Schema-Independent and Schema-Friendly Scientific Metadata Management[1]

Scott Jensen and Beth Plale
Indiana University Department of Computer Science
scjensen@cs.indiana.edu, plale@cs.indiana.edu

Abstract

Computational science is creating a deluge of data, and being able to reuse this data requires detailed descriptive metadata. Scientific communities have developed detailed metadata schemas to describe data products, but this metadata must be captured as workflows execute. Our research has identified characteristics of scientific schemas that can be exploited to efficiently capture and search this metadata based on the schemas specific to each community, but using an easily adaptable framework.

1. Introduction

Computational science grids generate huge volumes of scientific data through the workflow-orchestrated activities of their domain science users. These cyberinfrastructures, often architected as Service Oriented Architectures (SOA), support task composition through user facing and back-end orchestration workflow tools. The activities provided to a scientist can include access to public data repositories, large-scale computational model execution, data mining, image rendering, and analysis.

Grid compute resources are utilized for, at a minimum, the computationally intensive pieces of a workflow, and grid software services are essential for moving the large volumes of data products off the big computational machines on which they were derived. Our research focus in support of computational science is on the management of the long-term life of the data. That is, our research does not focus on having the data in the right place at the right time, though this is an important problem, but on capturing and representing the right metadata such that the data products can be published, shared, and used not only tomorrow but in decades to come.

The word "metadata" is broadly defined as data about data. It is closely related to "provenance", and in fact we view provenance as a subset of metadata. Determining *what is the right metadata to capture?* is an important question that we have dealt with. In the current version of our system, we place priority on capturing the metadata that is needed for short-term reuse of data products and collections by the owner or by another member of the virtual organization. But as a recent study by the U.K. e-Science Programme points out, if users are responsible for annotating their data, the task is often left undone [15]. So a second question is *how much metadata can be captured automatically?* Our focus on short-term reuse has resulted in technology for capturing rich domain-specific information about the data products generated during a workflow for purposes of complex discovery and search after-the-fact. For instance, the configuration parameters to a weather forecasting model are extremely valuable descriptive information for finding a data product after the fact. A meteorologist is interested in being able to find the results of a model run where the grid spacing is 1km, as these runs are more experimental than runs at larger spacing's.

A metadata catalog that is responsive to automatic metadata collection in a workflow system must also satisfy the following requirements:

1. Provide fast query responses, and speak the XML metadata schema of its scientific community.
2. Handle new attribute additions to an existing metadata object.
3. Does not require domain scientists to understand the structure of the XML or relational database structure used to communicate and store their metadata.
4. Can be used in diverse domain sciences.
5. Can be easily set up and integrated into an existing cyberinfrastructure framework.

The first version of our metadata repository responds to the first three requirements above. It is currently in use in the LEAD Science Gateway.

This paper discusses the last two, that is how a metadata catalog having a generalized database schema structure can handle diverse scientific XML metadata schemas through configuration instead of customization by exploiting the characteristics of scientific metadata. A key observation guiding our

