
Simplified Discovery Model: Key to Success

Claire Vishik

Abstract

Discovery of relevant objects and artifacts is an important topic in many disciplines. In economics, for example, numerous models have been developed for studying the effect of varying conditions on the search for products and general market efficiency. The emergence of the World Wide Web and the resulting prominence of open standards led to several attempts at standardizing discovery mechanisms, adapted for various usages and classes of applications. The importance of discovery to support higher levels of automation is understood by all and underscored by the number of specifications and proposals on discovery-related topics that appeared in the past several years.

This paper focuses on discovery paradigms that can be used in B2B frameworks. After describing the foundations of UDDI (Universal Description, Discovery and Integration) registry and registry/repository (RR) specifications in ebXML in order to assess their appropriateness for the creation of a large-scale general purpose registry of Web Services, the discussion moves to simpler models that could be used as a basis for building such a registry.

Based on the evaluation of the specialized registry specifications, the paper highlights the complexity of these highly structured discovery options. The complex structure can become a liability if used for a very large general-purpose Web Services registry. The paper further proposes a minimalist solution for generalized Web Services discovery mechanisms based on a very small number of required parameters and text indexing as well as the notion of trusted intermediaries. The paper doesn't provide technical specifications for the design and architecture of the discovery intermediary or techniques for processing of simple submissions; these aspects will be presented in subsequent papers. Instead, the paper concentrates on the rationale behind the idea of the simplification.

Table of Contents

1. Introduction	3
2. Statement of the Problem	3
3. Economics of Discovery	5
4. Discovery Contexts	6
4.1. UDDI Registry	7
4.2. ebXML RR (Registry/Repository)	9
5. Complex Registries and Search-Based Discovery Intermediaries: Coexistence	10
6. Why a Discovery Intermediary?	10
6.1. Submission, Processing, and Retrieval Process	12
7. Conclusions and Future Research	14
Bibliography	14

1. Introduction

As incentives to automate business processes and develop seamless collaboration continue to grow, so does the need to automate, to the extent possible, the discovery of services in multiple contexts. Many of the standards and state-of-the-art applications include a discovery component. Discovery-oriented standards ranging from BIB1 and Z39.50 to RDF have already been developed. And the work on specifications continues in other more specialized fields.

In the area of Web Services and related standards, Web Services (WS) framework emerged complete with UDDI (Universal Description, Discovery, and Integration). ebXML has defined registry/repository (ebXML RR) as an important part of the end-to-end B2B specification. In a different example, PKI (Public Key Infrastructure) is in dire need to devise a universal mechanism to enable discovery of credentials when they are used for collaboration, e.g. in secure email or secure document management. In a different field, Universal Data Element Framework (UDEF) is trying to define the components of semantic interoperability of already existing models in order to facilitate discovery and establish a global UDEF Registry and metadata registry/repository. The Liberty Alliance Project has expressed understanding that federated identities need a strong discovery mechanism to be fully functional. Semantic Web seeks to automate machine-to-machine communications by formalizing the discovery process of relevant elements. The list can be continued: interoperability is very important in today's technology, and discovery is its crucial element.

Yet even the most sophisticated discovery mechanisms containing semantic elements or possessing a very complex structure have not achieved wide adoption. UDDI standard emerged through a sustained development effort, and it is a strong specification, but it is now optional in the Web Services Framework. ebXML has a sophisticated information model for registry/repository, but the messaging components of the specification enjoy wider adoption. The idea of Semantic Web generated a lot of interest, but few large scale implementations or implementation ideas.

On the other hand, text search is the most common and the most successful discovery mechanism that has been used to process unstructured documents in a variety of formats and adapted for structured documents and databases. Text processing is based on pattern matching, not semantics, but modern technologies demonstrated that some pattern matching techniques can significantly increase precision in discovery. Although text search has become a de facto discovery mechanism for the Web, it is rarely considered as a foundation for discovery applications aimed at machine-to-machine communications. In the eyes of the developers, text search appears to lack precision necessary to execute automated tasks, but is it really so?

The discussion on whether it is necessary to define a domain in great detail in order to automate discovery or if minimalist descriptions are sufficient for the same purpose, is ongoing. The structuralists, those supporting detailed definitions for objects as a condition of future interoperability, believe that if most features remain optional, the standard will be meaningless because of a multitude of implementations. The minimalists think that a complex all-encompassing standard will not be used, and thus cannot be instrumental for achieving interoperability in discovery.

Why do the new discovery initiatives invariably grab attention, but don't achieve widespread adoption? What may be wrong with the current structured approach to a general solution? This paper attempts to answer this question and propose an alternative approach and mechanism for creating a general purpose discovery mechanism.

The paper highlights issues arising when the developers attempt to use current high quality, but complex and highly structured discovery options and proposes a simple solution for Web Services discovery based on a minimal number of required parameters and state-of-the art text indexing.

2. Statement of the Problem

Every concept is complex and can be defined in a variety of ways. The description is likely to be subjective, in that different "catalogers" will define these objects in different fashions, all of them easy to explain logically. In information science literature, many studies appeared describing the subjective nature of cataloging. There are very few objects that have unambiguous definitions, and the majority of these are in science rather than technology. In software especially,

there is little uniformity in describing similar artifacts, and there is no methodology available to normalize individual descriptions of objects. The issue of semantic interoperability is as burning now as it was thirty years ago.

Business applications rely on organization of information and search as the main means of information discovery. The organization of information could be achieved fully automatically through text processing and automatic categorization or constructed manually on the basis of the division of information into categories and registering the relationships between categories in a "directory." It is the manual construction of registries that has been the preferred method of design, although it is free text search that has acquired a far greater number of users. Even the registries that were designed for discovery automation, such as UDDI, have such a complex structure that it requires manual production by qualified individuals rather than reliance on machine generation. And the structural complexity makes automation nearly impossible. Those following the development and deployment of UDDI or ebXML registries may have noticed that the registries are used, in most projects, to promote governance and achieve control over referenced services rather than as a mechanism of discovery.

