SOA-PEDIA

SOAP

(Simple Object Access Protocol) The standard for web services messages. Based on XML, SOAP defines an envelope format and various rules for describing its contents. Seen (with WSDL and UDDI) as one of the three foundation standards of web services, it is the preferred protocol for exchanging web services, but by no means the only one; proponents of REST say that it adds unnecessary complexity.

WSDL

(Web Services Description Language) The standard format for describing a web service. A WSDL definition describes how to access a web service and what operations it will perform. Usually pronounced 'whizz-dul' (to rhyme with 'whistle'), WSDL is seen (with SOAP and UDDI) as one of the three foundation standards of web services.

UDDI

(Universal Description Discovery and Integration protocol) A directory model for web services. UDDI is a specification for maintaining standardized directories of information about web services, recording their capabilities, location and requirements in a universally recognized format. Seen (with SOAP and WSDL) as one of the three foundation standards of web services, UDDI is currently the least used.

XML

(eXtensible Markup Language) The data tagging language of web services. XML is not so much a language as a standardized set of rules for adding structure to any form of data using a system of markup tags. Anyone can create their own markup vocabulary (called an XML Schema), and XML ensures that the structure will be intelligible to anyone else who consults the XML Schema document. More importantly, it enables XML-aware software to automatically manipulate the data without needing advance knowledge of the structure.

WORKFLOW

Sequence of tasks. A workflow describes the order of a set of tasks performed by various agents to complete a given procedure within an organization. Repetitive workflows are often automated, particularly in organizations that handle high volumes of forms or documents according to fixed procedures. The co-ordination and management of workflow is an important part of business process management (BPM).

WfMC

The Workflow Management Coalition mission is:

-Increase the value of customers' investment in workflow technology

-Decrease the risk of using workflow products

-Expand the workflow market through increasing awareness for workflow

BPM

(Business Process Management) Activities that hone and direct business processes, with the aim of enhancing their responsiveness and efficiency -- such as design, deployment, instrumentation, direction and analysis. Important elements include business process modeling, orchestration and business activity monitoring (BAM).

BPML

(Business Process Modeling Language) Language for creating automated business processes. BPML makes use of WSCI (web service choreography interface), and claims interoperability with rival specification BPEL4WS. BPML processes can be graphically represented using the companion Business Process Modeling Notation (BPMN).

BPMN

The goal of BPMN is to provide a business process modeling notation that is readily usable by business analysts, technical developers and business people that manage and monitor these processes. One of the goal of BPMN is also to be able to generate execution definitions (BPEL4WS) that will be used to implement the business processes. As such BPMN positions itself as a bridge between modeling and execution and between people that run the business and implementers of systems that support the business.

BPMN allows us to create a Business Process Diagram which represent the activities of the business process and the flow controls that define the order in which they are performed.

BAM

(Business Activity Monitoring) Electronic equivalent of 'minding the shop'. BAM software builds up a picture of business activity by monitoring and recording metrics from various different sources within an organization's software infrastructure. Because it understands the business context of the information it collects, it can translate disparate software events, for example the creation of sales orders and invoicing, into information about how the business is performing, such as calculating inventory turns or days sales outstanding.

WSCI

The Web Service Choreography Interface (WSCI) is an XML-based language used to describe the flow of messages exchanged by a Web service in the context of a process. It allows the description of the observable behavior of a Web service in a message exchange. WSCI also describes the collective message exchange among interacting Web services, providing a global, message-oriented view of a process involving multiple Web services.

Wf-XML

This is a web services protocol that can be used to a process engine remotely for the purpose of sending or retrieving the process definitions. Most people are aware that various process definition languages have been developed in recent years; most famously is OASIS WS-BPEL, but equally notable is WfMC XPDL, as well as other specialized process definition languages. The process definitions that are described by these languages are expected to be installed into a process engine for execution. The language itself naturally does not define how to install the definition into the engine. Instead, this is the role of Wf-XML.

XPDL

This is a standard for exchanging process *design* between process modelling tools.