1 This work supported under NSF cooperative agreement ATM-0331480 and EIA-0202048.

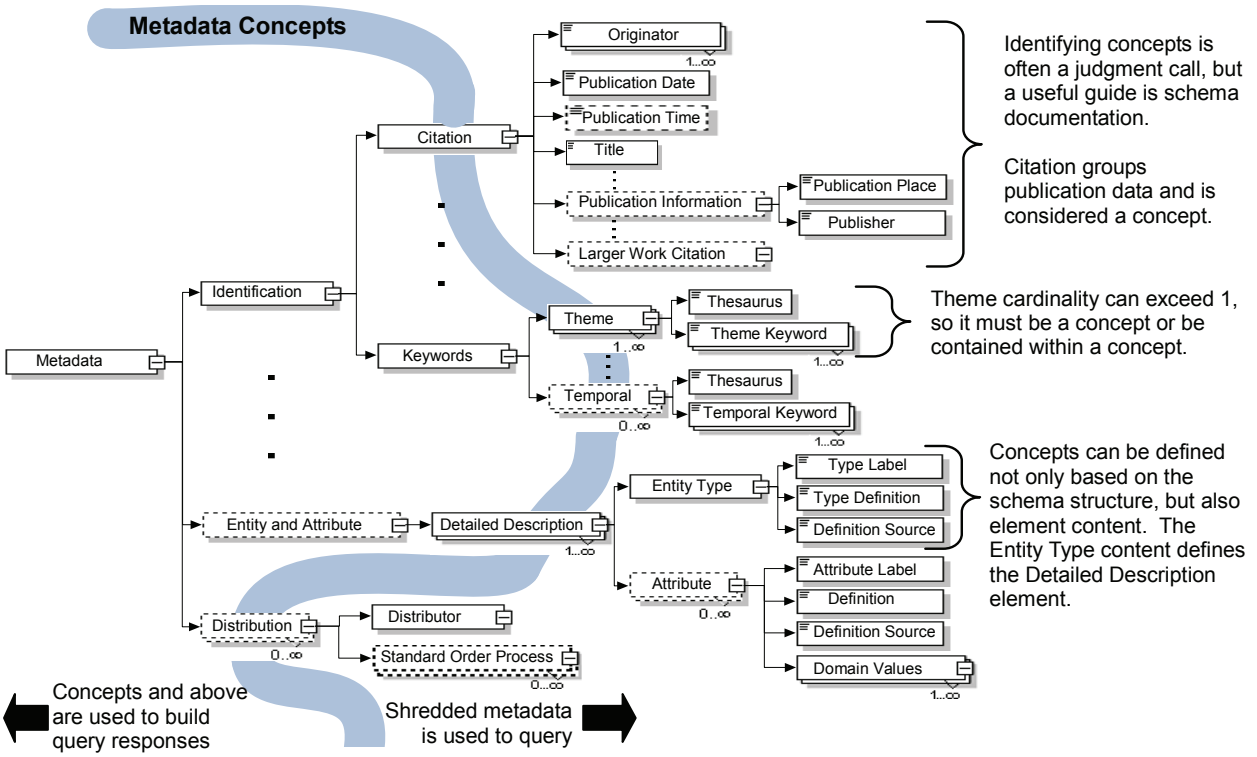

Fig. 1. Partitioning a metadata schema based on concepts

work is that across a broad spectrum of scientific communities, data sets are described using community tailored XML schemas, and while the semantics of each domain's schema may differ widely, there are structural commonalities to these schemas that can be exploited for efficient storage, while still retaining the system's ability to respond rapidly to queries and handle updates to existing metadata. Specifically, we propose extensions to the repository data model and implementation that will enable a scientist to mark up a community schema, then use the annotated schema as input to a tool that configures the repository for the community schema.

The remainder of this paper is organized as follows: Section 2 discusses the characteristics of scientific metadata schemas and Section 3 discusses the storing of scientific metadata. In Section 4 we provide a brief overview of the framework for our metadata catalog and in Section 5 we present our approach to a domain-independent metadata catalog. Section 6 discusses related work and Section 7 concludes with future work.

2. Identifying Characteristics of Scientific Metadata

In scientific communities as varied as marine sciences, meteorology, astronomy, and social sciences, [16,17,20,22] data sets are described using metadata that is communicated via XML schemas tailored to that scientific community. An early standard for geospatial data was published by the Federal Geographic Data Committee (FGDC) and mandated for the spatial data of U.S. government agencies and used by the NSDI clearinghouse – a cross-domain archive of geospatial data [8]. The LEAD Metadata Schema (LMS) used in the LEAD CI [16] to communicate metadata about data products is a profile of the FGDC schema. The LMS is based on the FGDC standard in large part to facilitate sharing of generated data products. Recently, a number of scientific domains have focused on schemas that are profiles of the ISO 19115 geospatial metadata standard (which is implemented as an XML schema in the ISO 19139 standard) [2,12,22].

A review of the FGDC and ISO 19115/19139 metadata schemas reveals common characteristics that apply across scientific domains. The most significant is that ***scientific XML schemas are composed of independent higher level concepts***. In the FGDC, these high level concepts are referred to as sections. Examples include identification, data quality, and distribution. Each section has one or more compound elements (which are complex types in the XML schema) and the compound elements can be composed of either compound elements or data elements, (where data elements are leaf elements in the XML schema). Figure 1 is a fragment of the FGDC schema showing

three of the seven major sections: (1) Identification, (2) Entity and Attribute, and (3) Distribution. An example of a compound element is the Theme element which contains two data elements: Thesaurus and Theme Keyword. This composition of the FGDC metadata schema as a set of independent concepts is not unique to the FGDC in that metadata is "data about data" [8].

