

Designing Web Services— The View From 60,000'



Copyright © 2003 Lincoln Stoller. Braided Matrix, Inc.. All rights reserved.

INTRODUCTION

This is an overview for developers interested in creating web services. It discusses what web services are, why you should be interested in them, what they are based on, what risks you should be aware of, and how you can start designing them.

SHORT DEFINITIONS OF WEB SERVICES

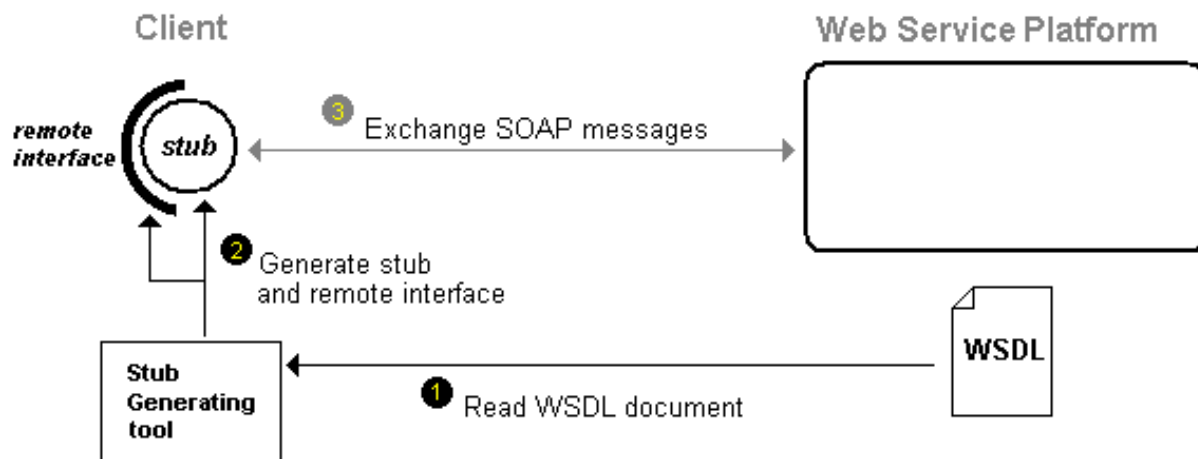
XML (Extensible Markup Language)

An XML markup language is a set of tags (called elements) that are used to describe information in Unicode text and organize it hierarchically.

SOAP (Simple Object Access Protocol)

An XML grammar used as an application protocol in both RPC and asynchronous messaging.

WSDL(Web Service Description Language)



An XML grammar used to describe Web services, including the kind of message format expected, the Internet protocol used, and the Internet address of the Web service.

Generating a web stub from WSDL, from Sun's "EJB2, Web Services: Part 1". For 4D replace "stub" with "proxy method."

UDDI (Universal Description, Discovery and Integration)

An XML grammar used to categorize, classify, and locate Web services according to their subject and function. What makes UDDI particularly powerful is that there are no limits on the number of different classification schemes that can be created.

Actual Services (i.e. Service Providers)

There is Web Services, and web services. The first is the concept, the second the actual providers using the concept. The two go together like a blue print and the building that's built from it.

SOAP

Formal Definition

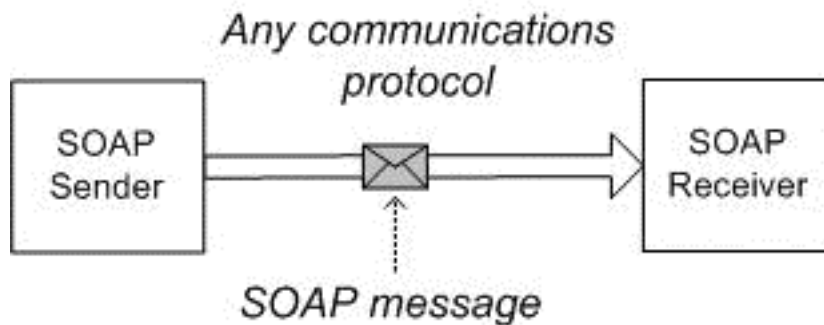
"SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment.... The framework has been designed to be independent of any particular programming model and other implementation specific semantics." — from the SOAP 1.2 specification

Small-Scale Definition

A protocol that we can use for exchanging simple or complex data between 4D and other applications.

Realistic Definition

The central component of a global system of standards that is large enough to support all forms of electronic collaboration.