Although some parameters in the business applications (e.g. protocols and ports used for communication) are easy to present in standard ways, many other components, including those pertaining to Web Services, have a semantic component that is very difficult to define in a uniform fashion. But is it necessary to capture 100% of attributes of a concept and match every query exactly to one concept? Experience shows that minimalist approaches to retrieval, based on pattern matching rather than exact definitions of structure and semantics work fine in most environments.

It has been frequently stated that once the community agrees on semantics and semantic relationships within a system or a domain, it is easy to enforce semantic interoperability. Subject headings and call numbers used by libraries are frequently mentioned to illustrate an example of success of such an "agreement." Yet many studies consistently showed that individual catalogers assign different headings and different numbers to the same items, placing them in varying positions of the cataloging system ([Kashyap & Sheth, 1995](#)). And the catalogers' descriptions of objects are different from those used by searchers ([Ojala 1990](#)).

In complex domains and situations, even those with expert knowledge are unable to provide an exhaustive description of concepts and their relationships. Frequently, it is a special kind of observation that is necessary to shed light on some aspects of such domains ([Leonard, 1995](#)); in technology especially emphatic design is frequently used.

Study of search terms used in different systems show that there is no consistency even if the subject of search is clear to the searchers. In some research ([Vishik, 1997](#)), the variability of terminology was shown to be very extensive, with 60% of terms in large systems used by human searchers only once on a large Intranet, even though the majority of discovery actions were aimed at a very limited number of topics. The diversity seems to be a strong indication that in a large discovery system, it will be difficult to achieve an agreement on semantics that is universally shared and can enable significant discovery automation.

It is the developers who define and set-up discovery mechanisms and compatibility with them in tools that are in need of discovery. But the developer's view cannot be accepted as the universally shared view in most semantically oriented systems. There is a world of difference between the developer's view and the view of millions of potential end users (and this group includes other developers) ([Vishik, Farquhar, & Smith, 1999](#)).

However, while the majority of search terms are used only once, a small number of terms accounts for a significant proportion of searches. So, while there is considerable diversity in discovery approaches, there is also a de-facto consensus about topics and terms that are important. This minimalist shared view of the domain gives hope to developers and architects in establishing a discovery mechanism that capitalizes on a small number of shared notions and concepts to create discovery artifacts that are almost universally recognized and provide sufficient precision to support easy deployment and sufficient precision to enable automation.

We contend that the complexity of structure of the current discovery mechanisms is a deterrent from their more common adoption and from the automation of discovery mechanisms. Rather than trying to alter the variability of the users' perceptions of a domain and their discovery strategies, we propose to adopt the free form indexing and search approach to Web Services discovery, and create and promote general purpose discovery intermediaries that can take care of indexing of submitted Web Services, replacing the expired entries, enforcing access control and security, and acting as

a trusted entity for providers and consumers of Web Services. Although intermediation mechanisms have been described in many publications over the last two decades, few of them focused on discovery, but research in intermediation is applicable to the discovery issues.

3. Economics of Discovery

Efficiency in discovery mechanisms can be studied from various points of view, including exploration in terms of economics. Some general features of goods markets can be extended into the discovery “market,” thus highlighting the factors influencing the efficiency for information and Web Services discovery.

Among the factors affecting direct exchange of goods, double coincidence of wants is seen as a source of inefficiency in direct exchanges. First described by Jevons in 1875 ([Jevons, 1920](#)), double coincidence of wants relates to the fact that both traders involved in an exchange transaction not using a recognizable currency have to find the other agent’s offerings useful and desirable in order for that transaction to take place. Instead of simply acquiring what is needed, an agent has to locate another agent that not only has the desirable commodity, but also wants to exchange it for the commodity that the first agent offers. Such situations increase the waiting periods before transactions can occur for all traders, and therefore make commerce less efficient. The analogy in the discovery “market” is, for example, interaction between two unconnected systems with complex semantics where the transaction (=service discovery) cannot take place until the definitions used by the two systems can be “translated” and then matched for retrieval. In this case, discovery includes consecutive unconnected steps associated with each system that can’t be linked for greater efficiency. Technically, this market for discovery can work only on the basis of case-by-case customization.

However, it is not double coincidence of wants alone that leads to decreased efficiency of exchange markets. In order to correct, to some extent at least, the double coincidence of wants in exchanges, the market can create special places where pairs of products can be exchanged ([Bannerjee and Maskin, 1996](#)). This method will increase the number of markets and is costly, but it is feasible. Continuing the analogy with the discovery services “market,” we can liken this method to installing a service mapping application between any two discovery systems: if you map the expressions in one discovery mechanism to the other, the interaction becomes possible. In this case, each two services have a separate map, without which discovery cannot take place. Mapping is the approach currently used to overcome incompatibility between multiple discovery-oriented specifications. Numerous prototype and commercial systems that, e.g., map tables and columns in different databases in order to search them concurrently have been created, such as Infosleuth agent system at MCC in the late 1990s.

If double coincidence of wants can be remedied, although expensively, there are other reasons why direct exchange of goods and information is inefficient. According to Banerjee and Maskin, the main problem of barter resides in asymmetric information, unequal knowledge of buyers and sellers (in our case, developers and users) about the value of products. Because of the lack of expertise and ability to recognize intrinsic and market value of products, potential customers assume that they are running a high risk of making a mistake ([Akerlof, 1973](#)). The results of the research of influences of asymmetric information on markets are fully applicable to the study of discovery mechanisms. Indeed, asymmetric information is inevitably present in markets for informational products and other experience goods (e.g. Web Services), since their full value (and potential harmful effects) can be appreciated only when the product has been acquired (or tested) and its value transferred to the other party. Even if mapping is made available to the discovery mechanisms using other semantic models, lack of established trust mechanisms may render the interaction impossible. As an organization, you are not likely to allow your mission critical systems to interact with services of unknown origin and quality even if semantic interoperability is not an issue. Hence, a trusted partner that can guarantee quality and mitigate risks is necessary to enable the automation of the discovery process.