The ISO 19115 metadata standard, profiles of which are being created by varied scientific domains or governmental units, is composed of 14 top-level "packages" which are similar to FGDC sections; including packages for concepts such as identification, data quality, and distribution. Although the FGDC and ISO 19115 are both metadata standards for geospatial data (and are undergoing a harmonization effort), this pattern of independent concepts is also found in metadata schemas from other scientific domains as diverse as astronomy and the social sciences [17,20].

In astronomy, the International Virtual Observatory Alliance (IVOA) has developed the VOResource schema that can be used to describe data collections or other objects [17]. The VOResource describes metadata in language similar to the FGDC, and defines core metadata as being in one of four categories: identity, curation, quality, and general content [17]. The VOResource schema differs from the FGDC and ISO 19115 in that instead of defining a root element that is then used to define a data object, it is expected that schemas based on the VOResource schema will extend the resource type (which includes the core metadata) using other global types defined in the VOResource schema. Although the implementation approach used by the IVOA differs from the FGDC and ISO, the structure of the metadata schema as a set of concepts is similar. In the social sciences, the Data Document Initiative (DDI) is a metadata specification for social science data resources [20]. Although the domain differs from FGDC or ISO 19115, DDI also partitions metadata into sections with each section composed of subsections.

While the semantic content of the metadata cataloged by each of these three approaches differs substantially, their composition of independent high-level concepts which are in turn composed of sub-concepts and metadata elements is what differentiates them from other data that is expressed in XML. XML is often used to exchange data between systems as part of a transaction – examples are the standards developed by the Transaction Workflow Innovation Standards Team (TWIST) for financial transactions and supply-chains [30]. The Bank Services Billing (BSB) standard from TWIST is used for financial transaction billing statements, and although the standard is comprised of sections, these sections are not independent as in the case of metadata schemas, but are closely interrelated.

Our position is that we can exploit the general observation that each section of a metadata schema describes a different aspect or facet of a data product with no dependencies on other sections of the schema. The spatial domain section of the FGDC metadata schema describes the spatial domain of the data – not the spatial domain of the organization that generated or published the data. If the location of the publisher is relevant in a particular domain, such information is included as a sub-concept within the citation section. This observation can also influence the types of queries issued against a metadata catalog. Metadata queries search for data products, collections of products, or collections of collections (i.e., experiments) that have properties such as spatial or temporal coverage, data format, or model configuration parameters. The typical query response for a metadata catalog is the full metadata for a data product. We found though that users frequently only need a subset of the full metadata document, and this was largely some number of distinct concepts. Responses to such queries differ fundamentally from queries executed directly over data in that queries for metadata do not result in the creation of new XML documents but instead return a subset of concepts from the metadata schema.

2.1 Identifying Metadata Concepts

How does a scientific community identify the concepts in its schema? Determining which schema elements represent concepts can be guided by rules based on the structure of the schema:

1. Every leaf element in a document conforming to the schema must be contained within a concept.
2. Any element that might have cardinality greater than one (e.g., the maxOccurs attribute for the element is set to "unbounded"), must either be a concept or be contained within a concept.
3. Schema extensions are concepts but are often defined based on content instead of structure.

The first rule is obvious in that it requires that any element containing actual data and not just providing structure must be contained within a concept. In Figure 1, the Title element contained within the Citation element must either be a concept or be contained within a concept since it contains a string representing the title of the data product. The Citation element itself does not directly contain data but instead is a complex type that provides structure – grouping together all of the elements that represent "the recommended reference to be used for the data set" [8]. Since the Citation element is not a leaf

element, it is not required to be contained within a concept based on the first rule – it could be above the concept level in the schema. Under the second rule, elements such as the Theme element must either be a concept or be contained within a concept.

While these two rules could be applied automatically to push the concept definitions as low in the schema hierarchy as possible, such an application of these rules would result in losing the benefit of grouping related elements into concepts. Under such an approach, in Figure 1, the "Publication Place" and "Publisher" elements contained within the "Publication Information" element under Citation would be separate concepts since no parent element on the path to the root Metadata element can have cardinality greater than 1.

