

Altova, Inc., The XML Spy Company  
900 Cummings Center, Suite 314-T  
Beverly, MA, 01915-6181, USA  
Tel: 978-816-1600  
Fax: 978-816-1606  
Web: [www.xmlspy.com](http://www.xmlspy.com)

ALTOVA

[www.altova.com](http://www.altova.com)

*Author:* Larry Kim  
© 2002 Altova, Inc. & Altova GmbH



xml spy

**XML** Integrated Development Environments

***Accelerating XML Application Development in the Enterprise***

---

## ***Table of Contents***

<b>Executive Summary</b>	<b>3</b>
<b>XML Defined</b>	<b>4</b>
<b>XML Technology Overview</b>	<b>5</b>
XML Schemas	5
XSL/XSLT	6
Simple Object Access Protocol (SOAP)	7
<b>XML in the Enterprise</b>	<b>8</b>
Web Services	8
The World Wide Web	8
XML Publishing & Document Management	8
Database Integration	8
Microsoft .NET Framework	9
Java & XML	9
Summary of Industry XML Usage	9
<b>The Software Development Lifecycle Model</b>	<b>10</b>
Classic Software Development	10
Modern XML Software Development	10
<b>XML Integrated Development Environments (IDE)</b>	<b>11</b>
<b>The XML Spy 4 IDE</b>	<b>12</b>
XML Schema Development	13
XML Editing	13
XSL/XSLT Stylesheets	14
SOAP/Web Services	15
Database Integration	15
<b>XML Spy 4 Suite Product Line</b>	<b>16</b>
<b>Industry Partnerships</b>	<b>17</b>
<b>About Altova</b>	<b>18</b>

## ***Executive Summary***

**XML is today widely accepted** as a key technological innovation to enable the next generation of software applications. XML as a technology is unique in many aspects compared to classic procedural programming languages (C/C++, Java etc.) with respect to its recursive structure and rules-based processing model. In spite of XML's paramount importance and all of the technological differences between XML and other classic procedural programming languages, classic IDE software vendors have implemented only the most rudimentary development support for XML technologies.

**This whitepaper examines** the most common usages of XML and explains the benefits driving the rapid rate of XML adoption in the enterprise. Both pre and post-XML software development practices are investigated. The concept of an XML Integrated Development Environment is introduced, referring to a comprehensive set of tools built specifically to accelerate working with all XML technologies and to help ease the transition from an XML document to the most common XML processing frameworks or databases. The explosive growth of new XML technologies and its widespread adoption across every vertical industry makes the rudimentary XML support offered by classic IDE's insufficient for building the next generation of distributed XML applications and web-services; and there is a clear need for an XML-specific IDE to complement a classic procedural language IDE.

**The benefits** of adopting XML Integrated Development Environments include enormous productivity gains and competitive advantages due to substantially reduced time-to-market and associated development costs. Especially in today's challenging economic climate, an XML IDE is a necessary requirement for designing and deploying any software application which creates or consumes XML documents.

*“The  
e X t e n s i b l e  
M a r k u p   L a n g u a g e  
(XML) is the universal file  
format for describing and  
representing absolutely any-  
thing imaginable, in such a  
manner that it is easily  
under standable to both  
people and com-  
puters.”*

*Alexander Falk,  
President & CEO, Altova Inc.*

## **XML Defined**

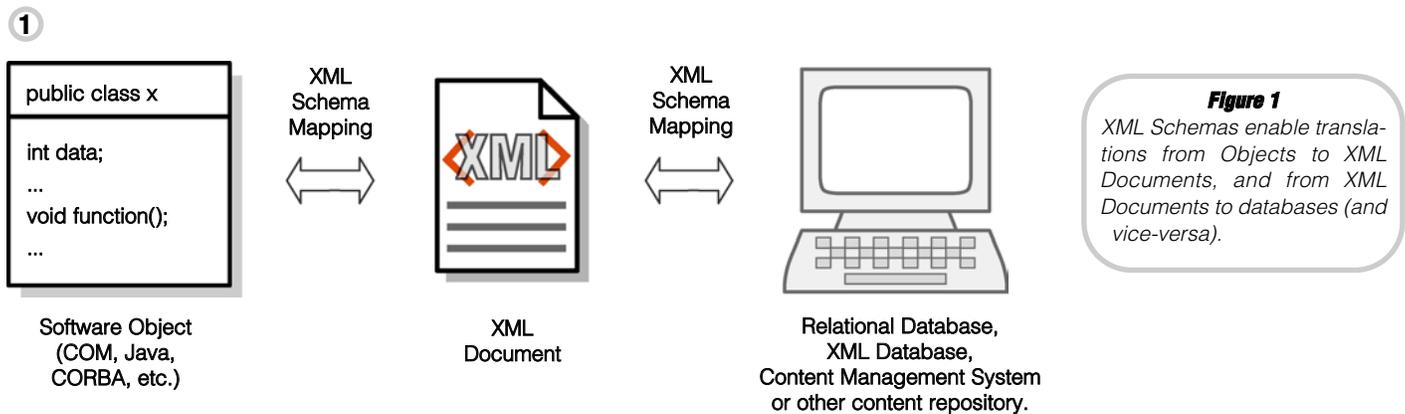
**XML technologies are numerous,** far-reaching and employed in so many different applications that the all-encompassing definition of XML is often misunderstood. As an example, a software engineer might view XML as new distributed wire protocol; a web developer typically associates XML as a means for separating data from presentation markup; a database administrator could view XML as a new way of describing data and content for improved storage and retrieval – the list of possible usages is endless. While no individual example is wrong, XML is all this and so much more! And in taking only a local view of the XML landscape, it is easy to overlook a far deeper understanding of how all XML technologies work together to solve today’s business challenges. The next sections examine the most powerful XML technologies and how they are currently being employed in industry to solve today’s business challenges.

