Exploring JXTASearch for P2P Learning Resource Discovery

Changtao Qu
Knowledge Based Systems and
Learning Lab Lower Saxony
University of Hannover
Appelstr. 4, D-30167, Hannover, Germany
qu@kbs.uni-hannover.de

Wolfgang Nejdl
Knowledge Based Systems and
Learning Lab Lower Saxony
University of Hannover
Appelstr. 4, D-30167, Hannover, Germany
nejdl@kbs.uni-hannover.de

ABSTRACT

In this paper we discuss a Peer-to-Peer (P2P) application aimed to improve the discoverability of learning resources distributed typically over different institutions. We investigate JXTASearch, a distributed search engine constructed upon Sun's open source P2P platform: JXTA, and extend it to enable Dublin Core (DC) meta-data based P2P searching. While the JXTA platform provides the essential interoperability among various peers at "network" level, DC meta-data constitutes the minimal interoperable basis for learning resource discovery at "meta-data" level. We then discuss the advantages and shortcomings of our prototype implementation, as well as future work building upon our experiences with JXTASearch prototype.

Keywords: Peer-to-Peer, Learning Resource, JXTASearch, Dublin Core Meta-data

1. INTRODUCTION

Central-server based approaches to learning resource sharing have received lots of attention in recent years [5][16][19]. However, because many institutions are reluctant to hand their own resources to centralized "E-Learning portals" due to the worry about losing their control over these materials, the central-server based approaches have run into increasing predicaments. This unpleasant situation motivates the application of a new computing style: Peer-to-Peer (P2P) in E-Learning area.

For this application area, the key to the sharing of learning resources is their discoverability in a distributed environment. In comparison with central-server based approaches, P2P networks provide some novel techniques for resource discovery implemented in a number of new, distributed P2P search engines [8][9], one of which is JXTASearch [20], a distributed P2P search engine constructed upon Sun's open source P2P platform: JXTA.

JXTASearch is originally designed to realize both "wide" (the ability to access various devices on the Web) and "deep" (the ability to access various backend repositories) searches based on certain meta-data sets [20]. It is an interesting approach also for learning resource discovery mainly because of its two capabilities:

First, widespread interoperability among various peers.

As one of the core services of JXTA P2P platform, JXTASearch directly benefits from JXTA's interoperability among various peers. Regarding provider peers, JXTASearch is able to not only work well with pure JXTA provider peers within the JXTA network, but also can bring Web-based provider peers into the searching scope without great overhead. This capability facilitates the shaping of a content provider scenario for E-Learning, where various learning resource repositories co-exist. Additionally, concerning consumer peers, JXTA can integrate pure JXTA peers, Web peers, and some low-end distributed devices, e.g., PDAs, handhelds, and cell phones, etc., into JXTASearch network. In general, at "network" level, the JXTA P2P platform basically ensures a widespread searching scope and potential large user community of JXTASearch.

Second, good adaptability to various meta-data sets.

In P2P networks, using meta-data is almost the only way for a peer to expose its own resources or discover and retrieve shared resources from other peers. While a simple meta-data definition is already sufficient for some domain-specific P2P search engines, e.g., Freenet [8] and Gnutella [9], discovering learning resources in P2P environments requires more complex meta-data support, having to involve in most cases several advanced learning resource meta-data sets, e.g., IEEE LOM (Learning Object Meta-data)[11], IMS Learning Resource Meta-data Specification [12], and SCORM (Sharable Content Object Reference Model)[1]. At "meta-data" level, JXTASearch outmatches most of other P2P search engines through introducing a novel searching concept: the queryspace, together with an open design of its searching algorithm. Utilizing queryspaces, JXTASearch can be extended to adapt to various meta-data sets. Also the channel-like searching approach enabled by queryspaces can greatly improve the searching efficiency in complicated P2P environments.

