Managing Scientific Metadata

Metacat is a network-enabled database framework that lets users store, query, and retrieve XML documents with arbitrary schemas in SQL-compliant relational database systems.

Matthew B. Jones, Chad Berkley, Jivka Bojilova, and Mark Schildhauer
National Center for Ecological Analysis and Synthesis

Investigators in the ecological sciences use a wide variety of protocols to collect data on complex topics such as marine bacterial community functions and global carbon flux. The resulting heterogeneous data are stored in autonomous database systems dispersed throughout the research community. There is growing recognition that these data should be networked and preserved for future studies to reuse in replicating and validating scientific conclusions, enlarging spatiotemporal scale, and so on. Ideally, these archived data should be stored in a framework that enables rapid, powerful access and discovery.

In response to this situation, we at the National Center for Ecological Analysis and Synthesis (NCEAS) at the University of California, Santa Barbara, have developed the modular Metacat framework (short for "metadata catalog"). The system (available from the Knowledge Network for Biocomplexity homepage at http://knb.ecoinformatics.org/) incorporates RDF-like methods for packaging data sets to allow researchers to customize and revise their metadata. It is extensible and flexible enough to preserve utility and interpretability working with future content standards.

Metacat solves several key challenges that impede data-confederation efforts in ecological research — or any field in which independent agencies collect heterogeneous data that they wish to control locally while enabling networked access. This distributed solution integrates with existing site infrastructures because it works with any SQL-compliant database system. The framework's open-source-based components are widely available, and individual sites can extend and customize the system to support their data and metadata needs.

Barriers to Confederation

Major research networks such as the Organization of Biological Field Stations (180 sites) and the Long-Term Ecological Research Network (24 sites) are increasingly concerned with synthetic analyses and integrative studies that could greatly benefit from access to additional existing data. It is currently an arduous task to locate and access data for these types of analyses. A framework for data confederation would let individual researchers and

1089-7801/01/$10.00©20010 IEEE

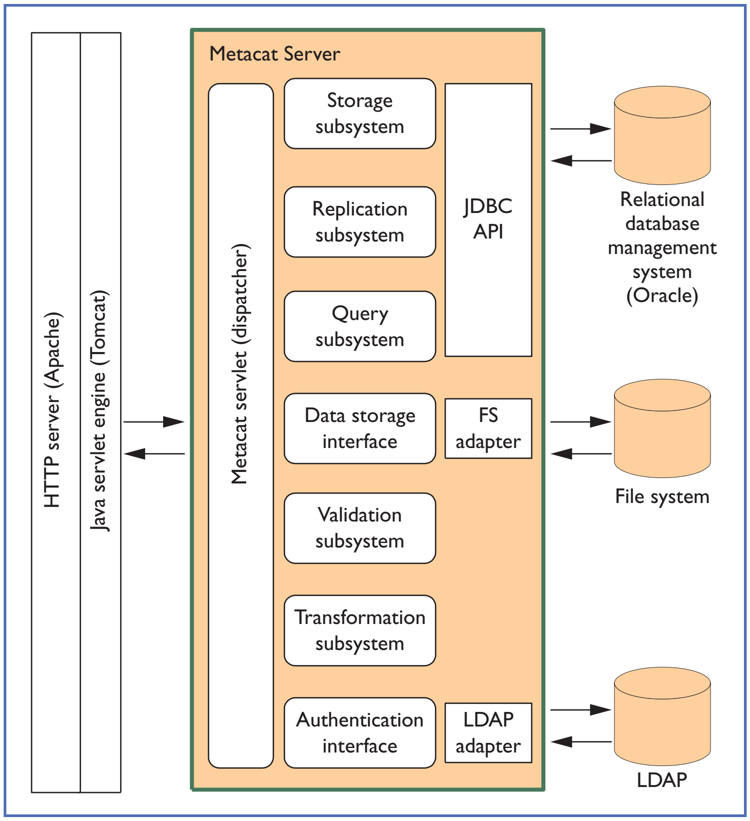

Figure 1. Architectural overview of the Metacat framework. Storage, query, replication, validation, and transformation functions are mediated via a Java servlet called the "Metacat Server," which has interfaces to RDMS-housed metadata, raw data storage (File system), and authentication services (LDAP).

sites continue collecting data by their own methods, while enabling them to share the documented data with researchers anywhere on the Internet. Despite the desire to confederate, ecological data still bear the strong imprimatur of the individual, field station, or organization that supported their collection. This trend gives rise to three main barriers that any system must address:

- *Data heterogeneity.* Data structures representing entities and attributes in ecological data sets tend to be diverse and dynamic because collection efforts and storage methods are often dictated by independent researchers' immediate needs.
- *Data dispersion.* Most ecological studies are carried out by individual investigators focused on specific research problems. This individualistic tradition can generate isolated islands of data controlled by individual scientists, although some investigators are members of ecological field stations and research labs that might aspire to manage their data within a common infrastructure.
- *Local control.* Each site in a research network manages data from tens to hundreds of independent investigators. Sites use their own data management systems with varying metadata standards and different objectives, which renders centralized decision-making and standardization nearly impossible. Efforts at confederating data across research institutions and among investigators must therefore provide a mechanism for maintaining local control of dispersed data.

