XML and the Future of Business

By Barry Briggs
Chief Technology Officer
Interleaf, Inc.

That the World Wide Web has revolutionized the very underpinnings of modern commerce seems now to be beyond question. Nearly every aspect of business, from marketing to sales to management to fulfillment has been irrevocably transformed by the existence and availability of a globally accessible computer network. Information, transactions, indeed, currency itself now stream at the speed of light across this great digital way.

What has driven the success of the Web? Principally, the key factor has been the general agreement among software developers upon a set of published standards for data exchange. This consensus has enabled them to create software tools both interoperable and powerful – running on servers, browsers, handheld devices, even television set-top devices. Because of these standards, the notion that these heterogeneous devices can indeed communicate with one another -- an idea very nearly unimaginable a scant decade ago -- now hardly raises an eyebrow.

Most important of these standards is HTML, the Hypertext Markup Language, the *lingua franca* of the Web. HTML, a broad, versatile presentation language, can describe everything from elementary school menus to nuclear power plants to online book catalogs.

Yet as significant as the Web has become in our lives it has yet to attain its full potential. A new standard, XML, encompassing all the functionality of HTML but also much more, will propel the Web, and the enterprises that adopt it, into new dimensions of global reach, organizational effectiveness, capability, and, ultimately, profit. Applications employing XML as their communications and content format will dramatically increase the usefulness of the Web to both consumers and businesses, and will tie them more closely together. This article describes the advantages of XML-based solutions and the impact XML will have on businesses.

XML: A Primer

Let us begin by understanding precisely what XML *is*. What follows is not an exhaustive description but one intended to illustrate XML's business benefits.

Consider a typical e-Commerce Web site, in this case one selling stereo components over the World Wide Web. When a user navigates to the site, information encoded in Hypertext Markup Language is sent to the browser, which then uses the HTML instructions to display it. One of the pages from the site describing stereo speakers might appear as depicted in Figure 1:

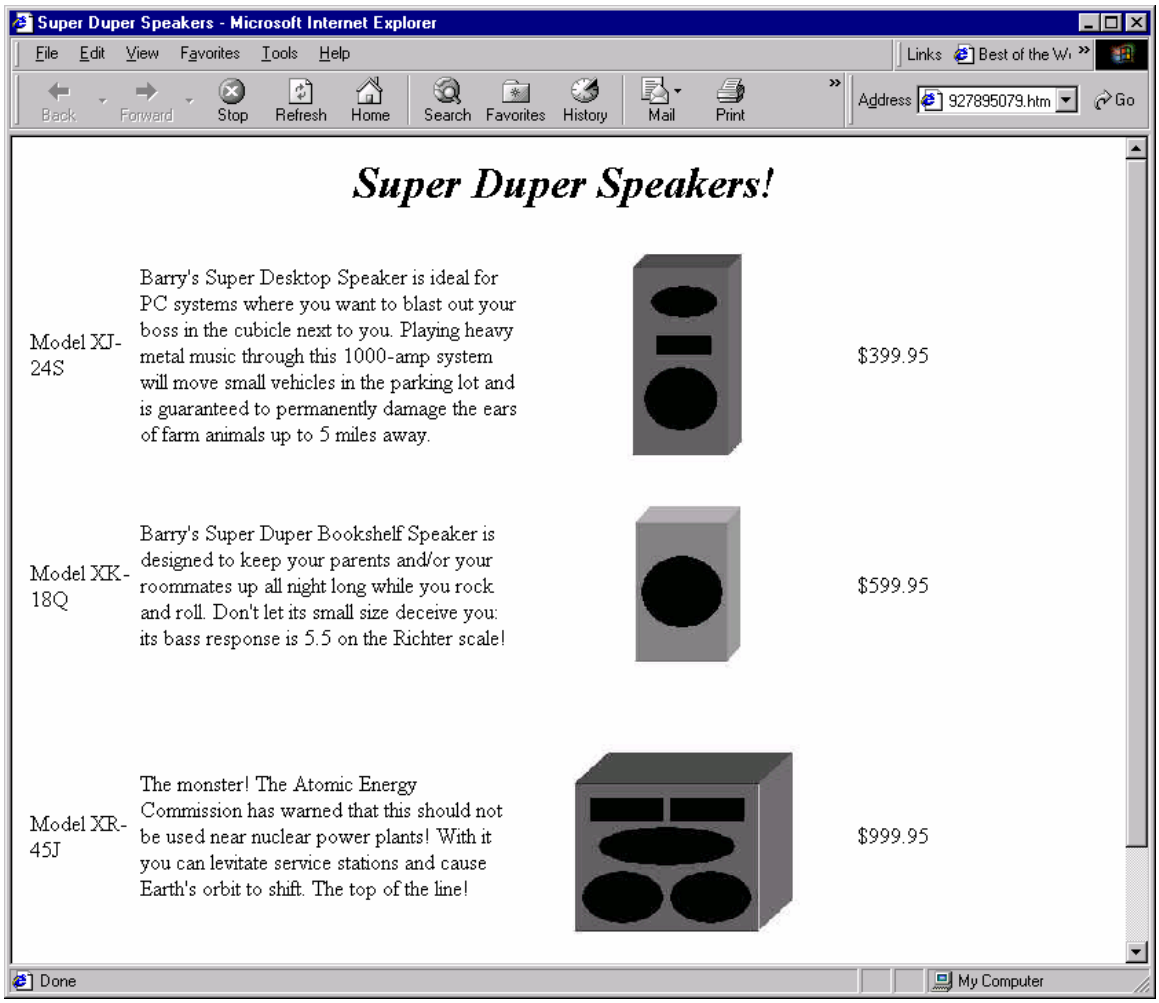


Figure 1. HTML Page

For reference, the underlying HTML code – the language the browser uses to create this page – is shown below. It's certainly not necessary for the purposes of this article to examine every line in detail; the point to be made is that HTML's syntax – its tags (<h1>, <p>, etc.) it uses – is directed at *presentation*. For example, the <h1> tag tells the browser that the following text should be rendered in a relatively large font. The <tr> tag, similarly, implies that a new row of a table is about to begin.

```
<html>
<head>
<title>Super Duper Speakers</title>
</head>
<body bgcolor="#FFFFFF">
<h1 align="center"><i>Super Duper Speakers!</i></h1>
<table width="91%" border="0" height="323">
  <tr>
    <td width="11%">Model XJ-24S</td>
    <td width="39%">Barry's Super Desktop Speaker is ideal for PC systems where
      you want to blast out your boss in the cubicle next to you. Playing heavy
      metal music through this 1000-amp system will move small vehicles in the
      parking lot and is guaranteed to permanently damage the ears of farm animals
      up to 5 miles away.</td>
    <td width="34%">
      <div align="center"><img src="Speaker.jpg" width="109" height="171"></div>
    </td>
```

```
    <td width="16%">$399.95</td>
  </tr>
  <tr>
    <td width="11%">Model XK-18Q</td>
    <td width="39%">Barry's Super Duper Bookshelf Speaker is designed to keep
      your parents and/or your roommates up all night long while you rock and
      roll. Don't let its small size deceive you: its bass response is 5.5 on
      the Richter scale!</td>
    <td width="34%">
      <div align="center"><img src="speaker2.jpg" width="108" height="156"></div>
    </td>
    <td width="16%">$599.95</td>
  </tr>
  <tr>
    <td width="11%">Model XR-45J</td>
    <td width="39%">The monster! The Atomic Energy Commission has warned that
      this should not be used near nuclear power plants! With it you can levitate
      service stations and cause Earth's orbit to shift. The top of the line!</td>
    <td width="34%"><img src="speaker3.jpg" width="218" height="200"></td>
    <td width="16%">$999.95</td>
  </tr>
</table>
<h1 align="left"> </h1>
</body>
</html>
```

Figure 2. HTML Source for Speaker Page

But here's the rub: nothing in HTML provides any information about the content itself; HTML knows nothing about speakers or sound systems or stereos; try, for example, to find from the HTML code all the speakers whose price is less than $500.

