Introduction to UDDI:
Important Features and Functional Concepts

THE STENCIL GROUP

The Stencil Group, Inc.
912 Cole Street, PMB 287
San Francisco, California 94117

www.stencilgroup.com

October 2004

STATUS OF THIS DOCUMENT

This is a REVISED DRAFT of the UDDI Technical White Paper, and it incorporates the feedback of the technical committee to the initial draft. Please note that this document is a work-in-progress and should not be distributed outside the UDDI member committees, except to solicit comments before publication.

REVISION HISTORY

Draft 3: October 21, 2004

- Incorporated committee feedback to make several descriptions of UDDI functions more precise.
- Restricted use of word "key" in paper to refer exclusively to UDDI key management infrastructure. Goal is to reduce reader confusion with either PKI or with general sense of "important, critical, or major."
- Moved discussion of important functional concepts in specification ahead of discussion of new features of v3.

Draft 2: October 18, 2004

- Incorporated committee feedback on several descriptions of UDDI functionality.
- Clarified status of UDDI v2 and v3 with regard to OASIS and updated general information about OASIS.
- Clarified scope to be an introduction to UDDI, not just v3.
- Introduced proposed title for paper.
- Corrected several typos and word choices.

Draft 1: August 6, 2004

- Initial version.

OVERVIEW

The Universal Description, Discovery, and Integration (UDDI) protocol is a central element of the group of related standards that comprise the Web services stack. The UDDI specification defines a standard method for publishing and discovering the network-based software components of a service-oriented architecture (SOA). Its development is led by the OASIS consortium of enterprise software vendors and customers.

This paper provides a concise overview of the UDDI standard and highlights significant architectural changes in the recent Version 3 specification. In a companion white paper, we describe business scenarios that UDDI and related Web services infrastructure are well suited to help address.

TYPICAL APPLICTIONS OF A UDDI REGISTRY

A UDDI registry's functional purpose is the representation of data and metadata about Web services. A registry, either for use on a public network or within an organization's internal infrastructure, offers a standards-based mechanism to classify, catalog, and manage Web services, so that they can be discovered and consumed by other applications. As part of a generalized strategy of indirection among services-based applications, UDDI offers several benefits to IT managers at both design-time and run-time, including increasing code re-use and improving infrastructure management by:

- Publishing information about Web services and categorization rules specific to an organization
- Finding Web services (within an organization or across organizational boundaries) that meet given criteria
- Determining the security and transport protocols supported by a given Web service and the parameters necessary to invoke the service
- Providing a means to insulate applications (and providing fail-over and intelligent routing) from failures or changes in invoked services

A BRIEF HISTORY OF UDDI

When UDDI first was conceived, much of the attention was focused on the "UDDI

Business Registry" (UBR), a public implementation of the UDDI standard that represented a master directory of publicly available e-commerce services. In many ways, this public registry can be considered analogous to the root node of the DNS database, another successful example of a distributed registry infrastructure.

Although the UBR remains an important part of the UDDI project, it represents only one aspect of the overall effort. Just as the overwhelming majority of DNS activity occurs within the confines of a company's own network, so too do most UDDI implementations support a business' own Web services infrastructure.

This understanding is reflected as the UDDI specification has evolved to reflect the need for federated control in real-world operational requirements, as well as to further integrate the standard with other elements of service-oriented infrastructure. Highlights of the standard's progress are shown in the table below.

Figure 1: History of the UDDI Specification

UDDI VERSION	YEAR RELEASED	KEY OBJECTIVE
1.0	2000	Create foundation for registry of Internet-based business services
2.0	2001	Align specification with emerging Web services standards and provide flexible service taxonomy. Formally released under OASIS aegis in 2003
3.0	2004	Support secure interaction of private and public implementations as major element of service-oriented infrastructure. To be released by OASIS in late 2004

The current 3.0 specification represents a significant milestone in UDDI's evolution. Its feature definitions are stable and backwards-compatible with earlier versions of the standard. In fact, a prerequisite of its certification by THE OASIS standards group was the existence of several working commercial implementations. We examine major functional features of UDDI in more detail, below.