Here we describe a data confederation framework that addresses these technical and sociological impediments and enables efficient network-based discovery and access. Metacat uses structured metadata expressed in Extensible Markup Language[1] to provide rich descriptions of data content and structure. While we also provide a specific metadata vocabulary for ecological data, the framework can store and present metadata contained within any well-formed XML document.

A Metadata-Based Framework

In our framework, investigators describe data syntax and semantics using metadata vocabularies defined by their own communities (ecology or geology, for example). The system serializes the metadata using XML and stores the documents in a schema-independent XML database, which lets researchers store, query, and retrieve formatted metadata. Data managers can also replicate data to a set of centralized servers to enable investigators to search the entire network for related data. Metacat uses an SQL-compliant relational database management system (RDMS) to store the XML data, and each research site maintains local control over its data and metadata.

In response to W3C director Tim Berners-Lee's recommendation that scientific disciplines develop their own controlled vocabularies to help facilitate the "semantic Web," the ecological community has focused on a few metadata content standards — particularly the U.S.-based Ecological Metadata Language (EML) and the National Biological Information Infrastructure's Biological Data Profile. Because such standards evolve rapidly, a critical design requirement was for the Metacat database to handle changes in metadata content standards without additional programming efforts. We designed the system to be independent of particular metadata schemas; it can store documents specified by any valid XML DTD, which thus enables researchers to add new metadata types as needed.

Because metadata document types are dynamic and the relationships among data and metadata can be complex, we needed a flexible mechanism for associating specific metadata documents with data files. For our implementation, we used a derivation[2,3] of the Resource Description Framework (RDF)[4] model. We defined a data package as a collection of data entities and metadata documents that are useful for a particular purpose (such as analysis). Following the RDF model, each data package is represented as a labeled, directed graph that defines a set of relationships between data and metadata. Nodes in the graph are the data and metadata objects, and the arcs represent relationships between them. A typical package relationship might be written as "Metadata object A describes the attribute structure of data object B," for example. Because packages themselves can serve as objects within a relationship, the Metacat framework includes a flexible mechanism for adding new and more complex objects without modifying the underlying storage system.

Metacat Architecture

The Metacat framework is controlled by a Java servlet that acts as the interface to any SQL-compliant relational database with a Java Database Connectivity (JDBC) driver. This vendor independence — we have tested the system using Oracle and PostgreSQL on Linux (Redhat 6.2 and 7.0) and Microsoft SQL Server on Microsoft Windows 2000 — allows ecological field stations to readily integrate Metacat with their current infrastructures.

The Java servlet communicates using HTTP, which reduces inter-institutional issues, such as opening holes in firewalls, that are often associated with setting up communications channels. Figure 1 presents an architectural overview of the Metacat framework. The servlet acts as a dispatcher, passing commands and data from client applications to the various subsystems that handle Metacat's functions. The main subsystems are storage, replication, query, validation, and transformation. Metacat handles authentication through a generic interface that can employ various services, but most installations will find the lightweight directory access protocol (LDAP) adapter sufficient. The data storage interface allows Metacat to point directly at referent raw data that might be stored on a local or remote file system.

Storage Subsystem

The Metacat storage subsystem lets researchers store XML data with arbitrary schemas in a relational database. The system facilitates efficient path-based queries by employing the Document Object Model (DOM)[5] to model the hierarchical structure of XML itself, rather than any particular XML document's schema representation.

An XML document can be modeled as a tree in which the root node represents the document entity, and children of the root node represent elements, attributes, and character data. (The tree's leaves are typically character data nodes.) Although it does not yet implement the DOM API, Metacat uses the DOM to store XML documents in a relational database.

The relational model differs substantially from the DOM representation, however. To represent an XML document in a table, we first use a SAX2 (Simple API for XML, http://www.megginson.com/ SAX/) parser to decompose the XML tree into constituent nodes. Figure 2 (next page) illustrates the way Metacat's storage subsystem can parse an arbitrary XML document into a series of DOM nodes that can be housed as individual records in an RDMS table (xml_nodes). Thus, for the simple XML metadata document:

The system is designed to be independent of particular schemas.

```
<dataset>
 <ds_id>12345</ds_id>
 <creator>Jane Scientist</creator>
 <desc>
  <title>Red Abalone along the Santa
     Barbara Coast</title>
  <dept>Marine Biology</dept>
 </desc>
</dataset>
```

the subsystem inserts each node as a record into a database as shown in Table 1 (next page), which has a recursive foreign key (`parentnodeid`) that allows each record (representing a node in the XML tree) to point to its parent. Because XML's tree structure limits each node to only one parent, the rootnodeid field allows a query hit on a particular node to be quickly linked back to its document root.

Regardless of the schema it represents, any XML document can be stored in Metacat because the table structure implements the DOM. Thus, while it was designed to store metadata for the ecological community, Metacat's DOM storage model makes it a general-purpose XML storage system.

Table 1. XML document nodes in a relational table.

