A United Approach to Discover Multimedia Web Services

Qianhui Liang, Stanley Y.W. Su
Department of Electrical and Computer Engineering
University of Florida
P.O. Box 116200, Gainesville FL
{ qliang, su } @cise.ufl.edu

Haifei Li, Jen-Yao Chung
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, New York 10598
{ haifeili, jychung } @us.ibm.com

Abstract

The Web services technology has been making a steady progress since its initial emergence in the beginning of this century. Since multimedia data have become ubiquitous on the Internet, it is not surprising that multimedia Web services have been receiving much attention by the Web services community. In the Web services platform, UDDI is the current de facto service discovery approach. Yet, researchers have long noticed that the UDDI business model has not really achieved its designated goal. In this paper, we have proposed an approach to complement UDDI with WSIL in the Web services discovery. The idea behind ***U***nified ***W***eb ***S***ervice ***D***iscovery (UWSD) is to use both the brokering-based approach and the trust-based approach in the Web services discovery. Further, UWSD is designed for handling the multimedia service discovery with specific QoS considerations. The services discovered by UWSD are separated into two groups. The first group contains relatively limited number of services that are trustworthy and guaranteed. The second group contains a large number of services, but the content is not guaranteed to be trustworthy. A markup language is also designed to facilitate the discovery process. We believe that the UWSD approach can better meet the current demand of multimedia Web services discovery.

1. Introduction

According to the definition from W3C [1], a web service is "a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A web service supports direct interactions with other software agents using XML based messages exchanged via internet-based protocols." Multimedia web services generally involve transportation of multimedia contents over the Web, and management of composite devices for multimedia contents [2].

1.1. Multimedia Services

Synchronized and interactive access to multimedia content through the Internet has been a major service that the Web provides. Due to the unique features of multimedia data, for example, its large size, researches of Internet multimedia were mainly focused on compressing, caching and streaming of multimedia data. Now, with the emergence of Web services technology and the standard way to represent, deliver and transform data (XML/SOAP), the trend is to integrate multimedia services with the most recent Web service technologies.

A comprehensive infrastructure to support multimedia Web services was proposed in [2]. A SOAP-oriented component-based framework was designed to ease streaming and caching of multimedia data through proxies with both the service providers and the service requestors. Metadata and multimedia content are separated into a number of SOAP messages for the proxies to process. Besides, in order to adapt the multimedia services to different composite devices, composite capability/ preference profiles (CC/PP) [3] are managed by specific managing components in the proxies.

Other related researches include WebSplitter [4], which provides an XML framework for multi-device Web browsing. In WebSplitter, XML metadata policy files are defined to allow different parts of the Web pages to be accessed with different privileges and devices. MyXML [5] is another research effort using XML/XSL technology. An XML/XSL-based template engine was proposed for solving the issue of device independence in multimedia services by completely separating content from format information.

The above researches benefit multimedia service providers and requesters by improving the multimedia service deployment and multimedia service request processing. However, no effort has been made on improving the "service registry", which is responsible for service discovery, to better facilitate multimedia Web service discovery.

1.2. Web Service Discovery

Publishing, discovering and invoking Web services are the key functions that a Web services platform needs to support. Web service providers advertise their services in a public accessible place, for example, a service registry, to bring greater business opportunities. Service requestors search the registry for desired services, and then contact the service providers to invoke the Web services. Web service discovery is one of the central tasks of the Web services platform. Further, to provide a discovery technology that allows easy and precise service discovery by businesses or consumers all over the Internet is among several basic issues the platform designer needs to consider. Discovery methods can range from manual to automatic. [6] has summarized the discovery methods: Discovery is originally done manually through email, Web browsing, phone calls, and even word-of-mouth. Later, improved discovery technologies, which provide XML formats assisting the look-up, emerged. They are less static and less manual. Examples include Microsoft's DISCO [7] and IBM's ADS [8]. Most recently, the use of Web-service brokers and related protocols takes discovery technology to the era of "automation". With the supported specifications, Web services can now be discovered by machines. UDDI is a good example of the latest discovery technologies.

1.3. WSIL, UDDI and UWSD for Web Service Discovery

The WS-Inspection specification provides an XML format for assisting in the inspection of a site for available services and a set of rules for how inspection related information should be made available for consumption [9]. A WS-Inspection document provides a means for aggregating references to pre-existing service description documents which have been authored in any number of formats. These inspection documents are then made available at the point-of-offering for the service as well as through references which may be placed within a content medium such as HTML.

The success of the Web services technologies can be attributed in part to the Universal Description, Discovery and Integration (UDDI) protocol. UDDI creates a standard interoperable platform that enables companies and applications to quickly, easily, and dynamically find and use Web services over the Internet. UDDI also allows operational registries to be maintained for different purposes in different contexts. However, the current UDDI business model still has a couple of challenging problems to solve before it becomes a really feasible discovery solution in the Web services platform.