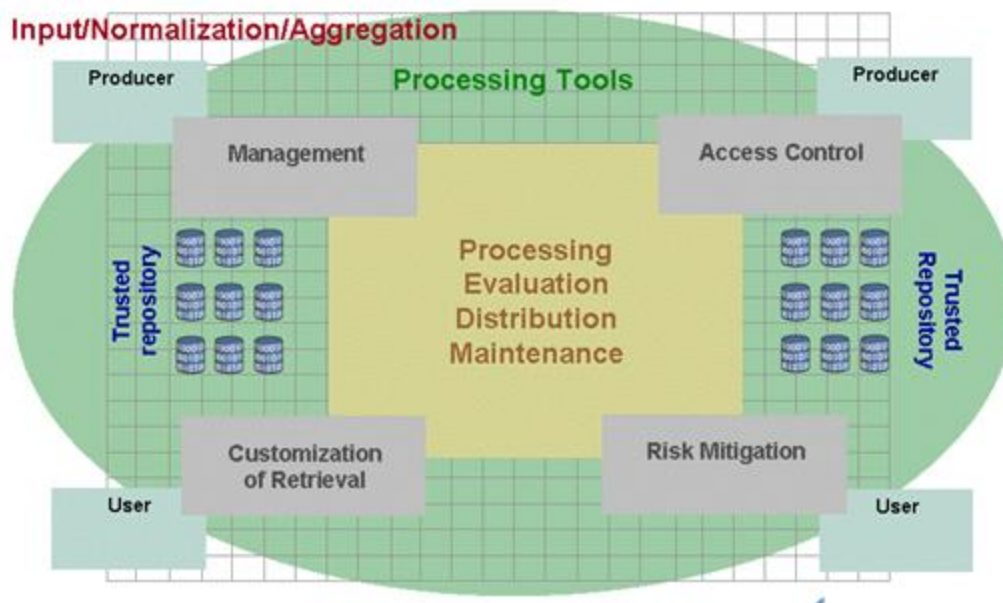


Figure 1. Digital Intermediaries

Marketing theory views intermediaries as “organizations that support exchanges between producers and consumers, increasing the efficiency of the exchange process by aggregating transactions to create economies of scale and scope” (Sarkar, Butler & Steinfield, 1996). According to Bose and Pingle (BP1996), the main function of intermediaries is to eliminate friction in markets (see also Williamson 1987). Intermediaries are also instrumental in shortening the length of time necessary for a transaction to complete, therefore decreasing the “search” efforts in a market. Bhattacharya and Hagerty (BH1995) see the role of intermediaries as “regulators.” Trust, rule setting, and decreased search efforts, associated with the role of intermediation in general, can and will bring serious improvements to the discovery business models (Vishik & Whinston, 1999).

4. Discovery Contexts

There is no doubt that the context (domain) knowledge is essential for the discovery process. At one end of the spectrum, there is the free text indexing of an extremely large repository, such as www.google.com [http://www.google.com/] where any topic can be explored using a variety of vocabularies and even languages. At the other end, there are systems that deal with very precise terms and structure to achieve discovery, e.g., chemical substances that have to be matched exactly for the discovery to be meaningful. The latest research by Google on using the text search engine for discovery in databases focusing on biology and genetics illustrates that the two paradigms are not as different as was previously concluded.

Precision matters little in some contexts, e.g. descriptions and narratives, and is essential in others, e.g. URLs. Uniqueness is key in some situations, but irrelevant in other cases (for example, in a text search, the fact that a source uniquely identifies Plutarch in Greek, practice not used in other English texts, probably has no additional discrimination value, while a reference expressed as a URI or URN is unique AND useful). Accessing the details about an object could be of paramount importance or not, depending on whether the discovery is focusing on the confirmation of existence of an object or more sophisticated tasks, such as that object’s properties. These differences are essential, but they could be built into an application based on its goals and projected functionality rather than incorporated in the discovery mechanism itself.

We envision an unstructured repository of information about public Web Services that is constructed automatically from submissions of very simple structures and retrieved by using a variety of mechanisms, with the emphasis on free

text search. Information about the Web Services is submitted by their owners or, if relevant, collected by a specially adapted Web crawler. The submission includes a unique ID number, service name, free text description, location information including URL, URI or references to related WSDL files. This information is maintained by a trusted intermediary and is available to all subscribers in automatic and manual modes. More information on the discovery intermediary will be presented in later sections. But before moving on to the detailed description, we present two Web Services related specialized discovery specifications (UDDI and ebXML RR) in order to highlight the complexity of these two standards and contrast them with the proposed approach.

4.1. UDDI Registry

UDDI was hailed as the “Yellow Pages for Web Services.” Started by an independent consortium, the development of the UDDI specification was passed on to OASIS after version 3.0 was completed. Since XML does not provide semantic interoperability, and WSDL (Web Services Markup Language) allows a provider to define end points for a Web Service, but contains no means to locate the WSDL descriptions, a dedicated discovery mechanism was conceived to support some level of automation. UDDI’s role was to support several aspects of discovery, but semantic interoperability remained elusive, even if the specification was commonly accepted. As a result, architecture for Web Services including an additional Interop Layer, placed on top of UDDI to support a higher level of interoperability circulated for a while.