Metadata standards such as the FGDC and ISO 19115 provide extensive documentation as to their structure through detailed UML diagrams and descriptions of the elements; such documentation can be useful in making judgments as to which elements in a metadata schema are concepts. When implementing the profile of the FGDC schema used in the myLEAD metadata catalog, for the example in Figure 1, the Citation, Theme, Temporal, Detailed, and Distribution elements were identified as concepts. A cursory review of the element names within the Citation element identifies these elements as belonging to the same concept – the time and date of publishing are clearly not separate concepts. According to FGDC, the Theme and Temporal elements respectively represent "subjects covered by the data set" and "time periods characterized by the data set". Based on the descriptions, these elements are concepts. Additionally, the concepts could not be defined lower in the schema hierarchy due to the cardinality of these elements. Likewise, the Distribution element is a concept due to the second rule based on the possible cardinality of the element. The Detailed element is a special type of element in the FGDC schema that makes it extensible.

In FGDC, the Detailed element is used to represent entities included in the data set and attributes about such entities. A schema that includes spatial data might also include entities such as roads that have attributes such as the road type. In LEAD, the Detailed element is used extensively to capture model configuration parameters and key notifications regarding stages in an experiment's workflow. Such metadata generated during workflow execution is critical to data reuse and the repeatability of an experiment, but is not captured directly in the structure of metadata standards.

As shown in Figure 1, the Detailed element in the schema is a concept, but it differs from the other concept elements in that the definition of the concept is captured in the data of the element itself – not in the structure of the schema. The concept definition in the metadata catalog for most elements identified as concepts is based on the element tag and the namespace of the element. For the detailed element, the definition is based on the "Entity Type Label" and "Entity Type Definition Source" element values.

3. STORING SCIENTIFIC METADATA

Metadata catalogs designed to be applicable across scientific domains such as the ICAT [21] use a relational database backend that can be extended to handle domain-specific metadata through name/value pairs (name/value/unit triples in ICAT) [21]. To communicate metadata using a community XML schema, scientific grids implementing a generic metadata catalog must handle the conversion of metadata between a schema-valid representation and the name/value extensions used in the metadata catalog. We illustrate this conversion with a short example.

Using the Theme keyword element from the FGDC as an example (see Figure 1), each Theme element contains: (1) a *Thesaurus* child element that identifies the controlled vocabulary, and (2) one or more *Theme Keyword* elements that contain a keyword from that controlled vocabulary. Possible values from the Climate Forecasting [7] controlled vocabulary include:

Thesaurus = CF-1.0
ThemeKeyword = soil_temperature
ThemeKeyword = air_pressure

If each element in this example were stored as a separate name/value pair, then the semantics of the theme keywords as defined by the CF-1.0 controlled vocabulary would be lost. Two alternatives that retain the information are:

1. Flatten the XML schema structure, resulting in two Theme entries:
   Name = Theme
   Value = CF-1.0:soil_temperature
   Name = Theme
   Value = CF-1.0: air_pressure
2. Include in the name a local ordering and path from the subtree rooted at the Theme element (since Theme is not used elsewhere in the schema):
   Name = "1:Theme/Thesaurus", Value = CF-1.0
   Name = "1:Theme/1:ThemeKeyword"
   Value = soil_temperature
   Name = "1:Theme/2:ThemeKeyword"
   Value = air_pressure

As the complexity of the element increases, collapsing the structure of the XML metadata document into the name field of a name/value pair becomes unwieldy. The Distribution element shown in Figure 1 contains

child elements within a hierarchy that has 5 to 6 levels where cardinality exceeds 1, so each level would need to be included in the name field of the name/value pair along with a local ordering of the elements at each level.

With the complexity of science schema profiles, reconstructing XML documents from name/value pairs is not efficient. The authors of the Globus MCS catalog (which also uses name/value pairs and a RDBMS backend), noted that the storage and reconstruction of XML was found to be a bottleneck to performance [25].

To manage metadata communicated by a community XML-schema, metadata catalogs need to be schema-aware and manage the storage of schema-valid metadata, handle the addition of annotations and additional metadata for existing data products, and reconstruct schema-valid metadata (including new metadata and annotations) in response to queries. Research to date on storing XML in relational databases has focused on general solutions to storing and querying XML documents.

4. METADATA CATALOG FRAMEWORK

There are two approaches to storing XML in a relational database [6]:

1. The XML document can be stored as a single character large object (CLOB).
2. A "composed" representation can be used in which the individual data items in the leaf elements of the XML document are stored in relational tables.