As the current JXTASearch implementation realizes only one simple and fixed queryspace (though the general design of JXTASearch is quite flexible), we've modified it by implementing a Dublin Core (DC) [6] meta-data queryspace as the first attempt to extend the searching capability of JXTASearch. Since the DC meta-data set, especially its RDF (Resource Description Framework)/XML (eXtensible Markup Language) binding [2], is the minimal interoperable basis of several popular learning resource meta-data sets, such as IEEE LOM, IMS, and SCORM,

JXTASearch plus DC provide us with a very useful rapid prototyping environment, which can benefit our exploration into some more advanced P2P applications such as the RDF-based Edutella network [7][14][15].

2. GENERAL DESIGN

2.1 JXTA and JXTASearch

JXTA is Sun's P2P initiative focused on constructing an interoperable P2P network [10]. In general, the JXTA platform is divided into three layers: Core Layer, Service Layer, and Application Layer. From its basis, JXTA is nothing more than six XML-based protocols [17] defined at a much lower level in comparison to some other P2P platforms such as Gnutella and Freenet, which can provide essential interoperability among peers. These six protocols are implemented at the JXTA Core Layer in the form of several building-blocks that can be used by almost all P2P applications regardless of their intended users, platforms, devices, and specific implementations. Upon the JXTA Core Layer we have the Service Layer and the Application Layer, where some higher-level interoperable services and applications can be built. In figure 1 we illustrate four types of JXTASearch services and three types of JXTASearch applications, including their corresponding locations on layered JXTA platform.

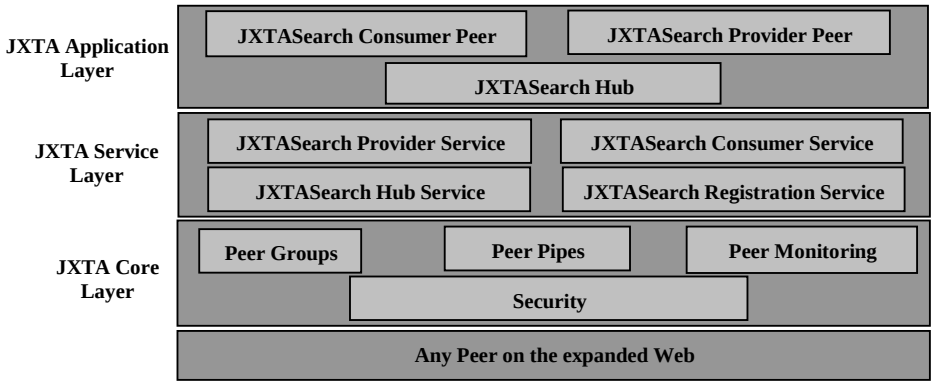


Figure 1. JXTASearch and its location on layered JXTA platform

The kernel of JXTASearch services is QRP (the Query Routing Protocol)[13] which fully specifies the message types, message formats, and message routing rules that must be supported by any JXTASearch peer. Based on QRP, JXTASearch defines four types of services [20]:

JXTASearch Provider Service: a service which accepts queries written in the QRP from JXTASearch hub (or JXTASearch consumer peer directly) and responds in the QRP to the requestor.

JXTASearch Consumer Service: a service which sends queries written in the QRP to the hub (or to JXTASearch provider peer directly) and awaits QRP responses.

JXTASearch Registration Service: a service which sends requests for registration to JXTASearch hub and maintains the registration file for the provider.

JXTASearch Hub Service, a service which performs routing of queries from consumers to providers. The Hub Service consists of two sub-components: (1) JXTASearch Router, which routes and manages query connections, collates results and returns results to consumers; and (2) JXTASearch Resolver, which matches queries to providers using a full text search engine that indexes meta-data specified by the provider during registration.

At JXTA application layer there are three types of JXTASearch peers: JXTASearch Provider, Consumer, and Hub. Because QRP has two transport bindings respectively for JXTA platform (XML over JXTA using either HTTP or TCP) and the Web (XML over HTTP), JXTASearch Provider can be further divided into two sub-types, either a pure JXTA peer or a Web server with the JXTASearch adapter. Likewise, a JXTASearch Consumer has also two sub-types, either a pure JXTA peer or a Web browser with HTTP client interface to the JXTASearch network. In figure 2 we illustrate the general architecture of JXTASearch, which describes the interaction between JXTASearch peers and different JXTASearch services.

