Pasta: A Network-level Architecture Design for Automating the Creation of Synthetic Products in the LTER Network

Mark Servilla, James Brunt, Inigo San Gil, and Duane Costa
LTER Network Office,
Department of Biology,
University of New Mexico

18 October 2006

Abstract

The LTER Network is now within its "Decade of Synthesis". Providing Network-level synthetic data products, however, is still a challenge for researchers in the Network and its 26 research sites. The Network Information System group at the LTER Network Office has designed and prototyped an automated Network-level synthesis architecture called "Pasta". The Pasta architecture extends the data warehouse notion of extraction and loading of external data into a centralized data store by building on key technology already in use at the LTER Network – primarily the Ecological Metadata Language and the Metacat database. Once loaded, the source or site data are transformed from the local site schema and into a global schema, and a new metadata document cast as EML is generated and inserted back into the Metacat database. Finally, the data that conforms to the global schema, called "synthetic" data, are exposed to the community through a number of different interfaces, including HTML and web services. This architecture is currently being developed through a "proof-of-concept" approach for the "Trends" project, and has recently been demonstrated at the LTER 2006 All Scientists Meeting in Estes Park, Colorado.

1 Introduction

The LTER Network has now been in operation for twenty-six years, and has grown to 26 sites that are distributed throughout North America, Antarc-

tica, Puerto Rico, and the French Polynesia. Funded by the National Science Foundation, the original call for proposals in 1980 specifically addressed the need for comparative analysis of long-term ecological research at sites that represent major biotic regions of North America. Although comparative analysis is performed today, the process by which it happens is cumbersome and often ad-hoc. With the exception of ClimDB and HydroDB[1] [1, 4], synthetic data sets are generated and managed by individual or small groups of investigators, along with their respective information managers, who provide some level of data quality and integrity. A consistent framework for synthesis across all 26 research sites is, however, non-existent.

In early 2005, Dr. Debra Peters of the Jornada LTER proposed the development of the "Trends" project [7] – "a large synthesis effort focused on improving the accessibility and use of long-term data." Participants in the Trends project include the 26 LTER sites, 8 USDA-Agricultural Research Service rangeland sites, 9 USDA Forest Service Experimental Forests, and 1 University of Arizona site. The project utilizes four primary data categories for synthesis: 1)climate and physical variability including disturbances, 2) human population and economy, 3) biogeochemistry, and 4) biotic structure, including biodiversity.

The end result of the Trends project will be a book containing a collection of synthetic data products (and their associated plots) that represent the most significant long-term observational trends within and among the sites. This book will be published by Oxford University Press as part of the LTER book series. An accompanying website will complement the book by providing online access to all of the synthetic data products and their plots.

The major effort of the Trends project is to collect and integrate these time-series data into a format that is consistent between the different data providers. This process generally begins with a data request to individual researchers or information managers at the site. Data are generally returned as Excel spreadsheets or comma delimited text files, often without supporting metadata or information about their origin. This data is then transformed to a synthetic product through a number of processes, which may include converting site-specific units to standard units, performing quality checks and assurance, and mapping the site-specific format to a Trends standard format. Paramount to this process is the creation of metadata that describes every synthetic data product. Each step is time consuming and has the potential for error due to the repetitive human interaction during data

[1] Both ClimDB and HydroDB are Network-wide synthetic data products for site climatological and hydrological data.

manipulation.

The ability to add new time-series data collected from ongoing experiments into the synthesis process, however, is missing from the original project goals. The LTER Information Management Executive Committee and Network Information System Advisory Committee recognized this deficiency and recommended the development of a prototype system to automate the synthesis process, and more specifically, to automate the integration of new time-series data. The following paper proposes both an architecture for generating synthetic data products within the LTER Network, as well as describes a prototype implementation as applied to the Trends project. This architecture is known by the moniker "Pasta"[2], and is being developed by the Network Information System group at the LTER Network Office.

2 Background

Each LTER site independently collects, documents, and archives their data for analysis and publication. In most cases, these data are made available for community scrutiny and further analysis after 2 years from the date of collection. Today, almost all data collected are documented by using the Ecological Metadata Language (EML), an XML-based data structure for describing scientific data that can be validated by using a community published XML schema. These EML documents are then uploaded into the LTER's data catalog, Metacat [2, 5], a schema-independent XML database that is optimized for storing and retrieving EML documents[3]. The LTER Metacat contains just over 25,000 EML documents as of October 2006.

Although not required, LTER sites comply with an EML document naming-convention [8] that is comprised of a *scope*, a unique *identifier*, and a *revision* number, joined together to form the string `SCOPE.ID.REVISION`. This name is referred to as the *Document ID* and provides a unique reference to the EML document within the Metacat database. The document scope is a string value that is the concatenation of the site's three letter acronym to the end of the broader domain identifier of "knb-lter", thus forming a complete scope hierarchy (e.g., `knb-lter-sev` is the scope associated with

[2]The name "Pasta" originates from a passing comment made regarding the original "Phase II" project title. The comment, "*Phase II* is about as informative as *Spaghetti and Linguini*", resulted in a more tangible noun, "pasta", but without any more meaning. It has since been accepted as our internal architecture moniker.