The CLOB approach works well for applications where whole documents are returned in response to queries, queries are not based on the contents of the XML documents, and documents are only archived and not updated [6,19]. Since metadata catalogs must allow scientists to query over the metadata, add annotations to existing documents, and insert additional metadata as experiments execute, a metadata catalog that stores only CLOBs does not address the needs of any grid community.

When a composed representation is used, the XML is parsed using a process known as "shredding" and the subsequent reconstruction of the XML in response to queries is known as "XML publishing" [19,23]. Multiple approaches have been proposed for shredding XML [6,9,24,28,29], but one of the most common approaches to shredding (for schema-based XML as in the case of a metadata catalog) is known as inlining [24] and one of the most efficient [6] approaches to XML publishing is the sorted outer union approach described in [23].

With the inlining approach, XML documents are shredded and leaf elements are stored in relational tables. Starting at the root element of the schema, as many child elements as possible are included (inlined) in that same relation. Additional tables are required for recursive elements and elements where cardinality can exceed one. Researchers have shown that inlining can be optimized based on the expected XML documents (e.g., the expected cardinality of elements and the frequency with which optional elements are populated), and the frequency of expected query patterns [4,5]. Inlining poses two significant problems when applied to managing scientific metadata:

1. Inlining exhibits a tight coupling of the XML and relational schemas [27].
2. Optimizations based on data and query patterns are not only schema specific, but specific to a VO's grid. Such optimizations also require continued monitoring and tuning as data patterns change (e.g., how many keywords a product has on average).

A third alternative that we proposed in [14] is a hybrid approach that combines shredding the XML with storing CLOBs. For a general XML-relational solution, this approach could result in a significant duplication of data since CLOBs must be stored for subtrees rooted at all elements in the XML schema except for leaf elements (which are stored as shredded data in the relational tables). Additionally, updates to the XML document (both insertions and deletions) would be inefficient and scale poorly since all of the CLOBs on the schema path containing the updated or deleted element must be regenerated. ***But the concept-based characteristic of scientific metadata makes the hybrid approach not only feasible, but efficient for storing and querying scientific metadata.*** Since scientific metadata schemas consist of distinct and independent concepts, and query results are composed from these concepts, CLOBs only need to be stored for those elements in the schema that represent concepts. The leaf elements within each CLOB are also shredded for use in the SQL queries that determine which data products meet the criteria specified in a scientist's query. Although each leaf element must be stored twice – once in a CLOB and again in the relational tables as shredded data, there is not the excessive duplication that arises when concepts cannot be utilized.

For metadata catalogs, query performance is more critical than insert performance. Inserts are primarily system generated as workflows execute and generate new data products or insert additional metadata regarding an experiment as it executes (e.g., model configuration parameters). Since workflows are often resource intensive and long-running, the time required to insert metadata is negligible in contrast to the time required to execute the workflows themselves and archive the data products generated. In contrast,

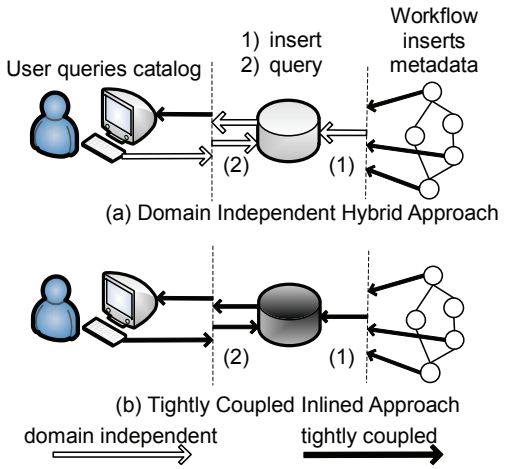

Fig. 2. Comparing XML and relational schema coupling

queries are often executed interactively as a user is browsing or querying their workspace through a portal interface or when a user is configuring a new workflow. Recent work [14] compared the hybrid and inlining approaches based on the workspaces of meteorological researchers using the LEAD gateway:

1. Hybrid query performance is up to 615% better than inlining.
2. Hybrid scales better. At 10 times the projected LEAD workload, hybrid query performance was still better than inlining under the base workload.

