U-P2P: A Peer-to-peer System for Description and Discovery of Resource-sharing Communities

Aloke MUKHERJEE, Babak ESFANDIARI, and Neal ARTHORNE

Abstract-- A simple method is proposed for peer-to-peer description and discovery of resource-sharing communities as well as the resources themselves. An XML Schema document describes a shared resource. By applying transformations, specified in XSL, the schema is used to generate an application with the ability to publish and search for the defined object. Meta-data is indexed and searchable allowing complex objects to be discovered. We propose to solve the problem of discovery for resource-sharing communities by treating a community as a shared resource. An XML Schema description of resource-sharing communities is used to generate a P2P system for community discovery.

***Index Terms*—peer-to-peer, resource description, resource discovery, XML Schema**

I. Introduction

THIS paper proposes a solution to the problem of resource description and discovery in peer-to-peer file-sharing communities. There are a multitude of file-sharing communities extant and they provide access to an exponentially increasing amount of data. Expanding the benefits of this technology beyond its current applications requires a simple and standard way to describe and discover file-sharing communities.

U-P2P, the framework we have developed, addresses the following problems:

1) The difficulty of creating file-sharing applications devoted to a specific class of objects

2) The lack of a standard way to discover the existence of file-sharing communities

Peer-to-peer file-sharing is a robust, bandwidth efficient means of distributing information. Simplifying the creation of file-sharing communities will benefit those with domain knowledge who do not possess the expertise to design their own applications. This allows applications to be tailored to smaller real-world communities and enables the sharing of virtually any object - for example:

- XML descriptions of chemical molecules for chemists or chemistry students
- descriptions of species for scientists studying biodiversity
- descriptions of genes for researchers studying the human genome
- design patterns for computer science students
- software components for application developers

The focus of existing communities can be narrowed by specifying additional attributes - for example:

- MP3 trading sub-communities focused on the work of a single artist or genre

Some existing file-sharing applications restrict the objects that can be shared. Other applications share any file but fail to take advantage of meta-data. Without support for searching this meta-data it is impossible to extend the benefits of file-sharing to complex object types.

U-P2P is used to generate a customized application from a description of the attributes of the object without additional programming. Object descriptions can be written in any text editor, based solely on domain knowledge, using Extensible Markup Language (XML) - a standardized language for data representation. Separating the object description from the application makes U-P2P flexible enough to create file-sharing applications devoted to any class of object.

U-P2P consists of many communities: each community must define the object shared - for example: **an MP3-sharing community shares MP3 objects**. This is analogous to the idea of a class in object-oriented programming which specifies the structure of objects. In object-oriented languages a class is merely another type of object whose structure is specified by a metaclass. Traversing the analogy in the

Manuscript received February 14, 2002.

Aloke Mukherjee is with Carleton University, Ottawa, Ontario, Canada (e-mail: alokem@cisco.com).

Babak Esfandiari is a professor in the faculty of Systems and Computer Engineering at Carleton University (e-mail: babak@sce.carleton.ca)

Neal Arthorne is also with Carleton University (e-mail: narthorn@chat.carleton.ca)

opposite direction, a specific U-P2P community can be seen as a class instantiated by a more general metaclass: **a Community-sharing community shares Community objects**.

metaclass is to a_*class* is to an_*object*
what
community is to *mp3 community* is to *mp3*

Similarly to a class, a specific community (e.g. a CML community or a MP3 community) is just another object. In U-P2P the problem of discovering the existence of a community is thus reduced to the problem of finding an object. This provides a standard way to discover the existence of resource-sharing communities.

II. Prior work

Napster opened up a huge distributed library of files to thousands of users. By downloading popular files, users increased the robustness of the network by increasing the probability of finding a host sharing the file. Unfortunately, the benefits were limited to those sharing MP3 audio files. This assumption was built into the system making it difficult to extend the system to share other objects.