[3]Refer to http://www.ecoinformatics.org for more information on the EML specification and Metacat.

EML documents from the Sevilleta LTER, which is part of the LTER Network and part of the informatics research funded through the Knowledge Network for Biocomplexity (KNB)[4]). Both the identifier and the revision number are integer values. The identifier need only be unique to the site, while the revision number marks the version of the EML document being reviewed. A higher revision number is added to the Document ID if a change or modification occurs to either the data, the metadata, or both as documented in the EML. This number must be an increasing and ordered value for all documents using the same scope and identifier.

To simplify and automate the upload process of EML documents to the Metacat, the Harvester application was developed as part of the KNB project. The Metacat Harvester allows individuals to register *http accessible* EML documents into a database table, along with the frequency of which the set of documents should be checked for updates. If the version of the EML document found at the site is more recent then the version found in the Metacat database, it is automatically copied from the site and inserted into the Metacat database. To date, the LTER Metacat Harvester application automatically checks for updates of more than 5,000 EML documents from 24 sites.

Both EML and Metacat are fundamental components of the Pasta architecture, providing a metadata standard capable of fully describing synthetic data products and a database system optimized for storing and retrieving such metadata.

3 Architecture

The Pasta architecture is based on a hybrid data warehouse model (see discussion in Section 4.1), which collects and organizes distributed data into an integrated data store. The following section illustrates the Pasta architecture through a dissection of its work-flow process, from the data source at the site to the end storefront where "synthetic" data products are made available to the community. The work-flow allows us to partition the architecture into individual modules that can be succinctly described performing their unique task. In fact, each module is designed to be independent of the others, and can be replaced with a new module that provides the same functionality without impacting the entire system.

For clarity, we define data that is extracted from the site as "source" data. The physical structure of how source data is represented at the site, as

[4]Funded by the National Science Foundation under Grant No. DEB99-80154.

defined in the site EML document, is the "local" schema. The local schema is replicated in the Source Database. Data that is produced by transforming source data is defined as "synthetic" data. The physical structure used to store synthetic data is the "global" schema. The global schema makes up the framework of the Synthetic Database.

3.1 The Work-flow

The Pasta work-flow can be characterized by dividing the architecture (Figure 1) into six separate steps, each of which can operate independent of one another, but as a whole perform the goal of generating synthetic data products[5]. The process begins at the individual research site and ends at the storefront that exposes the synthetic data to the community.

The work-flow steps are:

1. Site data collection, quality checking, and EML compliant metadata generation.

2. Harvesting new or revised site EML by the Metacat Harvester and inserting into the Metacat.

3. Recognizing new or revised EML documents that are specifically registered as part of the synthesis process. These EML documents are then parsed into their entity and attribute objects, which are then used to create a new database table in the Source Database. Finally, the data is extracted from the site and loaded into the new table.

4. The source data are then transformed from their local schema (captured by the Source Database table) and into the global schema as a new data product in the Synthetic Database.

5. The synthetic data product is documented with new or revised metadata, which is then inserted into the Metacat database as an EML document.

6. The synthetic data product is then made available for the consumer through a warehouse storefront, either a web-browser or web services interface, which accesses the Synthetic Database.

[5]It should be empahsized that this work-flow, and the operation of the Pasta architecture, depends on metadata that succinctly describes entity and attribute objects. Not all EML today meet this goal.

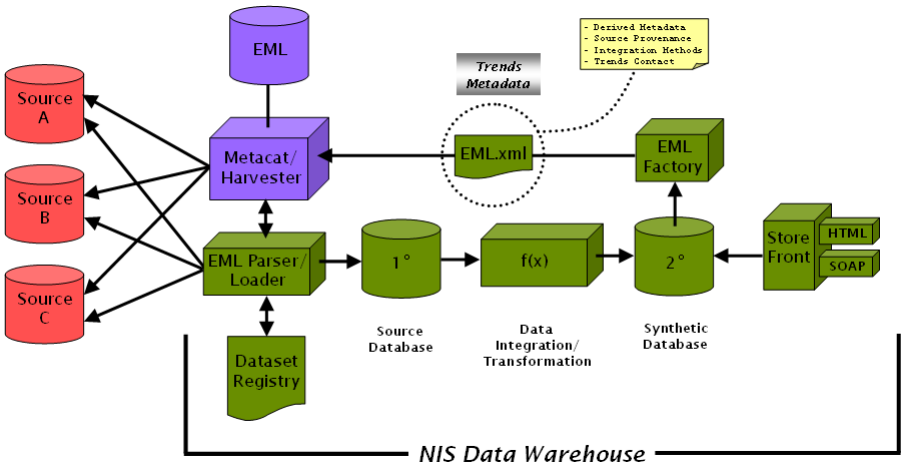

Figure 1: Major components of the Pasta architecture.

3.2 Steps 1-2: Site Data Management and EML Harvesting

Although steps 1 and 2 of the above work-flow are not technically part of the Pasta architecture, they are significant modules, and their absence would prohibit the execution of the complete work-flow. For this reason, they are described as modules of the architecture.