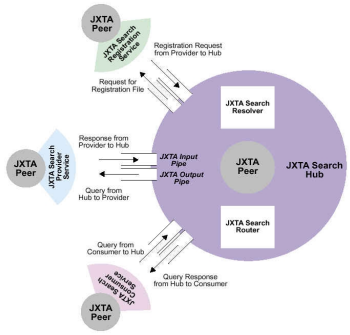


Figure 2. General architecture of JXTASearch [20]

In a JXTASearch network, each JXTA peer can run instances of the Provider, Consumer, and Registration service on top of its JXTA core. Each peer interacts with the JXTASearch Hub service, which itself also runs on top of the JXTA core. In section 3 we'll further describe interactions between different peers together with an introduction to our prototype implementation.

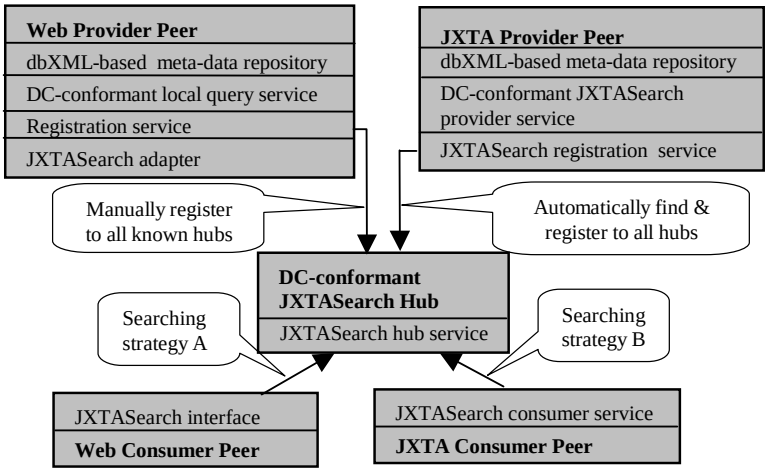
2.2 Dublin Core Meta-data Set and its RDF/XML Binding

The DC meta-data standard is a simple yet effective meta-data set for describing a wide range of networked resources. Since the original JXTASearch only defines one queryspace (http://search.jxta.org/text) which enables a full-text-like distributed searching, extending JXTASearch with DC meta-data can refine searching scope so as to improve searching efficiency. Because JXTASearch adopts an open design style, the core of JXTASearch, e.g., query resolving and routing algorithm, etc., did not need to be touched during the extension process. We only need to modify part of the JXTASearch source code in order to adapt it to DC meta-data schema and then construct a new DC queryspace (http://www.learninglab.de/DC) that defines DC-conformant query structure and its associated registration. Within the JXTASearch network, various queryspaces can co-exist.

In the DC queryspace, the DC meta-data set is represented using its RDF/XML binding [2]. An important consideration here is that all three popular learning resource meta-data sets: IEEE LOM, IMS, and SCORM can be mapped to DC meta-data, and also their RDF bindings are defined in a consistent fashion with DC's RDF/XML binding. As a matter of fact, when the meta-data sets become as complex as above three learning resource meta-data sets (each has over 60 meta-data entries), RDF is almost the only feasible way to format distributed queries and query results [14][15]. From this viewpoint, at "meta-data" level, DC meta-data and its RDF/XML binding provides an ideal starting point for the implementation of more advanced P2P applications in E-Learning area.

3. PROTOTYPE IMPLEMENTATION

In figure 3 we illustrate the system architecture of our prototype implementation.



Searching strategy A: For a Web consumer peer, hubs' addresses must be known before searching. Searching must employ hubs as mediators. After discovering, results returned from both Web provider peers and JXTA provider peers must use hubs as mediators in order to reach Web consumer peer.