## XML Technology Overview

There are hundreds of XML technologies in industry with new XML standards are being developed every day. XML technologies are being used in so many different ways across every imaginable vertical industry, however the vast majority are related to one of three key technologies: XML Schema, XSL/XSLT and SOAP.

### XML Schemas

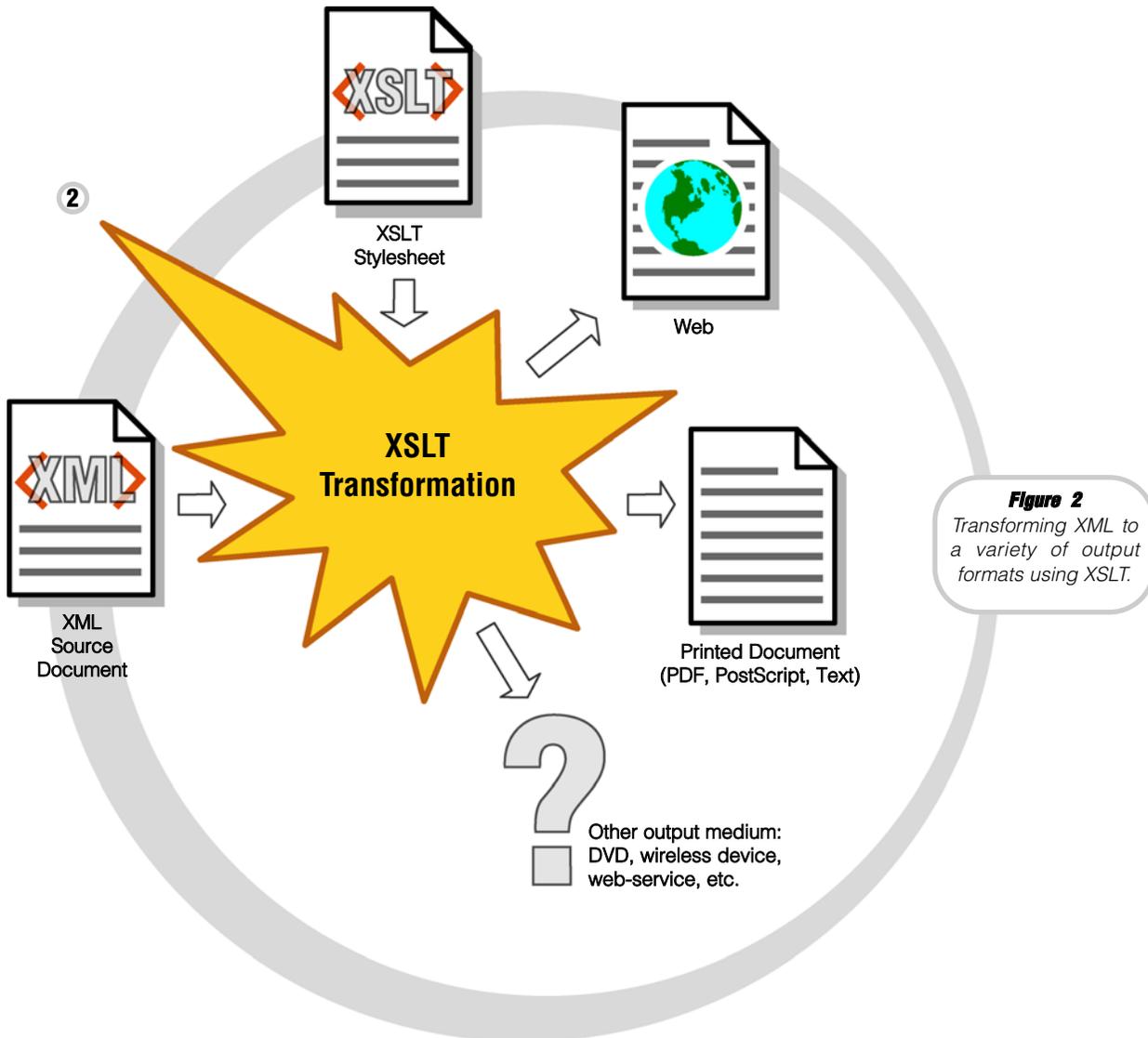
XML Schema is the World Wide Web Consortium's (W3C) official XML data definition language; it addresses many shortcomings associated with Document Type Definitions (DTD), and has massive industry support from all of the major software corporations. XML Schemas define the structure and allowable permutations that an XML document can take on, to be considered a member of common family of XML documents. Industry consortiums are joining together to develop XML Schemas that define common file formats for describing everything from mathematical formulas, research documents, news articles, credit card transactions, accounting audits, medical prescriptions, and much more. The ultimate goal is to enhance software application interoperability through the use of common XML-based file formats to express data and content. Once an XML schema is completed, a developer can offload much of the burden of validation code from the application to an XML validator.



XML Schema technologies are also used to bridge the gap between databases and software objects as illustrated below in figure 1. XML Schema is object-oriented in nature and allows for easy mapping between XML documents and objects written in any object-oriented programming language. XML Schema is strongly-typed, and its tree structure is ideal for preserving relationship information; this makes XML Schema a requirement for mapping of XML documents to any underlying data store.