As described earlier, LTER sites document their data by expressing their metadata content in the Ecological Metadata Language. A revised version of the EML document is created if, for example, new data is collected and added to the package (as with time-series data), data was modified to correct for errors, or if any metadata content has changed (e.g., the collection end-date or contact information has been revised). Changes to the EML revision number in the Document ID will cause the Metacat Harvester to copy the EML document from the site and insert it into the Metacat database. Older versions of the same document are deprecated from the active system, but are never deleted completely. Such deprecated versions become an integral part of the overall package provenance. This process, represented in Figure 2, is ongoing within the LTER Network and operates independent of the Pasta architecture.

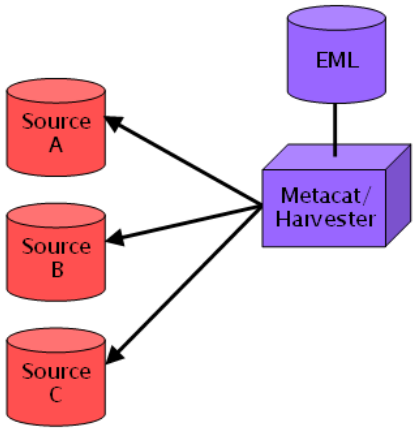

Figure 2: Work-flow steps 1 and 2 corresponding to architecture components for Site Data Management and EML Harvesting.

3.3 Step 3: Extraction and Loading

The third work-flow step includes the Dataset Registry, EML Parser, and Loader components (Figure 3) of the Pasta architecture. This module is responsible for identifying, parsing, and loading site data into the Source Database.

3.3.1 Dataset Registry

The Dataset Registry is used to register source data that are used during the synthesis process. Technically, it is the EML Document ID that is recorded in the registry, along with information relating source data to synthetic data. Specifically, the registry identifies what source data are used to create a synthetic product and the appropriate routine that performs the transformation. The current implementation of the Dataset Registry uses PostgreSQL as its relational database.

3.3.2 EML Parser

The EML Parser is a Java-based library that reads and parses EML documents that are specified in the Dataset Registry. There are two critical functions of the EML Parser: 1) to create a new table in the Source Database that represents the local schema defined in the EML document and 2) to

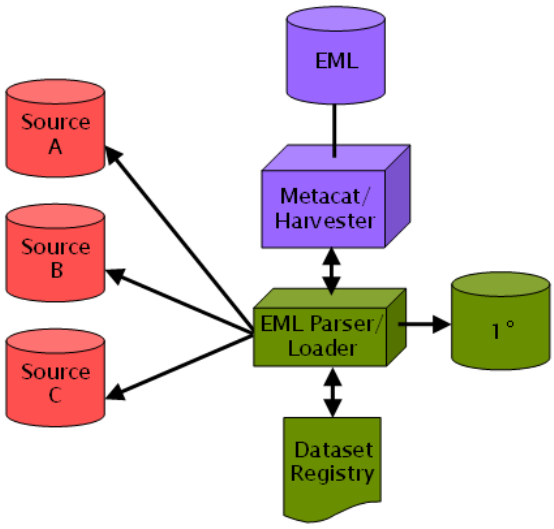

Figure 3: Work-flow step 3 corresponding to architecture components for identifying, parsing, and loading site data into the Source Database (1°) .

open a connection to the site's data store and copy the referenced data into the new table. These two steps depend on correct metadata in the EML document that describe the entity and attribute objects being created, and an unfettered network connection to the data. Development of this library is a collaboration between the LTER Network Office and the National Center for Ecological Analysis and Synthesis. The library, when complete, will become part of the Ecological Metadata Language software distribution.

3.3.3 Loader

The third component of this module, also a Java application, is the EML Loader. The Loader acts as the controller process between the Parser and Dataset Registry. Its primary responsibility is to read the Dataset Registry and compare the revision number of EML documents in the registry to those in Metacat. If the revision number in Metacat is greater than the one found in the registry, the Loader will assert a new extraction and loading sequence by calling the appropriate functions in the Parser. In this prototype, the Loader is also responsible for invoking specific transformation routines of the Transformation Engine and calling the EML Factory to generate a new EML document for the synthetic data. These two steps occur immediately

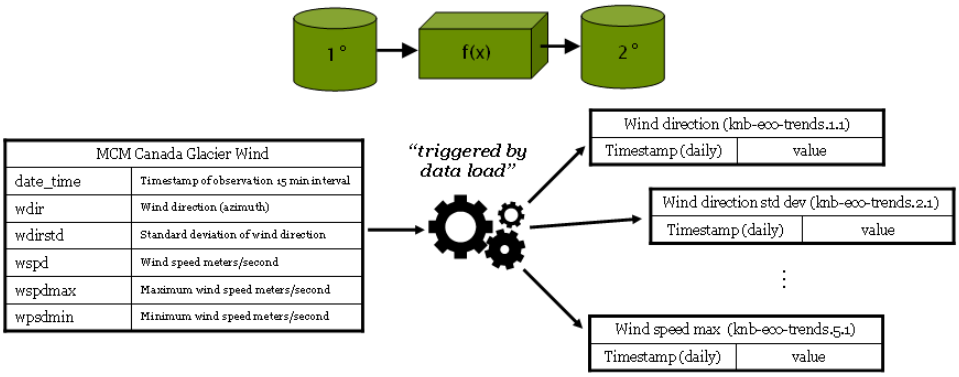

Figure 4: Work-flow step 4 captures the transformation of the local schema in the Source Database (1 °) to the global schema in the Synthetic Database (2 °) through use a Transformation Engine ($f(x)$).

after loading new data into the Source Database and are only necessary to continue the work-flow; they are not critical to the successful operation of this module. This module is still under extensive development and testing.