This is why, as a matter of fact, search engines often turn up useless links to unrelated topics. For example, if you were to type in the word "speaker" into any of today's popular search engines you would be as likely as not to receive links to after-dinner keynoters as to stereo speakers. (Try it.) The core reason is that HTML provides no clues to the search engines as to the *meaning* of the content; only its presentation.

XML solves this problem. XML in and of itself is not really a language, but rather provides a *framework* for structured languages. Anyone can create an XML "dialect," or *markup language*, for their use. (HTML itself can be thought of as an XML dialect.) For example, a developer might create a markup language for stereo systems. Using such a language the stereo speaker page might be coded as follows:

```
<?xml version="1.0">
<!DOCTYPE stereos SYSTEM "stereospeaker.dtd">
<stereos>

<speaker>
        <model>XJ-24S</model>
        <description>Barry's super desktop speaker is ideal for PC systems where
        you want to blast out your boss in the cubicle next to you. Playing heavy
        metal music through this 1000-amp system will move small vehicles in the
        parking lot, and will permanently damage the ears of cattle in the next
        county.</description>
        <price currency="US Dollars" amount=299.95/>
</speaker>

<speaker>
        <model>XK-18Q</model>
        <description> Barry's Super Duper Bookshelf Speaker is designed to keep
        your parents and/or your roommates up all night long while you rock and
        roll. Don't let its small size deceive you: its bass response is 5.5 on
        the Richter scale.</description>
        <price currency="US Dollars" amount=599.95/>
</speaker>

<speaker>
```

```
        <name>Barry's Desktop Speaker</name>
        <description>The monster! The Atomic Energy Commission has warned that
        this should not be used near nuclear power plants! With it you can
        levitate service stations and cause Earth's orbit to shift. The top of the
        line!</description>
        <price currency="US Dollars" amount=999.95/>
</speaker>

</stereos>
```

Figure 3. XML for Speaker Page

In this example it is obvious that we're talking about stereo speakers and not orators. The second line references a Document Type Definition (DTD) which formally specifies the dialect of XML used to describe stereo speakers – loosely analogous to a database schema. Anyone can create a DTD for a given application: for aircraft parts, real estate listings, microprocessors, or what have you. When industry groups agree upon them as standards, as we shall see, DTD's can become particularly useful and powerful.

Also straightforward is the nature of the information: it's easy to pick out the name of the product, its description, price and so on; finding all the speakers whose price is under $800 requires little effort. (Of course, if it's easy for us humans to read and extract meaning from this XML text, it's even more so for computers; and following this chain of logic then search engines indexing XML instead of HTML can return much more useful results.)

In computer science terms we say that XML is *self-describing;* in other words, you don't need an external tool (such as a proprietary word processor or spreadsheet) to understand the content. This is of tremendous value because it means that the information structure is *open;* anyone can build a tool to read, edit and manipulate XML.

Further, it should be noted that XML does *not* in any way describe the presentation of the data: there are no hints to a browser or any other program suggesting what font or paragraph style any given text should be rendered with. While at first glance this may seem limiting, in fact, XML's independence of presentation style gives it enormous power because now the content can be *redirected* to many different formats.

XML uses the notion of *style sheets* to render content aesthetically pleasing ways. Style sheets describe how XML should be displayed, or printed (or even spoken!), and, most interestingly, can contain different instructions for different output devices. Indeed, style sheets can *transform* the content into a form appropriate for the device. Consider then a document containing references to another document. On the Web, of course, these references would appears as clickable hyperlinks. But such things have little use on a printed page. A print-specific style sheet might then transform these links into (say) footnotes in print. The raw XML only knows that other documents are referenced; it is up to the style sheet to render them in a manner appropriate to the output.