**XSL/XSLT**

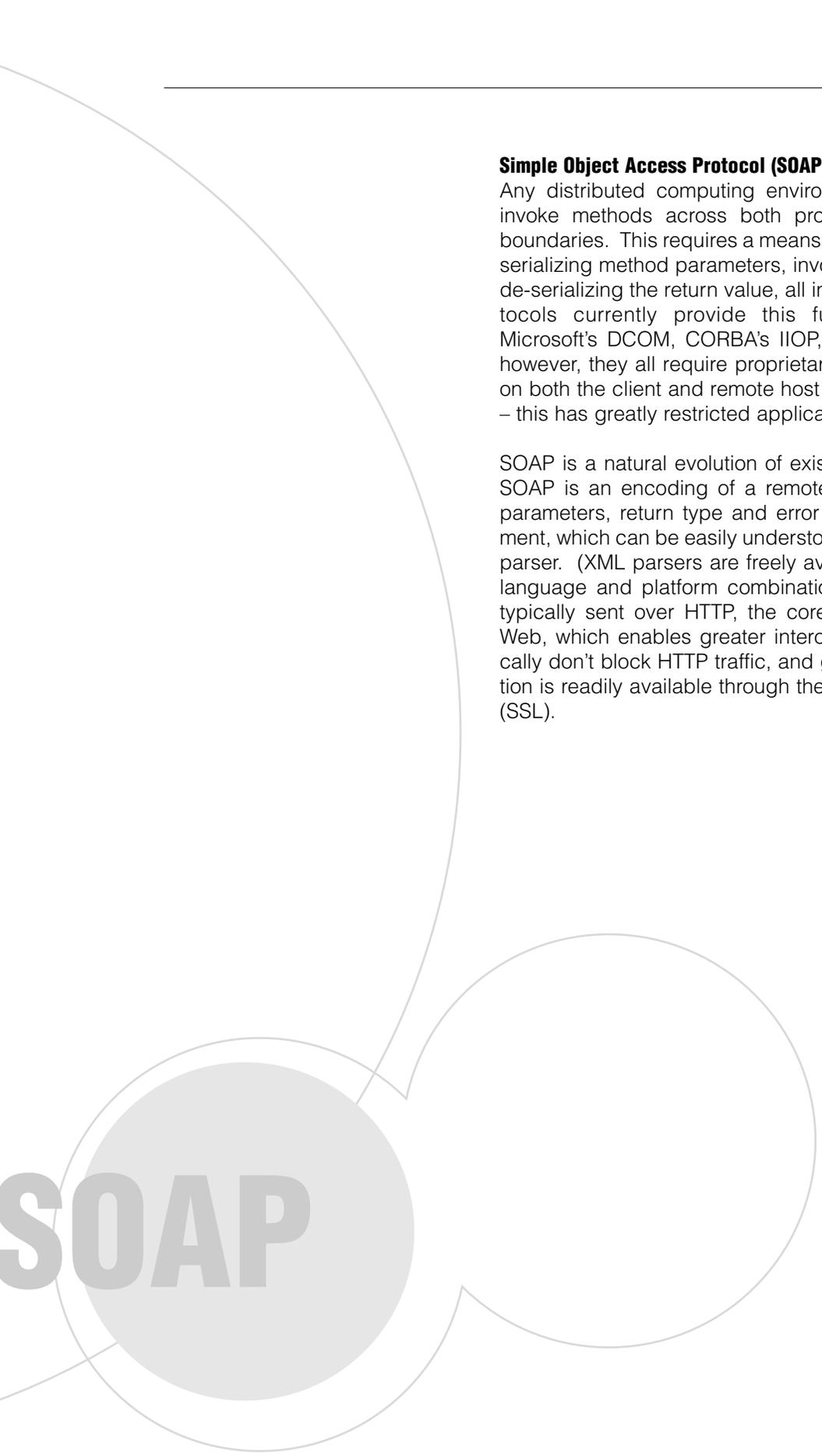
The eXtensible Stylesheet Language (XSL) and the eXtensible Stylesheet Language Transformations (XSLT) are standardized languages for transforming the content and data stored within an XML document, to a different output form. Using XSL, content saved in an XML format can be transformed into any output media (HTML, WML, PDF, PostScript, plain text, etc), by applying an XSLT stylesheet. The XSLT transformation process is illustrated below in figure 2.



### **Simple Object Access Protocol (SOAP)**

Any distributed computing environment requires the ability to invoke methods across both process and physical machine boundaries. This requires a means for locating a remote process, serializing method parameters, invoking the remote method, and de-serializing the return value, all in a secure manner. Many protocols currently provide this functionality including DCE, Microsoft's DCOM, CORBA's IIOP, Java RMI and many others; however, they all require proprietary class libraries to be loaded on both the client and remote host in order to inter-communicate – this has greatly restricted application interoperability.

SOAP is a natural evolution of existing distributed technologies. SOAP is an encoding of a remote method invocation (method parameters, return type and error codes) within an XML document, which can be easily understood by any client using an XML parser. (XML parsers are freely available in every programming language and platform combination.) The SOAP document is typically sent over HTTP, the core protocol of the World Wide Web, which enables greater interoperability since firewalls typically don't block HTTP traffic, and greater security since encryption is readily available through the use of Secure Sockets Layer (SSL).