Nodeid	Nodetype	Nodename	Nodedata	parentnodeid	rootnodeid
0	Document	dataset			0
1	Element	dataset		0	0
2	Element	ds_id		1	0
3	Text		12345	2	0
4	Element	creator		1	0
5	Text		Jane Scientist	4	0
6	Element	desc		1	0
7	Element	title		6	0
8	Text		Red Abalone along the Santa Barbara Coast	7	0
9	Element	dept		6	0
10	Text		Marine Biology	9	0

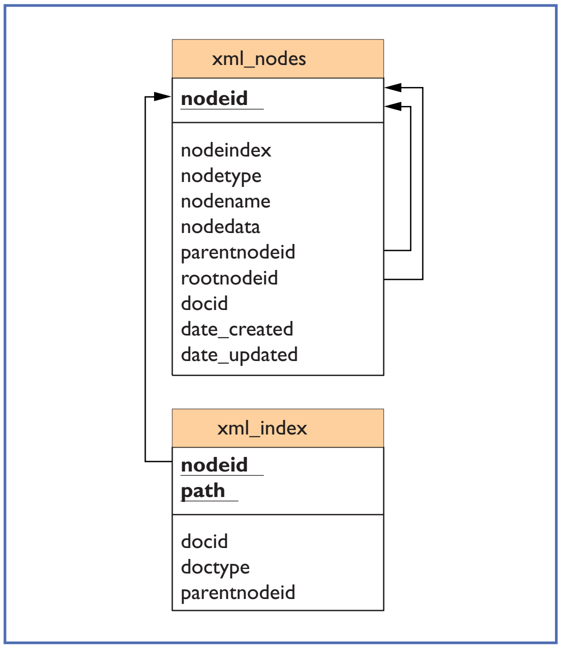

Figure 2. Entity relationship diagram. Metacat's core XML data storage model stores each node of the XML document as a record with a foreign key that indicates its parent node.

Replication Subsystem

We first envisaged Metacat as a centralized metadata server, but soon realized that research stations would need local servers that could share data with Metacat servers at other sites. Individual stations could thus maintain local autonomy over metadata and datasets, while gaining greater fault tolerance and network accessibility by replicating their metadata to other sites.

Metacat's specific role as an ecological research server allowed us to put two significant restrictions on the replication requirements.

- Replication can occur at the document level because the unit of interest is generally a complete XML document, which would correspond to the metadata for a raw data set.
- It is reasonable to restrict data control to a document's "home" or source site because data management in ecology is generally site based. This level of control would be useful in other scientific disciplines characterized by highly individualized and thematically broad research efforts, and where standardized protocols and public data repositories are not yet broadly available.

In the resulting Metacat replication design, we focused on data consistency, locking, and replication control.

Data consistency. A major requirement for the replication service was that metadata remain consistent on each server — even when Internet outages and server downtime temporarily interrupt connectivity. Figure 3 illustrates the two mechanisms we used to accomplish this goal: *timed checkpoints* and *event-based change notification*.

In the first, the destination server checks the accession numbers —which are incremented every time documents are updated on the source server — for all metadata documents on the source server and requests any missing or changed XML documents at startup. These checkpoints are designed to synchronize servers that have been disconnected. They are also timed to check periodically after startup to ensure that network outages haven't caused drift in the document store.

The second mechanism allows the source server to propagate changes by signaling destination servers when a document change event occurs (`insert, update, delete`). We could improve this mechanism's performance by transmitting only the differences between versions when a document is updated, as in the Concurrent Versions System (http://www.cvshome.org/) and rsync.[6] The event-based change notification could also be used to communicate the list of documents on the source server, which would improve replication performance as the number