5. A Domain Independent Metadata Catalog

Based on our hybrid approach to storing XML, a metadata catalog can efficiently communicate via the XML schema specific to a scientific domain while at the same time having a science domain-independent architecture. The domain independence under the hybrid approach applies not only to parsing and storing metadata, but also to a query interface that can adjust automatically to the schema being used.

Going back to Figure 1, a line is drawn through all of the concept elements in the metadata schema. This line partitions the schema – on the left (towards the root of the schema) is the portion of the schema that responds to queries. For these items, a CLOB is stored for each of the concepts. Query responses are built from the CLOBs and the higher elements in the schema that are on the left side of this partition.

On the right, each concept is shredded. However, the shredded metadata is not used to reconstruct XML in response to queries as is done in the inlining approach but is only used to determine which data products satisfy a user's query. Eliminating the need to reconstruct XML from the shredded metadata eliminates the need for a tight coupling between the XML and relational schemas.

Figure 2 illustrates the gains that can be had from viewing metadata schemas as decompositions of concepts. Under any approach that communicates via schema-valid XML, both the metadata inserted and the metadata returned in response to queries are expressed in the community schema (otherwise it would not be schema valid). However, under the hybrid approach, the relational schema is independent of the community schema and the structure of the XML schema is instead captured as data. As metadata is added, the operations as shown in Figure 2(a) are domain independent since the tables use a domain independent schema. Additionally, although the catalog service returns a schema-valid result, the SQL queries issued for a user are also domain independent since all of the data needed to build the response is contained in the schema mapping table and the CLOBs stored for each concept. This is unlike Figure 2(b) where the domain-specific nature propagates down to the database, limiting the tool's portability to other scientific domains.

The concept-based nature of scientific metadata and a hybrid approach provide efficient metadata management that would not be feasible for general XML content. However, science and scientific metadata schemas are a growth area, so the class of applications to which this approach applies is large.

5.1 A Domain Independent Query Interface

The domain independence of the concept/hybrid approach provides benefits that can help the user's interaction with the catalog service when specifying a catalog search:

1. Concept and element definitions can be used to present a dynamically constructed domain independent portal interface.
2. From a user's perspective, schema extensions are indistinguishable from concepts defined in the structure of the schema.

When a scientist issues a search, the shredded metadata is queried to determine the data products matching the search criteria the scientist specifies. The definitions for the concepts and elements contained in the schema are stored as data and not part of the database structure. In the LEAD portal, when a user calls up the search interface, the database is queried for the concept and element definitions and the query interface is dynamically created based on the definitions returned. When the catalog service is loaded with the concept and element definitions of a different domain's schema, the query interface is automatically built based on the new definitions.

The approach supports attribute extensibility too.

In the myLEAD metadata catalog, namelist parameters from the FORTRAN files used to configure the models are stored as metadata using the Detailed element. Under a tightly coupled approach such as inlining, the relational database would contain a table for Detailed elements which has columns for the entity label, definition, and source. A user searching for experiments that used a grid spacing of less than 5km would need to specify the name of the specific parameter they are searching for (dx) as the "Type Label" and the namespace of the model as the "Definition Source". Searching for extension elements based on the schema structure requires the user to know both the schema structure and details of the model parameters they are searching for.

6. Related Work

SRB's MCAT [18] and the Globus MCS metadata catalog [25], are applicable across scientific domains, but neither is designed to communicate metadata based on community XML schemas. However, MCS was an early inspiration for our work. iRODS [21] includes the ICAT metadata catalog which is similar to its predecessor MCAT in that it uses a generic approach that can be extended using name/value/unit triples for domain-specific metadata [21].

Beyond scientific metadata, researchers are also looking at managing personal dataspaces [10] – including not only data products, but also sensor data and streaming data. Unlike scientific metadata where there is an awareness that metadata is lost if not captured when it is generated [11], dataspace research emphasizes a "pay as you go" approach where the user adds metadata as they find a need for it [10].

Two recent alternatives to managing scientific data are the Maitri [26] and Data Ring [1] approaches. The Maitri approach differs from ours in that while it aims to manage scientific data in different file formats, it sits between the scientific tools and data libraries. Maitri maintains metadata regarding the data repository (e.g., is there an index over the data a user is looking for) in contrast to metadata regarding the data itself (such as the namelist model configuration data in LEAD). Alternately, Data Ring builds on P2P technology to develop a system for scientists to share data. While it envisions bringing together data from different domains, it brings together varied data formats (mainly file-based) and allow for a declarative query language that is simpler than SQL. Based on our experience with feedback early on in the LEAD project, scientific users may not be willing to tolerate the slow query response times such an approach would entail.