3.4 Step 4: Transformation

Transformation is the fourth step of the work-flow and defines the "Data Integration and Transformation" module (Figure 4) of the architecture. This module performs the mapping of data from the local schema, as represented in the Source Database, to the global schema through the use of a Transformation Engine - a collection of programs that perform the mapping operation (labeled as $f(x)$ in Figure 4). We assume that individual transformation routines are *a priori* aware of both schemas, including input and output data types, and their semantic meaning, thereby eliminating the need for more advanced knowledge-based reasoning algorithms.

3.4.1 The Global Schema and Synthetic Database

For the Trends prototype, we have constrained our working data model to point-location time-series data, such as climate observation data (e.g., a fixed sensor at a known geographic latitude and longitude that collects temperature values every 30 minutes). Such data is common within ecological observatories and provides a simple model to design and evaluate

the architecture. As such, the global schema data table consists of only two important attributes – the observation *time stamp* and the *value* being recorded[6].

Mapping from the local schema to the global schema consists of three basic steps. First, tables that are composed of multiple attributes are separated into individual tables containing only the time stamp of the observation and the observation value of interest. This step is illustrated in Figure 4 where a local schema table describing weather data from the McMurdo LTER is separated into product specific tables of the global schema. Second, observation values are converted to the appropriate standard unit. This step may include data type conversion (e.g., integer to float) or an actual conversion from a non-standard unit (e.g., Fahrenheit) to a standard unit (e.g., Celsius). Third, the time stamp and observation value are scaled to comply with the time-scale defined for the synthetic data product (e.g., hourly to daily or monthly). Further processing, such as combining two or more source data into an aggregate product, may take place at this point.

It is important to note that execution of the Transformation Engine for synthesis of new or revised data results in a new data table being created in the Synthetic Database. For these tables, we follow the same document naming convention used for EML – that is, `SCOPE.ID.REVISION`. To manage the synthetic data tables, the global schema uses two additional support tables - the "product" table and "revision" table (Figure 5). The "product" table contains scope and identifier information for the synthetic products and the "revision" table records the lineage of product revisions. With these two support tables, any synthetic data product may be identified and accessed, including those that have since been deprecated by a new revision. For the Trends project, the table name scope is `knb-eco-trends`[7].

Additional tables may be added to the global schema to support ancillary functions of the system, including to provide content for the EML metadata documents. The Trends prototype has tables to hold information about the station location (where data is being collected) and biotic relationships between individual synthetic products, neither of which are critical to the operation of the system.

3.4.2 Transformation Engine

The Transformation Engine is simply the software code that performs the mapping between data in the local schema to data in the global schema (i.e.,

[6]There are, of course, additional attributes allocated for data table management.

[7]This scope string is only a place-holder for testing purposes.

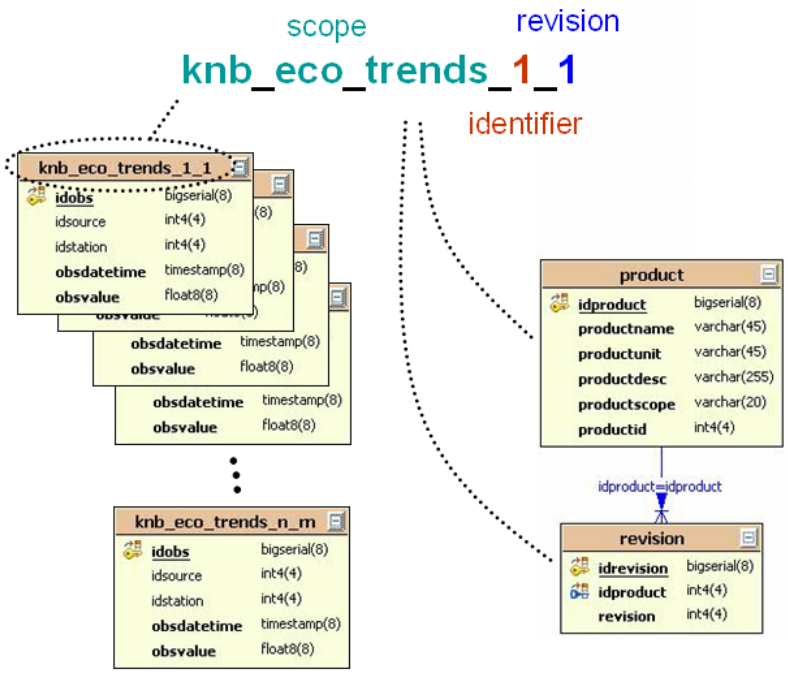

Figure 5: Global schema entity relationship between synthetic data product tables and support tables.

the synthesis process). It is noteworthy that transformation routines have the flexibility of being written in any programming language that support reading and writing relational database tables.[8] In fact, individual routines can be written in different languages. Our current implementation utilizes the "R" environment for statistical computing, and has only been tested in a few examples. In this case, individual R programs are invoked from standard shell scripts that are referenced from the Dataset Registry. The R language provides database support, including ODBC and SQL operations, both of which work well with the PostgreSQL RDBMS version that we use for the Source and Synthetic Database.