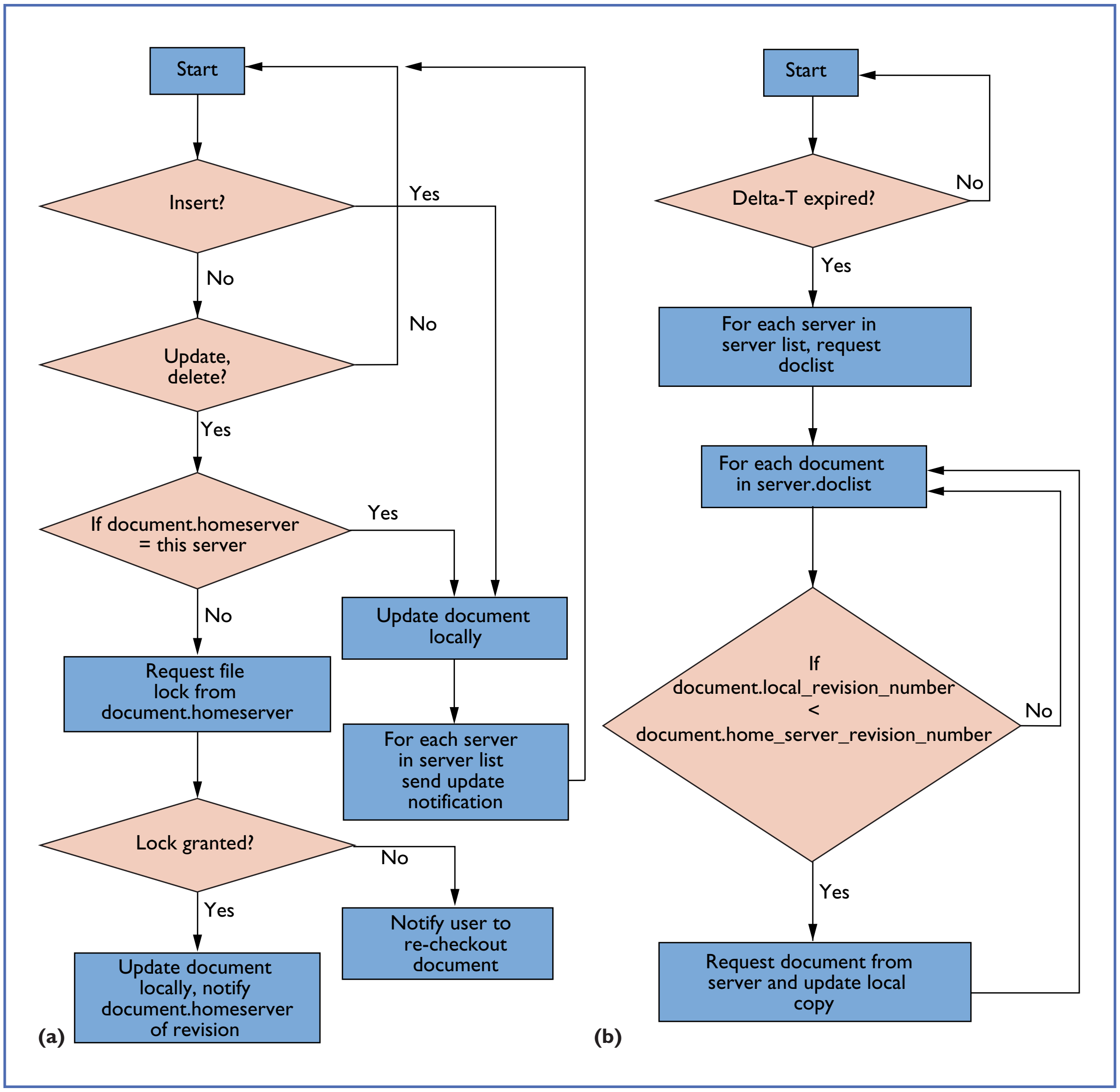
Start
Insert?
Yes
No
Update, delete?
No
Yes
If document.homeserver = this server
Yes
No
Update document locally
Request file lock from document.homeserver
For each server in server list send update notification
Lock granted?
No
Yes
Notify user to re-checkout document
Update document locally, notify document.homeserver of revision
(a)
Start
Delta-T expired?
No
Yes
For each server in server list, request doclist
For each document in server.doclist
If document.local_revision_number < document.home_server_revision_number
No
Yes
Request document from server and update local copy
(b)

Figure 3. Decision trees for replication services. (a) Event-driven replication allows the source server to propagate changes when a document change event occurs. (b) Timed-checkpoint replication is designed to synchronize servers that have been disconnected.

of distributed sites and documents in the system grows.

Locking. Each XML document is associated with a home or source server that controls write access to the document. The server employs a locking mechanism to keep users from changing out-of-date revisions, and to ensure that replicating servers cannot update locked documents over the network. When a user wants to update a remote document, the replicating server requests the lock from the home server, which then verifies that the update will be made against the most recent revision. If the request was made with an older version of the document, the lock is denied.

Replication control. The Metacat framework includes a mechanism that lets field sites restrict which other Metacat servers can replicate their XML documents; it also lets sites choose which other servers' metadata to replicate locally. Documents can only be replicated from their home server, and the destination server must initiate the request, which means the source and destination servers must both agree to the transfer before it occurs.

Metacat also enables sites to configure servers for one-way replication. This allows administrators to mirror metadata out to a centralized metadata clearinghouse without replicating the entire clearinghouse onto their local Metacat. Of course, any publicly accessible XML document could be read and redistributed (possibly illegally), but these

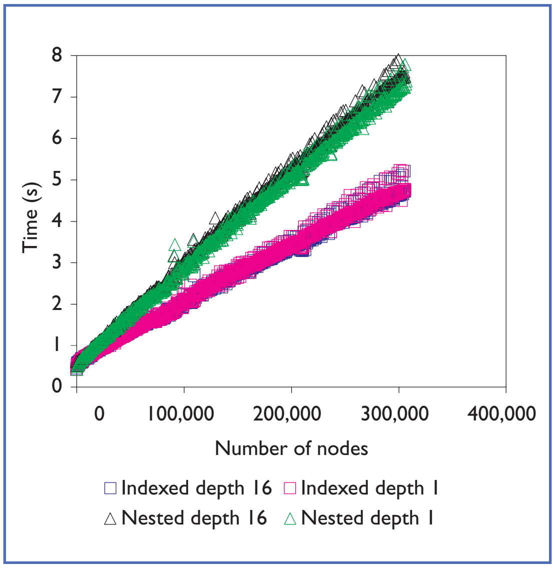

Figure 4. Query performance results with a uniform document set. In this test, the precomputed path index (squares) outperformed the nested SQL query (triangles) method as the number of nodes increased. Shorter times are better.

controls ensure that only installations trusted by the site administrator can access private XML data.