Gnutella, another interesting file-sharing application, lies at the opposite end of the spectrum. Aside from differences in network architecture Gnutella allows the sharing of any type of file. There is no explicit meta-data handling: attributes of objects are not specified and cannot be searched. In Napster and many Gnutella implementations search is implemented by filename matching against a simple search string. This acts as a barrier to sharing of complex objects. For example, a design patterns community requires the ability to search not just name but purpose, keywords, applications, etc.

This has led to proposals in the Gnutella developer community for richer meta-data searches [1, 2]. As with U-P2P, these revolve around using schemas for describing shared objects. One schema defines a structured format for searching MP3 meta-data that is sent as a search string to other Gnutella nodes. Responding clients use the query to search local files annotated using the schema and returns the results using the same structured format. Other P2P systems such as FastTrack, Opencola or Bitzi propose variations on that idea. They are all limited to a number of predefined schemas.

Increased use of meta-data is a step in the right direction. But meta-data must be separated from the application to allow file-sharing to become a general solution to a wide class of problems. Additionally, none of the existing proposals addresses the problem of discovering communities.

III. XML and XML Schema

Clearly, there is a need for shared ontologies or structure if we want to allow search for any type of file. In the Internet world, XML Schema [3] is the new format of choice to represent meta-data structures, and to a certain extent ontologies, replacing the older Document Type Definition defined for XML. XML Schema supports the creation of complex custom data types for XML tags, which is essential to describe resources of a composite nature.

There is also a need for rich, semantic descriptions of relationships between resources that will allow software agents to perform the laborious task of information search and retrieval. The Resource Description Framework [4] and more generally the Semantic Web [5] effort both address that need. RDF is specified using XML so it is compatible with U-P2P, but XML Schema is used because the RDF Schema language is still in flux. XML Schema also has better support for data-types. Another important benefit is the existing base of XML Schema or DTDs of objects such as species [6], genes [7] and molecules. [8]

It is worth mentioning at this point the Edutella project [9], which aims at creating a P2P network for sharing meta-data, described using RDF, as a technology for distributed learning. However in Edutella meta-data is only the resource, not also the means to describe one.

IV. Architecture

A. Generative role of XML Schema and XSLT

File-sharing applications have a few basic functions that benefit from knowledge of meta-data. In U-P2P, the Creation Function presents a form for entering attribute values. The Search Function presents a form used to specify the attributes of the desired objects. Finally, the View Function should allow all of the attributes of an object to be displayed. These functions require knowledge of attributes and attribute types. U-P2P tries to factor this knowledge out of the application by expressing it in the form of an XML Schema. In U-P2P, the fundamental information about a community is expressed in its schema. For this reason, U-P2P allows users to create their own schemas for shared objects and then share the schema with other users to form a "community".

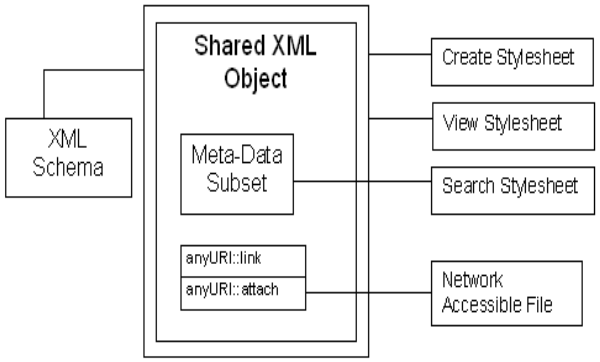
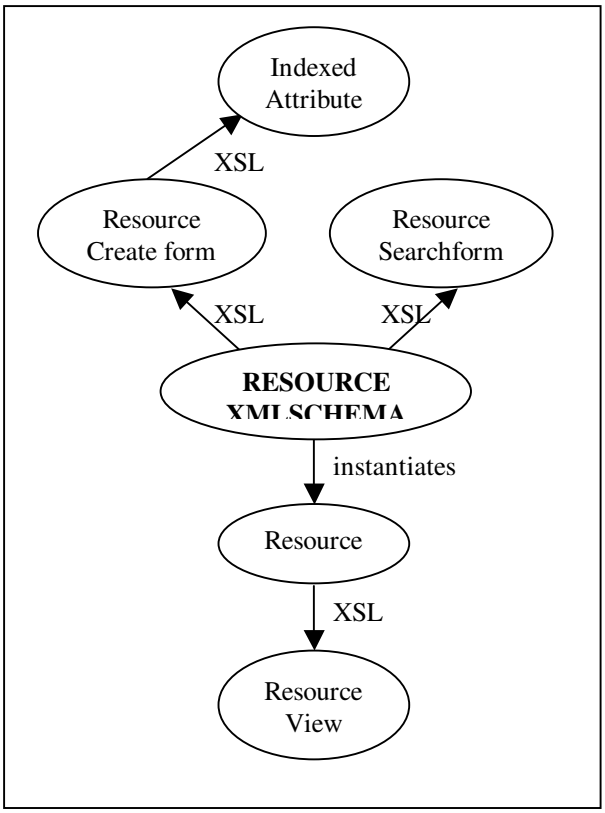