SOAP

## ***XML in the Enterprise***

Now that we have discussed XML Schema, XSL/XSLT, and SOAP, we now examine how these key technologies are commonly being applied to solve today's business challenges.

### **Web Services**

Using SOAP, XML Schema and other related technologies (collectively referred to as web-services) companies can expose programmatic access to business logic over the web, to any device, remote process, desktop application or web-application. Web services are transforming the World Wide Web from simple business-to-consumer applications which require human interaction to a distributed federation of loosely-coupled services. A key area for growth will be enhancing business-to-business (B2B) application infrastructure, enabling for the creation of virtual marketplaces, as well as streamlined order processing and back-office operations.

### **The World Wide Web**

XSL/XSLT stylesheets are commonly used by web-developers to separate data from presentation markup on web pages. This separation will greatly simplify indexing, sharing and retrieving data on the web, and is also used to enable the internationalization and localization of a website, and deliver website content to web-enabled mobile devices.

### **XML Publishing & Document Management**

XML has many benefits as storage format for rich, structured content because it more effectively preserves the semantics and context of the information being stored, allowing for efficient retrieval and re-purposing of content. XML technologies including XML Schema and XSL/XSLT have the potential to unlock proprietary content management systems through the use of standards-based interfaces. In addition XML is the underlying document layout language employed by popular file formats such as PDF, PostScript, and Scalable Vector Graphics (SVG).

### **Database Integration**

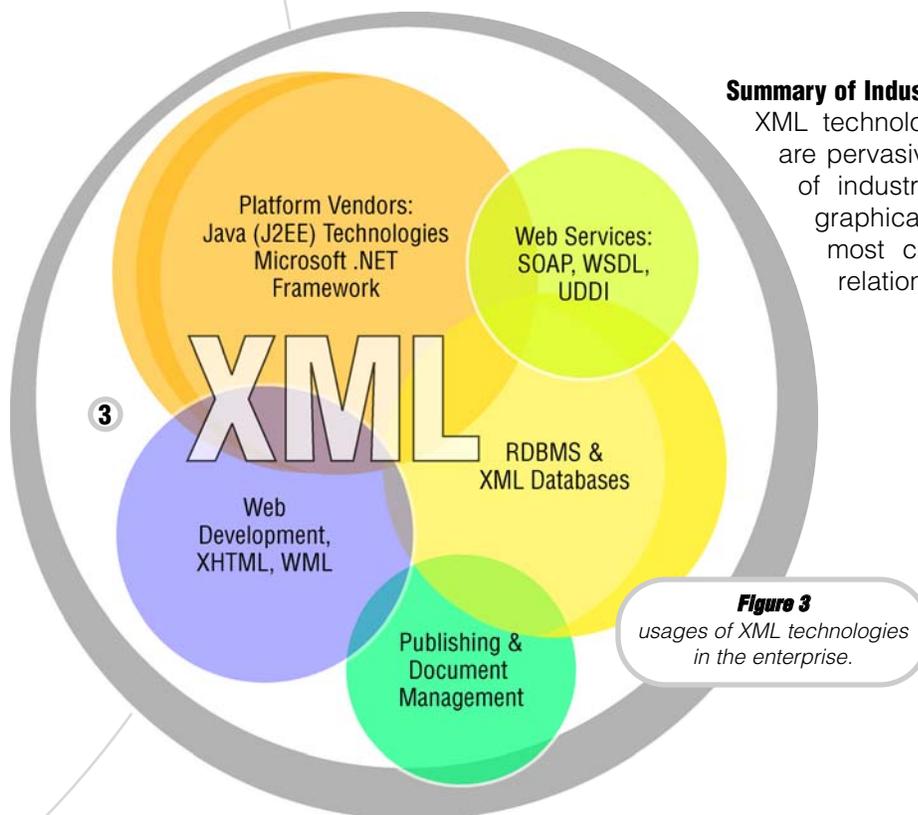
XML can be employed as an intermediary format (or adaptor) for Electronic Data Interchange (EDI) messages enabling easy integration with any relational database or native XML database system. Both Oracle and Microsoft have added support for XML schema into their respective databases.

### Microsoft .NET Framework

The recently released Microsoft .NET Framework is Microsoft's vision of the future of building next-generation of business applications, it encompasses various application servers, SQL Server 2000, the Windows operating system, multiple programming languages, mobile devices and more. SOAP and XML Schema bring all the pieces together allowing for tremendous application interoperability.

### Java & XML

The Java platform enables platform interoperability at a binary level - Java programs are compiled to an intermediate language and subsequently executed on any operating system through a native Java Virtual Machine. The combination of Java and XML has the potential to further improve interoperability by further decoupling the application from the underlying data storage format and opening up the application's communication protocol; these are important milestones in realizing true application portability. At the time of this writing, Sun is currently developing powerful new Java standards for Web services, XML bindings and XML messaging (currently available in beta, due out Q3/Q4 2002) which will greatly improve Java application server interoperability.



### Summary of Industry XML Usage

XML technologies are inter-related and are pervasive across a wide spectrum of industry applications. Figure 3 graphically summarizes some of the most common usages and their relationships.