In this paper, we propose a multimedia Web service discovery approach, called ***U***nified ***W***eb ***S***ervice ***D***iscovery (UWSD), by combining UDDI and WSIL. UWSD is expected to get the best from both UDDI and WSIL, as well as to take into consideration the special features of multimedia Web services. The remainder of the paper is organized as follows: we first present the necessity of combination in Section 2. In Section 3, we discuss the quality issue in multimedia Web services discovery. Then we explain the UWSD discovery mechanism and UWSD system architecture in Section 4. After that, we present the language for UWSD discoveries in Section 5. Section 6 gives a sample scenario of using UWSD to discover a movie preview service. Section 7 concludes this paper.

2. United web service discovery

2.1. Pros and Cons of UDDI and WSIL

The UDDI protocol is designed based on the Server/Client model. UDDI uses a centralized repository to store information about Web services. Its directory service relies on a server (operator) operating on the centralized repository. This centralized approach could have several shortcomings such as "single point of failure" [10]. In order to alleviate the burden of a single server, the UDDI organization has created several mirror sites and replicated registration data across the mirror sites. Currently, there are four UDDI operators: IBM, Microsoft, SAP and NTT-Com. However, how to keep all these data consistent is a resource-consuming task.

UDDI has promised to provide a platform where buyers and sellers of services could easily connect to each other and access information about potential trading partners. On this platform, service providers could easily publish their services. Service consumers could easily discover services and finally invoke these services. Unfortunately, this promise seems to be not fully fulfilled. According to [11], UDDI has two design flaws that prevent itself and the Web services architecture from achieving the expected goals: lack of moderation and inadequate Quality of Service (QoS) guarantees. Without a reliable and effective moderation, the registry could contain out-of-date and even untruthful records concerning the Web services. If the information held by the registry is questionable, scalability and other designated good features of the registry are meaningless. Quality of Service is the other concern. UDDI brings opportunity of new partnership; however, businesses need to establish the service quality agreement before they are willing to buy a service. UDDI does not take this issue into consideration. Neither QoS characteristics nor other SLA (Service Level Agreement), such as security, reliability, availability, were included into the UDDI business model. Businesses thus become very hesitated to use the service based on the information in UDDI [12]. Things have been improved as far as business identity fraudulence is concerned. The latest version of UDDI specification, UDDI Version 3, expands the foundation of UDDI Version 1 and Version 2 and brings desirable security features into the UDDI protocol. Service providers digitally sign the published data during registration, so that each entity in UDDI is now attached with an XML digital signature. Service requestors are allowed to find UDDI entities that are signed. Some level of data integrity and authenticity is delivered by a UDDI 3 registry together with the delivery of the brokering service. Despite of this important security advancement of UDDI, to include some form of central control for keeping the business and service data updated and posting the quality index of the services is still preferred.

WSIL is an aggregation of service descriptions. It allows the service requestor to discover Web services deployed on a Web server through inspection.wsil, the fixed-named top-level WSIL document at the root of a Web site, and all other WSIL files that inspection.wsil links to. These WSIL files are moderated by each business and exist in the business's web site. The "find" requests are processed against each business's own Web site in a decentralized fashion. The single failure point of the UDDI registry is now scattered to each Web server hosting the Web services. Also, when discovering Web services by accessing the business Web sites, a service requestor can trust the information presented in the WSIL. It's similar to your trust of a friend's words on what merchandise his store carries and what quality the merchandise has.

WSIL helps to form a web of services of a service provider in a very structured way. All these Web services could be discovered by traversing the links from the top level Inspection.wsil file. Since the location and convention of this file is well known, the discovery can be performed in a systematic way. However, WSIL only allows discovery of services of known businesses. WSIL, by itself, does not define a Web services repository and a mechanism to interact with such a repository like UDDI does. Web services on different servers are still isolated despite the presence of physical connection through the Internet. As a result, new connections between the providers, consumers and marketplaces cannot be quickly made through a single access point. As such, WSIL discovery is not enough for efficient cross-enterprise integration that the Web services technology is promoting.

2.2. The Combined Approach

We propose a combined approach, called ***U***nified ***W***eb ***S***ervice ***D***iscovery (UWSD), to get the best from both UDDI and WSIL by using two complementary information sources to discover Web services. UWSD makes Web services visible to anyone on the Internet through the UDDI brokering service. Meanwhile, it provides a trust-based searching facility through the WSIL inspection. Searching the UDDI registry is based on a multi-criteria query and is performed on a single or multiple registries. A multi-criteria query is an aggregated query consisting of several individual UDDI queries, each of which may be on names, identifiers and categories. Aggregation operations can be defined for "and," "or," and some other script operators. Searching is thus made more convenient and easier. WSIL, complementary to UDDI, makes up the other half of the UWSD discovery. A WSIL discovery always starts from the entry point of the business's Web site. As far as the discovery is done at the service provider's site, the requestor does not have to worry about the credibility of the services discovered. All information from the Web site is moderated by the business that owns these services thus should be integrate and updated. The inspection documents are well organized to be easily traversed.