Metacat's flexible replication scheme allows individual scientists and organizations to share data or metadata without relinquishing autonomous control over it. The replication mechanism can be used to create a centralized Metacat server that stores up-to-date metadata and data from all registered research sites, providing a cross-organizational metadata search facility, which addresses the data dispersion problem.

Query Subsystem

The Metacat framework's target audience consists largely of ecological researchers who will be registering their own metadata, but also searching for others' data contributions to complement their own work. These scientists thus need a simple but powerful querying system to assist in locating useful data sets registered within the Metacat system.

Query efficiency. The ability to store documents encoded using arbitrary XML schemas complicates querying the Metacat document store. Storing XML documents in an RDMS as a decomposed DOM structure forces us to use a nested SQL query in order to search by a specific path in the document:

```
SELECT rootnodeid FROM xml_nodes
WHERE nodedata LIKE ‘Marine Biology’
AND parentnodeid IN
(SELECT nodeid FROM xml_nodes
WHERE nodename LIKE ‘dept’
AND parentnodeid IN
(SELECT nodeid from xml_nodes
 WHERE nodename like ‘desc’
 AND parentnodeid IN
 (SELECT nodeid from xml_nodes
 WHERE nodename like ‘dataset’)));
```

Alternatively, we could execute this query by precomputing an index such as the xml_index table in Figure 2, which allows Metacat to quickly locate paths in the XML data. The query subsystem creates the records in this table when a document is inserted or updated. During this indexing phase, Metacat constructs all possible absolute and relative paths through the XML document and writes them to the index table along with a pointer to the deepest node's nodeid in the xml_nodes table.

Using an index table, we can reformulate each path-based SQL query as a select from the xml_nodes table and from the xml_index table:

```
SELECT DISTINCT rootnodeid FROM
xml_nodes
WHERE nodedata LIKE ‘Marine Biology’
AND parentnodeid IN
(SELECT nodeid FROM xml_index
WHERE path LIKE
‘/dataset/desc/dept’);
```

We expected a significant difference in performance between the two methods. As Figure 4 shows, the indexed query outperformed the nested query — especially with increasing numbers of nodes — for a document corpus with uniform structure (text nodes are evenly distributed). We also expected the nested query to become slower as the depth of the paths increased, but we found a linear relationship between the number of nodes and query time; node depth had little disproportionate impact on performance.

Figure 5 shows the results of querying a document corpus with a clustered structure — an XML markup of the Old Testament with a maximum node depth of 6, and approximately 80 percent of the 3.32-MByte file tagged as "verse" (depth-5 elements). For this test, we found that nested queries actually outperformed index queries.

These tests indicate that the database's performance depends greatly on the structure of the documents stored in the system. Metadata documents in the ecological community tend to be moderately structured (less than the document in the Figure 4 test, but much more structured than the Old Testament document). Test queries up to 20 nodes deep showed qualitatively similar behaviors to those described here, and metadata schemas for ecology generally reach a maximum depth of less than 10 nodes. We thus believe that Metacat should perform well on the intended document base, although query profiles might be different in other application domains.

Query specification format. A client accesses the Metacat query system by passing an XML-encoded version of the path query to the servlet. The XML query is marked up according to a document-type definition (DTD) that allows for Boolean logic and partial string matching (see the pathquery.dtd in the distribution for details). The mark-up process itself can be completed via simple client-side querying wizards that construct valid path queries for the user by presenting simple form field templates based on the XML DTDs within the metadata content specification. We have created one such client-side application, Morpho, which uses the EML content specification to assist in query construction. For a simple case-insensitive substring match on the string “12345” in the path “/dataset/ds_id,” for example, we would submit the following query:

```
<pathquery>
 <querygroup operator=“UNION”>
  <queryterm casesensitive=“false”
        searchmode=“contains”>
   <value>12345</value>
   <pathexpr>/dataset/ds_id</pathexpr>
  </queryterm>
 </querygroup>
</pathquery>
```

To specify free-text searches (not constraining the search to particular paths), we simply omit the `<pathexpr>` element.