Commercial RDBMS also handle XML to varying degrees [19] including IBM, Oracle, Microsoft SQL Server, and recently to a limited degree – MySQL. Oracle's implementation provides the option of either storing XML as a CLOB or shredding into an object database instead of a relational database [13]. In IBM's DB2 the XML data type was added which can contain an entire XML document. Instead of a CLOB, a tree representation is used with XML tags being assigned an internal integer ID [3]. These solutions aim to provide a general solution to managing and querying XML data – including XML that is well-formed but schema-less. The approach we present in this paper differs in that it is specialized to perform well for the general class of scientific metadata.

7. Conclusion and Future Work

In this paper we present an approach to managing scientific metadata that is communicated via domain science schemas. We identify characteristics that are common in schemas across varied scientific domains. We present an approach that is tailored to scientific schemas, yet retains as one of its underlying principles the flexibility to be deployed using any domain science schema. This approach has been applied in the LEAD grid and is being used on a consistent basis by students and researchers to manage their workspaces and review the results of their experiments.

An aim of our research is to show that an efficient schema-based metadata management approach can be generalized to be easily adaptable across varied scientific domains. The only schema-specific aspects of the hybrid approach are the XSLT template used to shred the metadata and the definitions loaded for the schema mapping, concept definitions, and element definitions. We are developing a software prototype that uses schema annotations to automate the deployment of a customized metadata catalog based on a metadata schema. Identifying metadata concepts will involve a combination of schema structure and judgment calls. Our research is looking at interfaces that could be used to identify the concepts in a schema. We believe viewing scientific schemas as decomposable independent concepts is a powerful notion for supporting domain science catalogs in the future. The decomposable concept notion also provides a rich opportunity for optimization as we have shown with the hybrid approach to metadata storage.

Acknowledgments

We thank the meteorological researchers at the University of Oklahoma and other institutions participating in LEAD for their help.

REFERENCES

[1] S. Abiteboul and N. Polyzotis, "The Data Ring: Community Content Sharing," CIDR 2007, Asilomar, CA, January 2007.

[2] ANZLIC – the Spatial Information Council, "ANZLIC Metadata Profile: An Australian/New Zealand Profile of AS/NZS ISO 19115:2005, Geographic information – Metadata," Draft Version 1.1, August 2007.

[3] K. Beyer, R. Cochrane, M. Hvizdos, V. Josifovski, J. Kleewein, G. Lapis, G. Lohman, R. Lyle, M. Nicola, F. Özcan, H. Pirahesh, N. Seemann, A. Singh, T. Truong, R. C. Van der Linden, B. Vickery, C. Zhang, and G. Zhang, "DB2 goes hybrid: Integrating native XML and XQuery with relational data and SQL", *IBM Systems Journal*, vol. 45, no. 2, pp. 271-298, 2006.

[4] P. Bohannon, J. Freire, P. Roy, and J. Siméon, "From XML Schema to Relations: A Cost-Based Approach to XML Storage," in *Proceedings of the ICDE 2002*, San Jose, CA, 2002.

[5] S. Chaudhuri, Z. Chen, K. Shim, and Y. Wu, "Storing XML (with XSD) in SQL Databases: Interplay of Logical and Physical Designs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1595-1609, 2005.

[6] D. Draper, "Mapping Between XML And Relational Data," in *XQuery from the Experts*, H. Katz, Ed. Boston: Addison Wesley 2004, pp. 309-352.

[7] B. Eaton, J. Gregory, B. Drach, K. Taylor, and S. Hankin, "NetCDF Climate and Forecast (CF) Metadata Conventions", Version 1.0, 2003.

[8] Federal Geographic Data Committee, Washington, D.C., Content Standard for Digital Geospatial Metadata Workbook Version 2.0, 2000.

[9] D. Florescu and D. Kossman, "Storing and Querying XML Data Using an RDBMS," *Bulletin of the IEEE Technical Committee on Data Engineering*, vol. 22, no. 3, pp. 27-34, 1999.