3. Quality-enabled service discovery

As illustrated before, one of the challenges that UDDI faces is that it lacks for QoS guarantees. The experience of using multimedia Web applications, especially those interactive and responsive ones are highly quality-dependent. It is necessary that we make some quality considerations during the service discovery process. In this section, we discuss the need for and the value of performing a quality-enabled service discovery. We also introduce two quality related criteria for the service discovery.

3.1. QoS

Consumers continue to increase the range and complexity of the quality provision on multimedia applications. Meanwhile, there have been, and continue to be, efforts to provide different quality of multimedia services upon different application quality provisions. In such a business environment, the match between the quality requirement of the user (or user application) and the quality specification of the services is important in many business activities. Quality of Service, i.e. "QoS" defines a set of quantifiable and non-quantifiable parameters of a network system (an application, a host, a network device etc.) necessary to achieve the level of assurance that its traffic and service requirements can be satisfied. QoS has been discussed a lot by the Internet community.

The Web services platform gives a service-centric view of Web applications to allow easy and homogeneous discovery. Only with quality matching, could a discovered multimedia Web service be truly usable. Therefore, it is desirable that "service quality" checking is included as part of the discovery process. With QoS, Web service providers have been able to offer carefully tailored and finely differentiated services for different customers. If the discovery request of a Web service can also be differentiated by using the QoS, the differentiation of the provided then could be easily mapped to the differentiation of the requested.

In the Web services platform, the requestor may specify the quality requirements in a discovery request for various reasons. Here are two situations of specifying the QoS in the discovery requests. In the first situation, the user signals his/her network quality to the service discovery agent, in our case UWSD, which would search for the Web services whose QoS requirements are not violated. Service users usually know the offerings of their network environment where they are going to use the service. In the UWSD query, he/she may describe his/her network resources through QoS parameters. By a comparison between his/her description and the service QoS, the discovery agent can tell if the service is good for the user. In the second situation, the user has specific performance requirements for using the service. If these requirements are not met, the service is considered not usable under his/her operating environment, although a service of same quality may be still acceptable to most users' tastes. In this case, he/she could request a premium Web service experience by describing the requirements through QoS parameters. The discovery agent would discovery the services that meet his/her superior requirements.

Below, we use the Internet multimedia streaming service as an example to show why QoS information in the description of an offer and demand of multimedia Web services can be helpful. It is well known that Internet performance is largely differentiated. With streaming, this issue is more prominent because streaming media usually contains large quantities of quality-sensitive information. Streaming service receivers connect to the Internet via various connection approaches. Well-connected business users receive stream service through high-bandwidth connections. Their access performance is primary due to the major backbone and ISP peering delays and availabilities. On the other hand, connected through lower-bandwidth, higher-latency lines, such as T-1 lines, DSL connections, cable modem lines and dial-up connections, the stream performance experienced by small business and home users is mainly decided by the bandwidth of the access link, the throughput of the local service provider's connection and the caching capability. For both types of stream consumers, the performance can be formulated as some function of a set of QoS elements. For this reason, streaming content providers and streaming distribution providers need to establish some QoS elements as the basis of a Service Level Agreement (SLA) to ensure a consistent quality, together with a certain measurement scheme to get and analysis the QoS data. For example, Keynote [13] uses the following pertinent factors to calculate the streaming quality: connect time, redirect time, initial buffer time, video frame rate, packet loss rate, bandwidth utilization and etc. For adaptive/intelligent streaming, e.g. Windows Media Services 9 Series [14], the QoS data, such as bit rates and bandwidth are a range of values, within which the media server can work with the client to optimize the experienced quality of the content delivered. From the point of view of service requestors, to find streaming services that fit their specific needs relies on properly specifying the needs in terms of services and their quality constraints. A consumer of the streaming service sometimes has a limited capability in consuming the media data or a constraint on the bandwidth dedicated to this usage. Including a QoS specification in the description of the request helps to avoid deleterious effect on the reception of streaming data due to link overload and to maintain a well-managed network concerning bandwidth distribution and to eliminate harmful contention.

However, QoS is left out of the current UDDI Web service discovery picture. In order to allow real satisfactory on the discovery result, we introduce QoS specification into UWSD. From the OSI perspective, QoS can be defined in a layered specific way [15]. QoS parameters of one layer should be directly mapped to the QoS parameters at the next layer. And end-to-end coordination on QoS parameters of two communicating parties is also necessary. In the UWSD context, we are basically talking about the application layer from a service discoverer's point of view. QoS parameters of the requestor at the application layer would have to be consistently imposed over the Internet to achievement an end-to-end performance level. UWSD permits specification of user requirements of service quality. From the standpoint of service requestors, it ensures that the services discovered are useful and satisfactory. From the standpoint of service providers, it ensures that the users do not violate the services' resource requirements. It is reasonable to include in UWSD four general QoS parameters that are common to several layers: bandwidth, latency, jitter and reliability.