BPEL

An XML-based language for the formal specification of business processes and business interaction protocols. BPEL extends the Web Services interaction model and enables it to support business transactions. It is the result of a cross-company initiative between IBM, BEA and Microsoft to develop a universally supported process-related language.

SOA

A service-oriented architecture is a collection of services that communicate with each other. The services are self-contained and do not depend on the context or state of the other service. They work within a distributed systems architecture.

SCA

SCA gives you a model to define interfaces, implementations, and references in a technology neutral way, letting you then bind these elements to whichever technology specific implementations you choose. Focusing on SOA, several major software vendors announced, on Nov’05, a specification called Service Component Architecture (SCA), intended to foster development of composite applications.

SCA will allow developers to focus on writing business logic in the building and packaging of applications, according to a source familiar with the announcement. Participating vendors include BEA Systems, IBM, Oracle, SAP, Iona, Siebel and Sybase, the source said. The specification is expected to be submitted to an industry standards body, possibly OASIS, for consideration as an industry standard.

SCA is designed for SOA, unlike platforms such as J2EE, which have been adapted to SOA. SCA is intended to allow development of application assemblies without regard to middleware or language. It features the notion of a service, which is to be defined by an interface that includes a set of operations.

With SCA, a developer, for example, could build a quote to cash application that unites a CRM system with an order management system. Service Data Object technology, for simplifying how applications handle data, is key to SCA.

SCA uses the notion of declarative policies for elements such as security, transactions and reliable messaging. Built for composite applications, it is able to describe assemblies of components that have been written in a variety of programming models and protocols.

SDO

SDOs (Service Data Objects) specify a standard way to access data and can be used to modify business data regardless of how it is physically accessed. Developers and architects do not need to know the technical details of how to access a particular back-end data source in order to use SDO in their composite applications. Consequently, they can use static or dynamic programming styles and obtain connected as well as disconnected access.

REST

Representational State Transfer could be considered as the architecture style of the Web itself. REST is considered by some to be the underlying framework of the next generation of Web services.

There are two common perspectives on the process of architectural design, whether it be for buildings or for software. The first is that a designer starts with nothing--a blank slate, whiteboard, or drawing board--and builds-up an architecture from familiar components until it satisfies the needs of the intended system. The second is that a designer starts with the system needs as a whole, without constraints, and then incrementally identifies and applies constraints to elements of the system in order to differentiate the design space and allow the forces that influence system behavior to flow naturally, in harmony with the system. Where the first emphasizes creativity and unbounded vision, the second emphasizes restraint and understanding of the system context. REST has been developed using the latter process. Such “constraints”, in the design of a distributed system, are likely to be (a) the client-server architecture style, (b) the –usually stateless- client-server interaction, (c) the cache constraints to form the client-cache-stateless-server style, aiming to improve network efficiency, (d) the uniform interface between components, (e) layered system constraints, (f) code-on-demand, where client functionality is extended by downloading and executing code in the form of applets or scripts; this simplifies clients by reducing the number of features required to be pre-implemented.

SRM

The Service Component Reference Model (SRM) is a business and performance-driven, functional framework that classifies Service Components with respect to how they support business and/or performance objectives.

The SRM is intended for use to support the discovery of government-wide business and application Service Components in IT investments and assets in USA. The SRM is structured across horizontal and vertical service domains that, independent of the business functions, can provide a leverage-able foundation to support the reuse of applications, application capabilities, components, and business services.

OWL

The OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full.

The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web. The Semantic Web will build on XML's ability to define customized tagging schemes and RDF's flexible approach to representing data. The first level above RDF required for the Semantic Web is an ontology language what can formally describe the meaning of terminology used in Web documents. If machines are expected to perform useful reasoning tasks on these documents, the language must go beyond the basic semantics of RDF Schema. The OWL Use Cases and Requirements Document provides more details on ontologies, motivates the need for a Web Ontology Language in terms of six use cases, and formulates design goals, requirements and objectives for OWL.

OWL has been designed to meet this need for a Web Ontology Language. OWL is part of the growing stack of W3C recommendations related to the Semantic Web.