The core of the UDDI project is business registration that is defined using white and yellow pages metaphors. White pages describe known identifiers of a business, such as its name and address. Yellow pages provide controlled vocabularies based on industry-related or geography-related taxonomies. The last layer, sometimes called “green pages,” references technical specifications for Web Services, such as pointers to files. UDDI is based on the idea of a centralized registry or registries that can be synchronized.

UDDI specification consists of several documents ([UDDI](#)), but we will only describe, in general terms, the information model to illustrate its complexity of the standard. It consists of four key XML elements: `tModel`, `businessEntity`, `businessService`, and `bindingTemplate`.

businessEntity contains the description of a business. In addition to obvious elements such as name and description of the business, `businessEntity` incorporates supports for taxonomies, including those based on geography, (`categoryBag`), and unique identifiers (`identifierBag`), such as tax IDs, DNB numbers, etc.

businessService and **bindingTemplate** are substructures of `businessEntity`. **businessService** is a descriptive container used to group Web Services according to commonality of business processes or category of service. One or more technical descriptions of Web Services can occupy one `businessService` container. The technical description contains specifications about the Web service (URL, routing options, such as load balancing, additional parameters needed to discover the Web service, such as query terms or values).

bindingTemplate describes additional information necessary to connect to the Web service. In many cases, discovery alone is not sufficient, and a business needs not only to locate a Web Service and connect to it, but also exchange business documents, such as RFQs or POs. `bindingTemplate` contains information allowing the registered participants to carry out transactions at a higher level. In order to exchange business documents, for example, developers can include in `bindingTemplate` the address of the document processing Web Service and a reference to its **tModel**. **tModels** contain metadata about specifications, including attribution information and URLs to locations where the actual specifications are stored. A `tModel` indicates that a Web Service is compatible with certain specifications. The relationship among the elements of the information model is illustrated in Figure 2 below.

Although the specification has tremendous potential for sophisticated integration scenarios, such as seamlessly passing acceptable business documents among business partners, its complexity, illustrated in the diagram, presupposes that a significant manual effort will be required to build and maintain the registry. However, a simpler method to achieve the same result can be possible and is already used in many implementations where full UDDI is substituted with a browsing interface for WSDL files, some with tags that support a certain level of automation. Attempts to simplify UDDI are a testimony that the functionality is useful, but proposed models are too demanding for implementers.

Moreover, tModels have significant conceptual overlaps with XML namespaces. Proposals were circulated to eliminate tModels in favor of XML namespaces. Additionally, many discovery functions in UDDI can be substituted with text search of the businessEntities and a system linked to information about endpoints and bindings.

In fact, if only a few parameters of a Web Service, such as location and protocol information as well as category of service could be defined by the producers and if unique identifiers could be assigned automatically, much of the additional functionality in UDDI could be provided by searching and retrieving descriptive documents associated with the Web Service in question. An agent automatically generating useful information for inclusion in a larger repository available to authorized users could be easily constructed. Extending the registry with an obligatory description is a subject of several proposals and would lead to the availability of richer content.

Although some of the ideas presented here are part of other frameworks, such as Semantic Web, all these frameworks presuppose that an ontology of a domain will be constructed and reasoning features in all applications will be defined first, resulting in some level of semantic interoperability that can enable machine-to-machine communications at a higher level. However, human behavior in discovery, where minimalist, not very precise systems (like Google search) are favored over more complex applications where greater precision requires the use of controlled vocabulary and syntax (such as library catalogs) leads us to believe that greater coverage and flexibility can compensate for lower precision in general purpose registries as it does in other applications focusing on discovery.

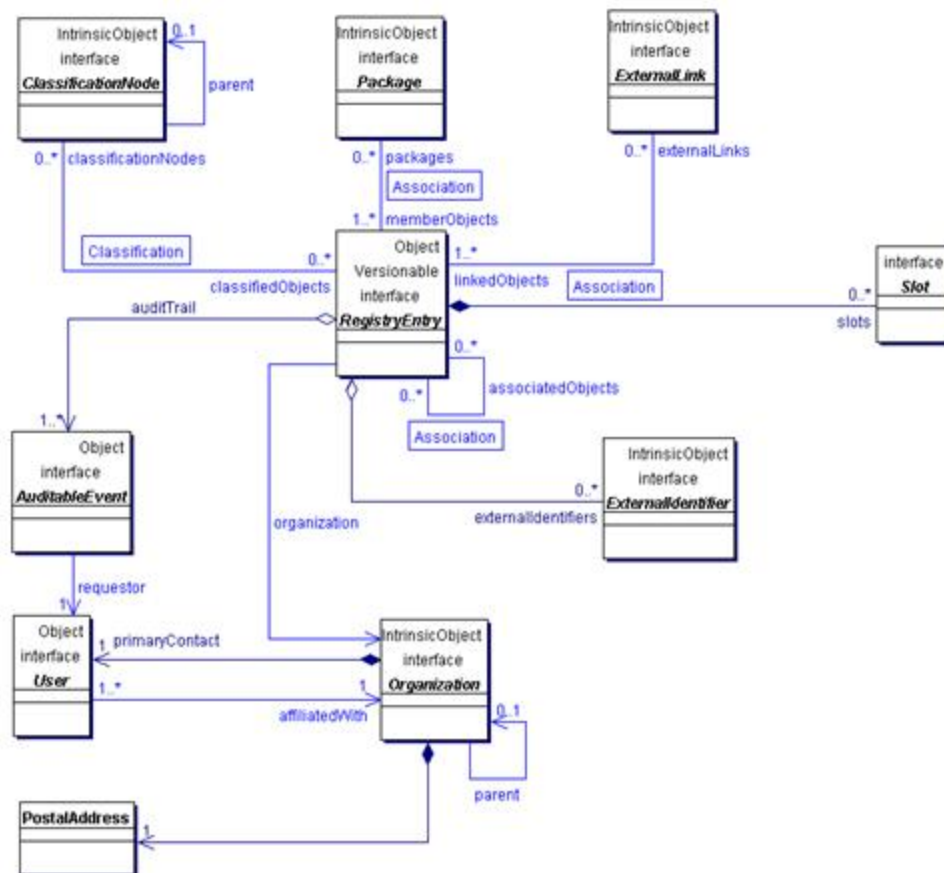


Figure 2. UDDI Components (from UDDI Technical white paper at <http://www.uddi.org/>)