3.5 Step 5: EML Factory

The EML Factory module (Figure 6) of work-flow step 5 supports the metadata documentation process for all synthetic data products. Planned as a Java-based application, the following section describes how the EML Factory is envisioned to operate.

The EML Factory would be invoked as a side-effect of creating a new synthetic data product. The current design uses the EML Loader component as the calling process. An alternative is to have a database trigger invoke the EML Factory module, but this capability is not available in all relational database applications. Yet, another possibility is to have each transformation routine execute the EML Factory directly after completing its processing.

Content for metadata is divided among four physical entities: 1) plain text files storing static content that changes infrequently, such as project management information or the LTER Data Access Agreement; 2) database tables of the global schema storing dynamic content that is directly associated with the synthetic data product, including date ranges, data statistics, and online access information; 3) programming code that is used to create the synthetic data product; and 4) the EML metadata of the source data product.

The EML Factory would use a predefined XML schema template to generate the EML document fields. It would then use metadata content from the four entities described above to complete the EML document. Static metadata from the plain text files and the dynamic metadata from the database would be placed in the canonical fields defined for such data in the EML specification. Both the programming code for the transformation routine

[8] In the current prototype, however, they must also be callable by the EML Loader component.

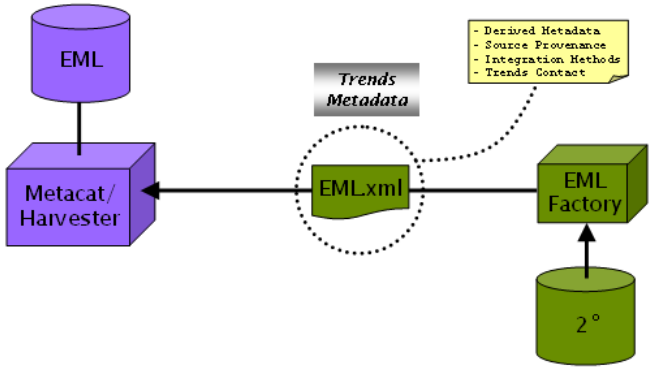

Figure 6: EML Factory module of work-flow step 5.

and metadata content from the source data EML document would be stored in fields of the "methods" subtree (Figure 7). Specifically, a "methodStep" element would be used to capture both types of metadata. In the case of the source data's EML, minimal content must be copied directly from the original document to the required fields within the "methodStep/dataSource" element, but the remainder (which, in most cases, is the complex metadata) can simply be referenced by placing the appropriate Metacat URL for the EML document in the "online" element of the same subtree. The final EML document would then be inserted into the Metacat for access through either the Storefront or the standard Metacat interfaces.

3.6 Step 6: Storefront

The final step of the work-flow is the "Storefront". The Storefront module of the architecture defines the community access point to the Synthetic Database and its content. The core table structure of the Synthetic Database, including the product, revision, and data tables, forms the primary scope of interaction to the synthetic data. Ancillary tables may be accessed to develop a well rounded human-interface that provides a variety of query tools to interact with both the data and the metadata stored in the Synthetic Database and in the Metacat as EML.

The current interface of the Trends prototype consists of a website that connects to the Synthetic Database for data access and the Metacat for displaying the corresponding EML document in its native XML format. The website provides a basic query interface that allows the user to select one or

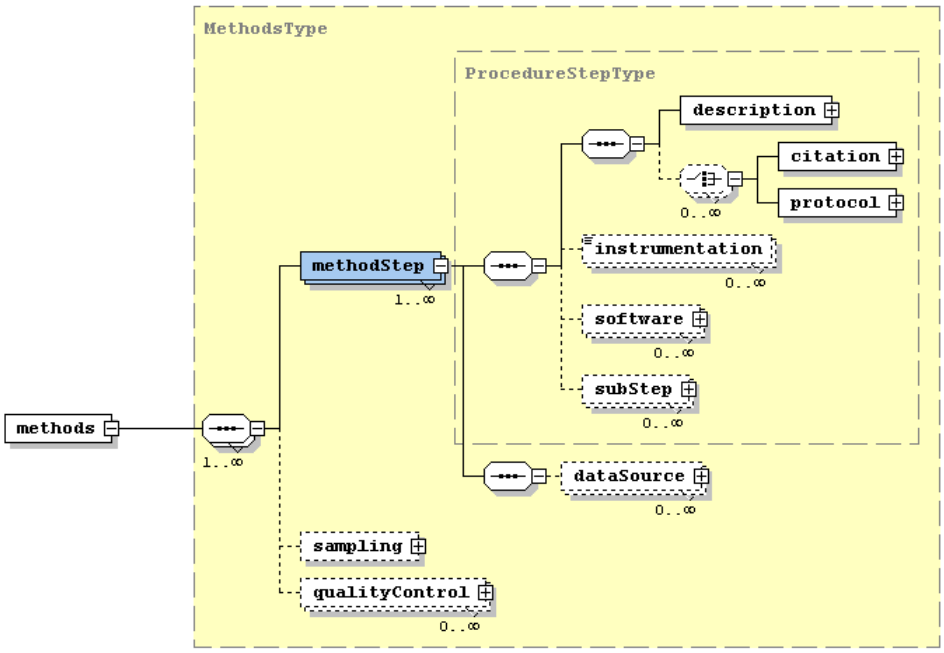

Figure 7: The "methods" subtree of EML 2.0.1 as viewed in XMLSpy 2005.

