

World Wide Web Consortium Workshop on Web services

11-12 April 2001
San Jose, CA – USA

Marwan Sabbouh, The MITRE Corporation
Stu Jolly, The MITRE Corporation
Dock Allen, The MITRE Corporation
Paul Silvey, The MITRE Corporation
Paul Denning, The MITRE Corporation

Abstract

The Web has evolved into islands of services that cannot interoperate easily with each other. Interoperable Web services are the fundamental building blocks of the next evolution of the Internet. Collaboration of the services will be the value-added service model. As a result, web site interactions will become transparent to the end user. In this position paper, we address issues that directly affect this evolution.

Introduction

In the world of technology, where evolution is certain, program managers are constantly challenged to make wise investments in technology. The need for interoperability is raising many questions such as: Should the system be based on DCOM or CORBA? Should the system support multiple platforms? All answers agree that the systems must be web enabled. Within the government environment, no one technology prevails. Information systems are implemented using different technologies and by different contractors. Yet, it is highly desirable to have these systems integrate and interoperate with one another. Web services enable interoperability of heterogeneous systems and allow the use of best of breed. In the next section, we introduce specifications that are used to implement Web services. Then we address the issues of interoperability, description and discovery, quality of service (QoS) and context.

Web Services

A Web service is a service that supports Web standards. It exposes its functionality and method of access in an XML file. Typically, this file contains a description of the message pattern and operation supported by the service. Also, this file contains a section that binds these operations into a concrete protocol. One example of such file is the Web Service Description Language (WSDL) [1]. This file is published in the Universal Description, Discovery, and Integration (UDDI) [2] registry repository. A Web service uses an XML based protocol for the exchange of information. This protocol specifies conventions for packaging a message and its processing rules. The Simple Object Access Protocol (SOAP)[3] is an example of an XML based protocol. The SOAP Messages with Attachments [4] specification defines the binding for SOAP messages to be carried within a MIME multipart/related message.

The current W3C XML Protocol (XMLP) [5] Working Group has been chartered to define a simple, extensible protocol standard based to a great extent on SOAP 1.1. In order for Web services to reach their potential, standards organizations need to address the areas of discovery, description, quality of service, and context. Also, extensions to XMLP that provide added value, such as reliable delivery semantics, must be addressed.

Interoperability

As the use of web services becomes pervasive, interoperability becomes increasingly important. Simply stated, a web service deployed and published, must be discovered and consumed by client programs, irrespective of the tools used in writing the service and the client. For example, a SOAP client program that uses the Microsoft's SOAP Toolkit for Visual Studio must consume a web service that uses IBM's Development Toolkit. Currently, SOAP1.1 specifications make the assumption that a SOAP processor is SOAP1.1 compliant if the on the wire format of the packet is compliant with the specification. This requirement, is a necessary condition for interoperability, but is not sufficient.

As more organizations adopt the SOAP1.1 specification, it is becoming clearer that not all aspects of the SOAP specification are being adopted. In particular, section five of the specification, titled SOAP Encoding is not being supported as is. A second issue is the character set encoding, both sender and receiver must support the same encoding of the character set. This highlights the question of which encoding specification must we use? A third issue is the protocol binding issue; both the client and the server must bind to the same transfer protocol. Standardization on a protocol binding, such as HTTP, enhances interoperability and makes headways in the movement to thin clients. Note that we are not advocating the exclusion of other protocol bindings to the XML protocol layer. Clearly, a balance will have to be maintained between flexibility and interoperability.

Context

A Web service, or the aggregation of Web services, may behave differently depending on some metadata that either accompany a web service request or be imported to the web service environment using an out of bound mechanism. Thus, the behavior of web services is determined at runtime. This has the benefit of the code adopting different behavior. This begs the question, as to what constitutes context. Context is any information that describes the user who is making the request. Examples of context include, but not limited to, a user's identity, location, role, and type of access device and interests. In this paper, we advocate a uniform vocabulary, based on XML, that describes a user's context. Additionally, context information must also travel on the wire with the request. For example, assume a user has access to two providers of web services; Provider A and provider B. Having authenticated with provider A, a user must be able to consume web services of provider B without authenticating a second time. Also, provider A can make a web service request of provider B on behalf of the user. This is possible if

context information always travel with the request. It should be possible for a web service to ignore context information it is not interested in.

Universal Description, Discovery, and Integration

UDDI is about to become the de facto standard for web services management on the web. It provides an XML/SOAP standards based framework for describing, discovering and managing the world of web services that will facilitate the growth of an integrated e-commerce environment. Within the government environment there is a need for the development of a virtual marketplace of data and services that will allow the evolution into a distributed environment for integrated planning and control. No single organization or contractor will be able to envision or accomplish this integrated vision. It will evolve from knowledge of the available resources and services by all involved parties and from experimentation and selection of best of breed. The UDDI standards and tools being developed by the commercial world can provide a non-proprietary market place where agencies and contractors can describe their mission related roles and the types of services and data that they provide. The searchable central registries will provide a publish and subscribe mechanism to store the agency and service descriptions and to point to detailed technical specifications that define the interfaces to these services. Metadata for the organizations, the services and the specifications will all be stored in standard XML schemas so that the information can be discovered using standard search tools. The registries will maintain technical fingerprints for specifications so that any change in the status of the service interface is immediately advertised on the web. While UDDI will not solve the problem of controlling data formats or force standard service interfaces, it will provide a uniform system for organizations to define their role in the overall government society, a uniform way for these organizations to present their resources (services and data), and a uniform way for the rest of the community to discover, develop integration strategies, and manage their access to distributed services in the web environment.