Searching strategy B: For a JXTA consumer peer, JXTASearch registration service can automatically find all available hubs and all available JXTA provider peers. Searching can employ hubs as mediators. In this case the working scenario is the same as searching strategy A. Searching can also directly point to any available JXTA provider peer. In this case Web provider peers are not in the searching scope. After discovering, results returned from JXTA provider peers can directly reach JXTA consumer peer without the need of using hubs as mediators.

Figure 3. System architecture of the prototype implementation

3.1 Constructing Provider Peers

Since JXTASearch is basically a meta-data based search engine, each JXTASearch provider peer must possess one or several meta-data repositories which are used to store meta-data of learning resources. While it does not matter how peers construct their meta-data repositories, using either relational databases or object-oriented databases, provider peers registering to the DC queryspace must be able to understand DC meta-data schema in order to provide DC-conformant local query services. For Web provider peers, local query service is usually implemented as a so-called JXTASearch adapter using a Java Servlet, while for JXTA provider peers, the local query service can be directly integrated into the JXTASearch Provider Service utilizing pipes, the natural communication mechanism defined by JXTA. In our prototype implementation, we employ an open source native XML database: dbXML [4] to construct meta-data repositories hosted by both Web and JXTA provider peers. The local query against dbXML is realized utilizing W3C XPath [3]. Since the source code of the JXTASearch adapter and JXTASearch Provider Service interface are nicely encapsulated in an object-oriented style, it is relatively easy to integrate various local query services corresponding to different types of repositories into JXTASearch network.

3.2 Registering Provider Peers

Web provider peers and JXTA provider peers adopt different method to register to JXTASearch hubs. Since Web provider peers cannot achieve natural, pipe-based communication with other JXTA peers (including hubs), they must know hubs' addresses before registration. In their registration file, they must also inform hubs about their local query service's URL in order to get these local query services to be called by hubs. Additionally, Web provider peers' registration is a one-way, "push" process, only from provider peers to hubs.

JXTA provider peers can automatically find all available hubs and then finish the registration process through "push"

or “pull”. Because JXTA provider peers can naturally interact with all JXTA peers utilizing pipes, they need only declare their IDs while registering. Both types of provider peers must declare their intended queryspaces and what sort of queries they wish to respond within each queryspace in their registration file. In figure 4 we show a simplified registration file of a JXTA provider peer. It declares its discovering capability within the DC queryspace.

```
<?xml version='1.0'?>
<register xmlns="http://search.jxta.org"
xmlns:dc="http://purl.org/dc/elements/1.1/">
<title>JXTA Provider Peer: dbXML</title>
<link>http://www.learninglab.de</link>
<query-server>
jxta://59616261646162614A757874614D50476AEB85094D5B4
AD89D969442DDCF4E55000000000000000000000000000000
00000000000000000000000000000000401</query-server>
<query-space uri='http://www.learninglab.de/DC'>
 <predicate>  <or>
  <query><dc:title>Java applet swing etc.</dc:title></query>
  <query><dc:creator/></query>
  <query><dc:subject>API applet etc.</dc:subject></query>
  <query><dc:description/></query>
  <query><dc:publisher/></query>
  <query><dc:contributor/></query>
  <query><dc:date/></query>
  <query><dc:type> Dataset Event Image etc.</dc:type></query>
  <query><dc:format>text/html etc.</dc:format></query>
  <query><dc:identifier/></query>
  <query><dc:source/></query>
  <query><dc:language>de en</dc:language></query>
  <query><dc:relation/></query>
  <query><dc:coverage/></query>
  <query><dc:rights/></query>
 </or> </predicate>
</query-space>
 </register>
```