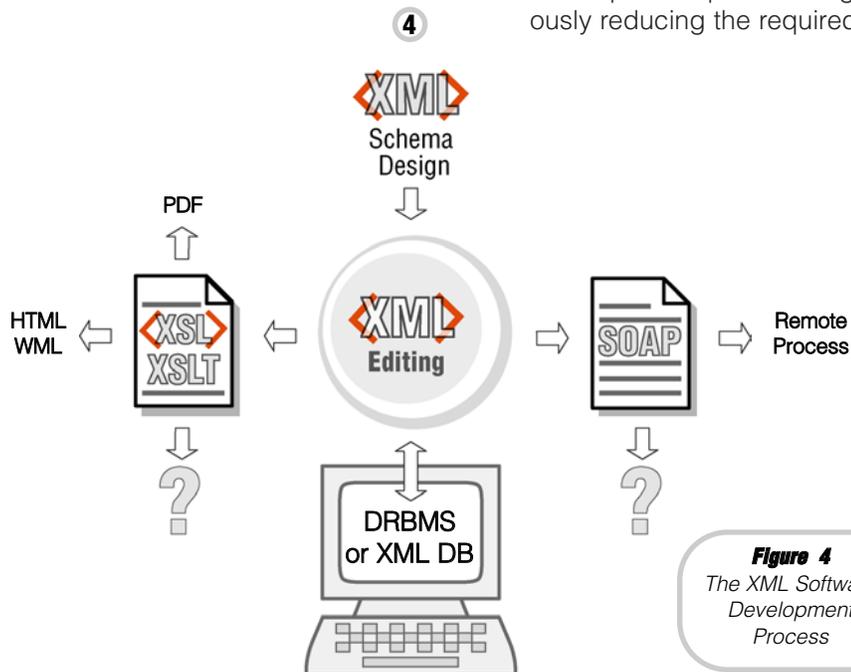
**Figure 3**  
*usages of XML technologies  
in the enterprise.*

## The Software Development Lifecycle Model

### Classic Software Development

Classic Software Development refers to developing applications using procedural programming languages such as C/C++, Java, C#, Cobol, Perl, etc. The classic software development process typically begins with a high-level architecting process which includes modeling the software objects and their interactions. Next, an editor is used to write the source code according to the proper syntax, and compiler is invoked to translate and link the software to an executable binary format. Finally, a debugger is used to catch any errors, thereby ensuring the correct program behavior.

Classic IDE's such as Microsoft's Visual Studio or Borland's JBuilder have revolutionized the software development process by providing enhanced tools support for editing source-code, as well as modeling and debugging tools which have enabled developers to produce higher quality software while simultaneously reducing the required effort.



### Modern XML Software Development

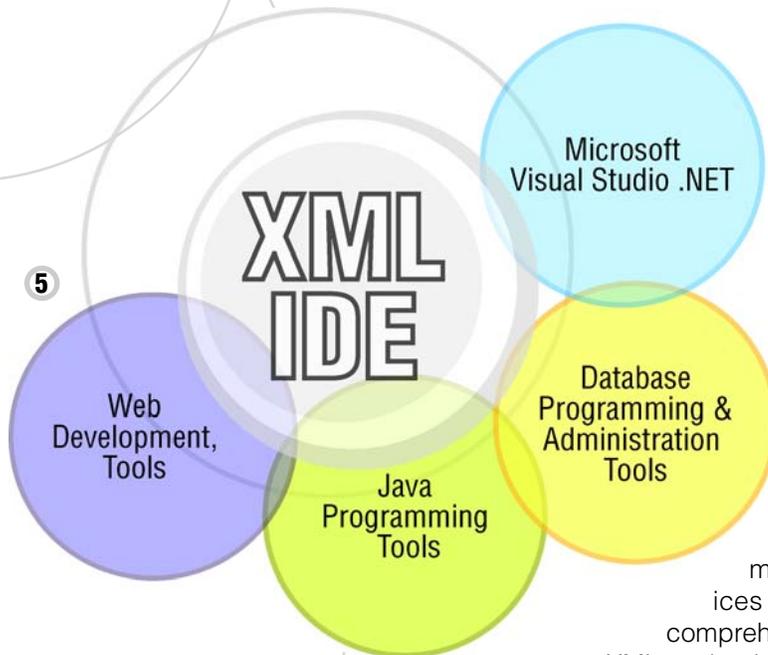
XML technologies are significantly different from classic procedural programming languages both in structure, syntax and nature. XML application development is also a completely different process. XML application development begins with developing the content model using XML Schema or DTD's.

Next, one must edit and validate XML documents according to the content model. Finally, the XML document is ready to be

consumed or processed by an XML enabled framework, and as previously discussed, this typically involves transforming the XML document to another format, saving the XML document to a database, or transmitting the XML document to a remote process. The XML Software development process is illustrated in figure 4. XML Schema modeling tools are required to define XML content, and XML editing and validation tools are needed to author the XML documents. Once an XML document has been created, XSL/XSLT development and debugging tools are essential for transforming XML. Equally, XML-to-Database mapping tools are vital for database integration applications, and SOAP development and debugging tools are necessary for building web services.

## XML Integrated Development Environments (IDE)

An XML IDE must provide development support for critical XML technologies: XML Schemas & DTD's, XSL/XSLT, SOAP & Web services, Database Integration, as well as XML editing & validation. An XML IDE is not meant to be a replacement for an existing classic software programming IDE, web-development tool, or database programming/administration tool, but rather, the XML IDE complements and enhances any developer tool, by providing the comprehensive support for XML development for any potential application. An XML IDE must also provide tools and features to help cross the boundary from a pure XML technology to a particular language binding, server run-time environment, or database. Figure 5 below illustrates how an XML IDE complements existing software development tools.