4.2. ebXML RR (Registry/Repository)

As the Internet use for B2B reached some level of maturity, a significant interest in Internet-wide back-end integration has emerged. At the center of related efforts is the use of XML and other open standards as the foundation of B2B integration. ebXML, started as a UN/CEFACT initiative and supported by OASIS, was tasked with defining the new generation B2B. ebXML Registry/Repository (RR) contains registry as well as repository of elements, although the current emphasis of the specification is on protocol-related pointers rather than semantics ([ebXML](#)).

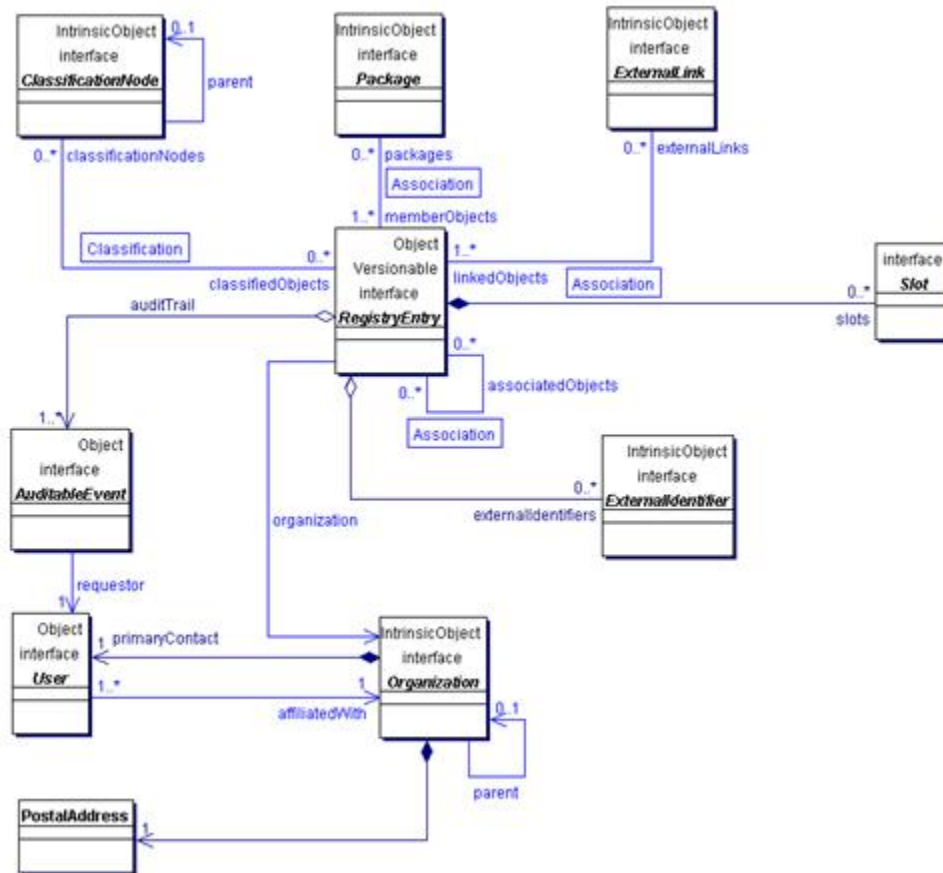


Figure 3. Objects in RIM and their relationships (from RIM specification version 1.00 at <http://www.ebxml.org/specs/ebRIM.doc>)

ebXML RR occupies a central place in the ebXML framework and is the subject of two of the ebXML specification documents. Other specifications, notably CPP/CPA defining trading partner profiles and agreements, are intimately connected with the registry/repository. The ebXML Registry Information Model (RIM) provides a high level schema for the registry, describing registry objects (types of metadata stored in the registry) and relationships among these objects. Figure 3 above outlines the main elements included in RIM.

ebXML registry stores information about items that reside in the repository. Items in the repository are created, updated, or removed through requests made to the registr. The repository can house multiple types of objects. Examples of objects to be stored include:

- CPP/CPA (partner profiles and agreements) templates;

- Business process documents;
- Core components and their aggregates (core components are reusable units that can be combined to form other objects);
- DTDs and schemas.

Even from a considerably simplified illustration of the ebXML RIM (Figure 3), we can see that it is a complex specification. Because of its complexity and positioning, it is used more frequently for governance or content management of underlying applications rather than discovery. ebXML RR can work very well to streamline integration of ebXML frameworks, but, due to its complexity, it is hard to extend the same approach to a general purpose discovery mechanism. Complete automation is not essential for the environments where ebXML RR specification may be used, but it is needed in order to develop a viable general purpose global registry of Web Services.

5. Complex Registries and Search-Based Discovery Intermediaries: Coexistence

The approach adopted by UDDI and ebXML RR is similar in that they design a complex discovery mechanism that is well suited for the manual registration of services or their elements combined with semi-manual nature of retrieval. The specifications seek to address most of the known cases of operations with the greatest possible precision.

There are differences between the specifications as well: UDDI Business Registry (UBR) is designed to provide global coverage, but there is no global ebXML registry; ebXML registries represent “local” business communities. According to the specifications, ebXML registries can be represented in UBR. UDDI centralization-based approach assumes that the participants share the same schemas; ebXML RR is built on the premise that most business registries will be private and may have to use different schemas. Both specifications are applicable to highly structured business environments, but are too complex for general-purpose Web Services discovery frameworks.