- **XML** provides a surface syntax for structured documents, but imposes no semantic constraints on the meaning of these documents.
- **XML Schema** is a language for restricting the structure of XML documents and also extends XML with datatypes.
- **RDF** is a datamodel for objects ("resources") and relations between them, provides a simple semantics for this datamodel, and these datamodels can be represented in an XML syntax.
- **RDF Schema** is a vocabulary for describing properties and classes of RDF resources, with a semantics for generalization-hierarchies of such properties and classes.

OWL adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes.

RDF

Short for Resource Description Framework. RDF is a general framework for describing a Web site's metadata, or the information about the information on the site. It provides interoperability between applications that exchange machine-understandable information on the Web. RDF details information such as a site's sitemap, the dates of when updates were made, keywords that search engines look for and the Web page's intellectual property rights.

Developed under the guidance of the World Wide Web Consortium, RDF was designed to allow developers to build search engines that relay on the metadata and to allow Internet users to share Web site information more readily. RDF relies on XML as interchange syntax, creating an ontology system for the exchange of information on the Web.

AJAX

(Asynchronous Javascript And Xml) Technique for dynamically updating web pages. AJAX is the term coined in February 2005 to describe a collection of technologies used to automatically update and manipulate the information on a web page while it is being viewed in a browser (ie without the user having to manually refresh the page). This allows developers to create more sophisticated web pages and applications without having to add to the native capabilities of the browser. A key component is the use of XMLHttpRequest, a function originally added to browsers by Microsoft, to exchange data in the background with one or more web servers.

XAML

Transaction Authority Markup Language (XAML) is a vendor-neutral standard that enables the coordination and processing of online transactions in the rapidly emerging world of XML web services – the revolutionary new model of Internet-based computing that is now being adopted by all major systems and software vendors. XAML is intended to be a completely open standard for web-based business transactions.

On October 25, 2000, leading proponents of e-business interoperability, as Bowstreet, Hewlett-Packard Company, IBM, Oracle Corporation, and Sun Microsystems, announced that they are leading an initiative to define a vendor-neutral industry standard that will enable the coordination and processing of on-line, multi-party transactions in the rapidly emerging world of XML-based web services. The initiative is called XAML (Transaction Authority Markup Language). The XAML initiative addresses coordinated processing of transaction-supporting web services between internal fulfillment services (the chemical provider's inventory system) and external services such as: (1) An insurance policy service to insure the product being shipped; (2) A financing service to ensure payment according to vendor terms; (3) A transportation service to guarantee timely shipment/delivery of product; (4) A regulatory service to ensure compliance with government safety requirements. The XAML standard will: (1) Provide a specification for the XML message interfaces and interaction models of web services to support the coordination and processing of multi-stage transactions on the Internet; (2) Specify interfaces and protocols that preserve investment and strengths in transaction monitors and resources; (3) Specify interfaces and protocols that can be 'added on' to existing and emerging web service interfaces and protocols; (4) Specify interaction models for software systems to provide business-level transactions that coordinate the processing of multiple distributed web services; (5) Build on existing and emerging industry standards. The XAML initiative is so-named because it is an extension of XML, the common language of e-commerce, which supports transactional semantics as defined by the widely adopted standard for two-phase commit, XA (Transaction Authority). XAML intends to provide a means for transaction supporting web services to participate in higher-level business transactions. The XAML proposal will be submitted to one or several standards bodies that may include the W3C, OASIS (Organization for the Advancement of Structured Information Standards) and/or the IETF (Internet Engineering Task Force).

ebXML

ebXML is a joint initiative of the United Nations (UN/CEFACT) and OASIS, developed with global participation for global usage. It is a set of specifications that together enable a modular electronic business framework. The vision of ebXML is to enable a global electronic marketplace where enterprises of any size and in any geographical location can meet and conduct business with each other through the exchange of XML based messages.

MDA

The Model Driven Architecture (MDA) provides an open, vendor-neutral approach to the challenge of interoperability, building upon and leveraging the value of OMG's established modeling standards: Unified Modeling Language (UML); Meta-Object Facility (MOF); and Common Warehouse Meta-model (CWM).