KEY FUNCTIONAL CONCEPTS IN THE UDDI SPECIFICATION

UDDI describes a registry of Web services and programmatic interfaces for publishing, retrieving, and managing information about services described therein. In fact, UDDI itself is of set a Web services! The UDDI specification defines services that support the description and discovery of (1) businesses, organizations, and other Web services providers, (2) the Web services they make available, and (3) the technical interfaces which may be used to access and manage those services. UDDI is based upon several other established industry standards, including HTTP, XML, XML Schema (XSD), SOAP, and WSDL.

In this paper, we will highlight several important technical characteristics of an UDDI registry, but this introductory discussion necessarily is neither exhaustive nor definitive. The official UDDI specification formally describes the Web services, data structures, and behaviors of a registry that complies with the UDDI standard.*

The UDDI Data Model

The core information model used by a UDDI registry is defined in several XML schemas. XML was chosen because it offers a platform-neutral view of data and allows hierarchical relationships to be described in a natural way. XSD was chosen because of its support for rich data types and its ability to easily describe and validate information based on information models represented in schemas.

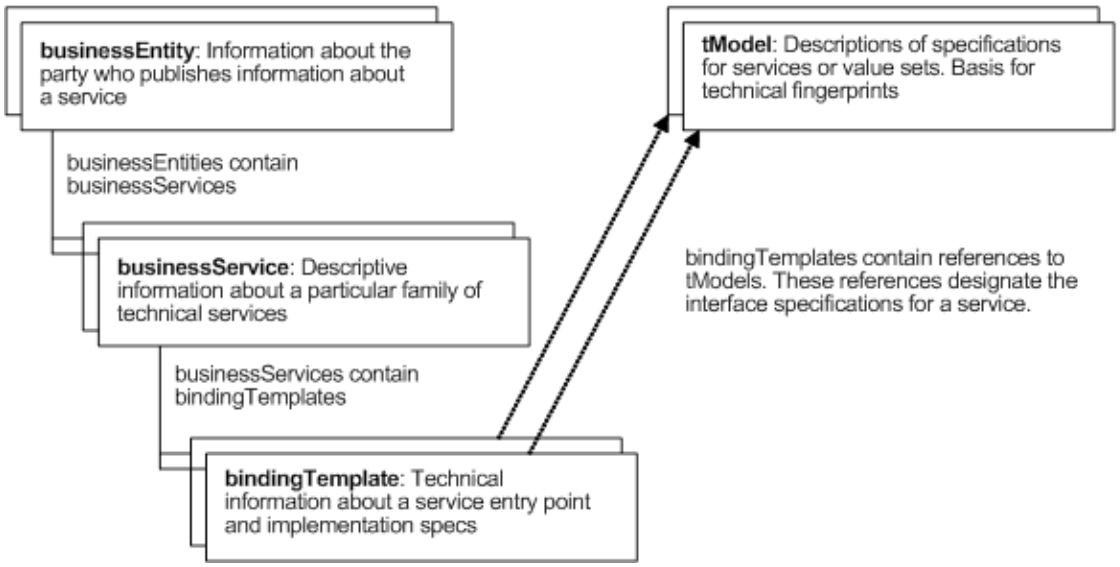
The UDDI XSDs define several core types of information that provide the kinds of information that that users and applications would need to know in order to use a particular Web service. Together, these form a base information model and interaction framework of UDDI registries. They are:

- A description of a service's business function (called the *businessService*)
- Information about the organization that published the service (*businessEntity*),
- The service's technical details (*bindingTemplate*), including a reference to the service's programmatic interface or API, and
- Various other attributes or metadata such as taxonomy, transports, digital

* The official UDDI specification can be found online at *http://www.uddi.org/specification.html*. Several technical notes and best practices documents also are available on this web site.

signatures, etc. (*tModels*).

Figure 2: UDDI's Core Data Types



UDDI versions 2 and 3 each add an additional data type to facilitate registry affiliation. Respectively, these are:

- Relationships among entities in the registry (*publisherAssertion*) and
- Standing requests to track changes to a list of entities (*subscription*).

These, like all UDDI data types, are expressed in XML and are stored persistently by a UDDI registry. Within a UDDI registry, each core data structure is assigned a unique identifier according to a standard scheme. This identifier is referred to as a UDDI *key*.