With QoS in UWSD, the user expresses his/her concerns regarding bandwidth, latency, etc. in the service discovery request. These requirements of certain properties of a service are used to look for a service with consistent QoS specification. Below is an example of a UWSD service discovery request with QoS concerns: a user is looking for a videoconferencing Web service to be used for negotiating with an important customer. He/She requires the service is delivered through a network guaranteeing a bandwidth of 128Kbps, with a maximum end-to-end latency no higher than 50msec. (We assume that he is connected to the Internet by a leased line for very fast and qualified network connection.) Another example of a UWSD service discovery request is: someone is looking for a Web service to preview new movies on the Internet over a plain dial-up 28.8kpbs modem.

3.2. File Size

The encoding of multimedia information leads to a big file size. Even after the file being compressed with an efficient compression algorithm, such as MPEG-4, H.263, file size is still the major factor that affects the transmission of multimedia data over the Internet. Problems caused by transmitting big multimedia files undermine the performance of the Web applications in several aspects. One of them is the response time. Here, we refer to the total elapse between the instance when the request of multimedia information is issued and the instance when the entire piece of information is ready to use as response time. Keeping the QoS as a constant, the larger the size of the loaded information, the longer the response time would be. Users may trade in the details of the information he/she is going to receive for a better response time. In other words, the user chooses a file of a smaller size so he gets to load his information faster.

There are other situations that would shed a light to the importance of choosing multimedia files with a proper size. A user may have several applications (channels) to share the same physical connection. He/she wants the most important applications, so-called mission-critical applications, to be guaranteed of the majority of the resources. Examples of mission-critical applications include eBusiness, EPR (Enterprise Resource Planning), voice over IP (voIP), videoconferencing etc. These applications require performance guarantees so that they do not suffer from traffic contention from less critical applications, such as large (secure) FTP file transfers, uploading/downloading digital music files and personal emailing or Internet messaging. A simple solution to the traffic contention between critical and non-critical applications may be minimizing the traffic volume (by minimizing the file size) of the less important applications to allow a commitment of a higher quality for the important one. There are other more sophisticated solutions such as policy-based traffic control. In policy-based traffic control, traffic is grouped into different categories, such as "mission-critical enterprise resource allocation group" and "delay-sensitive streaming group." Different policies are then defined and applied to each group concerning the traffic volume allowed. In case of outsourcing the functions provided by any of these applications, the policies can be used as the SLA for the registry to pick a proper provider for the requestor.

Below is an example service discovery request that have a file-size requirement. A teenager wants to preview movie clips of "The Core" by Paramount Pictures and is looking for a multimedia streaming service for the preview purpose. His Web connection is very slow (through a 28.8kbps modem) and he dose not like to experience a long loading time for a large trailer. Rather, as a smart Internet user, he prefers to preview several small trailers or short clips, shorter than one minute each. In this way, he has a better idea of what the movie is about without incurring a long waiting time. Here is another example: A sales person is videoconferencing with his boss. At the same time, he was also trying to download the catalog from one of their suppliers' Web site for reference in his conversation. Not wanting to impair the conference quality, he selects the supplier that has a small catalog file.

In summary, "QoS" and "File size" matching are included as a part of the WSIL discovery in UWSD. Thus the discovery of a multimedia service with some constraints on "QoS" or "File size" is supported. In our proposed query language for UWSD, there are two corresponding elements: "ContentSize" and "QoS." In Section 5, we describe the language in detail.

4. UWSD Architecture

4.1. UWSD Discovery Approach

UWSD is designed to improve the existing discovery approaches by combing two complementary ones, i.e. the UDDI and the WSIL approach, for a both credible and complete discovery. In Fig.1, we can see the relationship among the UWSD discovery, the UDDI discovery and the WSIL discovery. The user's UWSD discovery request is decomposed into two parts: the brokering-based discovery and the trust-based discovery. The brokering-based discovery relies on a repository that contains information about registered businesses and services. The UDDI protocol defines not only how the information is structured and organized in the registry, but also a set of standard APIs to retrieve the information. In our case, we use USML [16] in the brokering-base discovery. USML (UDDI Search Markup Language) extends the original UDDI by allowing searching multiple UDDI registries with multiple queries, each of which has a different criterion. Trust-based discovery allows business to directly query a known business with which trust of some degree has already been established before. This fits into the current Web services situation because the broker does not provide enough moderation on the registration information it is holding. Trust-based discovery is expected to return meaningful and usable services.