[10] M. Franklin., A. Halevy, and D. Maier, "From Databases to Dataspaces: A New Abstraction for Information Management," *SIGMOD Rec.*, vol. 34, issue 4, pp. 27-33, December 2005.

[11] J. Gray, A. S. Szalay, A. R. Thakar, C. Stoughton, and J. vandenBerg, "Online Scientific Data Curation, Publication, and Archiving", Tech. Rep. MSR-TR-2002-74, Microsoft, 2002.

[12] International Organization for Standardization, Geographic Information – Metadata (ISO19115:2003), 2003.

[13] M. Krishnaprasad, Z. H. Liu, A. Manikutty, J. W. Warner, and V. Arora, "Towards an industrial strength SQL/XML Infrastructure," in *Proceedings of the 21st International Conference on Data Engineering*, April 2005.

[14] S. Jensen and B. Plale, "Using Characteristics of Computational Science Schemas for Workflow Metadata Management", IEEE 2008 Second International Workshop on Scientific Workflows (SWF 2008), in *Proceedings of the 2008 IEEE Congress on Services*, July 2008.

[15] S. Newhouse, J. M. Schopf, A. Richards, and M. P. Atkinson, "Study of user priorities for e-infrastructure for e-research (SUPER)", in *Proceedings of the UK e-Science All Hands Conference*, 2007.

[16] B. Plale, R. Ramachandran, and S. Tanner, "Data Management Support for Adaptive Analysis and Prediction of the Atmosphere in LEAD", 22nd Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology (IIPS), January 2006.

[17] R. Plante, K. Benson, M. Graham, G. Greene, P. Harrison, G. Lemson, T. Linde, G. Rixon, and A. Stebe, "VOResource: an XML Encoding Schema for Resource Metadata", Version 1.02, 2006, at: http://www.ivoa.net/Documents/cover/VOResource-20061107.html

[18] A. Rajasekar, "Managing Metadata in SRB", SRB Workshop, San Diego, CA, February 2-3, 2006, at: http://www.sdsc.edu/srb/Workshop/Talks

[19] M. Rys, D. Chamberlin, and D. Florescu, "XML and Relational Database Management Systems: the Inside Story," in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, New York, June 2005.

[20] J. Ryssevik, "The Data Documentation Initiative (DDI) metadata specification, at: http://www.ddialliance.org

[21] F. Schintke, "iRODs I Rule Oriented Data System", D-Grid All Hands Meeting, Göttingen, September 2007.

[22] SEA-SEARCH European Directory of Marine Environmental Data, at: http://www.sea-search.net/edmed/welcome.html

[23] J. Shanmugasundaram, E. Shekita, R. Barr, M. Carey, B. Lindsay, H. Pirahesh, and B. Reinwald, "Efficiently Publishing Relational Data as XML Documents," *The VLDB Journal*, vol. 10, nos. 2-3, pp.133-154, 2001.

[24] J. Shanmugasundaram, K. Tufte, C. Zhang, G. He, D. J. DeWitt, and J. F. Naughton, "Relational Databases for Querying XML Documents: Limitations and Opportunities," in *Proceedings of the 25th VLDB Conference*, Edinburgh, Scotland, 1999.

[25] G. Singh, S. Bharathi, A. Chervenak, E. Deelman, C. Kesselman, M. Manohar, S. Patil, and L. Pearlman, "A Metadata Catalog Service for Data Intensive Applications" In *Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, Washington, DC, 2003.

[26] R. R. Sinha, S. Mitra, and M. Winslett, "Maitri: Format Independent Data Management for Scientific Data," In *Proceedings of the 3rd International Workshop on Storage Network Architecture and Parallel I/Os (SNAPI)*, 2005.

[27] D. Suciu, "On Database Theory and XML", *SIGMOD Rec.* vol. 30, no. 3, pp. 39-45, 2001.

[28] I. Tatarinov, S. D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang, "Storing and Querying Ordered XML Using a Relational Database System", in *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, June 2002.

[29] F. Tian, D. J. DeWitt, J. Chen, and C. Zhang, "The Design and Performance Evaluation of Alternative XML Storage Strategies," *SIGMOD Rec*, vol. 31, issue 1, pp. 5-10, 2002.

[30] Transaction Workflow Innovation Standards Team (TWIST) at: http://www.twiststandards.org