**Figure 5**  
XML IDE provides comprehensive XML development support and complements other software development tools.

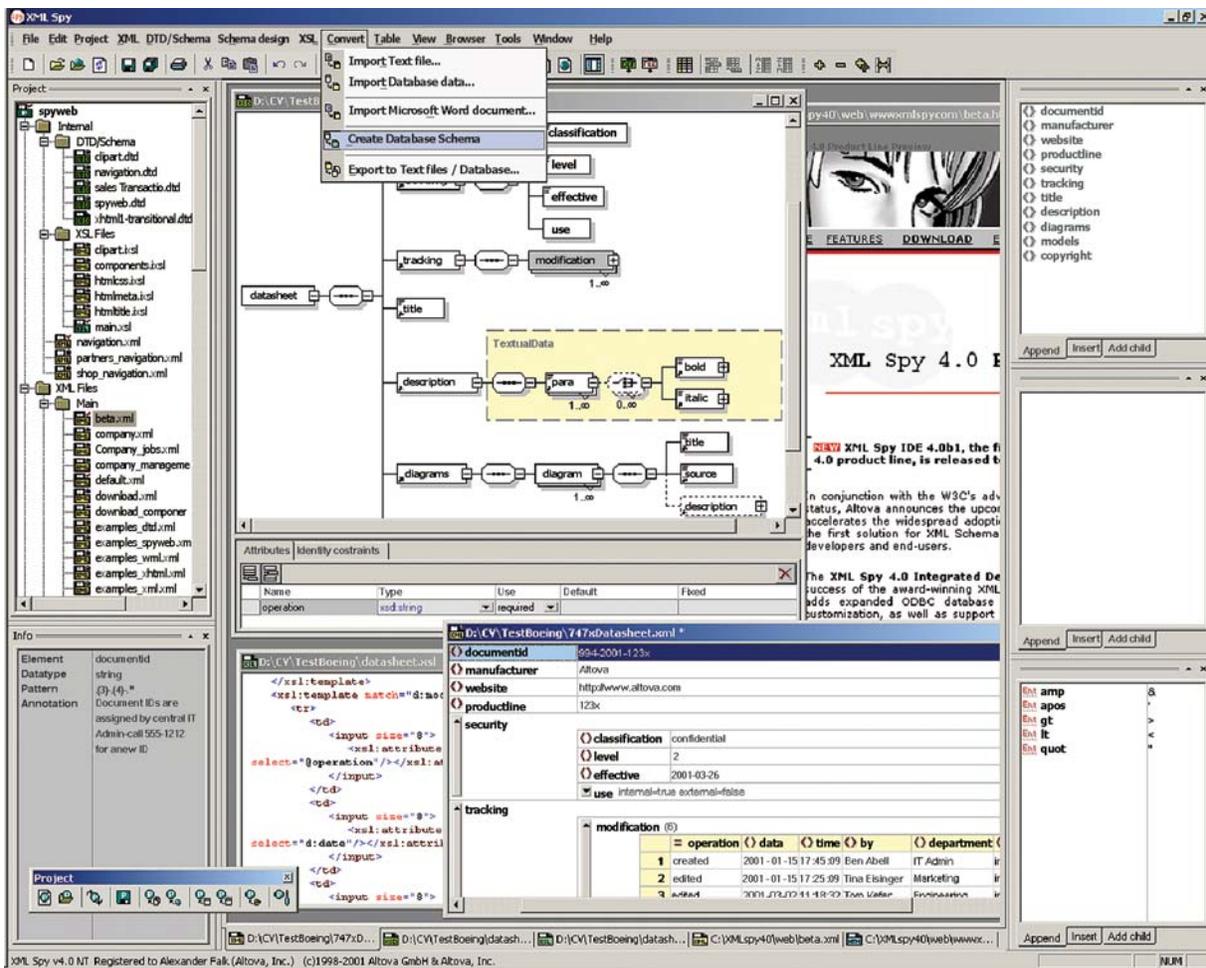
Just as classic IDE's have revolutionized the software development process over the past 2 decades, XML IDE's are modernizing XML & web-services development by providing comprehensive tools support for all XML technologies enabling developers to produce higher quality XML applications while simultaneously reducing the required effort.

## The XML Spy 4 IDE

The XML Spy IDE is the industry standard for developing software applications based on eXtensible Markup Language (XML) technologies.

The XML Spy 4 IDE was built from the ground up to fully support designing and editing all XML technologies such as XML Schema, XSL/XSLT, and SOAP, to accelerate development of web services, publishing & document management applications, database integration applications, Microsoft .NET and J2EE applications, and more. Figure 6 below shows the main editing window of the XML Spy 4 IDE.