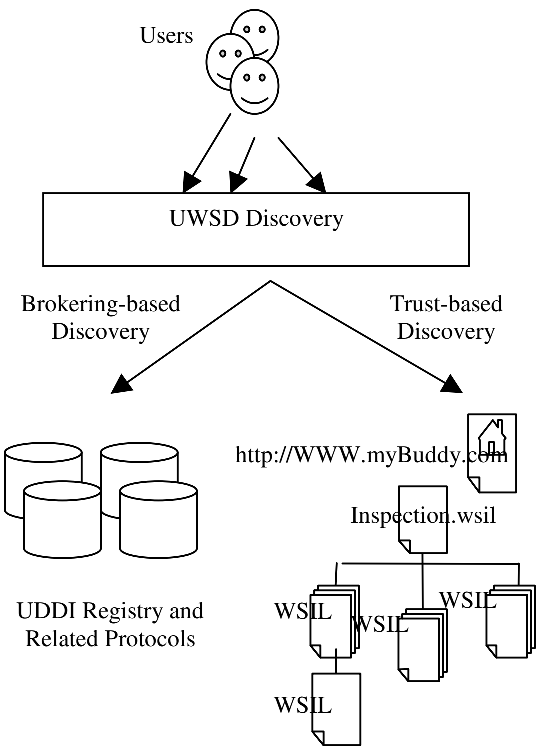

Fig.1. UWSD Web Service discovery

4.2. System Architecture

Fig.2. shows the system architecture of the UWSD discovery operator. We call the operating component ***UWSD Processor***, which is shown as a "U" shape in the figure. UWSD Processor takes the UWSD discovery request and returns a list of discovered services. In this system architecture, ***UWSD Request Interpreter*** is the module that parses the UWSD discovery request and decomposes the query into a USML query and a WSIL search. The USML query is passed to ***USML Operator***. The WSIL search command is passed to ***WSIL Operator*** that searches the WSIL hierarchy only to the prescribed depth. Remember that we have added some quality requirement parameters as additional criteria in the WSIL discovery. As such, the query result from WSIL Operator needs to be processed in order to match the services' QoS with requestor's quality specification. The module of ***Quality Matchmaker*** takes the discovered service information form WSIL Operator and the quality requirements in the UWSD discovery request from UWSD Query Interpreter. It matches the service's quality with the requirement of the user, and only passes the qualified ones. The discovery result from WSIL discovery is relatively credible, while the discovery result through UDDI discovery contains none-moderated information. Therefore, the job of the ***Query Result Aggregator*** module is to tag two sets of results with "trustworthy" and "not guaranteed to be trustworthy" respectively, and to integrate these results. The list of services is returned to the requestor.

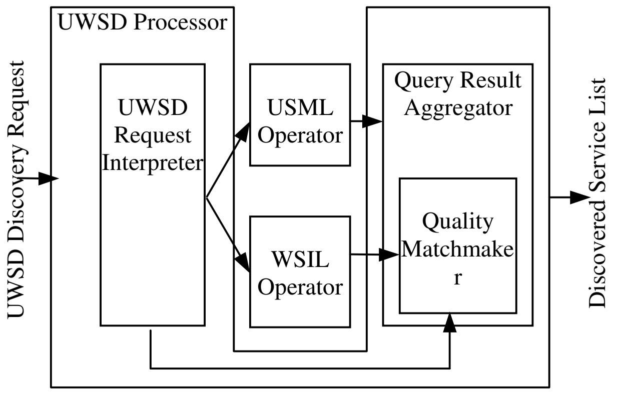

Fig.2. UWSD System Architecture

A user trying to find a multimedia Web service should benefit from this combined service discovery approach. With USML, the query is more flexible, and the result is more complete. With the WSIL discovery extended by the quality matching, not only the result is trustable, but also the requirement of the requestor can be accurately matched to achieve more satisfying results. Furthermore, thanks to the combination of USML and WSIL, both completeness and accuracy are achieved. The requestor is presented with both the services that are safe to be used and the services that could be examined for more business opportunities.

4.3. Implementation

The proposed approach is implemented using IBM Emerging Technology Toolkit 1.1 (ettk1.1) and Apache Tomcat 4.1.24. UWSD Processor includes a parser, which interprets a UWSD service request into a WSIL search command and a USML query, and a query result aggregator, which transforms results from both sources into a canonical form with two dimensions of rating data. One dimension of rating shows how trustable the information is. The other shows how well it is matched to the request. For example, result from WSIL search is 100% trustable and result from UDDI search is 100% matched. WSIL Operator uses wsil4j in the package and USML Operator makes use of BE4WS (Business Explorer 4 Web Services).

5. Language for UWSD

The language for UWSD, referred to as the Language hereafter, is an extension of USML (Unified Search Markup Language) as described in [16]. In addition, the Language also includes constructs for specifying a WSIL discovery command and some quality requirements of the Web service requested. In this section, both parts of the Language, i.e. the USML query and the WSIL discovery specification are discussed.