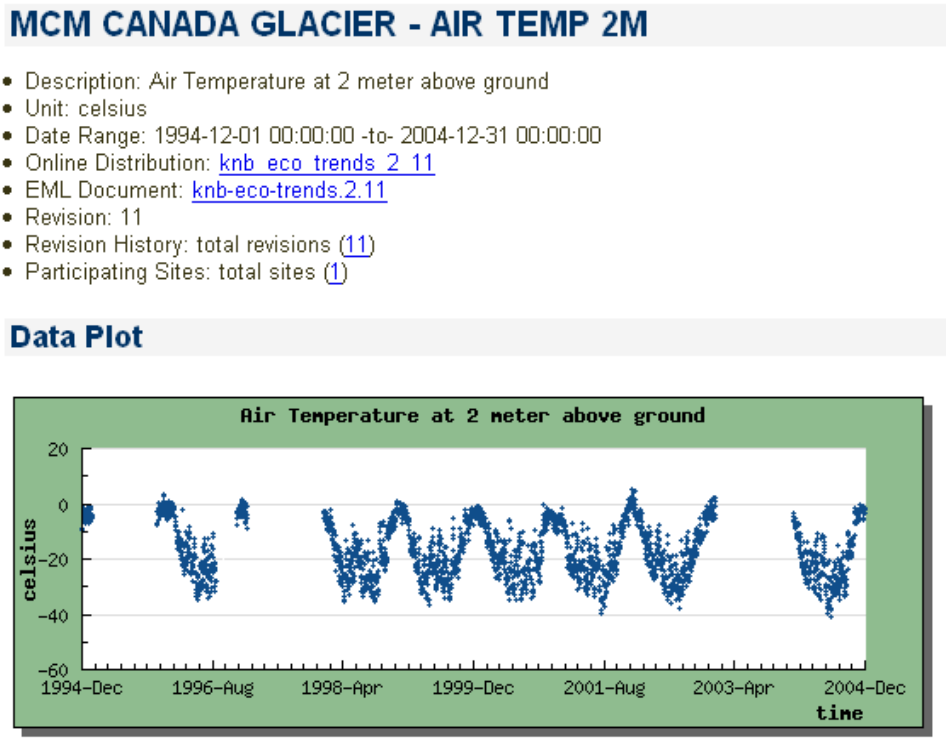

Figure 8: Example screen shot of work-flow step 6, a synthetic data product with plot.

more of the synthetic data products for viewing. Once selected, the website displays a brief summary of metadata, along with a dynamically generated plot of the data (Figure 8). The metadata includes links that lead to site information, a data download page to directly access the synthetic data, and a product lineage list that displays all revisions of the selected product.

The website utilizes the PHP server-side scripting module for the Apache web server to provide programmable logic (i.e., database connectivity and program execution control) to the overall web site. The plotting application, which is an add-on PHP library called "JpGraph", is tightly-coupled to the web server. We also explored the use of more loosely-coupled packages that reside outside of the web server scope - specifically, we tested the efficacy of calling an "R" script from PHP to generate the plots. Although this approach seemed to work reasonably well, due to time constraints, we opted

for using the JpGraph library for the prototype.

Our plans include the development of a web-service interface over Simple Object Access Protocol for direct access to synthetic data and metadata. This aspect of the project is still under early development.

4 Discussion

Although the Pasta architecture is still in the design and prototype phase, its current blue-print provides a solid foundation for meeting the goal of generating Network-level synthetic data products. Its overall design incorporates a number of salient features, including 1) the use of data warehousing strategies for data integration and archiving, 2) providing support for tracking data lineages and provenance, 3) a design that easily integrates new replacement modules with little or no impact on other components of the system, and 4) a simple and compact global schema that leverages already proven informatics tools used within the LTER Network and broader community.

4.1 A Data Warehouse Approach

Data warehouses are focused around a centralized store of information that conforms to a physical global schema [6, 10]. Data from outside of this store must be mapped from their local schema to the global schema through a transformation process. It is common within a data warehouse to use the phrase "extraction, transformation, and loading", or by its acronym ETL, to describe the process in which data external to the central store is "extracted" from the source location, "transformed" to the global schema, and "loaded" into the final store database. The Pasta architecture, however, utilizes two "loading" steps in its work-flow – one to load the initial site data into the Source Database, and another to load the transformed data into the Synthetic Database.