Fig. 1. U-P2P shared object model. This diagram shows the UP2P components and their relation to the shared XML object. The Create, View and Search stylesheets shown are strictly for presentation in an HTML web browser interface and are not essential for other GUIs.

The schema for a community provides a template for both searching, creating and viewing shared objects. In U-P2P the template is instantiated using Extensible Stylesheet Language Transformations (XSLT). XSLT, also specified in XML, specifies the transformation of one XML document into another [10]. In U-P2P, XSLT is used to transform XML Schema into another format: HTML. In a web browser, this allows XSLT stylesheets to render screens for creating, viewing and searching for shared objects. U-P2P provides default stylesheets that operate on any community schema, but users are encouraged to create their own stylesheets to customize their community.

Fig. 2 Relationship between schema, stylesheets and U-P2P functions.



Using XML Schema provides a standard way of describing objects (creating meta-data) that is easy to understand and easy to deploy. U-P2P addresses the difficulty of creating file-sharing applications devoted to a specific class of object by allowing users to easily define their own objects. For this reason, U-P2P allows users to create their own schemas for shared objects and then share the schema with other users to form a "community". A schema is an important attribute of a community: it is used to instantiate the objects that are being shared.

U-P2P provides one default schema as a bootstrap: a schema for community objects. Thus through the same facility, users can search for objects *within* a community or search for a community itself. Under this model, a user must join a community by downloading its schema in order to conduct searches in that community. All users are members of the global or root community by default and can search, create and view other communities at any time. This provides a standard way to discover the existence of resource-sharing communities.