Does all this mean HTML is dead? No. HTML remains a superlative rendering (presentation) language for the Web. Over the next few years systems will be built that maintain and manage raw XML content but are able to dynamically create finished documents in a variety of formats, including HTML. In other words, HTML becomes one of many output renderings of content; it is the one most appropriate for Web browsers whereas (say) PostScript is more appropriate for a printer.

Let us now turn to how XML will transform the world of business in the connected world; what, specifically, is the value proposition of this new technology?

XML Accelerates the Delivery of Information

In today's world the dynamo of commerce is information. Information drives buy and sell decisions at every level. The more current information is, the more value it brings to both vendor and customer. But how is information created and transmitted?

In almost every enterprise today the production of documentation is a *process* involving multiple departments and people. Information in the organization has a *life cycle*, beginning with each department generating documents in the area for which it is responsible: pricing data from marketing, test specifications from engineering, and so on.

Yet the documentation which is *published* -- to paper, or to the Web – usually represents an admixture of the contributions of each of those departments. For example, a catalog may include some of the original engineer's specifications, combined with pricing data from the marketing department, and so on. Thus in many organizations there is a person or group whose responsibility it is to bring all of this data together in a way intelligible to its many consumers.

How much faster it would be if this process could be automated!

Consider again our speaker manufacturer. The information process begins with design specification for a speaker created by an engineer. A test engineer adds data collected during system test; and marketing adds pricing data.

Each of these resources – engineers, testers, marketing – are expensive resources. Thus it behooves the organization to make maximum use of the content they create, and not force them to recreate it for each different publication or media type.

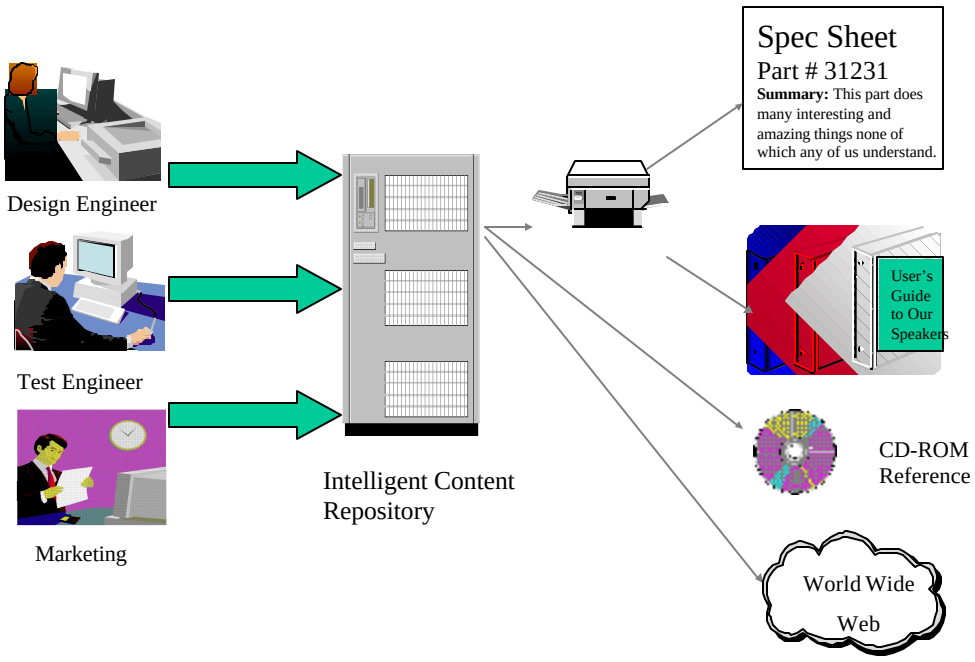


Figure 4. Information Life Cycle

XML enables this. XML content can be reused, recombined and retargetted at any number of output media and devices.