However, the existence of the complex and multi-faceted specifications for Web Services registries does not have to be at odds with much simpler mechanisms adopted by a discovery intermediary. The two can be complementary to each other. The first can be used to support specialized well defined framework serving organizations or consortia, while the second can emerge to create a general purpose registry of Web Services descriptions and pointers to specifications. Specialized registries and repositories and general purpose discovery intermediaries can be linked bi-directionally, if needed, with submissions marked for propagation to either discovery intermediary or specialized registry. This approach will support both the more open discovery mechanism that is broad in scope and a more secure and structured private discovery mode adapted for internal organizational uses.

6. Why a Discovery Intermediary?

Many general purpose Web Services applications have been developed and made available for reuse, reference and sharing. Collecting and maintaining an up-to-date registry of Web Services is a daunting task that has to be performed with skill and consistency. Moreover, the trusted aggregation service will benefit from additional efficiencies intermediaries possess, as described in Section 4. When a critical mass of sharable Web Services becomes available in an easy to use and secure Web Services Registry (WSR), submissions will flow, as has happened in the Web environment in the past. The users (including developers and architects) typically adopt the best and most informative free tool.

But a public registry of Web Services will be always held back by the issue of imperfect information, even if the discovery problem is solved in a satisfactory fashion, unless a trusted intermediary is established. Today’s security conscious organizations need a guaranty that a public Web Service they utilize is not going to turn malicious or affect performance of essential systems. The intermediary can take care of the evaluation of the Web Services and provide a guaranty to users that the submission was inspected; this way, an acceptable level of service can be expected and guaranteed.

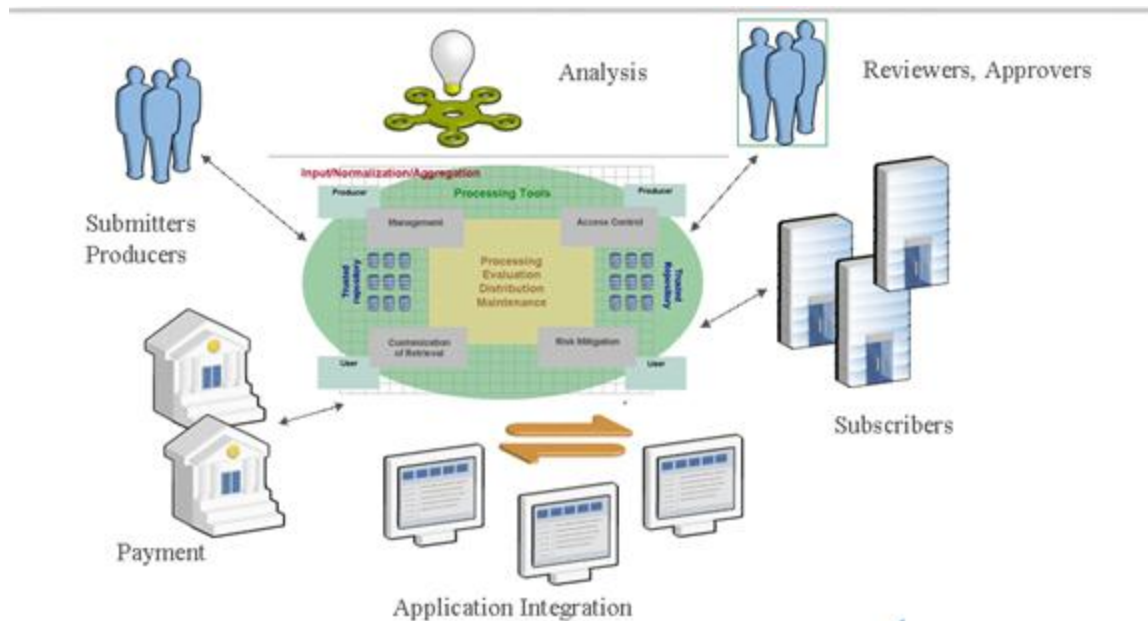


Figure 4. Functions of Intermediaries

In addition to the intermediary itself, inspection can be performed by trusted reviewers, as a public or professional service. Thus, the WSR intermediary will perform the following functions:

1. Hosting of discovery indices;
2. Processing of WS submissions and creation of WS indices, using a modification of a common text indexing engine;
3. User and application registrations;
4. Access control, security, and user/application privacy management;
5. Management of user submissions;
6. Aggregation of materials;
7. Review and quality control of submissions;
8. Compilation of statistics and audit;
9. Advertising the services and regulating the user community.

What organization can take the leading role of a Web Service discovery intermediary? There are many possible choices. Telephone companies have maintained, online and offline, a directory of businesses, and a registry of Web Services seems a logical addition to the existing electronic phone books. Universities can also play a role. But the need to develop general purpose discovery tools (e.g. Web Services discovery oriented search engine) points to a commercial operator that will provide technology and operational support for collection and distribution of Web Services as a public service and derive revenues from adjacent business opportunities.

6.1. Submission, Processing, and Retrieval Process

For a WSR application like the one described here, usability for heterogeneous audiences with different goals is the primary focus. In this context, it is important to remember the crucial value of simplicity. “To be effective, - says P. Drucker, - an innovation has to be simple, and it has to be focused. ... Indeed, the greatest praise an innovation can receive is for people to say, ‘This is obvious!...’” (Drucker, 1998, p. 156).

Consequently, we propose a simple process for registration and submission of Web Services. A submitter registers using the registration interface. Registration is hierarchical, and if the submitter belongs to an organization that is already registered with the discovery intermediary, he/she can be referred by a member with referral rights. Being referred shortens the clearing process for registration and subsequent submission. Organizations can acquire referral rights by being certified, and individuals can acquire such rights by following rules and submitting high quality Web Services.