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema">
 <element name="community">
  <complexType>
   <sequence>
    <element name="name" type="xsd:string"/>
    <element name="description" type="xsd:string"/>
    <element name="keywords" type="xsd:string"/>
    <element name="category" type="xsd:string"/>
    <element name="security" type="xsd:string"/>
    <element name="protocol" type="protocolTypes"/>
    <element name="schema" type="xsd:anyURI"/>
    <element name="displaystyle" type="xsd:anyURI"/>
    <element name="createstyle" type="xsd:anyURI"/>
    <element name="searchstyle" type="xsd:anyURI"/>
   </sequence>
  </complexType>
 </element>
 <simpleType name="protocolTypes">
  <restriction base="string">
   <enumeration value=""/>
   <enumeration value="Napster"/>
   <enumeration value="Gnutella"/>
   <enumeration value="FastTrack"/>
  </restriction>
 </simpleType>
</schema>
```

Fig. 3. An XML Schema for resource-sharing communities

This is an attempt at defining the attributes of a file-sharing community. In the context of U-P2P a community is defined by a schema and a set of stylesheets – these attributes can be seen above. Attributes such as “keywords”, “description” and “category” can be used to filter searches and provide more meaningful information about a community’s purpose. In addition there are attributes referring to the underlying peer-to-peer network such as “security” and “protocol”. These will eventually be used to correctly format and route queries in the system but are not implemented today.

B. Implementation

U-P2P is a web-based application. Any browser can be used to interface to a U-P2P servent. UP2P is currently implemented using JavaServer Pages (JSP) and several helper Java classes. The Tomcat Web server was used as the foundation of the servent because of its support for Java ServerPages. Java was chosen because of the Xerces and Xalan libraries: Java libraries for XML parsing and XSL transformation. Information about objects in the system is stored in a database based on Magenta, a Java package developed by the authors as an agent framework [11]. Transactions between the servent and the database are formatted as CMIP queries.

The primary objective of U-P2P is to facilitate the exchange of information or resources using meta-data rich searches over a peer-to-peer network. U-P2P does not focus on the underlying network architecture or discriminate between centralized or distributed approaches to searching, peer discovery, message routing or security. It is meant to be layered on top of any peer-to-peer network organization and expose some basic features of searching, sharing and viewing objects. Although the current implementation is tied to a centralized database, the basic primitives - create, search and view - can be implemented in any peer-to-peer network.

C. Important Functions

1) Create

A user can create and share an object on the peer-to-peer network. For the user, this means selecting a schema, writing out the meta-data for the object (or using an automated meta-data extraction tool) and submitting the object to the UP2P client. The UP2P client will then send off the meta-data to some either centralized source or in a distributed scenario, the meta-data will be kept locally for distributed searches.

The shared object will always be an XML object described by the community schema. It may or may not have links to network accessible files that are flagged as attachments. Attachments are only downloaded when the object is retrieved from a peer on the network. In the case of creating a community, the user will fill out the necessary meta-data according to the community schema provided by UP2P. This ensures that all communities are uniform in their description and can be found in searches.

2) Search

A user can join a community and then search within that community by filling out a search form derived from the community schema. Fields defined in a community schema must be marked searchable for them to form part of a search query. This allows only fields with small portions of content to be present in the search engine instead of the entire XML object. Schema authors will be required to mark fields as searchable.

After filling in meta-data fields the search will be submitted to the UP2P client. The UP2P client will send the search out on the underlying peer-to-peer network implementation. Results will be returned from the network and will consist of full meta-data for each search result. All the search results will be processed and presented to the user for further scrutiny. The user can select a search result and UP2P will download it from the peer who is providing the shared object. This download involves retrieving from the network, any file(s) attached to the shared object and storing them locally for the user.

3) View

UP2P provides a view for any objects that the user has provided for sharing or has downloaded from the network. In the case of the web browser interface, this view is generated by an XSLT stylesheet provided by the creator of the community. This feature also includes browsing through downloaded files and organizing the downloaded files using filters or other tools.

V. Case Study: Design Patterns

The Carleton Pattern Repository [12] was started in 1999. It serves as a repository for software design patterns and provides extensive search capabilities over an as yet, small list of patterns. The patterns are represented in XML using a DTD designed for especially for the repository project.