5.1. WSIL Discovery Specification

The major difference between the Language of UWSD and the USML is the addition of the element called "WSILDiscoverySpec." This element represents a WSIL service discovery based on starting URLs. A starting URL needs to be set to the entry point of the Web server, where the root level WSIL is located. Since the WSIL files in a Web server are organized in a hierarchical manner through the element of "link" with the top-level inspection document at the entry point, WSIL files in a Web server could be retrieved for discovery of services. To trade off between the number of services discovered and the time spent in the discovery, it is a good practice to limit the depth of the link-traversal. This is presented in the element of "DepthofSearch". As we discussed in section 3, both QoS and file size are useful criteria when selecting a service. We define a "QoS" element, with which the service requestor may express his network quality requirement such as the bandwidth, latency, jitter and reliability. "BWU" is used to mark up the upper bound of the bandwidth and "BWL" the lower bound of the bandwidth. We also define a "ContentSize" element for the requestor to limit the size of the multimedia files being transmitted.

WSIL requests are usually issued towards business partners who are known to or even familiar to the service requestor. We could expect that this discovery return credible result. And the services discovered could be marked with a higher preferable rating.

5.2. USML Query in the Language

The results by a UDDI discovery could return a large amount of businesses that might become new partners later on. However, due to the absence of enough moderation, how many of them are really trustworthy is unknown. Detailed inspection needs to be imposed before a decision of selection is made on any of them. These services can be marked "not guaranteed to be trustworthy".

The UDDIQuery element of the Language defines the format of a UDDI query based on an aggregated condition.

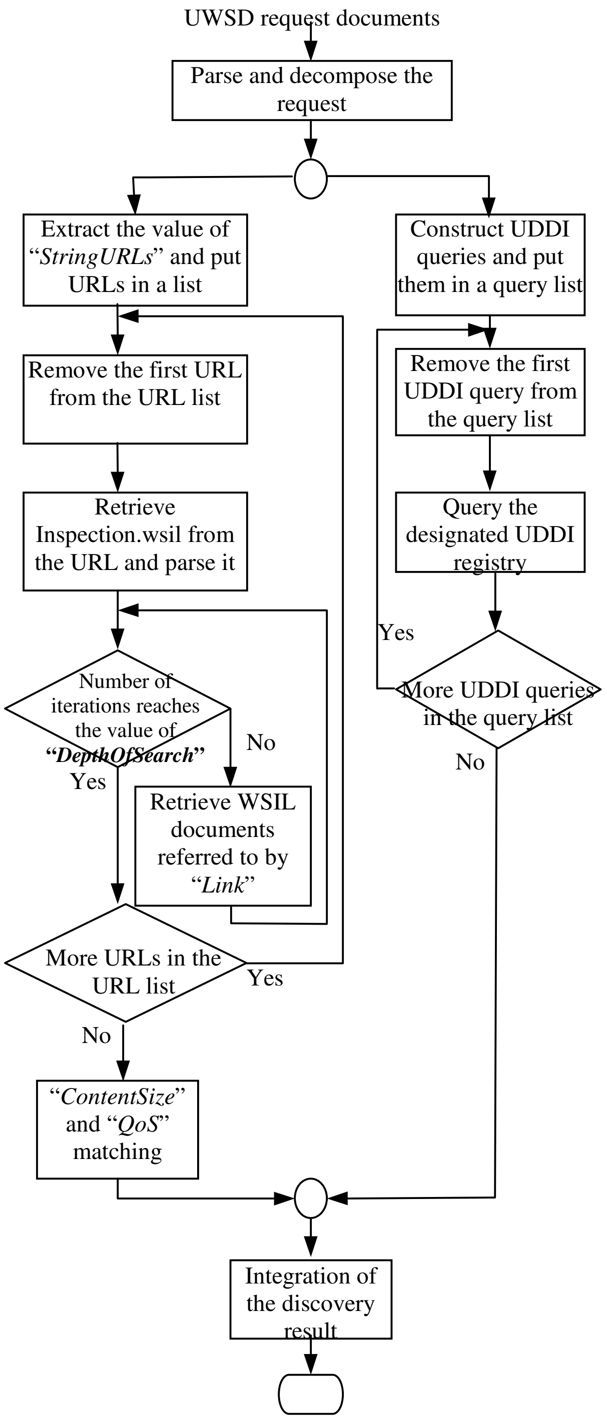
UWSD request documents
Parse and decompose the request
Extract the value of "*StringURLs*" and put URLs in a list
Construct UDDI queries and put them in a query list
Remove the first URL from the URL list
Remove the first UDDI query from the query list
Retrieve Inspection.wsil from the URL and parse it
Query the designated UDDI registry
Yes
Number of iterations reaches the value of "*DepthOfSearch*"
More UDDI queries in the query list
No
Yes
No
Retrieve WSIL documents referred to by "*Link*"
More URLs in the URL list
Yes
No
"*ContentSize*" and "*QoS*" matching
Integration of the discovery result