Accessing applications can also be registered by their owners with sufficient rights, which enable these applications, for example, to switch to alternative Web Services when the one they are ordinarily using is not operational.

Although typical submission is carried out following registration, automated collection of Web Services is also possible for the situations when a trusted relationship can be established between the intermediary responsible for data collection and the sites or organizations where data collection takes place.

After completing the registration, the members receive log-in credentials and can submit their Web Services. The submission form includes the system-assigned ID number and three required fields: name, location identifier, and extensive description. In addition, optional fields, containing anything from supported protocol information to keywords and classification terms can be included, but this information can also be included in the description. These fields can be custom-added by some members with sufficient rights.

After the Web Service is registered, it is submitted for review. Review can be performed by the intermediary or by qualified members as a public or professional service. If the WS passes the review, it is published, and the submitter is notified. Publication involves listing the services in all applicable locations, but mainly adding it to the text search index created by the discovery intermediary. Search is the most common way of retrieving the services. Form-based search, with predefined parameters, is also available.

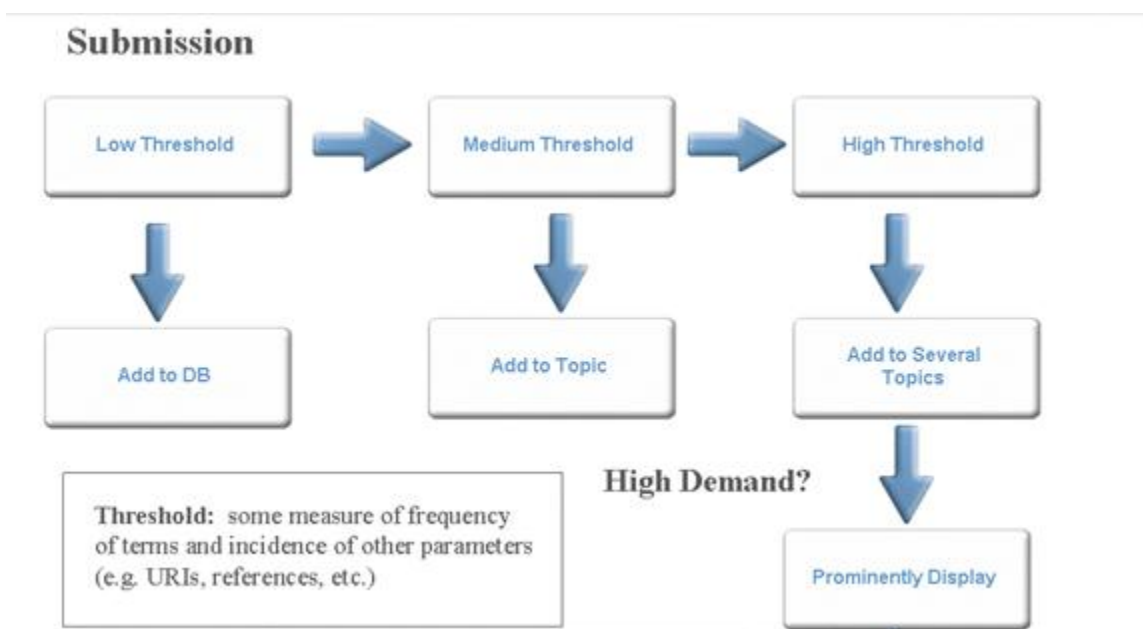


Figure 5. Sample automatic indexing and display process

The new service can be added manually to a group of services by the submitter or automatically linked with a group (e.g. stock tickers) based on the description or other information. Sophisticated automatic cataloging routine based on the statistical analysis of the submitted document and the document collection can be introduced as in the simplified process illustrated in Figure 5 above. The new services are also advertised by the intermediary in the appropriate locations in the service discovery channel. If the service doesn't pass the registration, a notification is sent to the submitter.

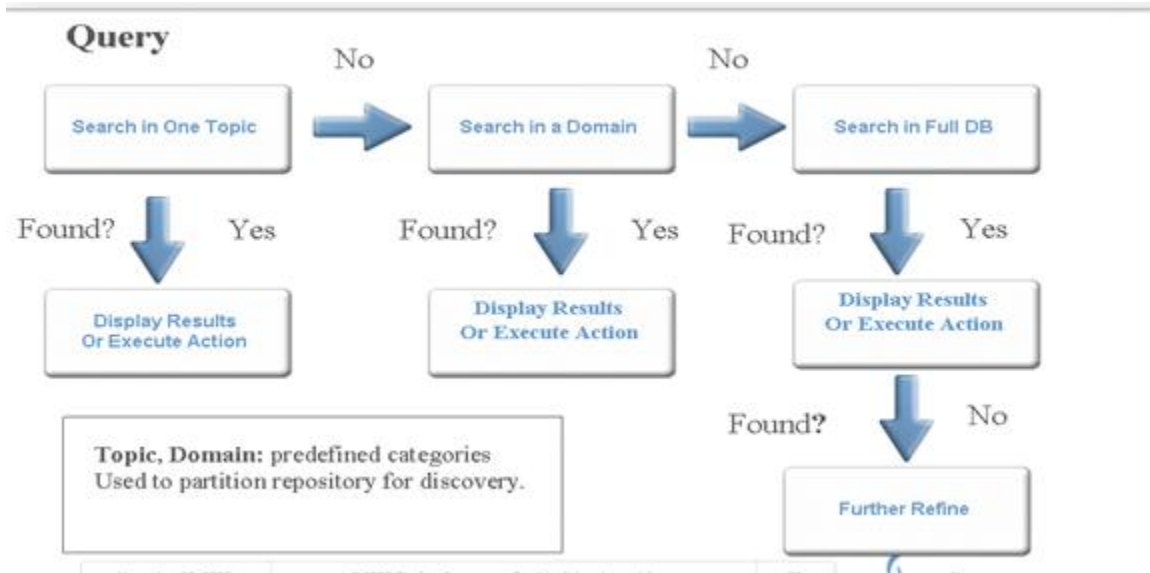


Figure 6. Sample retrieval process

Posted Web Services can be ranked based on their popularity or positive reviews and can be adapted for other uses if such a permission is provided by owners. They can be retrieved in a variety of ways, either through a simple process or using a multi-step retrieval process as illustrated in Figure 6.