Web Services Can Be Private

XML allows for proprietary structures. This “loop hole” in the paradigm of self-explaining data structures can operate at any web service level. This means that you can create a valid web service that completely unique, private, proprietary. You can create a valid web service that fails to provide any of the benefits listed below.

BENEFITS OF WEB SERVICES

The following benefits are part of the promise of web services.

Interoperability in a heterogeneous environment

Web service permit different distributed services to run on a variety of software platforms and architectures. *The key benefit.*

Business services through the Web

Leverage the advantages of the World Wide Web for the operations supported by any web service-enabled application.

Integration with existing systems

Most enterprises have an enormous amount of data stored in existing enterprise information systems, and the cost to replace these systems is such that discarding these legacy systems may not be an option.

Freedom of choice

Web service standards have opened a large marketplace for tools, products, and technologies.

Support more client types

Since a main objective of Web services is improving interoperability, exposing existing applications or services as Web services increases their reach to different client types.

Programming productivity

By creating a common programming standard Web services provides standard means to access the services required by multi-tier applications and standard ways to support a variety of clients.

Web service standards provide you with a means of delivering these benefits, but they are neither guaranteed nor required. Following the web service standards *will not* guarantee web service benefits.

THE ROLE OF STANDARDS

Web Services (the theoretical underpinning of web services) is nothing more than a set of potentially comprehensive, potentially unlimited standards. Standards have not played a big part in the world of 4D developers. Web services changes this.

4D Developers have lived a sheltered and somewhat isolated life. We have developed our own standards, programs, and solutions. This is true to various extents for developers of other platforms. It's a consequence of developing applications for private use within local networks.

Web services provides developers with, and requires that developers adhere to standards of:

- A. Discipline — standard designs and coding practice.
- B. Manners — service-oriented guidelines for what is provided, how it's provided, and how it's supported.
- C. Sociability — inter-connection with other platforms, programs, and objectives.



4D Developers have lived a somewhat isolated life.

THE NEED FOR STANDARDS

There can be no program for inter-operability without standards. Web Services objective of unlimited inter-operability creates a potential need for *unlimited* standards. It also forces users with little need for standards to work in conjunction with users that have a great need for standards.

This has already generated a tremendous volume of rules and recommendations. For most 4D projects, and most 4D developers, these standards will be a crushing burden. And while you are not required to comply with anything beyond the basic web service standards, there are also consequences for those applications that do not follow the extended standards.

What do *you* need to do?

- A. Follow standards to maximize inter-operability, or
- B. Create working applications that provide limited services quickly and cheaply.

Web services can support either objective.

STANDARDS AT DIFFERENT LEVELS

Web services requires that you broaden your concept of standards to include data and services at both higher and lower levels.

Lower Levels

Bit streams, communication protocols, rules for the formatting of rules (i.e. SOAP, WSDL, and UDDI are written in XML).

Higher Levels

Complex data, complex content: movies, music, commerce, human nature. Standards are being developed for how information of this kind is to be packaged and processed.

Web services potentially exposes us to a much larger universe of standards for describing and handling data.

To retain our effectiveness we must shield ourselves from as much of these “promising opportunities” as possible. Web services offers us tremendously expanded opportunities, but the costs are not clear. We only know that there may be many new requirements.



Odysseus and the sirens: ... we must shield ourselves from ... “promising opportunities”

SUPPORT FOR STANDARDS

Industry groups, standards organizations, academic subcommittees, and vendor partnerships are coming out of the woodwork to create standards. Some of the URLs of the leading standard setting organizations are:

1. Accredited Standards Committee (ASC X12): www.x12.org
2. Electronic Business XML (ebXML): www.ebxml.org
3. Java 2 Platform, Enterprise Edition (J@EE): java.sun.com/j2ee/
4. Organization for the Advancement of Structured Information Sources (OASIS): www.oasis-open.org/home/
5. World Wide Web Consortium (W3C): www.w3.org
6. Web Services Interoperability Organization (WS-I): www.ws-i.org

Address: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ebxml-cppa



ABOUT | MEMBERS | JOIN | NEWS | EVENTS | MEMBERS ONLY | COVER

TECHNICAL PROCESS ■

- Overview
- TC Process
- IPR Policy
- TC Guidelines
- Liaison Policy
- Liaisons
- Spec Templates
- Approved Work