Fig. 3. UWSD XML document processing

We adopted the constructs defined in USML as the language constructs to support UDDI queries. Below, we briefly review the key concepts and mechanisms of the USML. USML supports searching for three types of entities defined in UDDI, Businesses, Services and Service Types. A business can be searched by a business name, an identifier with its identified name, a category name with its category code, or a URL of the business (usually the home page of the business). A service can be searched by a service name, or a category code. A service type can be searched by a service-type name, or a category code. "Source" is used to mark up the name of UDDI source. "SourceURL" is used to mark up URL of the UDDI registry. "BusinessName" is used to mark up a business name. "Identifier" together with an attribute of "IdentifierType" is used to mark up an identifier. "IdentifierType" can take either "D-U-N-S" or "thomasRegister" as its value. "Category" together with an attribute of "CategoryType" is used to mark up a category code. "CategoryType" can take one of the following four values: "NAICS," "UNSPSC," "GEO," "UDDITYPE" and "SIC." "ServiceTypeName" is used to mark up a service-type name. And "ServiceName" is used to mark up a service name. "FindBy" identifies the data type of the retrieved object. It can take one of the three values: "service", "serviceType" or "business."

5.3. The Language Schema

The schema of UWSD requests is defined as UWSDRequest.xsd (The xsd file is not listed here. However, we show an instance document of this schema in section 6). This schema defines the basic building blocks of the UWSD XML documents. The root element is "Schema". Its attribute of "targetNamespace" indicates that elements defined in this schema come from "http://www.cise.ufl.edu/UWSDSchema" name space. The segment of *xmlns:us="urn:uddi-org:api_v3"* indicates that all elements and data types prefixed with "us" come from the "uddi-org:api_v3" name space, which is UDDI schema version 3. The "UWSDRequest" element is used to mark up a UWSD discovery request. It is composed of four child elements: "UDDIQuery", "WSILDiscoverySpec", "AggOperator" and "RequestTypeName". As we discussed in section 5.1 and 5.2, UDDIQuery is used to specify UDDI queries with specific query criteria on specific UDDI registries, and "WSILDiscoverySpec" is used to specify a WSIL discovery command. "AggOperator" identifies the aggregation among multiple UDDI queries and "RequestTypeName" identifies the type of resources requested. Fig. 3 shows a flow chart of how a request XML document is processed.

6. Sample Scenario

We would like to present a sample scenario of the service discovery using UWSD, the united discovery approach. In this scenario, a teenager plans to go to the theater to see the movie "The core" by Paramount Pictures ®. Before that, he needs to preview this movie on the Internet. He uses UWSD to search for a movie preview service. Below lists the UWSD discovery request in discoveryTheCorePreview.xml:

```
<?xml version="1.0"?>
<UWSDRequest
    xmlns="http://www.cise.ufl.edu/UWSDSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
    xsi:schemaLocation="http://www.cise.ufl.edu/UWSDSch
ema  http://www.cise.ulf.edu/schemas/UWSDRequests.xsd">
<UDDIQuery>
<Query>
    <Source>
        Public UDDI
    </Source>
    <SourceURL>
    https://uddi.ibm.com/ubr/inquiryapi
    </SourceURL>
    <Category CategoryType="UNSPSC">
        <tModelKey>
            uuid:CD153257-086A-4237-B336-
6BDCBDCC6634
        </tModelKey>
        <KeyName>
            Entertainment service
        </KeyName>
        <KeyValue>
            90.15.00.00
        </KeyValue>
    </Category>
    <ServiceName>
        Trailer of  "The core"
    </ServiceName>
    <FindBy>
        Service
    </FindBy>
</Query>
</UDDIQuery>
<WSILDiscoverySpec>
    <StartingURLs>
        http://www.thecoremovie.com/
    </StartingURLs>
    <DepthOfSearch>
        2
    </ DepthOfSearch>
    <ContentSize>
        5
    </ContentSize>
    <QoS>
        <BWU>
            28.8
        </BWU>
    </Qos>
</WSILDiscoverySpec>
</UWSDRequest>
```

On one hand, the requestor directly searches the official Web site of the movie, www.thecoremovie.com, for preview services. He requests that the maximum searching depth is two. He specifies that the size of the trailer is around 5 Megabytes. Since he has a slow dial-up connection and doesn't want to jam his network traffic, he requests the upper bound of the bandwidth is 28.8kbps. On the other hand, he searches the IBM public UDDI registry for a "Trailer of The core" under the UNSPSC category of "Entertainment service". The service from the official movie Web site should be trustworthy. And the services obtained from the UDDI registry would provide him with additional choices.