Straightforward text indexing, the predominant method of entry processing in the registry described in this paper, doesn't allow to match a structured set of attributes with the entry in the registry. However, Web Services files submitted to the registry, are structured documents, and additional matching, e.g. by proximity or positioning, may allow the developers to approximate the complete matches based on structured attributes. These more precise matches hold a promise in supporting a significant level of discovery automation.

Compared with the complex registry specifications that were briefly described earlier, the method outlined in the paper provides several advantages for a global registry of Web Services:

1. Ease of use and automatic processing;
2. Introduction of a trusted intermediary responsible for maintenance, review, advertising, and access control;
3. Ability to co-exist with the complex discovery mechanisms that are better adapted for specialized environments;
4. Reasonable chance to acquire a critical mass of submissions necessary for common use.

In addition, the proposed approach to the general purpose registry appears to be consistent with the trends used in larger organizations where minimalist automated processing of documents and artifacts and manual creation of ontologies/taxonomies as well as elements of manual categorization coexist to create a more coherent information space. Although the approach in question is applied to textual documents rather than programmatic elements of Web Services, we can consider this trend as guidance in creating a new discovery paradigm for Web Services as well.

7. Conclusions and Future Research

Many good quality highly structured specifications have been created for the purposes of discovery, such as UDDI and ebXML RR. While these standards work well in specialized situations, they are too complex for general purpose applications and cannot support a sufficiently high level of automation.

Since automatic text indexing and search has gradually substituted fielded search based on human cataloging in so many areas, we think this approach is applicable as the main discovery mechanism for registries of services, identities, and credentials (e.g. certificates). Adaptations of the method will be required for different environments, and 20% of the cases may require human intervention, but overall, registries based on text processing are easier to create, maintain, and to use.

Outside of technology, issues of the certification of information and access control require the presence of a trusted intermediary, an approach that was validated in text search and general purpose online directories of various kinds.

Bibliography

[web1] Akerlof, G. A. *The Market for "Lemons": Quality Uncertainty and The Market Mechanism*. Quarterly Journal of Economics, 1973, 488-500.

[art2] Bannerjee, A and Maskin, E. *Fiat Money in the Kitoyaka-Wright Model*. *Quarterly Journal of Economics*, 111 (4), 1996, p. 955-1005.

[art3] Bhattacharya, S. and Hagerty, K. *Dealerships, training externalities, and general equilibrium*, In Prescott, E.C. and Wallace, N. (eds.). *Contractual Arrangements for Intertemporal Trade*. Minnesota Series in Macroeconomics, Minneapolis: University of Minnesota Press, 1989.

[art4] Bose, G. and Pingle, M. *Stores*. , Economic Theory, 6 (1995), p. 251-262.

[art5] Drucker, P. F. *The discipline of innovation*. Harvard Business Review, November-December (1998), 1049-166.

[art6] Jevons, William Stanley. *Money and the mechanism of exchange*. New York : D. Appleton, 1920.

[art7] Leonard, D. *Wellsprings of knowledge*. Boston: Harvard Business School Press, 1995.

[art8] Kashyap, V. and Seth, A. *Semantics-based information brokering: a step towards realizing the infocosm*. Technical Report DCS-TR-307, Rutgers University, Department of Computer Science, March, 1994.

[art9] Ojala, M. *The online searcher*. New York: Neal-Schumann Publishers, 1990.

[art10] Sarkar, M, Butler, B. and Steinfield, C. *Intermediaries and Cybermediaries: A Continuing Role for Mediating Players in the Electronic Marketplace*. JCMC 1(3), 1996. <http://shum.cc.huji.ac.il/jcmc/vol1/issue3/sarkar.html>

[art11] Vishik, C. *Internal Information Brokering and Usage Patterns on Corporate Intranets*. Proceedings of the 1997 International Conference for Group Computing, Nov. 16-19, Phoenix, Arizona, 1997.

[art12] Vishik, C and Whinston, A. *Knowledge Sharing, Quality, and Intermediation*. Proceedings of WACC '99 (February 22-25 1999, San Francisco, California)

[art13] Vishik, C., Farquhar A. and Smith, R. *Enterprise Information Space: User's View, Developer's View, and Market Approach*. In Proceedings of ASIS '99, November 1999, Washington, DC.

[art14] Williamson, S. D. *Recent developments in modeling financial intermediation*. Federal Reserve Bank of Minneapolis, Quarterly Review, 11, Summer, 1987, 19-29.

[art15] *EbXML specifications* at <http://www.ebxml.org/>

[art16] *UDDI specification* at <http://www.uddi.org/>

Biography

Claire **Vishik**

[Sterling Commerce](http://www.sterlingcommerce.com/) [<http://www.sterlingcommerce.com/>]

750 West John Carpenter Freeway

Irving

Texas

United States of America

claire_vishik@stercomm.com

Claire Vishik received her PhD from the University of Texas at Austin. She pursued the study of Internet technologies as a Research Scientist at Schlumberger Laboratory for Computer Science and as a Principal Member of Technical Staff at SBC Laboratories, focusing on B2B, multi-modality, and security. Now member of the Applied Research Group at Sterling Commerce, Claire concentrates on the security issues including regulatory compliance. She authored/co-authored many papers and has 16 pending and issued patents on e-commerce, security, and multi-modal applications.