Taxonomic Classification of UDDI Entities

An important part of UDDI is providing a foundation and best practices that help provide semantic structure to the information about Web services contained in a registry. UDDI allows users to define multiple taxonomies that can be used in a registry. In such a way, users are not tied to a single system, but can rather employ an unlimited number of appropriate classification systems simultaneously. UDDI also defines a consistent way for a publisher to add new classification schemes to their registrations.

Defining UDDI Nodes, Registries, and Affiliated Registries

The UDDI specification includes a specific definition of the hierarchical relationship between a single instance of a UDDI implementation and others to which it is related. Technically, there are three major classifications of UDDI servers:

- A *node* is a UDDI server that supports at least the minimum set of functionality defined in the specification. It may perform one or more functions on the UDDI data to which it has access. It is a member of exactly one UDDI registry.
- A *registry* is composed of one or more nodes. A registry performs the complete set of functionality as defined in the specification.
- *Affiliated Registries* are individual UDDI registries that implement policy-based sharing of information among them. The affiliated registries share a common namespace for UDDI keys that uniquely identify data records.

Essential Programmatic Interfaces in UDDI

Although describing UDDI's application programming interfaces (APIS) is well beyond the scope of this paper, it is worth noting several features that support the concepts described above.

- Publishing information about a service to a registry
- Searching a UDDI registry for information about a service

These inquiry and publishing functions represent the core data management tools of a UDDI registry. Additionally, we have described how multiple registries may form a group, known as an affiliation, to permit policy-based copying of core data structures among them. Some of the most important concepts that support registry interaction include:

- Replicating and transferring custody of data about a service
- Registration key generation and management
- Registration subscription API set
- Security and authorization

The UDDI specification divides these functions into "Node API Sets" that are

supported by a UDDI server and "Client API Sets" that are supported (naturally enough) by a UDDI client.

UDDI VERSION 3: A FOCUS ON PRIVATE REGISTRIES AND REGISTRY AFFILIATION

Although many aspects throughout the UDDI specification have matured in the version 3 release,* the chief architectural change is the concept of "registry affiliation." This shift reflects the increasing recognition that UDDI is one element of a larger set of Web services technologies that support the design and operations of myriad software applications within and among business organizations.

Figure 3: Several "Flavors" of UDDI Registries

REGISTRY TYPE	DESCRIPTION	EXAMPLE APPLICATION
Corporate/Private	An internal registry, behind a firewall, that is isolated from the public network. Access to both administrative features and registry data is restricted. Data is not shared with other registries.	Enterprise Web Service Registry
Affiliated	A registry deployed within a controlled environment, but with limited access by authorized clients. Administrative features may be delegated to trusted parties. Data may be shared with other registries in a controlled manner.	Trading Partner Network
Public	From an end-user's perspective, a public registry appears to be a service in a cloud. Although administrative functions may be secured, access to the registry data itself is essentially open and public. Data may be shared or transferred among other registries, and content may or may not be moderated.	UDDI Business Registry (UBR)

* An inclusive list of major new features in UDDI version 3 is available on the UDDI web site at *http://www.uddi.org/pubs/uddi_v3_features.htm*.

For UDDI, businesses' increasing focus on service-oriented architecture has led to a strong ongoing emphasis to support a variety of infrastructural permutations and to provide a means to define the relationships among a variety of UDDI registries. Although the UDDI specification from the start included concepts like delegation and distribution among server peers, earlier definitions of the standard relied upon various proprietary means of interaction. By contrast, the current version provides an open, standardized approach to ensure widely interoperable communication.