To specify Boolean logic of arbitrary complexity, we use a combination of `<querygroup>` and `<queryterm>` elements. Within a query group, all terms are combined using either logical AND or logical OR. Within a query term, attributes can specify whether substring matching is case-sensitive and whether to perform an exact match or one of several types of substring match (contains, starts-with, ends-with). As XML-based query standards mature, we intend to replace these path queries with an open standard such as XQuery.[7]

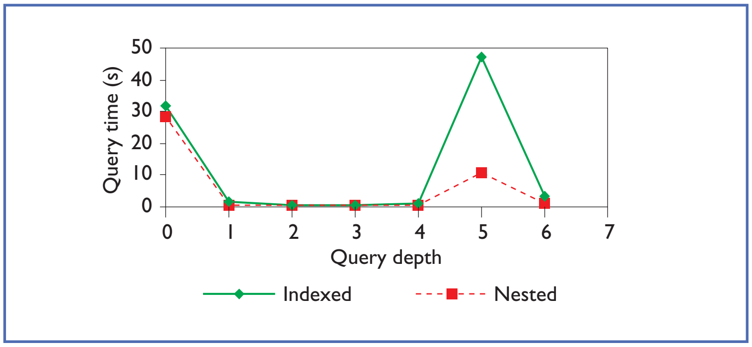

Figure 5. Query performance with an irregular document set. In this test, the repository consisted of 2.03×10^6 nodes (80 percent of which were five nodes deep), and nested SQL queries (dashed line) outperformed the precomputed path index (solid line) method.

Query result set. Metacat returns query results as XML documents that include both the query and a list of the documents that satisfy it. The default result set returns each document’s identifier (`docid`), name (`docname`), type (`doctype`), creation date (`createdate`), and modification date (`updatedate`). Because the XML documents stored in Metacat have arbitrary schemas, however, constructing a result set that contains additional document data is more complicated than in a fixed-schema system.

Clients can use the `returnfield` element to determine which elements the user receives from a document as part of the query encoding. The query subsystem returns the `returnfield` information in param elements and sets an attribute “name” to the name of the requested `returnfield` as shown in Figure 6 (next page). The value of the name attribute for each param element in the result set is the path to that node of the document. These `param` elements can be parsed out using an XSL stylesheet to display them in whatever form is required, including reconstructing part or all of the original XML document. A client can also retrieve any full XML document by its document identifier (`docid`).

Validation Subsystem

The Metacat framework makes extensive use of metadata standards, such as EML, that are formalized in XML DTDs. Users and software clients must therefore be able to validate documents against particular metadata schemas. We have implemented a validation methodology based on document type assertions, in which the PUBLIC identifier is used as the name of the document type and is associated with a particular XML DTD. We have also recently developed a set of XML schema documents that represent EML, and we intend to

```
<resultset>
 <query>
 <querygroup operator=“UNION”>
 <returnfield>/dataset/creator
 </returnfield>
 <returnfield>/dataset/desc/title
 </returnfield>
 <returnfield>/dataset/desc/dept
 </returnfield>
  <queryterm casesensitive=“false”
       searchmode=“contains”>
   <value>12345</value>
   <pathexpr>
    /dataset/ds_id
   </pathexpr>
  </queryterm>
 </querygroup>
 </query>

 <document>
 <docid>metacat.1</docid>
 <docname>dataset</docname>
 <doctype>dataset</doctype>
 <createdate>
  2001-02-09 13:58:21
 </createdate>
 <updatedate>
  2001-02-09 13:58:21
 </updatedate>
 <param name=“dataset/creator”>
  Jane Scientist
 </param>
 <param name=“dataset/desc/title”>
  Red Abalone along the Santa
  Barbara Coast
 </param>
 <param name=“dataset/desc/dept”>
  Marine Biology
 </param>
 </document>
</resultset>
```

Figure 6. Example XML result set. This document contains a query that specifies what additional data fields to return, as well as the resulting parameter names and values that satisfy the search criteria.

extend Metacat in the near future so that it can use XML schema documents in addition to DTDs to specify validation rules.

When a user submits an XML document for the system to insert or update, the validation subsystem scans the document's DOCTYPE statement for a PUBLIC or SYSTEM identifier. When neither identifier is found, Metacat will accept an untyped (but well-formed) document without validation. If the document is not well formed, it will be rejected.

If the system finds an identifier, however, it validates the document against the associated DTD from the cache. If Metacat has not previously cached a DTD for this identifier, the SYSTEM tag tells it to cache a copy of the DTD and then validate the document against that copy. If the DTD is unavailable for some reason (because of network outage, an invalid URL, or so on), or if the document fails validation, then the document insertion or update fails and the user receives an appropriate error code. Note that the first use of a PUBLIC identifier defines the type associated with it within the system. Users (and automated software applications) can thus be assured that documents associated with a specific type are always valid according to the type registered with the Metacat server.

Transformation Subsystem

XML documents stored in Metacat can automatically be transformed to other XML document types or HTML using Extensible Stylesheet Language Transformations (http://www.w3.org/TR/xslt). XSLT allows flexibility in presenting metadata and exchanging it among participating sites. This feature is particularly relevant to ecological research sites because data and legacy metadata are stored in many arbitrary and informal site-specific formats. Most sites will lack the technical personnel to map their metadata to emerging, formalized community standards; instead, we are working on an approach (similar to that used in the Z39.50 information-retrieval protocol standard for searching and retrieving information from distributed databases; http://www.loc.gov/z3950/agency) in which each site maps its metadata standard to the EML exchange format using XSLT. Then, Metacat automatically converts among standards.