When this service discovery request document is input to UWSD processor, it is first parsed by UWSD Request Interpreter and is decomposed into a USML query and a WSIL discovery command. Two lists are then constructed: a WSIL URL list and a UDDI query list. The WSIL URL list contains a single URL: "http://www.thecoremovie.com/". The UDDI query list contains a single UDDI query against IBM public registry. WSIL Operator would perform the WSIL discovery on the identified URL of the movie Web site. It retrieves Inspection.wsil from http://www.thecoremovie.com/ and parses this WSIL document. (We are simply using this Web site as an example of WSIL-enabled Web sites. Currently, wsil documents are not provided.) URLs of WSDL documents or other description files regarding the Web services are recorded. Also recorded are file size and Qos associated with each Web service discovered. Further, through the "Link" element in Inspection.wsil, more WSIL documents can be retrieved. Thus, more services and description documents can be discovered. This process is only repeated once, since "DepthOfSearch" is set to 2 in this example. All discovered Web services are passed onto Quality Matchmaker, where "QoS" and "ContentSize" of each service are compared with the requirements in the service discovery request. Not qualified services are filtered out. In this example, there are three of them from Paramount's official THE CORE web site: "large trailer", "medium trailer" and "small trailer." Since the user specifies the file size to be around 5 megabytes, only the "smaller trailer" is selected. At the same time, The UDDI query in the UDDI query list is sent to UDDI Operator. The "FindXXXX" APIs in UDDI are used to search for a business/service_type/service by the business name, the category, the identifier, and etc. Results from both WSIL Operator and UDDI Operator are aggregated by Query Result Aggregator and then sent back to the requestor. The "small trailer" from WSIL Operator is guaranteed to work. The streaming clips and trailers from UDDI Operator would have to be checked by the requestor personally to verify the truthfulness and the quality of the services.

7. Conclusion

In this paper, we have demonstrated the deficiencies of the current UDDI discovery. We argue that a combined approach making use of UDDI and WSIL would get the best out of both UDDI and WSDL. We have proposed in this paper Universal Web Service Discovery for Web service discovery. With UWSD, a service discovery request is decomposed into a brokering-based discovery and a trust-based discovery. The trust-based discovery is guaranteed to produce a result that is trustworthy and that matches quality requirement well. Queries against the UDDI registries provide new business opportunities. But further investigation on the result is recommended. USML is used in the combined approach for a more flexible UDDI search. WSIL discovery is also enhanced to better cater multimedia Web service discovery.

8. References

[1] Web Services Architecture working group, "Web Services Architecture, W3C Working Draft 14 November 2002", http://www.w3.org/TR/2002/WD-ws-arch-20021114.

[2] Jia Zhang, Jen-Yao Chung, "A SOAP-Oriented Component-Based Framework Supporting Device-Independent Multimedia Web Services, IEEE-MSE 2002, Dec. 2002.

[3] W3C Composite Capability/ Preference Profile, Http://www.w3c.org/TR/CCPP-struct-vocab/, W3C Working Draft, March 15, 2001.

[4] R. Han, V. Perret, and M. Naghshineh, "WebSplitter: a Unified XML Framework for Multi-Device Collaborative Web Browsing", Proceedings of the ACM 2000 Conference on Computer Supported Cooperative Work, 2000, Philadelphia, Pennsylvania, USA, pp. 221-230.

[5] "Web Engineering Device Independent Web Services", Proceedings of the 23rd International Conference on Software Engineering, 2001, Toronto, Ontario, Canada, pp. 795-796.

[6] Baljit Singh Chandhoke, Brent Heetland, Herb Turner, Ed Waitkaitis, "Will UDDI Succeed as the Web Service Description and Discovery Standard?", http://198.11.21.25 /capstoneTest/Students/Papers/docs/UDDIproceedings311231.pdf, 2003.

[7] Aaron Skonnard, "Publishing and Discovering Web services with DISCO and UDDI", http://msdn.microsoft.com/ msdnmag/issues/02/02/xml/default.aspx, Feb, 2003.

[8] W. Nagy, F. Curbera, S. Weerawaranna, "The Advertisement and Discovery of Services (ADS) protocol for Web services, http://www-106.ibm.com/developerworks/ webservices/library/ws-ads.html, Oct. 2000.

[9] WSIL, Web Services Inspection Language, available at http://www-106.ibm.com/developerworks/webservices/library/ ws-wsilspec.html

[10] Sitthichai Laoveerakul, Kittinarong Laongwaree, Sissades Tongsima, "Decentralized UDDI based on P2P," available at http://www.hpcc.nectec.or.th/C4/grid/UDDI.pdf, 2003.

[11] Tarak Modi, "WSIL: Do we need another Web Services Specification? Explaining the difference between UDDI," available at http://www.webservicesarchitect.com/content/articles/modi01.asp, 2003.

[12] SalCentral, "UDDI Weather report," available at http://www.salcentral.com/uddi/default.asp, 2003.

[13] http://www.keynote.com/solutions/solutions_pm_ streaming_perspective_tpl.html

[14] Microsoft Windows Media Technology http://www.microsoft.com/catalog/display.asp?subid=22&site=816;h ttp://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwmt/html/intelligent_streaming__qwes.asp

[15] Quality of Service: Motivations, Requirements, Architecture and Services, http://www.comp.leeds.ac.uk/si32/ lectures/si32.3.2.pdf

[16] Liang-Jie Zhang, Haifei Li, Henry Chang, Tian Chao, "XML-based Advanced UDDI Search Mechanism for B2B Integration," 4th International Workshop on Advanced Issues of E Commerce and Web-based Information Systems (WECWIS), 2002.