Another common approach to the integration of external data is to provide a virtual interface, or view, that "federates" each site's data store to appear as if a single store is being accessed [3, 9]. Data warehouses differ from federated systems by enforcing the mapping of external data to the global schema by loading and transforming the data before being accessed by the consumer. In this case, the transformed data reside in a store that is within the domain of the warehouse. A federated system, in contrast, relies on dynamic mapping of the external data to a more loosely-coupled

"mediated" schema through a common software interface, such as a web application.

Issues of data quality, including *persistence* and *freshness*, often arise between the two approaches. Data warehouses tend to promote persistence of data by acting as an archive, while federated systems are characterized by providing the most up-to-date data available to the consumer. From our perspective, the data warehouse model is more aligned with a science-based synthesis environment where reproducibility and open access to all data are critical. Our vision here is that any site data loaded into the Source Database will be permanently archived, thus making it always available to the synthesis process, and ultimately the consumer. A federated system cannot meet this level of service since it operates outside of a site's functional realm and cannot guarantee data availability (e.g., one or more sites may be unavailable) or date persistence. The North Inlet LTER site, no longer participating in the LTER Network program, is one example of a site whose data is no longer available for synthesis[9].

The most up-to-date data, on the other hand, is best served by a federated system. The Pasta architecture achieves reasonable "freshness" by utilizing the revision concept of the EML to proxy a site-based release schedule. In this instance, the latency of freshness may be quantified by summing the time between EML revisions at the site and the frequency of the Harvester schedule.

4.2 Data Lineage and Provenance

In this context, we define data *lineage* as the linear progression of a specific data product over time, as in a sequence of revisions that are generated by adding new time-series data to a product line. It is assumed that each revision is preserved as a discrete "snap-shot" in time, thereby having a fixed start and end date. Similar to lineage, we define data *provenance* as the origin or history of a specific revision. In other words, provenance records the series of steps that generate a data product, such as the transformation process that produces synthetic data, including the origin of the source data. We can illustrate these two concepts as a matrix (Figure 9) that shows lineage as a progression in time from $t = 0$ to $t = n$ and provenance as a product changes through transformations from S to S^m, with D as the final synthetic data product. We call this the "Heritage Matrix". Both lineage and provenance are crucial for scientific data integrity and reproducibility.

[9]The North Inlet LTER data is currently archived at the LTER Network Office.

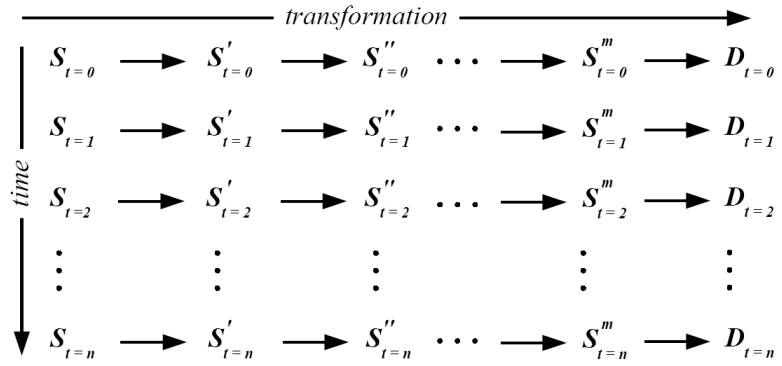

Figure 9: The Heritage Matrix showing both lineage and provenance as progression in time and product change, respectively.

Lineage provides the ability to work with data from a previous state in time, while provenance allows that data to be reproduced from its origin.

The Pasta architecture supports both lineage and provenance. Lineage is simply the by-product of keeping each data table that is created for every new revision of a synthetic data product. The lineage list can be seen in the Trends prototype when viewing "product details". Since revisions are never deleted, a user will always be able to access an earlier data product. This is especially important if overlapping data values change from one revision to the next.

Provenance information is recorded in the EML document for the synthetic data product. There are two parts of the provenance record as described earlier: 1) the software code of the transformation routine that is used to produce the synthetic data and 2) the content of the source data EML document, including the reference URL of the specific document located in the Metacat.

4.3 Module In, Module Out

The overall design of the Pasta architecture is defined by separate and autonomous modules that can operate independent of one another (as opposed to a single monolithic system that tightly-couples each process). This provides a great advantage in the development and testing of the modules since individual modules can be built in parallel. Front-end test cases can be easily developed for specific modules so that they may be demonstrated for

functionality and user acceptance without relying on those modules that are part of the early work-flow steps. An example from the Trends prototype is the development of the Storefront module, which was designed, implemented, and demonstrated using a manually generated testbed Synthetic Database for data access. Because the global schema will remain consistent between work-flow steps, we presume with confidence that the Storefront will operate continuously and unknowingly when the full system is put into place.

This modular approach also has the benefit of simplifying new module integration into the future. As new techniques for data integration and synthesis become available, individual modules can be replaced without concern for the other components of the system.

The opposite is also true. Individual modules can be taken out of the Pasta architecture and used else where in the ecoinformatics community. An example of this is the EML Parser library, which will become integral to both the Pasta architecture and to Metacat when it is complete.