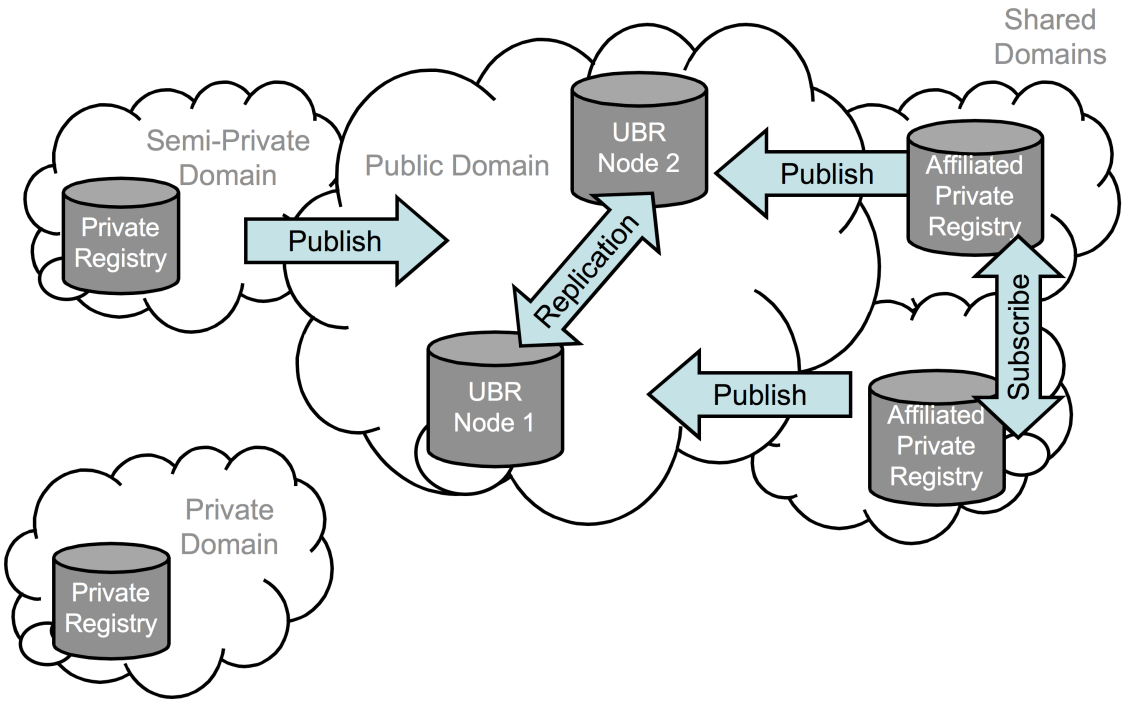
While the specification enables a technical interoperability of registries, it does not dictate the nature of or policies for such interaction. Rather, it leaves those issues to be decided upon by the registry operators. Obviously, the establishment of these policies, as well as a UDDI key management infrastructure, will become a critical element to successful distribution of registry responsibilities on not just a technical level, but also on a business process plane.

A Closer Look at Registry Affiliation

It is worth examining what is meant by the concept of registry affiliation in the UDDI specification. Simply put, affiliation refers to using UDDI to support a variety of network/infrastructure topologies. The possibilities have expanded from a stand-alone, single-registry approach to include hierarchical, peer-based, delegated, and others. In short, the structure of a UDDI registry (or registries) can now reflect the realities and relationships of the underlying business processes that it supports.

Managing multiple versions of registry entries presents a challenge, but it is a critical aspect of managing this sort of distributed infrastructure. The standard itself provides guidance to help facilitate the maintenance and mapping of UDDI keys and records across registries, but the specification is intended to do just that—facilitate, but not define, a wide range of business scenarios. It will be the registry operators, users, and software developers who design and implement a wide range of business policies and constructs on top of the basic UDDI infrastructure.

Figure 4: Conceptual Illustration of Registry Affiliation



Comment: This diagram illustrates several models of registry interaction enabled by Version 3 of the UDDI specification. Through mechanisms like publish/subscribe and replication among peer nodes of a registry, the information in UDDI servers can be fully public (like the UBR), semi-private (such as the affiliated registries shown here), or even fully private and isolated from the public network (as depicted in the "Private Domain" above).

HOW TO LEARN MORE

The UDDI specification is managed by OASIS, a member-led, international, non-profit standards consortium that concentrates on structured information and e-business standards. The organization's members include enterprise IT users, vendors, academics, governments, trade associations, and individuals. In addition to UDDI, OASIS is known best for shepherding Web services-related protocols such as EBXML, SAML, WS-Security, WSBPEL, and others.

To learn more about UDDI and OASIS, please visit *www.uddi.org*. In addition to the

specification itself, the UDDI Web site provides detailed technical notes, best practices, case studies, and information about how to contribute to UDDI's ongoing development. The site also provides links to several commercial and open-source implementations of UDDI registries that are available in the marketplace. Information about OASIS is available on *www.oasis-open.org*, and the work of the OASIS UDDI Specification Technical Committee can be found at *www.oasis-open.org/committees/uddi-spec*.