Platform-independent Application descriptions built using these modeling standards can be realized using any major open or proprietary platform, including CORBA, Java, .NET, XMI/XML, and Web-Based platforms.

The aim of Model Driven Architecture is to allow definition of machine-readable application and data models which allow long-term flexibility of:

- **implementation**: new implementation infrastructure can be integrated or targeted by existing designs.
- **integration**: since not only the implementation but the design exists at time of integration, we can automate the production of data integration bridges and the connection to new integration infrastructures.
- **maintenance**: the availability of the design in a machine-readable form gives developers direct access to the specification of the system, making maintenance much simpler
- **testing and simulation**: since the developed models can be used to generate code, they can equally be validated against requirements, tested against various infrastructures and can used to directly simulate the behavior of the system being designed.

OMG

The Object Management Group (OMG) is an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications. Our membership includes virtually every large company in the computer industry, and hundreds of smaller ones. Most of the companies that shape enterprise and Internet computing today are represented on our Board of Directors.

OMG’s flagship specification is the multi-platform Model Driven Architecture (MDA), recently underway, but already well known in the industry. It is based on the modeling specifications the MOF, the UML, XMI, and CWM. OMG's own middleware platform is CORBA, which includes the Interface Definition Language OMG IDL, and protocol IIOP. The Object Management Architecture (OMA) defines standard services that will carry over into MDA work shortly. OMG Task Forces standardize Domain Facilities in industries such as healthcare, manufacturing, telecommunications, and others.

CORBA

CORBA is the acronym for Common Object Request Broker Architecture, OMG's open, vendor-independent architecture and infrastructure that computer applications use to work together over networks. Using the standard protocol IIOP, a CORBA-based program from any vendor, on almost any computer, operating system, programming language, and network, can interoperate with a CORBA-based program from the same or another vendor, on almost any other computer, operating system, programming language, and network.

CORBA is useful in many situations. Because of the easy way that CORBA integrates machines from so many vendors, with sizes ranging from mainframes through minis and desktops to hand-helds and embedded systems, it is the middleware of choice for large (and even not-so-large) enterprises. One of its most important, as well most frequent, uses is in servers that must handle large number of clients, at high hit rates, with high reliability. CORBA works behind the scenes in the computer rooms of many of the world's largest websites; ones that you probably use every day. Specializations for scalability and fault-tolerance support these systems. But it's not used just for large applications; specialized versions of CORBA run real-time systems, and small embedded systems.

As of a tech insight: CORBA applications are composed of objects, individual units of running software that combine functionality and data, and that frequently (but not always) represent something in the real world. Typically, there are many instances of an object of a single type - for example, an e-commerce website would have many shopping cart object instances, all identical in functionality but differing in that each is assigned to a different customer, and contains data representing the merchandise that its particular customer has selected. For other types, there may be only one instance. When a legacy application, such as an accounting system, is wrapped in code with CORBA interfaces and opened up to clients on the network, there is usually only one instance.

For each object type, such as the shopping cart that we just mentioned, you define an interface in OMG IDL. The interface is the syntax part of the contract that the server object offers to the clients that invoke it. Any client that wants to invoke an operation on the object must use this IDL interface to specify the operation it wants to perform, and to marshal the arguments that it sends. When the invocation reaches the target object, the same interface definition is used there to unmarshal the arguments so that the object can perform the requested operation with them. The interface definition is then used to marshal the results for their trip back, and to unmarshal them when they reach their destination.

The IDL interface definition is independent of programming language, but maps to all of the popular programming languages via OMG standards: OMG has standardized mappings from IDL to C, C++, Java, COBOL, Smalltalk, Ada, Lisp, Python, and IDLscript.

MOF

The MOF is the OMG's Meta-Object Facility standard. It defined a suite of CORBA-based services for managing meta-information.

It is a generic framework for describing and representing meta-information in an CORBA-based environment. In this context, the term "meta-information" covers any information that in some sense describes other information. This is intended to include such things as:

Interface definitions for CORBA objects, COM objects, DCE services and so on, Service types for the CORBA Trader, Meta-data for databases and information retrieval systems, Models and project management information for software development tools, Mapping descriptions for interoperability tools; e.g. application level bridges.