4.4 Extensible Global Schema

The global schema of the Pasta architecture is unique, with regard to most local schemas, due to its compact nature – the core schema requires only a single data table structure (although, many actual data tables) and two support tables. This compact nature is possible because much of the content that most systems would require to store separately is already embedded in the EML documents. As a result, the global schema is quite portable from one database management system to another, and allows more superficial schemas to be constructed around it, thereby extending the overall functionality of the Synthetic Database. The current Trends prototype uses PostgreSQL and provides minimal content beyond the core global schema. We plan to move this instance of the global schema to the MySQL database system and augment the content with additional metadata for the completed Trends project, thus enabling a more comprehensive query environment.

4.5 Future Direction

The design and prototype development of the Pasta architecture provides only the foundation for automating synthesis at the LTER Network. Already, there are visions to expand the current architecture to take advantage of the work being performed by community-based information technology, like the informatics research taking place at the National Center for Eco-

logical Analysis and Synthesis, and the super computing centers, including the National Center for Supercomupting Applications and the San Diego Supercomputer Center.

One such improvement on the horizon is the use of the Life Science Identifier (LSID) protocol for identifying network accessible objects, such as an EML document and/or data. The LSID would be used as a universal identifier to uniquely name, reference, and retrieve distributed data objects, including an EML document. Object requests are never to the object directly (like a website address), but are to the LSID server, which is responsible for resolving the correct location of the object. In this model, if an EML document or data changes location, only the LSID server needs to know the new location. The rest of the world would simply continue to request the object through the LSID server by using the same object identifier.

Another technology that is being looked at closely is the use of public key certificates (X.509) for authentication. There are two specific areas that certificate-based authentication would improve system security and integrity. The first is the connections that are made during the "loading" process of source data from the individual site. It is not difficult to provide address resolution authentication from the data warehouse to the site, however, certificates would allow the data warehouse to load balance through multiple servers without the need to always update Internet addresses. More importantly, the certificate process would enforce the authentication to an authorized user of the application – not just the host Internet address of the server. The second use of certificates would be applied to the Transformation Engine that is performing the synthesis process. By authenticating through certificates, the Transformation Engine could be off-loaded to high-performance computing resources that are not part of the local domain. This would enable the use of applications that reside outside of the LTER Network, but which are authorized to access LTER data.

5 Conclusion

The Pasta architecture described above is part of an evolving suite of technology being investigated to address the challenges of Network-level synthesis. This architecture is only in its early stage of development, but promises to meet some long-standing goals of the ecoinformatics community - namely, an automated process to access site data and produce a synthetic data product that is available to the community.

6 Acknowledgments

The Network Information System group at the LTER Network Office would like to thank the LTER Network Information System Advisory Committee, and especially Wade Sheldon and Don Henshaw, for motivation and direction on this project. We sincerely appreciate our collaboration with the National Center for Ecological Analysis and Synthesis, including the insight provided from Mathew Jones, Mark Schildhauer, and Jing Tao. A sincere appreciation goes to the Trends project team and the editorial committee, with special thanks to Debra Peters, Christine Laney, and Ken Ramsey. Work completed on the Pasta architecture is funded by the National Science Foundation under Cooperative Agreement #DEB-0236154.

References

[1] Baker, K., B. Benson, D. Henshaw, D. Blodgett, J.H. Porter, and S.G. Stafford. 2000. "Evolution of a Multisite Network Information System: The LTER Information Management Paradigm", Bioscience, vol. 50, no. 11, pp. 963-978.

[2] Berkley, C., M. Jones, J. Bojilova, and D. Higgins. 2001. "Metacat: A Schema-Independent XML Database System", 13th Intl. Conf. on Scientific and Statistical Database Management, p. 171.

[3] Halevy, A.Y. 2001. "Answering queries using views: A survey", The VLDB Journal, pp. 270-294.

[4] Henshaw, D.L., M. Stubbs, B. Benson, K. Baker, D. Blodgett, and J.H. Porter. "Climate Database Project: A Strategy for Improving Information Access Across Research Sites", presented 9 August 1997, workshop on Data and Information Management in the Ecological Sciences: A Resource Guide, Albuquerque, NM.

[5] Jones, M.B., C. Berkley, J. Bojilova, M. Schildhauer. 2001. "Managing scientific metadata", IEEE Internet Computing, vol. 5, no. 5, pp. 59-68.

[6] Lenzerini, M. 2002. "Data Integration: A Theoretical Perspective", PODS 2002, pp. 243-246.

[7] Peters, D. and C. Laney. 2006. "Trends in Long-Term Ecological Research Project", Jornada Trails, Jornada Basin Long-Term Ecological Research Program, vol. 10, no. 1, p. 2.

[8] Sheldon, W., J. Brunt, D. Costa, C. Gries, J. McGann M. O'Brien, K. Ramsey, and M. Servilla. 2004. "EML Best Practices for LTER Sites", working document, LTER Network Office, 21 p.

[9] Ullman. J.D. 1997. "Information Integration Using Logical Views", ICDT 1997, pp. 19-40.

[10] Ziegler, P. and K.R. Dittrich. 2004. "Three Decades of Data Integration - All Problems Solved?", WCC 2004, pp. 3-12.