Pip3A4PurchaseOrderAcceptance>と、それぞれの内容を表わすものになっている。また、注文請書(リスト2)には<requestingDocument ~>というタグが付け加えられて、もとの注文書に関する情報が記述されているという点が異なる。

それぞれの過程における情報をXML文書に埋め込みながら処理が進められており、これ以外の部分では、もともとのXML文書から情報がそのまま引き継がれ

ていることがご理解いただけると思う。

今回ご紹介するデモでは、取引先サイトの役割を果たすサーバーにもwebMethod B2Bサーバーを利用している。もちろんこれはMicrosoftのBizTalkサーバーやInfoteriaのAsteriaなど、webMethod B2Bサーバーで利用できるフレームワークをサポートしているものであれば、どのようなサーバーとの間でも接続可能だ。

図4：webMethod B2Bサーバーの開発画面



豊富な機能モジュールによる迅速な開発

システムを構築する段階では、入出力されるXML文書の各要素が、社内システムのどのノードとして利用されるのかをGUI上でマッピングすることで、システムと各フレームワークとの接続を行なっている。またRosettaNetなど主要なB2Bフレームワークについては、サポート用のモジュールが標準で装備されている。

このため、特別なコーディングを行なうことなく、各フレームワークに対応することが可能だ。図4はXML文書のスキーマを、実際にwebMethod B2Bへとマッピングするデベロッパーの画面である。画面右下のペインのように、対応するノード同士を結び付けることで、XML文書のハンドリングが行なえるようになっている。

入力されたデータに、プログラミング的な処理を施す場合は、Javaによって必要な機能をカスタムモジュールとして実装できる。これは一度作成してしまえば、他のモジュールと全く同じように再利用可能だ。

XMLの実データを操作するような部分に関しては、モジュールに隠蔽されてしまい、webMethod B2Bサーバーの利用者にとって、XMLに関するコーディングレベルの知識は不要になる。そのため今回のデモでは、非常に短期間で実装が行なうことができた。ただしこれは、上流設計がきちんと行なわれているという前提での話である。

コーディングのないシステム開発へと移行するのは？

webMethod B2Bのデベロッパーは、システムの設計ツールと共通する要素を持っており、各モジュール間の関係性を

記述することが、そのまま機能の実装となるという側面を持っている。

このようにソフトウェアの機能を抽象化し、設計作業がそのまま実装へと近づくようなアプローチは、BizTalkサーバーなどの製品でも見られるものであり、

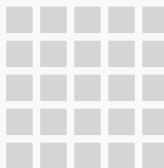
このようなフレームワークをサポートした製品では、今後ポピュラーな手法になることも予想される。

さらにebXMLなどでは、やり取りされるメッセージの中に自己に関する情報（メタデータ）が埋め込まれ、その情報

リスト1：注文書のXML文書（抜粋）

```
<?xml version="1.0" encoding="Shift_JIS"?>
<?xml-stylesheet type="text/xsl" href="../../RosettaNet/orderSheet.xsl"?>
<Pip3A4PurchaseOrderRequest>
<PurchaseOrder>
<ProductLineItem>
<shipFrom>
<GlobalLocationIdentifier>9876543210000</GlobalLocationIdentifier>
</shipFrom>
<ProductQuantity>10</ProductQuantity>
<LineNumber>1</LineNumber>
<productUnit>
<ProductPackageDescription>
<ProductDescription>
<GlobalProductIdentifier>00000000000000</GlobalProductIdentifier>
</ProductDescription>
</ProductPackageDescription>
</productUnit>
<countryOfOrigin>
<GlobalCountryCode>JP</GlobalCountryCode>
</countryOfOrigin>
<requestedShipDate>
<DateStamp>20001106Z</DateStamp>
</requestedShipDate>
:
<requestedPrice>
<FinancialAmount>
<GlobalCurrencyCode>JPY</GlobalCurrencyCode>
<MonetaryAmount>180000</MonetaryAmount>
</FinancialAmount>
</requestedPrice>
</ProductLineItem>
:
</PurchaseOrder>
:
<thisDocumentGenerationDateTime>
<DateTimeStamp>20001106T110634.458Z</DateTimeStamp>
</thisDocumentGenerationDateTime>
<thisDocumentIdentifier>
<ProprietaryDocumentIdentifier>200008290001</ProprietaryDocumentIdentifier>
</thisDocumentIdentifier>
<GlobalDocumentFunctionCode>Request</GlobalDocumentFunctionCode>
</Pip3A4PurchaseOrderRequest>
```