Figure 4. Registration file of a JXTA provider peer

3.3 Searching and Discovering

Similar to JXTASearch provider peers, there are two types of consumer peers in the JXTASearch network: either a pure JXTA peer or a Web peer (usually web browser) with JXTASearch interface. The essential difference between the both is that JXTA consumer peers can directly send queries to any provider peer (including JXTASearch hubs) and get direct response from these peers, while Web consumer peers must use JXTASearch hubs as mediators in order to accomplish searching and discovering. In our prototype implementation, we simply use an extended JXTA Shell [18] as JXTA consumer peer, and use a JSP (Java Server Pages) running on JXTASearch hubs as the JXTASearch interface for Web consumer peers. In figure 5 we illustrate the query interface for a Web consumer peer. Nine commonly used meta-data entries are selected from DC’s 15 entry definitions in order to format a query. The user can also define the logical relationships between each meta-data query entry. Note that these logical relationships only make sense for local query services, they do not influence the query resolving and routing process.

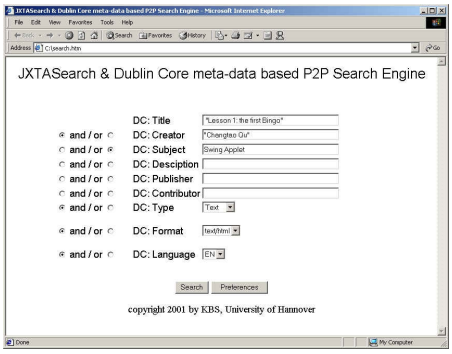


Figure 5. Query interface (Web consumer peers)

After discovering, the query results, which are formatted in XML syntax according to QRP, can be rendered by consumer peers in different styles. For Web consumer peers, JXTASearch hub (precisely speaking, HTTP Router) is responsible for rendering the results in a table-like style and sending them back to the consumers. For JXTA consumer peers, the JXTASearch Consumer Service running on each peer is responsible for presenting the query results. In our prototype implementation, JXTA Shell simply renders the query results into formatted texts.

3.4 Resolving and Routing

Query resolving is the process of determining to which set of providers a given query should be routed. The minimal condition for matching a query to a provider peer in the JXTASearch network is that the query must have the same queryspace as the provider registration. In JXTASearch, query resolving is implemented by a Java Servlet (JXTASearch Resolver) running on the hubs. It can be directly called by JXTASearch Routers.

In general, the JXTASearch Router sequentially performs three functions: (1) receiving queries from consumer peers; (2) calling Resolver to find appropriate providers and further routing the queries to the providers; and (3) merging query results and sending them back to consumer peers. Corresponding to JXTASearch’s two transport bindings, there are two types of JXTASearch Routers: (1) JXTA Router, used to handle queries from JXTA consumer peers utilizing pipes; and (2) HTTP Router, used to handle queries from Web consumer peers utilizing HTTP. Because the query resolving and routing algorithm as well as their implementations in JXTASearch are shielded from end-user applications, they did not need to be touched during our prototype implementation.

4. DISCUSSION

In comparison to conventional crawler-based search engines, distributed P2P search engines such as JXTASearch have many notable advantages, e.g., the ability to provide ad hoc query results, the ability to cross firewalls and heterogeneous backend repositories, as well as high searching efficiency enabled by some P2P-specific designs,

such as JXTA's "peer group" concept and queryspace proposed by JXTASearch. Together with powerful meta-data support, these P2P search engines may essentially change the way information is shared, which makes special sense for resource sharing in E-Learning area.

Our early exploration of JXTASearch manifested also several shortcomings of JXTASearch, one of which lies in the QRP, which defines simple query, response, and registration formats using XML syntax. For more complex queries, the current QRP is insufficient and would have to be extended along the lines of current XML query languages, which unfortunately will have ramifications for the JXTASearch's current searching, resolving, and routing algorithms.

Another shortcoming of JXTASearch lies in its missing of multi-hub chaining definition. Although JXTASearch does contain mechanisms for supporting multiple hubs, the JXTASearch protocol does not define methods for chaining hubs. In fact, coordination among a number of hubs enabled by sophisticated mechanisms is one key to ensuring the wide-spread searching scope and high searching efficiency of JXTASearch. This point has to be addressed by further JXTASearch protocol and hub implementations.