TECHNICAL COMMITTEES ■

- Current TC List
- Join a TC
- Mailing List Archive
- Access Control
- Application Vulnerability
- Asynch Service
- Auto Repair
- Biometrics
- Business-Centric Methodology
- Business Transactions
- Conformance
- Content Assembly
- Controlled Trade
- Customer Information
- Digital Signature Svcs
- Directory Services
- DocBook

OASIS ebXML Collaboration Protocol Profile and Agreement TC

- ebXML CPPA
- ebXML Implement
- ebXML Messaging
- ebXML Registry
- Education
- e-Government
- Election Services
- Electronic Procurement
- Emergency Mgmt
- Entity Resolution
- Extens Resource ID
- HumanMarkup
- LegalXML Court Filing
- LegalXML eContracts
- LegalXML eNotary
- LegalXML IntJustice
- LegalXML Lawful Intercept
- LegalXML Legislative
- LegalXML ODR
- LegalXML Transcripts
- Localization
- Materials
- Mgmt Protocol
- Open Office Format
- PKI
- Product Life Cycle
- Provisioning
- RELAX NG
- Rights Language
- Security Services
- Tax XML
- TM Pub Sbj Geo Lang
- TM Vocab for XML Stds
- Topic Maps
- Published Subjects
- Translation WS
- UDDI Spec
- Universal Business Language
- User Interface
- Web Application Security
- Web Services Business Process
- Web Services Distrib Mgmt
- Web Services For Interactive Applications
- Web Services Reliable Messaging
- Web Services for Remote Portlets
- Web Services Security
- XSLT Conformance

JOINT COMMITTEES ■

- ebXML Joint Committee
- Security Joint Committee

OASIS NETWORK ■

- CGM Open
- ebXML
- LegalXML
- PKI
- UDDI

OASIS INFO CHANNELS ■

- Cover Pages
- XML.org
- Sponsorship

NEWSLETTERS ■

- OASIS News
- Cover Pages Weekly
- XML.org Daily Newslink

Some of the OASIS subcommittees operating to develop standards for particular aspects of web services.

QUESTION: What standards do we need to follow?

ANSWER #1: The least that is required for your application.

ANSWER #2: Very few if you're building an "old style", private application.

If you serve only 4D clients, then use 4D's built-in SOAP and for whatever else is needed, roll your own.

If you are a public service, then you'll need to find and follow common standards, or what you predict to become common standards.

ANSWER #3: At minimum you'll need 4D's SOAP Wizard, XML, and some awareness of some of the services available through 4D's ITK (authorization, authentication, encryption, etc.).

CURRENT LIMITS ON STANDARDS

Basic services are largely encapsulated and automated. You don't need to know how they work, only how to use them.

- ❑ SOAP: message packaging.
- ❑ WSDL: disclosure.
- ❑ UDDI: directory of services.

But structural and functional standards are largely missing from basic services.

Structural Standards

Standards for how complex data is structured will be detailed using XML. We don't yet have tools for abstracting this information (some other platforms do, notably J2EE's Java Beans).

Current structural standards are sketchy. The lack of tools and standards means that 4D programmers will probably need to again "roll your own."