The repository website contains papers on representing design patterns in XML, searching over design patterns and even a small mention of a distributed model for the repository [13]. The distributed model proposed was for each author group to have a repository server with a fixed list of the other servers in the network. The servers would presumably form a highly distributed mesh and send out their searches to all other servers. This model was not implemented and evidently, no one else has pursued the idea.

Using the DTD as a basis we have developed an XML Schema for representing design patterns [14]. This is used as the basis of a file-sharing community for design patterns. In addition to the schema a custom stylesheet was required to render this complex object since the default stylesheet is tailored to more simple formats. Another design problem is deciding which parts of the design pattern should be indexed. The community designer can also control this by implementing a stylesheet to filter indexable attributes from the XML object before submitting them to the local or remote database.

To our knowledge, prior to our work there has been no way to share design patterns in a peer-to-peer fashion that incorporates meta-data search. When fully implemented this U-P2P based system will expand the benefits of peer-to-peer file-sharing to this area. Such a system would allow computer scientists and students to publish a rich collection of patterns into an underlying peer-to-peer network, search them using rich queries and replicate popular patterns to increase their accessibility. The community-discovery aspect could also be used to access sub-communities devoted to different classes of design patterns or based on different underlying networks.

VI. Conclusion

The current success of peer-to-peer file sharing applications has highlighted the benefits of distribution and redundancy of resources. Spreading these benefits to wider applications requires a simple way to describe a resource-sharing community. By describing the resource using XML Schema, U-P2P separates the description of the resource from the application. Using XSL transforms the description can be used to generate the interface of the application.

The problem of community discovery arises as communities proliferate. U-P2P proposes that treating a community as a resource resolves this problem. A schema for describing a resource-sharing community is used as the basis of a community discovery application. By treating a community as an object the community discovery problem becomes just a specific case of the more general problem of resource discovery.

Sharing of design patterns and other complex objects presents interesting directions for future work. Some design challenges in this area include: rendering an object creation interface less restrictive than HTML forms, replacing CMIP-based queries with richer languages such as the XML Query language and more robust methods of marking indexed attributes.

Another major direction for future work is in demonstrating the protocol independence of U-P2P. This involves removing the dependencies on Magenta and introducing a more generalized interface to the peer-to-peer layer. One approach may be to use JXTA, a peer-to-peer architecture proposed by Sun [15], as the network layer, and to use queries formatted according to JXTASearch. Another approach might be to model the peer-to-peer layer as providing a generic interface with primitives for create, search and retrieve.

We have also created a web-based tool for generating XML Schema. The benefits of integrating this with U-P2P will be to hide the underlying XML completely from the user. This tool could also be enhanced to create all the accompanying stylesheets required for rendering each community and to assist the user in making design choices such as which attributes will be indexed.

Note: U-P2P is an open source application licensed under the GPL. The source code for the prototype version 0.2 can be browsed and downloaded at http://sourceforge.net/project/showfiles.php?group_id=23652.

References

[1] Sumeet Thadani. "Meta Information Searches on the Gnutella Network" http://www.limewire.com/index.jsp/metainfo_searches August 2001.

[2] Sumeet Thadani. "Meta Data searches on the Gnutella Network (addendum)" http://www.limewire.com/developer/MetaProposal2.htm July 2001.

[3] XML Schema Part 0: Primer: http://www.w3.org/TR/xmlschema-0/

[4] Resource Description Framework: http://www.w3.org/RDF/

[5] Tim Berners-Lee, James Hendler and Ora Lassila. "The Semantic Web," *Scientific American*, May 2001

[6] Robert D. Stevenson, Robert A. Morris. "Electronic Field Guides: Leveraging Biodiversity Data with the Web and Mobile Tools; Project Description", http://www.cs.umb.edu/efg/dbi

[7] Brian King. "Introduction to AGAVE Genomic Annotation XML," *Abstract for Bioinformatics Open Source Conference (BOSC) 2001.*

[8] Peter Murray-Rust and Henry S. Rzepa. "Chemical Markup, XML, and the Worldwide Web. 1. Basic Principles," *J. Chem. Inf. Comput. Sci.,* May 26, 1999.

[9] Nejdl, Wolfgang et al. "EDUTELLA: A P2P Networking Infrastructure Based on RDF" http://edutella.jxta.org/reports/edutella-whitepaper.pdf November 14, 2001.

[10] XSL Transformations: http://www.w3.org/TR/xslt

[11] B. Esfandiari, G. Deflandre, J. Quinqueton and C. Dony, "Agent-Oriented Techniques for Network Supervision", Annals of Telecommunications, Tome 51, No. 9-10, Sept 1996, pp 521-529. Magenta project website. http://magenta.sourceforge.net

[12] Dwight Deugo, Darrell Ferguson, Carleton Pattern Repository. http://muffin.nexus.carleton.ca/~darrell/repo/

[13] Darrell Ferguson. "Updates to the Pattern Repository." http://muffin.nexus.carleton.ca/~darrell/papers/UpdateReport.pdf

[14] Neal Arthorne. A XML Schema for Design Patterns. http://chat.carleton.ca/~narthorn/project/patterns/pattern.xsd.

[15] Gong, Li. "JXTA: A Network Programming Environment." IEEE Internet Computing Vol 5. No. 3. May/June 2001.