How is this possible? An *intelligent* repository holds the XML. Its intelligence lies in its ability to understand the structure of XML documents – that they can be broken down and decomposed into information *granules* (e.g,, an address, part descriptions, speakers) and then recombined into new documents. Thus by querying the repository for the appropriate data, and by then applying the correct style sheet, a spec sheet can be created, or a user's guide. Or, equally importantly, the repository could output HTML for use in browsers. There is *one* copy of the original content: but an infinite number of outputs, styles and formats.

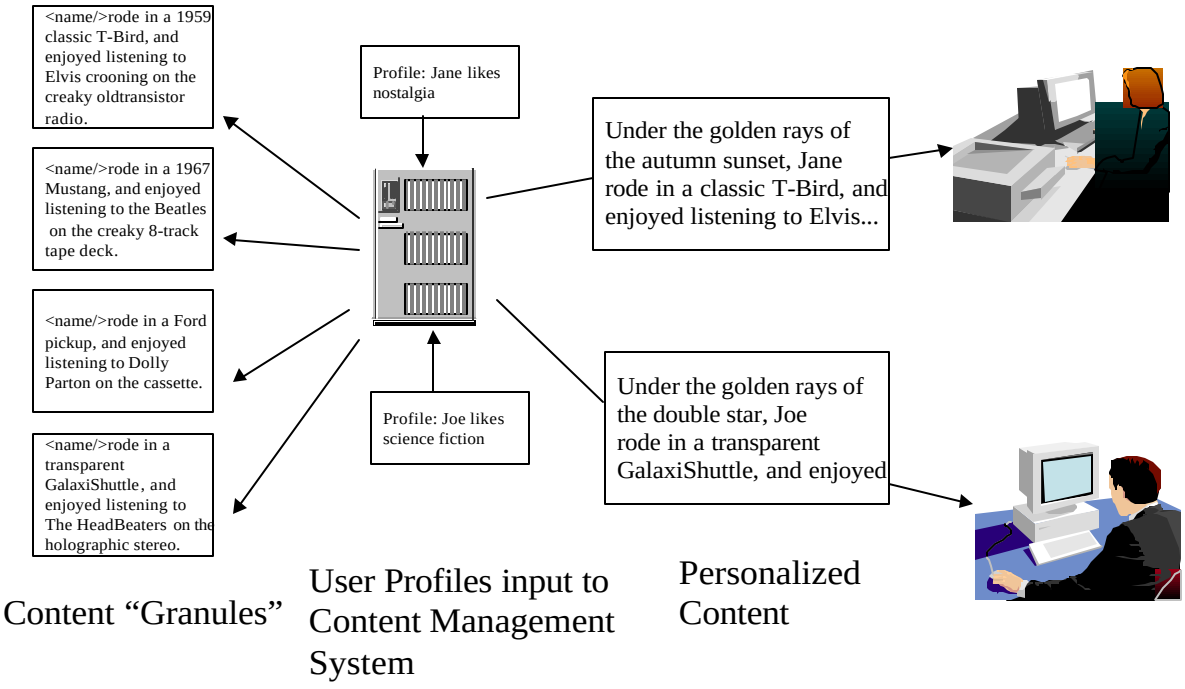
The notion that documents are no longer monolithic, but rather consist of components, is critical. It means that new documents, indeed, documents that may never have existed before, can be created by combining components. In fact the persistent form of a document may not be long strings of text and images, but rather a query!

XML Enables True One-to-One Marketing

"One-to-one" marketing is one of the catch phrases of today's Web. With XML, personalization and customer relationships can become infinitely richer and more meaningful for clients, and thus more valuable for enterprises.

Remember that XML is semantically rich, that is, its tags carry real meaning with them, not just instructions on how to present the data. These tags make possible rapid, intelligent queries which will return useful, relevant data. Hence documents will be generated and published as the result of a query submitted to the content repository.

User profiles and preferences, then, can be the criteria upon which these queries are based. In other words, Web pages and printed books are *dynamically* created based upon knowledge of the end user.