The MOF is designed to support many different kinds of meta-information. This is achieved by treating the meta-information as information, and formally modeling each distinct kind of meta-information. These formal models are expressed using the meta-modeling constructs provided by the MOF Model.

The MOF specification also defines an IDL mapping which allows models expressed using MOF Model constructs to be translated into interfaces CORBA-based meta-information services. These interfaces can be implemented by hand or using non-standard server generation tools.

The mapped interfaces for a meta-model all inherit from a standard Reflection module that supports "introspection" and meta-model independent access and update. The interfaces can be used within a MOF Repository framework, or deployed independently.

CWM

The Common Warehouse metamodel (CWM) specifies interfaces that can be used to enable easy interchange of warehouse and business intelligence metadata between warehouse tools, warehouse platforms and warehouse metadata repositories in distributed heterogeneous environments. CWM is based on three standards:

UML - Unified Modeling Language, an OMG modeling standard.

MOF - Meta Object Facility, an OMG metamodeling and metadata repository standard.

XMI - XML Metadata Interchange, an OMG metadata interchange standard.

SAML

SAML is a framework for exchanging authentication and authorization information. Security typically involves checking the credentials presented by a party for authentication and authorization. SAML standardizes the representation of these credentials in an XML format called assertions, enhancing the interoperability between disparate applications.

VoiceXML

VoiceXML is designed for creating audio dialogs that feature synthesized speech, digitized audio, recognition of spoken and DTMF key input, recording of spoken input, telephony, and mixed initiative conversations. Its major goal is to bring the advantages of Web-based development and content delivery to interactive voice response applications.

SCXML

SCXML is a flexible state machine language that combines concepts from CCXML and Harel State Tables. It enhances the basic concept of state machines with such powerful concepts as conditions on transitions and nested and parallel states. As a result, it provides a compact and intelligible representation of complex systems. SCXML is being developed in conjunction with VoiceXML 3, but it will be useful in a wide variety of applications involving the control and synchronization of asynchronous resources.

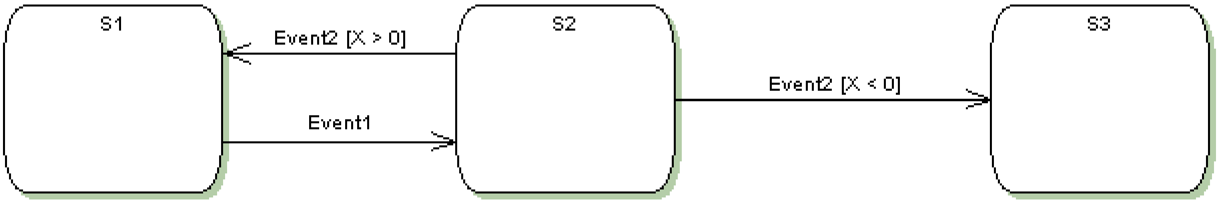
CCXML

Call Control eXtensible Markup Language (CCXML) is an XML standard designed to provide telephony support to VXML. Where as VXML is designed to provide a VUI interface to a voice browser, CCXML is designed to inform the voice browser how to handle the telephony control of the voice channel. The two languages are wholly separate are not required by each other to be implemented.

Harel State Table

The event processing model and the semantics of states and transitions are defined in detail in the UML specification. Nevertheless it is worth outlining some of their salient features here. The most basic concepts are states and transitions. Diagram 1 shows a simple state machine with three states S1, S2, and S3.