For example, we have already developed a DTD for the Biological Data Profile of the U.S. Federal Geographic Data Committee's content standard for digital geospatial metadata (CSDGM).[8] We intend to provide a mapping that will enable ecological research sites to transform their site-specific metadata expressed in EML into the Biological Data Profile. This will allow Metacat to act as a node on the U.S. National Biological Information Infrastructure metadata clearinghouse system (through a Z39.50 gateway), and therefore indirectly as a node in the U.S. National Spatial Data Infrastructure.

The transformation subsystem can automatically perform metadata format transformations when a user reads a document from the database: an internal lookup table links each document type, based on its PUBLIC identifier, to an XSLT stylesheet that defines the transformation. The end user need never see the document in its original XML form nor even know that the transformation has taken place. The transformation function can also be dynamically turned off so that advanced

Related Work in Ecological Metadata

Managing scientific data requires techniques from multiple disciplines. Our work on Metacat has been influenced by developments in metadata standards, XML databases, and managing heterogeneous data.

Metadata Content Standards

Several existing metadata content standards, such as the Dublin Core (http://dublincore.org/) and the Global Information Locator Service (http://www.gils.net/), are somewhat relevant to ecological and biological sciences, but none are deep and broad enough for effectively documenting biological data.

Perhaps the most relevant of these standards is the U.S. Federal Geographic Data Committee content standard for digital geospatial metadata (CSDGM),[1] but it still has notable omissions in taxonomic coverage and other biologically relevant information. The ecological community has thus been developing its own Ecological Metadata Language (EML) to effectively document metadata that are essential to researchers in this field. EML formalizes and expands on earlier work by several committees sponsored by the Ecological Society of America.[2]

Some of EML's core concepts have been incorporated into a biological profile of the CSDGM to make this standard more comprehensive and useful for ecological researchers.[3] One focus of research efforts on metadata at the Knowledge Network for Biocomplexity (http://knb.ecoinformatics.org/) involves integrating these various documentation standards.

XML Databases

A variety of systems, such as XML-DBMS (http://www.rpbourret.com/xmldbms/), use relational database systems to store XML data.[4] They generally either store the data as a large string object — in which case, XML data are searchable only through a free-text mechanism — or map the XML schema onto the relational schema using template-driven systems. In the latter case, the system's flexibility is limited because the XML documents' schemas must be predetermined to allow them to be mapped to a fixed relational schema. The trend with projects such as Lore (http://www-db.stanford.edu/lore/), XYZFind (http://www.xyzfind.com/), and Ozone (http://www.ozone-db.org/) is toward creating dedicated XML databases that allow storage of XML documents with arbitrary schemas,[5] but these databases have yet to mature to support the enterprise features commonly found in RDMSs.

Heterogeneity and Interoperability

There has been much work on integrating heterogeneous, distributed databases, with most developers focusing either on using prior knowledge to merge two or more schemas into a multidatabase (the Z39.50 protocol, for example, maps multiple databases onto a shared schema)[6,7] or on using automated mechanisms to determine appropriate global views of the schemas.[8,9]

Although attractive, the multidatabase approach requires all participating research sites to map their investigator-specific metadata and data schemas onto a global schema. Machine-generated global views also provide an appealing way to handle heterogeneity, but current methods require extensive, formal metadata for each of the data sources that will be members of the global view.

References

1. FGDC-STD-001-1998, *Content Standard for Digital Geospatial Metadata*, Federal Geographic Data Committee, Washington, D.C., June 1998.
2. W.K. Michener et al., "Non-Geospatial Metadata for the Ecological Sciences," *Ecological Applications*, vol. 7, 1997, pp. 330-342.
3. A. Frondorf, M.B. Jones, and S. Stitt, "Linking the FGDC Geospatial Metadata Content Standard to the Biological/Ecological Sciences," *Proc. Third IEEE Computer Soc. Metadata Conf.*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1999; http://computer.org/proceedings/meta/1999/papers/4/afrondorf.html.
4. J. Shanmugasundaram et al., "Relational Databases for Querying XML Documents: Limitations and Opportunities," *Proc. 25th Int'l Conf. Very Large Databases*. Morgan Kaufmann, San Francisco, 1999, pp. 302-314.
5. J. McHugh et al., "Lore: A Database Management System for Semi-structured Data," *Special Interest Group on Management of Data (SIGMOD) Record*, ACM Press, New York, vol. 26, no. 3, 1997, pp. 54-66.
6. A. Sheth and J. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," *ACM Computing Surveys*, vol. 22, 1990, pp. 183-236.
7. ANSI/NISO Z39.50, *Application Service Definition and Protocol Specification*, American National Standards Institute/National Information Standards Office, Bethesda, Md., 1995; available online at http://lcweb.loc.gov/z3950/agency/.
8. S. Castano, V. De Antonellis, and S. De Capitani de Vimercati, "Global Viewing of Heterogeneous Data Sources," *IEEE Trans. Knowledge and Data Engineering*, IEEE Computer Soc. Press, Los Alamitos, Calif., vol. 13, no. 2, 2001, pp. 277-297.
9. M.P. Reddy et al., "A Method for Integration of Heterogeneous Databases," *IEEE Trans. Knowledge and Data Engineering*, IEEE Computer Soc. Press, Los Alamitos, Calif., vol. 6, no. 6, 1994, pp. 920-933.

users or specific data management clients can directly access XML documents.