For example, a trivial application of a modern content management system would be customizing storybooks. Here the content of the story is centered around the reader: his or her likes and dislikes, friends, ambitions, social circumstances, and so on. The process of creating this dynamic content is depicted in the illustration below.

Figure 5. Recombinable Content

The user's profile is used, in effect, as a query into the content management system which then assembles appropriate granules of content, styles it according to a style sheet, and renders it through the web server. Note that the "stories" which have been automatically generated *might never have existed as such before* (imagine if Dostoevsky or Proust had such a tool!)

This example is somewhat frivolous, but it illustrates the point of dynamically generated documents. For example, the storybook above could have just as easily been a catalog of consumer electronic devices created for a specific user based upon knowledge of the user's likes, dislikes, previous purchases, and so on. Indeed, in combination with modern profiling and recommendations engines whole virtual communities of customers, tied together by bonds of mutual interest, can be built.

Consider as well that it is useful for a provider of dynamic content to track its usage. For example, as "content granules" are assembled into a document, a script can be invoked – a mechanism easily accomplished via a style sheet. That script could then log the granule's usage; it could cache it, if it determined it was being access frequently; or it could even invoke a micropayment from the user.

The bottom line: granularized content has enormous power.

XML Breaks Down Enterprise Barriers

But that is not the end of the story. XML is not merely a format for data kept in databases. It is rapidly becoming a format for data *transmission* as well, subsuming and replacing any number of older, restrictive formats such as EDI and others.

Consider a semiconductor manufacturer. As with the fictitious stereo company mentioned above, information content about chips is generated by many different individuals in the enterprise – design engineers, test engineers, and so on. What is particularly interesting, however, is that there presently exists not just a *corporate* standard XML format for chip descriptions but an *industry* standard. The Silicon Integration Initiative (SI2 for short) created ECIX, for Electronic Component Information Exchange, precisely to make information interchange as seamless as possible.

Now the promise of XML begins to be realized, for the design specifications created in one company can be *shared* over the Web – not just in presentation form, but in a form usable by tools. A product designer in one company can *directly* import a chip design from another company over the Web into a CAD/CAM system in order to test its suitability in a product. All the barriers and roadblocks that used to exist between enterprises – phone calls, sales meetings, presentations, and so on – are removed, or to use to phrase more in vogue, disintermediated. The consumer and provider are directly linked.

And of course this example serves only to illustrate a more general phenomenon. Consider another example, in which an e-commerce Web site – a bookseller, for example, or a toy vendor – aggregates catalog content from a multitude of suppliers. Each of those suppliers generates structured XML (using whatever tool they care to) and then transmits it to the central site. Here, ironically enough, the e-commerce site *re-intermediates*; it provides a convenient online superstore (or more accurately, a *superstore-front*) for customers.

One last scenario: today's forms and workflow packages usually utilize proprietary formats and graphical presentations. Imagine, however, if the forms were predefined XML documents. Since XML documents are easily decomposed and recomposed into new documents, a straightforward XML-based workflow system would consist of (say) a purchase requisition originated by the requestor, signed by various levels of management, and then transformed into an industry-standard purchase order and transmitted over the Internet to the supplier.

This is depicted in Figure 6: here an employee generates the purchase requisition, which, because it has been rendered with a style sheet, appears to her to be a well-designed form – but the application cares only for the XML content, shown at the top. The req is approved by her manager, and the relevant portions of the content are merged automatically into a standard PO, sent to the supplier.

Note that there is *no* conversion between legacy, incompatible data formats. Only the most minimal human intervention is required and thus XML speeds the transaction.