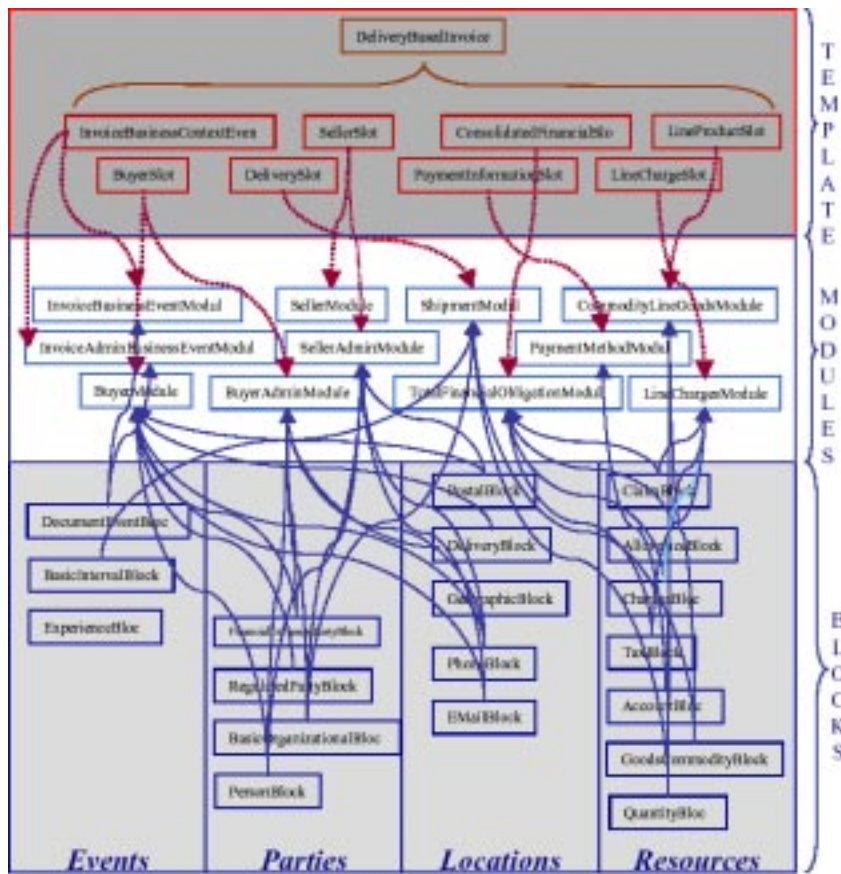


Figure 2
 ASC X12C/2002-61
 ASC X12 Reference model for XML Design

ASC X12's suggestion for the Data Flow Diagram of a delivery based invoice.

Functional Standards

These will also be specified using XML either in the header or the body of the SOAP messages. Since standards are lacking, you'll need to "roll your own."

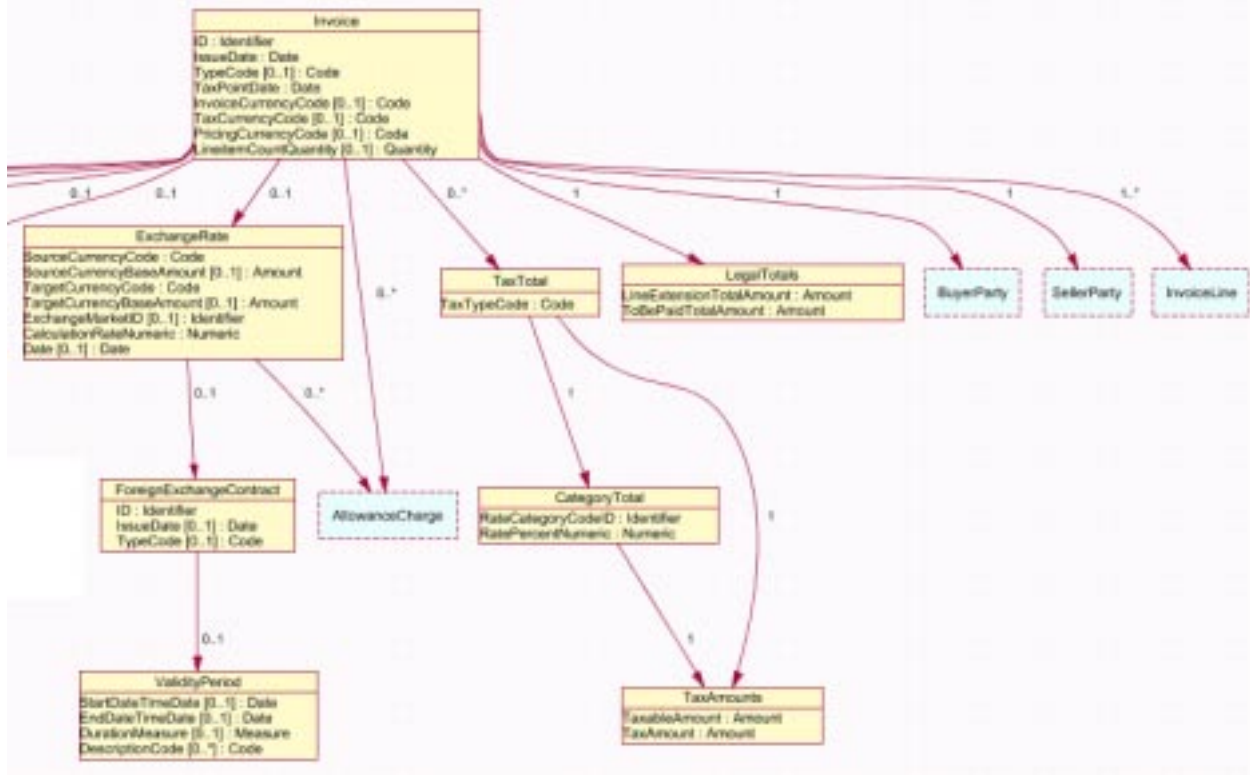
Useful References for Existing Standards

Extensive work exists and is publicly available. It is impressive, and complex. It is not self-explanatory.