Document transformation facilitates sharing of data and metadata among independent sites because one site's metadata can be automatically transformed into the format used by others, or into a standard exchange format such as EML. In addition, query results can be transformed dynamically into various HTML formats when the query is run. We have used this feature of Metacat to create customized user interfaces for three different projects based on one underlying data system.

Conclusion

Metacat provides an extensible and modular framework that should be useful to the growing community of users that need to store, manage, and retrieve structured XML data or metadata. The system couples the flexibility of a schema-independent XML-enabled database with the robust features of a mature SQL-compliant RDMS. By adhering to standardized SQL, we avoid committing to any proprietary implementation, and that allows ecological research stations with varying infrastructures and budgets to use their choice of RDMS.

Our future plans for Metacat include implementing the DOM API, fully complying with the XPath specification (http://www.w3.org/TR/xpath), and adopting a query standard such as XQuery. For most current DOM API implementations, the entire document must be memory-resident, but implementing the DOM API in Metacat will eliminate this requirement by establishing a persistent DOM. Users could then access and edit documents without first extracting them from the database, as is currently necessary. Full XPath compliance will also allow researchers to execute standard structured-path queries. These enhancements will bring Metacat more in line with emerging XML standards and methodologies, and will grant users greater flexibility and more capabilities.

Our structured metadata approach does not immediately solve the problem of integrating complex, heterogeneous data, but it represents a major step toward disclosing "arbitrary" data structures through explicit documentation in a standardized, discipline-specific vocabulary. Our EML metadata content specification, for example, includes elements that are of vital interest to ecological researchers, but also contains elements that provide general information about data table structure, variable typing, and so on. We believe this framework can help facilitate new methods for data sharing and accelerate the pace of scientific discovery by granting researchers access to rapidly growing data stores that can complement and enrich their analytical insights.

Acknowledgments

This article is based on work supported by the U.S. National Science Foundation under grants nos. DBI99-04777 and DEB99-80154 and by the National Center for Ecological Analysis and Synthesis, which is funded by NSF (grant no. DEB-0072909), the University of California, and the UC Santa Barbara campus.

References

1. T. Bray, J. Paoli, and C.M. Sperberg-McQueen, "Extensible Markup Language (XML) 1.0," World Wide Web Consortium (W3C) Recommendation, Oct. 2000; available at http://www.w3c.org/TR/REC-xml.
2. R. Daniel Jr. and C. Lagoze, "Extending the Warwick Framework: from Metadata Containers to Active Digital Objects," *D-Lib Magazine*, Corp. for Nat'l Research Initiatives, vol. 3, no. 11, Nov. 1997.
3. R. Daniel Jr., C. Lagoze, and S.D. Payette, "A Metadata Architecture for Digital Libraries," *Proc. IEEE Forum on Research and Technology Advances in Digital Libraries*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1998, pp. 276-288.
4. O. Lassila and R.R. Swick, "Resource Description Framework (RDF) Model and Syntax Specification," W3C Recommendation, Feb. 1999; available at http://www.w3.org/TR/REC-rdf-syntax.
5. L. Wood et al., "Document Object Model (DOM) Level 1 Specification," W3C Recommendation, Oct. 1998; available at http://www.w3.org/DOM/.
6. A. Tridgell and P. Mackerras, "The rsync algorithm," tech. report, Computer Science Dept., Australian Nat'l Univ., Canberra, Australia, 1998; available at http://rsync.samba.org/rsync/tech_report/.
7. D. Chamberlin et al., "XQuery 1.0: An XML Query Language," W3C Working Draft, work in progress.
8. M.B. Jones, FGDC-STD-001.1-1999, *FGDC Biological Data Profile of the Content Standard for Digital Geospatial Metadata*, Federal Geographic Data Committee, Washington, D.C., Oct. 1999; available at http://www.fgdc.gov/standards/status/sub5_2.html.

Matthew B. Jones is the database and information specialist for the National Center for Ecological Analysis and Synthesis (NCEAS). He has an MS in ecology from the University of Florida. His research interests include scientific computing, ecological informatics, and the integration of heterogeneous scientific data. Jones is a member of the IEEE Computer Society and the Ecological Society of America.

Chad Berkley is the metadata systems developer at NCEAS. He is involved with various research projects on ecological informatics and metadata systems. He has a BS in computer science from the University of Montana.

Jivka Bojilova is the database integration developer at NCEAS. She is involved with various research projects on ecological informatics and database development for scientific data management. She has an MS in computer science from Sofia University, Bulgaria.

Mark Schildhauer is director of computing at the NCEAS. He has a PhD in ecology and evolutionary studies from the University of California, Santa Barbara. His interests include scientific computing, data visualization, the semantic Web, and ecological informatics. Schildhauer is a member of the IEEE Computer Society and the Ecological Society of America.

Readers can contact the authors at {jones, berkley, bojilova, schild}@nceas.ucsb.edu.