XML - Based B2B Frameworks



によって、メッセージの処理モジュールを動的に構成することなども検討されている(102ページのクラス・ノック氏へのインタビューを参照されたい)

数年のうちに訪れるかもしれないこのような変化の片鱗を、今回のデモの中に見て取ることができるだろう。

リスト2: 注文請書のXML文書(抜粋)

```
<?xml version="1.0" encoding="Shift_JIS"?>
<?xml-stylesheet type="text/xsl" href="../../RosettaNet/POA.xsl"?>
<Pip3A4PurchaseOrderAcceptance>
<PurchaseOrder>
  <GlobalPurchaseOrderStatusCode>Accept</GlobalPurchaseOrderStatusCode>
  <GlobalShipmentTermsCode>Third party pay</GlobalShipmentTermsCode>
  <RevisionNumber>1</RevisionNumber>
  <purchaseOrderDate>
    <DateStamp>20001106Z</DateStamp>
  </purchaseOrderDate>
  <ProductLineItem>
    <ProductQuantity>10</ProductQuantity>
    <ProductQuantity>10</ProductQuantity>
    <ProductQuantity>10</ProductQuantity>
    <productUnit>
      <ProductPackageDescription>
        <unitPrice>
          <FinancialAmount>
            <GlobalCurrencyCode>JPY</GlobalCurrencyCode>
            <MonetaryAmount>180000</MonetaryAmount>
          </FinancialAmount>
        </unitPrice>
        <ProductDescription>
          <GlobalProductIdentifier>00000000000000</GlobalProductIdentifier>
        </ProductDescription>
      </ProductPackageDescription>
    </productUnit>
    <scheduledShipDate>
      <DateStamp>20001106Z</DateStamp>
    </scheduledShipDate>
    <GlobalPurchaseOrderStatusCode>Accept</GlobalPurchaseOrderStatusCode>
    <LineNumber>1</LineNumber>
  </ProductLineItem>
  :
</PurchaseOrder>
  :
  <requestingDocumentIdentifier>
    <ProprietaryDocumentIdentifier>3111108225437</ProprietaryDocumentIdentifier>
  </requestingDocumentIdentifier>
  <GlobalDocumentFunctionCode>Response</GlobalDocumentFunctionCode>
  <requestingDocumentDateTime>
    <DateTimeStamp>20001106T110822.000Z</DateTimeStamp>
  </requestingDocumentDateTime>
</Pip3A4PurchaseOrderAcceptance>
```

S H O E I S H A



EC-MAGAZINE

月刊DBマガジン2000年12月号増刊
定価1280円

特集

「eビジネスを
「収益」&「コスト」
で考える」

BIOBからBtoCまでの「ビジネス活用情報誌」

① 収益とモデルで見る

Case Study

日本サムスン、日経BP、ニッセン、
博報堂、ジャパンネット銀行 ほか

② 予算とポイントで見る

Case Study

日比谷花壇、大日本スクリーン、
ユニクロ、monoマガジン ほか

保存版 / サイトトラブル
完全対応マニュアル

(株)翔泳社 〒160-0006
東京都新宿区舟町5 出版局
TEL.03-5362-3810 FAX.03-5362-3817
http://www.shoeisha.com

翔泳社の本がWebで買えます
http://www.seshop.com
SEshop.com