Example: A Simple State Machine



SCXML Equivalent

```
 <?xml version="1.0" encoding="us-ascii"?>
<scxml version="1.0" xmlns="http://www.w3.org/2005/07/scxml">
  <state id="S1">
    <transition event="Event1">
      <target next="S2"/>
    </transition>
  </state>
  <state id="S2">
    <transition event="Event2">
      <target next="S1"
    cond="X&gt;0"/>
    </transition>
    <transition event="Event2">
      <target next="S3"
      cond ="X&lt;0"/>
    </transition>
  </state>

  <state id="S3">
  </state>

</scxml>
```

These states linked by transitions, which are triggered by events. In this example State S1 transitions to state S2 when the event Event1 occurs. This is the only way the system can move from S1 to S2, since it is the only transition between those two states. Any other events that occur while the system is in S1 will therefore be ignored. The logic in S2 is slightly more complex. It has two transitions, both triggered by event Event2. Both transitions have guard conditions that check the value of the variable X. If X < 0, S2 will transition to S3 when Event2 occurs. If X > 0, it will transition back to S2. Note that there is no condition covering the case where X = 0. Therefore the system will remain in state S2 if X = 0 when Event2 occurs. As in state S1, all events not mentioned in transitions will be ignored.

UML

The OMG's Unified Modeling Language™ (UML®) helps specifying, visualizing, and documenting models of software systems, including their structure and design, in a way that meets all of these requirements. (Can be used for business modeling and modeling of other non-software systems too.) Using any one of the large number of UML-based tools on the market, we can analyze future application's requirements and design a solution that meets them, representing the results using UML 2.0's thirteen standard diagram types.

XPS

The XML Paper Specification (XPS) describes the XPS Document format and how it is organized internally and rendered externally. The XML Paper Specification builds on the Open Packaging Conventions. The XML Paper Specification is intended for producers, who want to create document files in the XPS Document format, and consumers, who want to access, render, or process the contents of an XPS Document.

The XPS Document format describes a paginated set of related pages having a fixed layout organized in one or more fixed documents and stored according to the Open Packaging Conventions. The XPS Document format supports digital signatures and rights management of the contents. The XPS Document format also includes a provision for optional components that can group page contents together for easier streaming and provide information and control data for use when the document is printed.

ODF

OASIS Open Document Format for Office Applications (OpenDocument) is a standardized XML-based file format specification suitable for office applications. It covers the features required by text, spreadsheets, charts, and graphical documents.

DTD

Document Type Definition, part of XML specification, as inherited, in simplified form, from SGML.

DSDL

DSDL ("Document Schema Definition Languages") is a project of the ISO/IEC JTC 1/SC 34 (chair, Jim Mason) Working Group 1 (chair, Charles Goldfarb). The word "document" is meant to be read as "XML document-oriented applications", as opposed to data-oriented applications. "Languages" is used in the plural form because DSDL is not intended to point to a One True Schema Language.

DSDL is necessary because other XML schema languages (primarily W3C XML Schema) do not meet the needs of "document heads", and document validation is too complex to be done using a single language. The goal is to propose a set of specifications which will include a framework, several schema languages (including Relax NG and Schematron), a datatype system, and other pieces needed for document validation.

XSLT

Extensible stylesheet language transformation (XSLT) is a language for transforming XML documents into other XML documents. XSLT is designed for use as part of XSL, which is a stylesheet language for XML.

XSL

Extensible Stylesheet Language. A language for creating a style sheet that describes how data sent over the Web using XML is presented in a browser. XSL provides a programmer with the tools to describe exactly which data fields in a XML document are displayed and how they are displayed.

A W3C recommendation. This standard includes methods for adapting or changing information structures (transformation). Using XSL, it is possible to realize automatic transformation from XML to e.g. HTML or XSL-FO. XSL-FO (formatting objects) is a page description language that is used to generate printed formats with an XML notation.

Table of Contents

SOAP 1
WSDL 1
UDDI........ 1
XML........ 1
WORKFLOW........ 2
WfMC 2
BPM 2
BPML........ 2
BPMN 2
BAM 3
WSCI........ 3
Wf-XML 3
XPDL 4
BPEL........ 4
SOA........ 4
SCA........ 4
SDO........ 5
REST........ 5
SRM 6
OWL 6
RDF........ 7
AJAX 7
XAML........ 8
ebXML........ 9
MDA 9
OMG 10
CORBA........ 10
MOF........ 11
CWM........ 12
SAML 12
VoiceXML 12
SCXML........ 13
CCXML 13
Harel State Table........ 13
UML........ 14
XPS 14
ODF........ 15
DTD 15
DSDL 15
XSLT........ 16
XSL........ 16