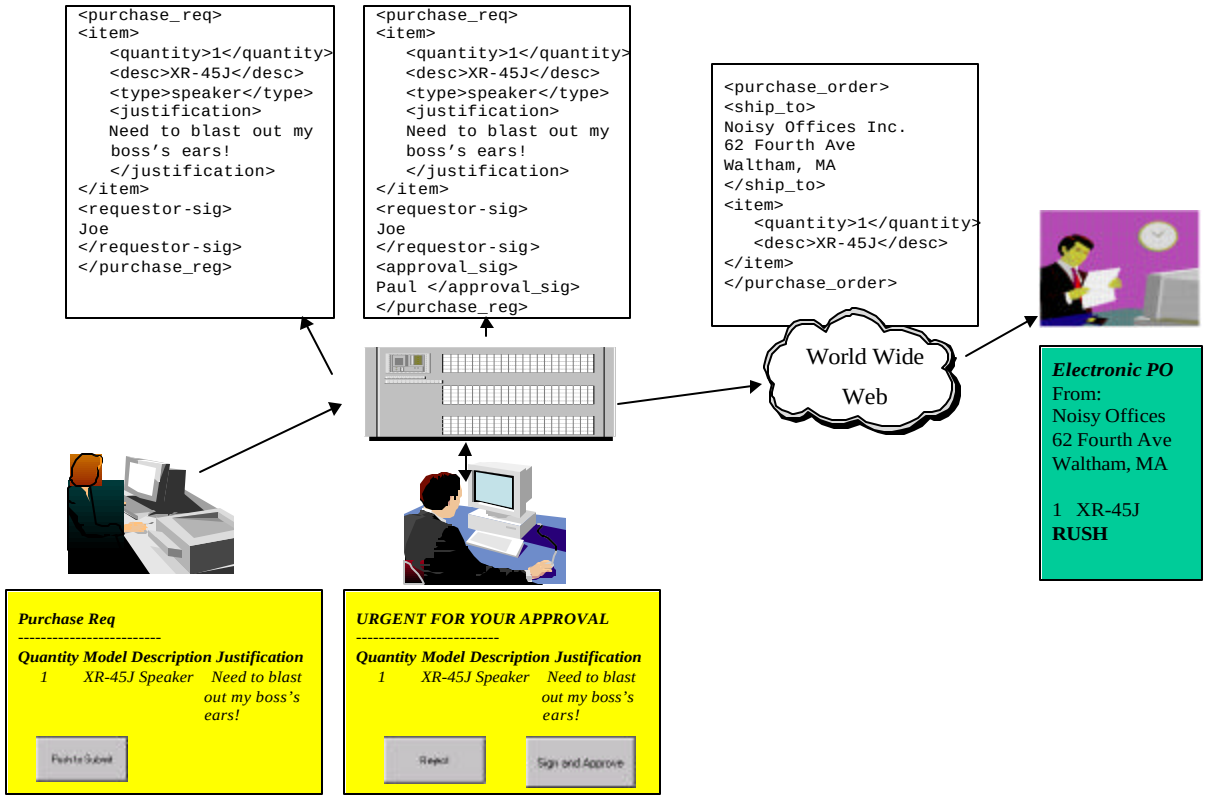


Figure 6. XML Powers the Extended Enterprise

The conclusion is clear, that XML *lowers the barriers* that have previously existed between organizations and enterprises.

Summary: XML is the Driver of 21st Century Commerce

The question has often been asked, "Is XML simply the next big fad?" HTML, of course, has in its very short life already evolved significantly and has undergone numerous revisions; since it was introduced in 1992 it has already seen four major upgrades.

XML is different, however. XML, a language for languages, describes real business objects and processes; it does not need to be revised simply because someone thought of a new way to scroll a blinking banner across the screen. It's true that XML requires some retooling of content, but the rewards, as we have seen, far exceed the small, one-time price to retool.

Unquestionably, information and communication drive commerce; anything that improves their quality and speed accelerates business. As we have seen, XML, because of its inherent richness of meaning, its

granularity, its ability to be repurposed, recombined and retargetted, and its flexibility, will bring supplier, vendor and customer closer together. As a result the speed at which business is transacted will again undergo a great transformation as the customer's experience of goods and services is made even more direct.