During the development process, we have also suffered from another "shortcoming" of JXTASearch. Because the JXTA platform itself is currently not yet finalized, its evolution has heavily influenced JXTASearch, which has to continuously experience change in order to keep pace with JXTA core. Currently several important aspects of JXTASearch (also our prototype), such as trust and security are still open to implementation. We expect that these functionalities will be present in the near future with the maturity of JXTA core and JXTA services.

5. CONCLUSIONS AND FUTURE WORK

Using JXTASearch and Dublin Core meta-data set, our prototype implementation explored the possibility of building a P2P framework which can be applied for improving learning resource discoverability in E-Learning area. Based on the work described in this paper, we are now working on a more advanced P2P framework: Edutella, a RDF-based P2P infrastructure for the exchange of educational media in the global area [7][14][15]. Edutella search extends our JXTASearch prototype by introducing the design of more general hubs and especially a more powerful query language (a Datalog-based language transmitted in RDF/XML syntax), which can express not only simple conjunctive queries as defined by JXTASearch QRP, but also relationally complete queries based on RDF namespaces to query arbitrary schemas over the net. As an official JXTA service, Edutella will co-exist with JXTASearch on the future JXTA platform, interfacing with JXTASearch using an Edutella/JXTASearch gateway which we are currently working on.

6. REFERENCES

[1] ADL Technical Team, SCORM Specification V1.2, http://www.adlnet.org/Scorm/scorm.cfm

[2] Beckett, D., E. Miller, and D. Brickley, Expressing Simple Dublin Core in RDF/XML, http://dublincore.org/documents/2001/09/20/dcmes-xml/

[3] Clark, J., and S. DeRose, XML Path Language (XPath), http://www.w3.org/TR/xpath

[4] dbXML Group, dbXML Native Database, http://www.dbXML.org/

[5] Dhraief, H., W. Nejdl, and B. Wolf, Open Learning Repositories and Metadata Modeling, in Proc. of Int. Semantic Web Working Symposium, Stanford, USA, Aug. 2001.

[6] Dublin Core Meta-data Initiative, Dublin Core Metadata Element Set, V 1.1, http://dublincore.org/documents/1999/07/02/dces/

[7] Edutella, http://edutella.jxta.org

[8] Freenet, http://freenet.sourceforge.net/

[9] Gnutella, http://gnutella.wego.com/

[10] Gong, L., Project JXTA: A Technology Overview, http://www.jxta.org/project/www/docs/TechOverview.pdf

[11] IEEE Learning Technology Standards Committee, IEEE LOM working draft 6.1, http://ltsc.ieee.org/wg12/index.html

[12] IMS Global Learning Consortium, Inc., IMS Learning Resource Meta-data Spec. V1.2.1, http://www.imsproject.org/metadata/index.html

[13] JXTASearch Engineering Team, JXTA Search Protocol Spec. 1.0 (DRAFT), http://search.jxta.org/protocol.html

[14] Nejdl, W., B. Wolf, S. Staab, and J. Tane, EDUTELLA: Searching and Annotating Resources within an RDF-based P2P Network, Technical Report, Dec. 2001.

[15] Nejdl, W., B. Wolf, Ch. Qu, S. Decker, M. Stintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch, Edutella: A P2P Networking Infrastructure Based on RDF, Technical Report, Nov. 2001, http://edutella.jxta.org/reports/edutella-whitepaper.pdf

[16] Open Archives Initiative, http://www.openarchives.org/

[17] Sun Microsystems Inc., JXTA v1.0 Protocols Spec., http://spec.jxta.org/v1.0/docbook/JXTAProtocols.html

[18] Sun Microsystems Inc., Project JXTA: Technical Shell Overview, http://www.jxta.org/project/www/docs/TechShellOverview.pdf

[19] Teachware On Demand, http://www.teachware-on-demand.de/index.php

[20] Waterhouse, S., JXTASearch: Distributed Search for Distributed Networks, http://search.jxta.org/JXTAsearch.pdf