6



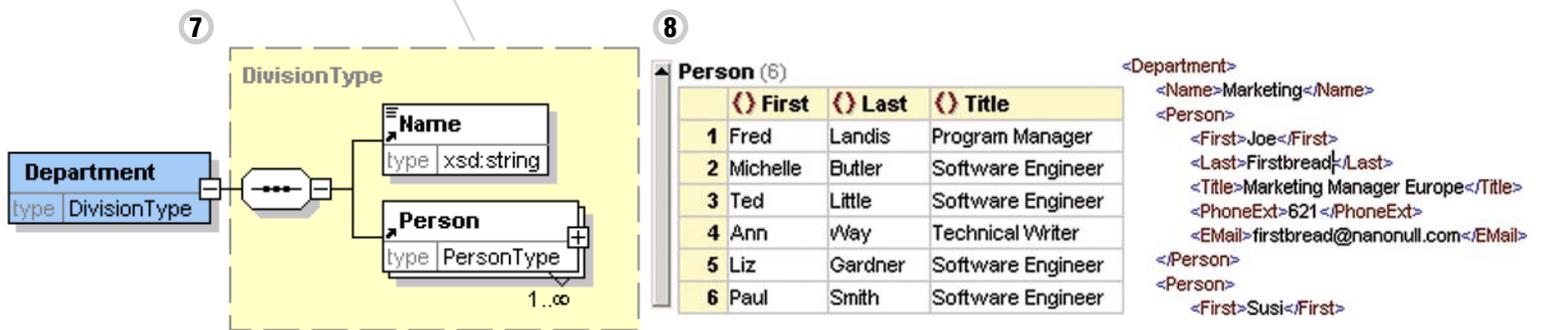
**Figure 6**  
The XML Spy 4 Integrated Development Environment

## XML Schema Development

The XML Spy 4 IDE includes a graphical schema editor which supports several mechanisms for generating XML Schemas:

- 1 The XML Spy 4 IDE includes a graphical schema editor which supports several mechanisms for generating XML Schemas. An XML Schema can be automatically inferred from one or more XML instance documents.
- 2 An XML Schema can be reverse-engineered from an existing database schema.
- 3 An XML Schema can be reverse-engineered from an existing database schema. An XML Schema can be converted from an existing content model such as a DTD.

The above mentioned schema generation and a powerful XML schema modeling tool illustrated below in figure 7, greatly accelerate the XML schema development process freeing developers to concentrate on developing business logic.



### XML Editing

One of the key benefits of the XML Spy IDE is its extensibility: whenever a new XML-based language is developed, any XML document in that language can automatically be edited by XML Spy by simply referencing a XML Schema or DTD. This is extremely important because while this whitepaper examined 3 key technologies in depth (XML Schema, XSL/XSLT and SOAP), there are indeed hundreds XML technologies currently being used in industry. XML Spy provides numerous advanced views for working with XML documents.

**Enhanced Grid View**, illustrated below in figure 8 (left-side) graphically represents repeating XML elements as rows within a table, allowing a developer to work with a complicated XML document in a tabular fashion, as though it were a simple Microsoft Excel spreadsheet.

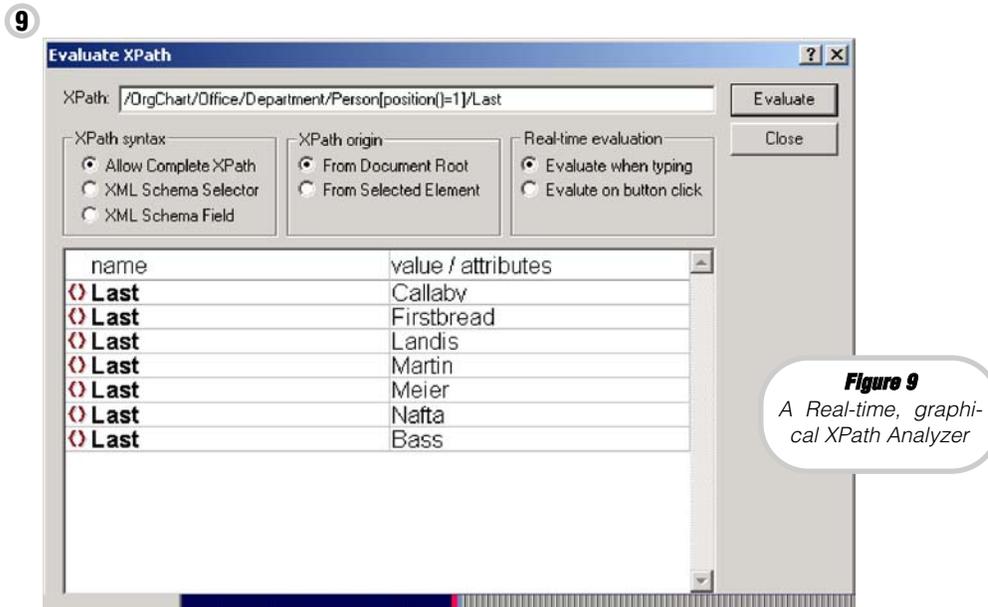
**Text View**, illustrated below in figure 8 (right-side) provides syntax coloring of XML elements and attributes, context-sensitive entry helpers, code-completion, and single-click document validation based on W3C's XML Schema or any DTD.

**Browser View**, provides a web-browser preview of HTML output.

## XSL/XSLT Stylesheets

XML Spy 4 IDE has many features to simplify XSLT stylesheets development, including an XPath analyzer, XSL syntax-help, color coding, and an integrated browser preview. XML Spy 4 IDE supports Formatting Objects (XSL:FO) and can generate PDF or PostScript files with a single-click. The open XSLT engine supports all of the leading XSLT processors including MSXML4, Saxon, Xalan, and more.

The XPath analyzer is shown in figure 9 below: simply type in an XPath expression and the resulting node-set will be evaluated and graphically displayed in real-time, eliminating the guesswork in XSLT stylesheet development.



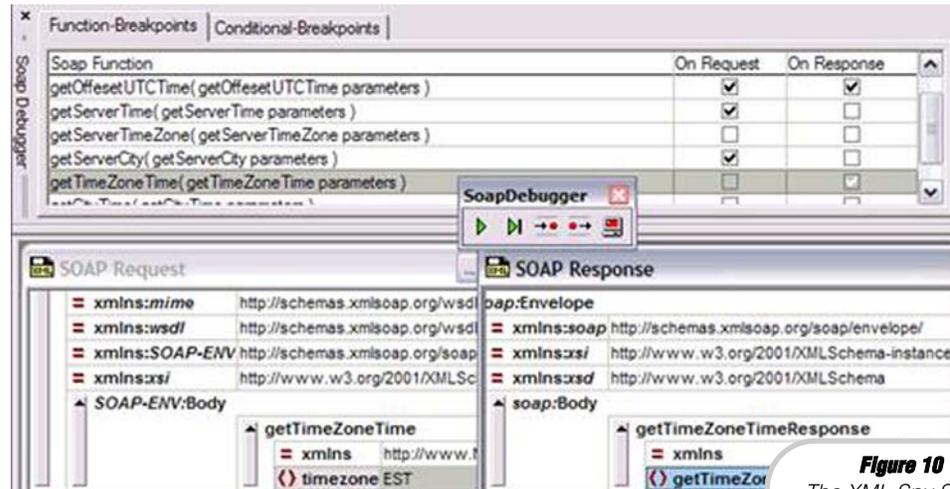
### XSL/XSLT Stylesheets

The XML Spy 4 XSLT Designer is a clear and easy to use tool. Unlike complicated, proprietary content management systems, the intuitive user-interface allows any web-designer (even with little or no understanding of XSLT) to leverage existing skills and design advanced stylesheets.

### SOAP/Web Services

XML Spy 4 Suite is the first and only tool to fully support testing & debugging of web-services! Automatically generate SOAP requests, step through SOAP requests & responses, set conditional breakpoints on functions or data! XML Spy 4 IDE supports development and debugging of web services on all the major web services platforms including Microsoft .NET and J2EE.

10

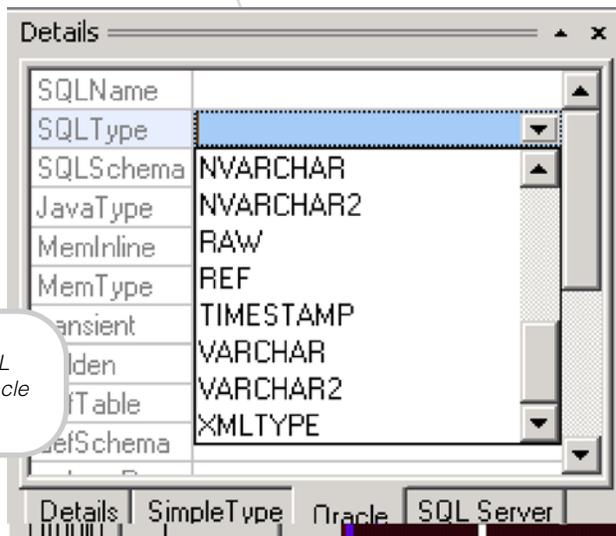


**Figure 10**  
The XML Spy SOAP Debugger.

### Database Integration

To bridge the gap between relational and native XML database storage models, the leading enterprise database companies have implemented *XML Schema extensions*; this additional XML meta-data enables for the seamless translation from an XML document to a relational database. XML Spy 4 IDE is the only developer tool to support XML Schema mappings to Microsoft SQL Server 2000 (i.e. SQLXML 2.0) and Oracle 9i XML schema extensions.

11



**Figure 11**  
Mapping an XML Schema to an Oracle XML Database.

XML Spy 4 IDE provides complete access to all relational databases through ODBC or ADO, including importing relational data into an XML file, and exporting XML data to a relational database. XML Spy can automatically create a database file from an XML Schema, and much more!

## XML Spy 4 Suite Product Line

The XML Spy 4 IDE is a part of the XML Spy 4 Suite, the ultimate software development tools suite for Advanced XML Application Development! The XML Spy 4 Suite consists of the XML Spy 4 Integrated Development Environment, the XML Spy 4 XSLT Designer, a WYSIWYG tool for generating XML-to-HTML XSLT stylesheets, and the XML Spy 4 Document Editor, a lightweight XML editing component for use in XML content applications.



**Figure 12**  
The XML Spy 4 Suite product line.

According to James Karney of [PC Magazine](#):

*"If XML's standards and syntaxes give you a case of acronymphobia, then Altova's XML Spy 4 Suite may be just what the doctor ordered.*

*This package provides the tools needed to develop and manage XML projects from schema design to document management ... it is a solid tool for developing sophisticated solutions without your having to learn all the nuances of XML coding."*

## Industry Partnerships

Altova is partnering with the world's leading software technology companies, providing technology integration to ensure a seamless transition from the XML Spy IDE to any XML processing environment. Technology integration has included Visual Studio .NET integration, Database Integration and much more.



Altova is an active participation in open industry consortiums such as the World Wide Web Consortium (W3C) and the Web Services Interoperability Organization (WS-I); this further ensures standards compliance and interoperability.



## **About Altova**

Altova produces and markets XML Spy 4 Suite, the world's leading development tools suite for Advanced XML Application Development. Founded in 1992, Altova has offices in Beverly, Massachusetts and in Vienna, Austria. XML Spy is used by over a half-million registered users across 50,000 companies worldwide, and is the leading choice of Fortune 500 and NASDAQ 100 companies. Altova is a member of the W3C and the WS-I.



---

**ALTOVA**

[www.altova.com](http://www.altova.com)