No doubt, few UDDI registries will surface on the Internet. If the evolution of the UDDI registries mimics those of search engines, one might expect that the UDDI registries will suffer similar disadvantages. Those disadvantages include broken links and invalid or irrelevant entries. To prevent these problems, there is a need of having a standard mechanism for removing outdated entries from the registry. Also, a mechanism for validating entries must be in place. In other words, the integrity of the data in the UDDI registry must be preserved. Another issue is whether UDDI registries are synchronized with each other. This implies that a registration is propagated to all other registries. The disadvantage is that an invalid registration must be removed from all registries. Alternatively, a web service can be registered in multiple UDDI registries.

Web Services Profile Languages and Matchmaking

In order to facilitate sophisticated matchmaking between providers of web services and candidate users of those services, a common language for "profiling" these services would be extremely helpful. This language would ideally be a declarative specification of constraints over distinguishing service characteristics, based on authoritative Community

of Interest metadata standards, such as (for example) an XML Schema definition of Dublin Core RDF metadata tags. This information could be included with the services' UDDI registration. The process of matching provider and recipient profiles should admit the possibility that matching is sometimes partial, and provide for some notion of candidate goodness of fit in results. Further, providers and recipients should be treated equally, so that a provider can be just as easily matched against a candidate set of recipients as can a recipient be matched against a candidate set of providers. If this profiling language is to be practically useful on a large scale, providers of web services will need a general mechanism to express relative "costs" (not necessarily just monetary costs), and recipients will likewise need a mechanism for specifying relative "benefits", so that trusted intermediary matchmakers can factor in concerns of service capabilities and preferences. In cases where providing this information would raise privacy concerns, it should be possible for a participant to hide its value system logic from the intermediary, thereby assuming more of the ranking and selection responsibilities directly.

Quality of Service

QoS is expected to become a value-added capability of emerging web services, which Internet businesses will want to advertise in registries, such as UDDI, to distinguish themselves from their competitors. A class of web services is expected to emerge that provides its service using XMLP, and also requires QoS extensions to XMLP. An appropriate forum is needed to define these QoS extensions in a manner consistent with other XMLP extensions. QoS extension of XMLP would be used in conjunction with other XMLP extensions, e.g., extensions for digital signatures; such extensions may be orthogonal to QoS, or tightly integrated and interdependent. QoS extensions to XMLP would allow propagation of QoS on an end-to-end basis allowing intermediaries to select appropriate lower layer QoS mechanisms when forwarding XMLP messages.

Any infrastructure supporting web services must be able to handle interactions which have temporal requirements. Examples include services that are required periodically, services that have completion deadlines, reconstruction of event sets, and construction of workflows. These requirements must be met within the context of other QoS characteristics such as the cost of the service, and its importance. Three foundation technologies are needed: a means to express temporal (and other) QoS needs and capabilities of services, a set of operations to support reasoning about temporality, and a means to construct temporally conditioned triggers and workflows.

Many well understood scheduling disciplines have a set of parameters that represent temporal QoS as it applies to the particular discipline. Commonly seen parameters include deadline for completion of a service, importance, utility functions, the time required to execute a service (worst case, average case, etc), priority (in the sense of scheduling priority or operating system priority), and required frequency. Some of these parameters characterize the "client" needs, whereas others characterize the service. Meeting temporal needs requires that both types of data be expressed so that the underlying infrastructure can use them.

Another aspect is temporal reasoning. This includes operations that allow reasoning about time stamps (is X within 5 minutes of Y, did X occur before Y, in what sequence did X and Y occur, did X occur while Y was true), the ability to attach time stamps to requests and data, and the ability to determine the accuracy of time stamps (synchronization of time is ideal).

Finally, you need expressions that allow services to be triggered by combinations of temporal and non-temporal conditions (if X occurs with 30 minutes of Y - do Z within 10 minutes, if X does not happen in the next 10 minutes - do Z as long as Z can be done within the next 20 minutes - otherwise do nothing). This capability supports triggering of services, and the construction of temporally conditioned work flows. If this support is omitted, the applicability of the service support is severely limited to interactions which either have no deadlines, or for which the deadlines are sufficiently lax to allow the deadlines to be met by brute force.

Conclusion

The XMLP Working Group is addressing Web services interactions. Standards initiatives must be undertaken to address the complex integration of the above issues. Web services won't reach their full potential unless these issues are properly addressed.

References:

- [1] Web Service Description Language (WSDL) 1.0 (<http://www.msdn.microsoft.com/xml/general/wsdl.asp>)
- [2] Universal Description, Discovery, and Integration (UDDI) (<http://uddi.org>)
- [3] Simple Object Access Protocol (SOAP) Version 1.1 (<http://www.w3.org/TR/SOAP/>)
- [4] SOAP Messages with Attachments (<http://www.w3.org/TR/SOAP-attachments>)
- [5] XML Protocol Working Group (<http://www.w3.org>)