ASC X12 Reference model for XML Design/ 2002-61

“The purpose of this document is to specify an approach to eBusiness messaging that: 1) links implementation with the structure, 2) enables cross industry differentiation and 3) supports industry needed ‘quick’ solutions.” — sounds great!

The following diagrams suggest data structures and their translation into XML. These are provided in the UBL library at <http://oasis-open.org/committees/ubl/lcsc/0p70/>.



About 1/10-th of ASC X12’s suggested data structure for a general sales invoice.


```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- 16 Aug 2002 -- source: UBL_Library_0p70_Reusable.xsd -->
<!-- Created by: Gunther Strober, SAP AG -->
<!-- 16 Jan 2003 -- source: UBL_Library_0p70_Reusable.xsd -->
<!-- Edited by: Tim H-Grath, UBL IC -->
<!-- 21 Jan 2003 -- source: UBL_Library_0p70_Reusable.xsd -->
<!-- Edited by: Gunther Strober, EBL HEGOC, SAP AG -->
<!-- 18 Jan 2003 -- source: UBL_Library_0p70_Reusable.xsd -->
<!-- Edited by: Tim H-Grath, UBL IC -->
- <!-- schema targetNamespace="urn:oasis:names:tc:ubl:Invoice:1.0:0.70" xmlns:sd="http://www.w3.org/2001/XMLSchema"
xmlns="urn:oasis:names:tc:ubl:Invoice:1.0:0.70" xmlns:ccts="urn:oasis:names:tc:ubl:CoreComponentParameters:1.0:0.70"
xmlns:cct="urn:oasis:names:tc:ubl:CoreComponentTypes:1.0:0.70"
xmlns:cat="urn:oasis:names:tc:ubl:CommonAggregateTypes:1.0:0.70"/>
<!-- report namespace="urn:oasis:names:tc:ubl:CoreComponentParameters:1.0:0.70"
schemaLocation="CoreComponentParameters.xsd" />
<!-- report namespace="urn:oasis:names:tc:ubl:CoreComponentTypes:1.0:0.70"
schemaLocation="CoreComponentTypes.xsd" />
<!-- report namespace="urn:oasis:names:tc:ubl:CommonAggregateTypes:1.0:0.70"
schemaLocation="UBL_Library_0p70_Reusable.xsd" />
<!-- element name="Invoice" type="InvoiceType" />
- <!-- complexType name="InvoiceType" id="UBL700001">
- <!-- annotations>
- <!-- documentation>
<!-- ccts:ABIS dictionaryEntryName="Invoice, Details" definition="complete information of the invoice."
qualifierTermObjectClassTerm="" objectClassTerm="Invoice" qualifierTerm_PropertyTerm="" propertyTerm=""
representationTerm="Details" xmlns:ccts="urn:oasis:names:tc:ubl:CoreComponentParameters:1.0:0.70" />
<!-- documentation>
<!-- annotations>
<!-- sequence>
- <!-- element ref="cat:ID" minOccurs="1" maxOccurs="1">
- <!-- annotations>
- <!-- documentation>
<!-- ccts:ABIS dictionaryEntryName="Invoice, Identifier" definition="the unique number assigned to the invoice
by the seller (invoice)" qualifierTermObjectClassTerm="" objectClassTerm="Invoice"
qualifierTerm_PropertyTerm="" propertyTerm="Identification" representationTerm="Identifier"
xmlns:ccts="urn:oasis:names:tc:ubl:CoreComponentParameters:1.0:0.70" />
<!-- documentation>
<!-- annotations>
<!-- element>
- <!-- element ref="cat:IssueDate" minOccurs="1" maxOccurs="1">
- <!-- annotations>

```

First 45 of 200 lines of Invoice specifications. UBL library.

ECONOMICS OF STANDARDS

The development of Web Services is driven by:

- ❑ Consumers — industry and government demand for inter-operability.
- ❑ Vendors — need to “grow” new markets. (Web Services is very good for software tool makers!)
- ❑ Vendors — desire to dominate developing markets with their own proprietary technology.

Collaboration

Consumer and vendor needs work together: consumers buy value, vendors develop and sell it.

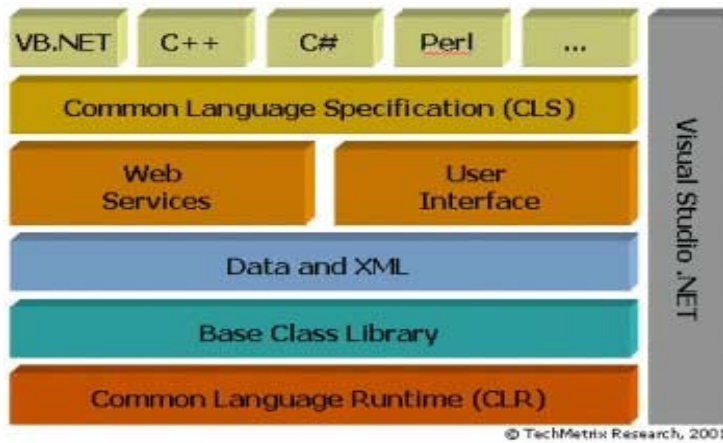
Conflict

Consumer’s desire for choice and secure, broad standards conflicts with vendors’ need to differentiate, and to pursue development programs that are more efficient than public forums, and that produce more profitable results for them.

All players want standards, but only consumers want comprehensive standards. Vendors want to retain some proprietary control over the development of some aspect of the product in order to make their offerings more attractive, and to exclude competitors.

The more domineering vendors want the greatest control over standards. For example, Microsoft and IBM will not join UBL in the development of ebXML standards.

Compare these Microsoft’s view of .Net with the structure of web services in general and you see that Microsoft would like to keep web services within their control.



“The primary goal of .NET is to provide developers with the means to create interoperable applications using "Web Services" from any sort of terminal”

— From <http://www.cswl.com/whitepr/white/microsoftnetdoc.html>



The structure of web services. — From “The Business Web Services Scorecard, Using Web Services for Business” by David Burdett, Commerce One Operations, Inc.

The structure of .Net matches the whole web service architecture except for two elements. One is the “Profiles, Policies & Agreements” which are determined largely by customers. These lie outside of the control of software vendors. The other is the operating system that lies at the bottom of the web service diagram. This is omitted from the .Net diagram for the obvious reason that Microsoft supports only one operating system.

Note that the OS plays an indirect role in web services, hence Apple’s absence from that group of vendors vying for control of standards.

POLITICS OF STANDARDS

There are built-in conflicts that will not go away.

Developers will have mixed allegiances.

Complex Standards — Wide markets for all, but also uncertainty, complexity, instability, incompleteness, and lack of availability.

Proprietary Standards — Disciplined, relevant, easy to use, advantageous to those who profit from them.

Competition — Just like vendors, developers also want tools that will make their offerings more attractive than the products of their competitors.

For example:

- ❑ **J2EE** offers easier XML programming (XML structures are mapped to Java language structures).
- ❑ **.Net** offers extended inter-operability within the Microsoft product line.

Vendors will never fully support open standards

W3C						
OASIS						
ISO		Microsoft's		Sun's		IBM's
ASC X12	versus	.Net	versus	J2EE	versus	OGSA
WS-I						(Open Grid Svc. Arch.)
etc.						

THE FUTURE

Because the marketplace requires new hype every few years, we know web service hype will be replaced by something else. If it were not replaced, then web services would create an entrenched hierarchy of users, vendors, and experts that would bore consumers and stifle innovation. Yet, to make web services work will require more than a few years. Will this great project reach some kind of completion, or will it morph into the next big thing?

Web services is an application-enabled version of the current World Wide Web. Both started with strong institutional backing and theoretical promise. The www has become highly commercial. I predict web services will do the same. It has yet to be seen if web services, like www, will end up driven by the same base and carnal customer interests.



“And they still ran faster and faster and faster, till they all just melted away, and there was nothing left but a great big pool of melted butter round the foot of the tree.”

— From “The Story of Little Black Sambo”

SUMMARY

Cautious Developer’s Approach

- ❑ Define your market.
- ❑ Discover what tools exist and what standards apply.
- ❑ Determine if the tools and standards support your markets’ need at an acceptable risk to you.
- ❑ Roll your own standards where ever necessary.
 - Use SOAP’s RPC mode and 4D variables until XML tools improve.
 - Program your own rules where usable standards are unavailable.
 - Encapsulate your variations and extensions to the standards to allow for future replacement.

Web services offer both new tools and new conflicts for developers. Battles will be fought over the setting of standards. Vendors will ask developers to invest in proprietary initiatives. Recognize the conflicts, let them work in your favor, design around them: