LTER
Long Term Ecological Research

01001100 01010100 01000101 01010010

LTER DataBits

Information Management Newsletter of
The Long Term Ecological Research Network

01001100 01010100 01000101 01010010

Feature Articles

News Bits

Good Reads

Calendar

DataBits: An electronic newsletter for Information Managers ----- Spring 2003 Issue (http://intranet.lternet.edu/archives/documents/Newsletters/DataBits/03spring/)

Featured in this issue:

Matt Jones explains the grid computing concept and describes a major new grid computing initiative for ecologists; Bill Michener fleshes out the context of that initiative with a sketch of SEEK, a wide-reaching grant for information technology in ecology. Peter Arzberger and others give us an insider's look at an international grid computing effort for environmental science. Chad Berkley and Peter McCartney bring us up to date on the latest tools for doing ecology "on the grid". Then, Matt Jones is back to unveil the long-awaited release of EML 2.0, and David Blankman gives us a candid introduction to the world of data modeling.

DataBits continues as a semi-annual electronic publication of the Long Term Ecological Research Network. It is designed to provide a timely, online resource for research information managers and to incorporate rotating co-editorship. Availability is through web browsing as well as hardcopy output. LTER mail list IMplus will receive DataBits publication notification. Others may subscribe by sending email to majordomo@lternet.edu with two lines "subscribe databits" and "end" as the message body. To communicate suggestions, articles, and/or interest in co-editing, send email to databits-ed@lternet.edu.

----- Co-editors: Tim Bergsma (KBS), Todd Ackerman (NWT)

Feature Articles

SEEK EcoGrid: integrating data and computational resources for ecology

-Matthew B. Jones, jones@nceas.ucsb.edu, National Center for Ecological Analysis and Synthesis, University of California Santa Barbara

The Science Environment for Ecological Knowledge (SEEK) is pursuing a vision of a seamless, easy-to-use infrastructure for scientific analysis and synthesis at national and international scales. The foundation of such an infrastructure must be a transparent and powerful system for accessing ecologically relevant data and for executing computationally demanding analyses and simulations. These data include the heterogeneous data collected at field stations (e.g., species monitoring, hydrology, meteorology, etc.) as well as remote sensing data, data from museum collections, and much more. Models and analyses that will need to be supported include well-known models such as GARP (Genetic Algorithm for Rule-set Production) and CENTURY (a trace gas model) as well as custom models and analyses written for a single experiment or study. The SEEK EcoGrid is being designed to provide the infrastructure for managing these data and computational resources.

The EcoGrid is intended to be a thin interface layer that allows various data and compute services already in existence to interoperate. For example, the Metacat system developed by the KNB Project is a networked data and metadata management platform with features that are similar to the Storage Resource Broker developed at SDSC. The EcoGrid will make both of these systems and others accessible through a high-level programmatic API. Anybody who has developed a data management system for ecologically relevant data will be able to implement the EcoGrid API and therefore be a full participant in the EcoGrid network. Thus, field stations and other sites with highly customized data management infrastructures will be able to exchange data through their common EcoGrid interfaces.

Grid Background

A Grid (formally, a Compute Grid) is a system that links multiple computational resources such as computers, sensors, data, and people. In contrast, a Data Grid denotes a network of storage resources, from archival systems to caches and to databases that are linked across a distributed network with an emphasis on high performance and throughput. Scientific computing presents several challenges not found in typical business applications (such as the need for high performance computing). Recent research into Grid computing has tried to create an infrastructure for science that provides solutions to these challenges. Information about the most successful of these research efforts can be found at the Globus project and the Global Grid Forum. Our work on the EcoGrid will be conducted with full knowledge of existing Grid efforts and we will strive for maximum compatibility with those efforts.

EcoGrid Design and Implementation

The EcoGrid will be an infrastructure that combines features of a Data Grid for ecological data management and a Compute Grid for analysis and modeling services. EcoGrid will form the underlying framework for data and service discovery, data sharing and access, and analytical service sharing and invocation. Specifically, the EcoGrid will provide:

- Seamless access to data and metadata stored at distributed EcoGrid nodes. Features include scalability, multiplicity of platforms (desktop to supercomputers) and storage devices, single sign-on authentication, and multi-level access control.
- Execution of analyses and models in a computational network.
- EcoGrid node registry for data and compute nodes based on UDDI.
- Rapid incorporation of new data sources as well as decades of legacy ecological data.
- Extensible, ecologically relevant metadata based on the Ecological Metadata Language.
- Replication of data to provide fault tolerance, disaster recovery, and load balancing.
- An EcoGrid Portal that provides a central access point for all EcoGrid data access services.

We envision that EcoGrid will tie together a variety of currently independent software systems and networks (Figure 1). Although we will not initially include all of the services displayed in Figure 1 due to resource and timing constraints, EcoGrid will be an open system, allowing others to develop systems according to the EcoGrid interfaces and participate in the grid.

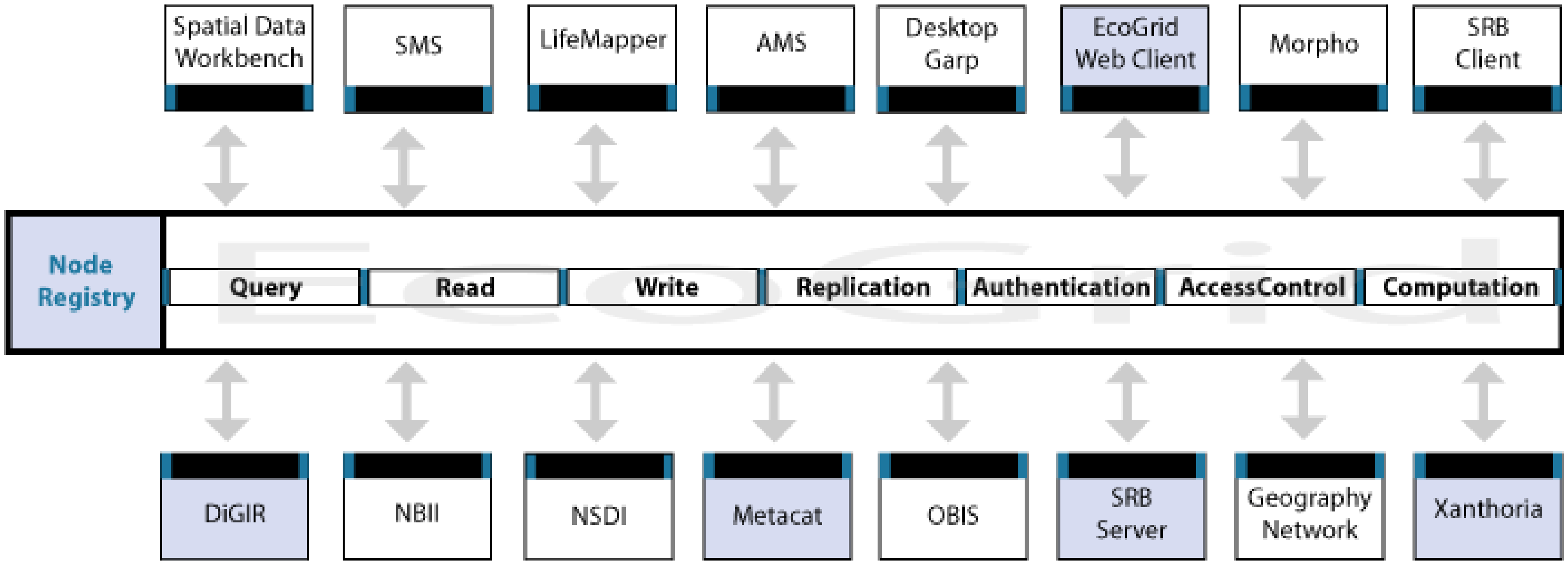


Figure 1: A schematic of the EcoGrid interfaces and participating services. Components below the EcoGrid interfaces largely function as data provider services, while those above largely function as data consumer clients. Shaded components will be products of the first phase of EcoGrid implementation, others as resources permit.

Deployment of EcoGrid

Initial deployment of the EcoGrid will occur through the implementation of the EcoGrid interfaces for Metacat, SRB, and DiGIR data management systems. This should provide a base EcoGrid network that includes data from the 24 LTER sites, hundreds sites in the Organization of Biological Field Stations, the 36 UC Natural Reserve System reserves, the PISCO network, over 80 collections from museums in the Species Analyst network, and a variety of data in SRB servers at SDSC. We also expect that a wide variety of independent researchers that are not site affiliated would be able to expose their data through the EcoGrid.

Conclusion

The EcoGrid will grow in its significance as it gains acceptance and participation by producers of environmental and ecological data. Initially we may have to actively recruit sites and scientists to develop systems and data for use in the EcoGrid. However, the EcoGrid will enable gains in synthesis and broad-scale research, and as these become apparent we expect to create a positive-feedback cycle that increases participation. The sites with the foresight to participate in the nascent EcoGrid will help to build a data and computational research system that will revolutionize the types of synthetic science that are possible in ecology.

We welcome your contributions to EcoGrid and the SEEK project in general, and your feedback on the technical approaches we are taking. Updates about EcoGrid and the initial EcoGrid portal will be available on the Science Environment for Ecological Knowledge (SEEK) web site.

This material is based upon work supported by the National Science Foundation under Grant No. ITR 0225676. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do

not necessarily reflect the views of the National Science Foundation (NSF).

Building SEEK: the Science Environment for Ecological Knowledge

- William K. Michener, wmichener@LTERnet.edu, LTER Network Office; Department of Biology, University of New Mexico

Knowledge of the natural world is limited not just by the complexity of natural entities and processes, but also by the complexity of the data that describe them. It is clear that enhanced understanding of the natural world depends on our capacity to access and integrate data from the biological, physical, and social sciences; mine those data for new knowledge; and convey new insights to decision-makers and the general public. The Science Environment for Ecological Knowledge (SEEK) represents a 5-year research investigation that will address many of the challenges associated with facilitating knowledge discovery across disciplinary, geographic, and methodological boundaries. The project encompasses information technology (IT) research in open architectures for data access, semantic mediation using domain-specific ontologies, and semantically integrated analysis and modeling systems.

The core infrastructure for SEEK will be developed by an extensive multidisciplinary team, led by the Partnership for Biodiversity Informatics - a consortium including the University of Kansas, University of New Mexico, National Center for Ecological Analysis and Synthesis, University of California San Diego, in conjunction with other partnering institutions (e.g., Arizona State University, Napier University, University of North Carolina, University of Vermont). The SEEK architecture is designed to include three well integrated components: an Analysis and Modeling System that provides users with the ability to capture scientific workflows as structured objects in digital libraries for re-use and extension; a Semantic Mediation System that enables discovery and automated integration of highly relevant but heterogeneous data via formal ontologies; and the EcoGrid which encompasses the grid infrastructure that manages the data, metadata and computational resources.

The overarching goal of SEEK is to provide for the integration of local desktop data with a larger network of data and analytical tools, enabling ecologists and other researchers to tackle complex research problems that were hitherto intractable. It is anticipated that the resulting Internet-based infrastructure will make it easier to derive ecological knowledge by flexibly composing and applying a spectrum of analysis techniques to heterogeneous data from geographically distributed sources. Several key computer science and IT challenges will be addressed by the distributed research and development team as they build the SEEK infrastructure:

- Analysis and Modeling (AM) System-Developing a modeling language for ecological analysis based on parameter ontologies, which provide an extensible and adaptable vocabulary of ecology concepts and their relationships, and analytical workflows, which chain together analysis steps and bind them to suitable datasets
- Semantic Mediation (SM) System-Extending XML-based data integration and mediation technology by merging it with formal ontologies and other logic-based knowledge representation formalisms to facilitate semantic mediation over hard-to-relate schemas of data sources
- EcoGrid (EG) - Providing the grid infrastructure for seamless access and manipulation of ecology data and tools by extending the expressiveness of service description languages for enhancing data and analytical service grid capabilities (for more information on EcoGrid, see the companion article by Matthew Jones in this issue)

In addition to the core development team of research scientists and programmers, multidisciplinary teams of scientists organized in collaborations-the SEEK Working Groups-will closely inform the design and development of the IT research areas described above. These collaborative working groups will engage approximately 60 scientists and informatics specialists from the domains of biodiversity, ecology, earth science, and human factors to address the most critical conceptual barriers for SEEK: biological classification and nomenclature semantics, knowledge representation for the biodiversity and ecological sciences, as well as biodiversity and ecological modeling and analysis. Drawing on broad, multi-disciplinary representation, the working groups are structured so as to identify community concerns and needs, and feed that information directly into the infrastructure design process.

The three Working Groups are:

- Biological Classification and Nomenclature (BCN)-investigates solutions to mediating among multiple, often competing or inconsistent, taxonomies for naming organisms, including integration of disparate data sources (museum collections with ecological data) via enriched semantics
- Knowledge Representation (KR)- develops and tests ontologies for expressing concepts in ecosystem and biodiversity science that will parameterize the semantic mediation system and provide a vocabulary for the analysis and modeling system
- Biodiversity and Ecological Modeling and Analysis (BEAM)-provides domain experts' knowledge in modeling and analysis to evaluate SEEK usability for addressing biodiversity and ecological questions

SEEK-BEAM will initially target the integration and synthesis of ecological and biodiversity data. This research frontier is critical to ecological and biodiversity forecasting and to enable managers and policymakers to anticipate environmental change and thereby deal with it in a more sustainable fashion. Research issues of critical interest to scientists and policymakers will serve to test the utility and effectiveness of SEEK. We will initially devote efforts to creating the IT infrastructure that will facilitate detecting and understanding patterns in living resources and biodiversity-i.e., ecological niche modeling. Future research themes will likely include understanding the interrelationships between biodiversity and ecosystem function and how they may be affected by global change, as well as other important scientific and societal issues.

SEEK will employ a multi-faceted approach to insure that the research products, software, and information technology infrastructure resulting from SEEK optimally benefit science, education, and the public. Outreach includes community involvement, a WWW portal, informatics training, and an innovative annual IT transfer symposium. Broad community participation in SEEK will be ensured through the direct inclusion of IT and domain scientists from the international scientific community in Working Groups and on the Science & Technology Advisory Board. Working Groups will include participants from the U.S. and International Long-Term Ecological Research Networks, the California Institute for Telecommunications and Information Technology, the Integrated Taxonomic Information System, the Organization of Biological Field Stations, Scripps Institution of Oceanography, the National Biological Information Infrastructure, BIOSIS and other relevant organizations.

A WWW portal, http://seek.ecoinformatics.org, will house and/or point to Internet-accessible resources (software, archives, research products and technical information) that are easily discovered, and freely accessible to the scientific community. SEEK is firmly committed to supporting the Open Source Initiative (OSI; http://www.opensource.org/).

Informatics training will be coordinated and provided through tutorials at the San Diego Supercomputer Center and an intensive two-week course in informatics (funded by an NSF RCN grant) for staff and students associated with biological field stations and marine laboratories, offered at the University of New Mexico. SEEK will support instructors and provide training materials and content for these courses, as well as a new distributed graduate seminar series that will be offered at

SEEK institutions and made available over the WWW.

A key element of our community outreach will be an innovative annual symposium and training program that focuses on information technology transfer to young investigators and students, particularly those from underrepresented groups. Young faculty members and post-doctoral associates will participate in a weeklong symposium in which the participants will gain hands-on experience with the latest information technology, including products resulting from SEEK. Participants will be provided with web-based materials that they can use in developing courses at their home institutions. Our objective in this regard is to "train the teachers", thereby extending our outreach to the broadest possible community.

It is anticipated that when complete, SEEK will enhance the national and global capacity for observing, studying, and understanding biological and environmental complexity in several ways. Through the development of intelligent analytical tools and an infrastructure capable of semantically integrating diverse, distributed data sources, it will remove key barriers to knowledge discovery. By expanding access to ecological data, information, and knowledge, SEEK will create new opportunities for scientists, resource managers, policy makers and the public to make informed decisions about the environment. Finally, it will provide an infrastructure for educating and training the next generation of ecologists in the information technology skills that will be critical for scientific breakthroughs in the future.

We welcome your contributions to the SEEK project as well as your feedback on the technical approaches we are taking. SEEK updates, as well as information about subscribing to mailing lists, is available through http://seek.ecoinformatics.org.

This material is based upon work supported by the National Science Foundation under Grant No. ITR 0225676. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

PRAGMA: the Pacific Rim Application and Grid Middleware Assembly

- Peter Arzberger, Tony Fountain, Philip Papadopoulos, UCSD, SDSC

In the 21st century advances in science and engineering (S&E) will, to a large measure, determine economic growth, quality of life, and the health of our planet. The conduct of science, intrinsically global, has become increasingly important to addressing critical global issues...Our participation in international S&E collaborations and partnerships is increasingly important as a means of keeping abreast of important new insights and discoveries in science and engineering [NSB].

The Pacific Rim Application and Grid Middleware Assembly (PRAGMA) was established via an inaugural workshop in March 2002 with a dual mission: 1) to establish sustained collaborations, and 2) to advance the use of grid technologies in applications among a community of investigators working with leading institutions around the Pacific Rim. PRAGMA was established based on the following observations and trends: Science is an intrinsically global activity, and problems in the environment need to be addressed globally; the grid is transforming computing and collaborations, creating an environment that allows pulling together geographically distributed resources (instruments, datasets, computing and visualization platforms, and people); the grid is still too complicated to use by members of the scientific community.

PRAGMA focused on creating testbeds for grid software and applications. One particular application is ecoinformatics, described below:

- The PRAGMA ecoinformatics collaboration aims to provide access via a simple web interface to global resources and application technologies developed in the Pacific Rim. The overall goal is an integrated knowledge management system that uses extensible and scalable infrastructure to provide data and analytical services to the national and international ecological research communities. By integrating and analyzing data from geographically distinct locations, new cross-site ecological hypotheses can be framed and explored, thus allowing scientists to analyze ecological patterns and processes that are currently unapproachable, e.g., correlating land cover changes with other environmental or anthropogenic factors. PRAGMA collaborators in the ecoinformatics project include members from the San Diego Supercomputer Center, UC San Diego, the Long Term Ecological Network, the National Agriculture Research Center of Japan (NARC), the Taiwan National Center for High-Performance Computing, and the Taiwan Ecological Network (TERN). Although individuals from various partner sites have specific interest areas and local resources, they all participate in the joint development of a web services testbed for data and resource sharing. The Spatial Data Workbench (SDW) (http://sdw.sdsc.edu) provides a portal into the current testbed. This system combines a variety of development efforts, including the NPACI project to manage LTER Network hyperspectral data and the web services ClimDB prototype. The SDW supports explorations and experiments of grid-based technology. Its role is to provide a laboratory for hands-on experience in web service applications.

By providing dynamic data access and integration, the system supports cross-site studies and collaborations. By basing the architecture on web services technology, the system is extensible and scalable, supporting the smooth integration of new data collections, new analytical services, and new sites and communities. The current on-line system is modest in its operational functionality; however, the lessons learned from this testbed will be invaluable in developing the next generation of production grid applications.

For more information about PRAGMA, the resources currently available, and our ongoing activities, please visit:

- PRAGMA: http://www.pragma-grid.org
- SDW: http://sdw.sdsc.edu
- MetBroker: http://www.agmodel.net/MetBroker/index.html
- Taiwan Ecogrid: http://ecogrid.nchc.gov.tw/ecogrid/index.htm
- Press release regarding the Taiwan Ecogrid: http://www.gridtoday.com/03/0317/101186.html

Monarch: metadata-driven analytical processing

- Chad Berkley, National Center for Ecological Analysis and Synthesis, University of California, Santa Barbara

Introduction

In any discussion of metadata, one inevitable question is "What can we do with metadata once we've gone to the trouble of collecting, formatting and storing it?" The typical answer is that metadata enable us to discover, search, access, and interpret data with far greater power and accuracy than is possible with full-text indexing, or more crudely, filesystem level information. But metadata can also create exciting new possibilities for the processing and analysis of data. Monarch is a demonstration of these possibilities. Monarch provides a flexible interface to an analytical engine that allows advanced processing of data files based on metadata descriptions of both the data themselves, and of a library of well-defined analytical procedures.

Currently, when a scientist wants to analyze data, he or she opens an analytical application such as SAS® or MatLab® and either writes the code to do the analysis, starting from scratch. Alternatively, existing code is modified to use the new data file as input. The data may start out as an Excel spreadsheet which is exported as a comma delimited text file for import into SAS, then the output of the SAS analysis may be transformed again into a data file that Matlab can read so that Matlab can produce a summary graph for publication. These tasks can be tedious and inefficient, especially when many heterogeneous datasets are analyzed in this way, since the input code must be changed every time a new data file is imported.

The Monarch Process

Monarch aims to automate the time intensive process of writing input code. Monarch will also allow scientists to store and share the analytical code that they do write and easily reuse that code on different datasets. Most of the actual analytical code that is written is independent of the data that is being used as input. Monarch formalizes this with the construct of an analytical "step". Each step defines the input data needed, the output data produced, and the native code required to run one analytical process. The code within a step is native to one of the back-end analytical engines (for example SAS, Matlab, R, etc.) to which Monarch has access. Thus, the output from a SAS step could be used as input to a Matlab step and the output from the Matlab step could be used as input to another SAS step, and so on.

Step Example

```
<?xml version="1.0"?>
<step>
  <identifier system="knb">sum.1.1</identifier>
  <name>sum</name>
  <description>Sum two attributes from an entity</description>
  <input>
    <table-entity id="t1">
      <entityName>sumdata</entityName>
      <entityType>table</entityType>
      <entityDescription>Data table containing the data to sum</entityDescription>
      <attribute id="a1">
        <attributeName>x</attributeName>
        <dataType>decimal</dataType>
    <attributeDescription>numeric data to be summed</attributeDescription>
   </attribute>
   <attribute id="a2">
    <attributeName>y</attributeName>
    <dataType>decimal</dataType>
    <attributeDescription>numeric data to be summed</attributeDescription>
   </attribute>
  </table-entity>
 </input>
 <output>
  <table-entity id="t2">
   <entityName>sumdata</entityName>
   <entityType>table</entityType>
   <entityDescription>Data table containing the summed data</entityDescription>
   <attribute id="a3">
    <attributeName>x</attributeName>
    <dataType>decimal</dataType>
    <attributeDescription>numeric data to be summed</attributeDescription>
   </attribute>
   <attribute id="a4">
    <attributeName>y</attributeName>
    <dataType>decimal</dataType>
    <attributeDescription>numeric data to be summed</attributeDescription>
   </attribute>
   <attribute id="a5">
    <attributeName>z</attributeName>
    <dataType>decimal</dataType>
    <attributeDescription>numeric data sum</attributeDescription>
   </attribute>
  </table-entity>
 </output>
 <code language="SAS"><![CDATA[
  DATA sumdata;
   set sumdata;
   z = x + y;
  RUN;
  ]]>
 </code>
</step>
```

Example 1: XML specification of a summation step.

Example 1 is the specification for a simple summation step. The inputs are defined as two columns (attribute) from a data table (table-entity) and the output is a three column data table. The output attributes include the two attributes that were added together (x and y) as well as the result of the addition (z). The code section defines the language as SAS, then defines the actual code that will be run by SAS to perform the operation. The names of the variables used within the code are the same as the names of the variables in the inputs and outputs.

Thus for the following example input:

x	y
1	2
3	4
5	6
7	8

the output would be:

x	y	z
1	2	3
3	4	7
5	6	11
7	8	15

Just because the step defines a two column table as input does not mean that the data file is limited to a two column table. Realistically, a data table could have 25 columns, only two of which would logically be the arguments of a summation. Because of this, Monarch allows the user to dynamically select which columns within a given table to bind to the step inputs. Monarch assists the user in knowing which variables are likely candidates by type checking each input variable in the step and matching the variable to the attribute data type defined in the metadata.

Once a formal step is defined, it can be linked to other steps via a pipeline. Monarch is built around the idea that an analysis really is a sequence of discrete analytical steps, where data flows from one step to the next just as water flows through a pipe. Thus, we call these chains of analytical steps 'pipelines'. Each pipeline terminates with a final step that produces the desired output. This output may be a textual report of summary statistics, a simple bar chart graph, a 3D density model, a GIS layer, or any other type of scientific data output.

Pipeline Overview

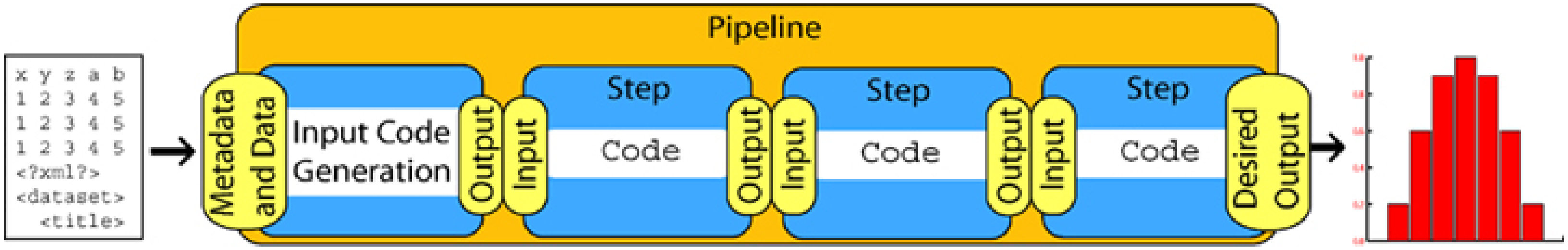


Illustration 1: High level view of a linear pipeline flow.

Illustration 1 shows a high level view of a pipeline starting with the input code generation and ending with the desired output, in this case, a bar chart. Each blue box is a step with defined inputs, outputs and code. The pipeline itself is made up of the set of steps. In reality, a pipeline is really a directed graph, allowing for steps to be linked into a tree structure. Illustration 1 shows a linear example for simplicity.

Pipeline Example

```
<?xml version="1.0"?>
<pipeline>
  <identifier system="knb">sumplot.1.1</identifier>
  <name>Graph Sum</name>
  <description>The graph of the sum of x and y versus y</description>
  <input>
    <scalar id="s1">
      <scalarName>plot_title</scalarName>
      <dataType>string</dataType>
      <scalarDescription>Title of the plot</scalarDescription>
      <default>Default plot title</default>
    </scalar>
    <table-entity id="t3">
      <entityName>sumdata</entityName>
      <entityType>table</entityType>
      <entityDescription>Data table containing the data to sum</entityDescription>
      <attribute id="a1">
        <attributeName>x</attributeName>
        <dataType>decimal</dataType>
        <attributeDescription>numeric data to be summed</attributeDescription>
      </attribute>
      <attribute id="a2">
        <attributeName>y</attributeName>
        <dataType>decimal</dataType>
        <attributeDescription>numeric data to be summed</attributeDescription>
      </attribute>
    </table-entity>
  </input>
  <output>
    <other-entity id="o1">
      <entityName>none</entityName>
      <entityType>image/gif</entityType>
      <entityDescription>the image produced from the plot</entityDescription>
    </other-entity>
  </output>
  <execute step="sum.1.1" stepID="0">
    <map inparam="t3" fromstepID="" param="t1" />
    <map inparam="a1" fromstepID="" param="a1"/>
    <map inparam="a2" fromstepID="" param="a2"/>
  </execute>
  <execute step="scatterplot.1.1" stepID="1">
    <map inparam="t2" fromstepID="0" param="t1" />
    <map inparam="a3" fromstepID="0" param="a1"/>
    <map inparam="a5" fromstepID="0" param="a2"/>
    <map inparam="s1" fromstepID="" param="s1"/>
  </execute>
  <outputmap inparam="o1" fromstepID="1" param="o1"/>
</pipeline>
```

Example 2: An XML pipeline specification.

The Example 2 pipeline uses the sum step that was defined above in Example 1 and maps it to a scatterplot step to produce a gif image of a summed scatterplot. The pipeline defines the inputs and outputs of the pipeline itself as well as mapping the outputs of the first step (sum.1.1) to the inputs of the second step (scatterplot.1.1). In each execute section, the "inparam" attribute tells the name of the input object that is being used as input to the step and maps it to a "param" which is the id of the input object in the step. Thus, in the first map statement, the table-entity t3 from the pipeline input declaration is mapped to the param t1 of the sum.1.1 step. When mapping between steps, the "fromstepId" attribute is used to show from which step the mapping is coming. When the mapping comes from the pipeline, fromstepId is empty.

The sum and scatterplot examples have been chosen to illustrate the basics behind Monarch. They are by no means the extent of Monarch's ability. Pipelines have been or are currently being written to do everything from summary statistics, to quantile plots, to EML metadata quality assurance, to simple means charts. The power in the pipelining system lies in its ability to take these individual pieces (steps) and link them together to form a much more complex analysis. Eventually, Monarch will have a simple graphic pipeline editor which will eliminate the need for the user to understand the pipeline and step specifications altogether.

Pipelines Currently Being Developed

The table below is a summary of the pipelines in development at the time of publication. Please see the Monarch CVS server for a current list of pipelines and steps.

Pipeline	Purpose
ComputeUnivariateStats	Compute univariate stats for all numeric fields in the input file.
MakeBivariatePlot	Make a bivariate scatterplot of each numeric data pair in the input dataset.
MakeBoxPlot	Make a box plot for each numeric attribute in the input dataset.
MakeQQPlot	Make a two-sample quantile plot of each pair of numeric attributes in the input dataset.
MakeQuantilePlot	Make a quantile plot of each numeric attribute in the input data set.

MeansChartPipe	Chart of the means of the data groups.
GraphSum	The graph of the sum of x + y versus y.
TestDataFormat	Test the format of the input dataset (make sure all data is readable in the EML-specified format).
TestEMLConsistency	Test the consistency of what EML metadata says about the data versus what is actually in the input dataset.
XmlToHtmlXSLTPipeline	Transform an XML document to HTML using a given XSLT stylesheet.
XmlToPlainTextXSLTPipeline	Transform an XML document to ASCII text with a given XSLT stylesheet.
XmlToXmlXSLTPipeline	Transform an XML document into a different XML document with a given XSLT stylesheet.

Benefits of Monarch

Besides the obvious benefits of allowing more generalized code reuse and freeing the scientist from having to write specific input statements for heterogeneous data, Monarch also provides several other services to the user. First, because the analytical code is marked up into steps and finally pipelines, the code and the analytical process itself are fully documented. The pipeline and steps can be added to the metadata for a dataset to document the transformations the data has gone through in its lifetime. Like metadata in the Metacat system, pipelines and steps have strict version control so that changes in code and analytical processes can be tracked over time. Creating steps and pipelines also facilitates the sharing of code between scientists.

Flexibility

Monarch does not assume that a scientist will want to use one of the back-end analytical engines that it currently supports (SAS, MatLab): it has been designed with a flexible plugin interface. Virtually any analytical engine can be configured to be used with Monarch. Even non-analytical programs can be used. For instance, Monarch currently has an XSLT plugin to facilitate XML transformations. Likewise, Monarch does not make assumptions about the metadata standard that a scientist is using. Any metadata standard that can properly define the form and function of a data file can be used. This is facilitated through another plugin system. By default, Monarch is set up to use EML 2.0.0 and EML 2.0beta6 metadata.

Since Monarch is a server based system, it can be utilized from a variety of clients. For instance, there are plans to add Monarch support to Morpho so that data packages can be analyzed directly from the Morpho interface. There is also a web interface that is integrated into the standard Metacat web interface. Once a scientist has data described with EML, they will be able to run any number of analytical processes from within their current interface.

Future Development

Monarch development will continue under the new Science Environment for Ecological Knowledge (SEEK) project. Semantic capabilities will be added. Additionally, Monarch may one day be capable of distributing analytical processes not just across different analytical engines but across different specialized machines. For instance, a high end graphics application could be sent off to an SGI machine or a number crunching application could be run on a super computer at a different location.

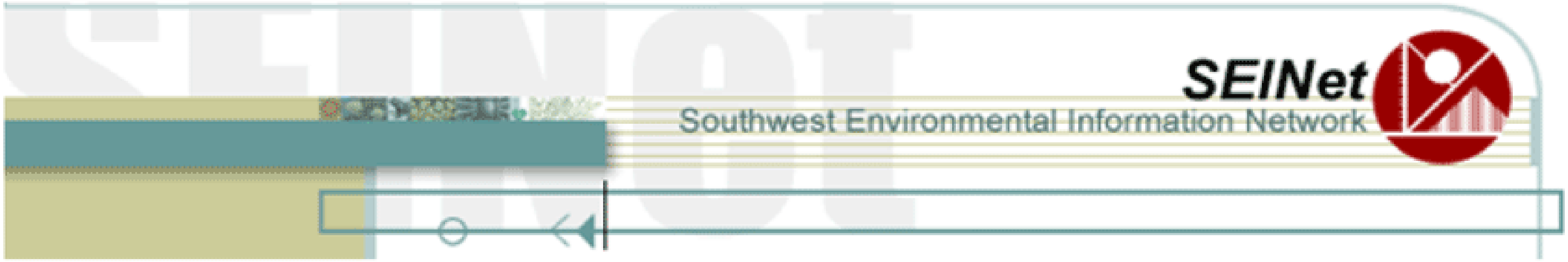
Contacts

If you would like to try out an early release of Monarch, please visit the online demo.

If you would like more information on Monarch please visit the Monarch website or contact Matt Jones, Rich Williams or Chad Berkley.

SEINet: metadata-mediated access to distributed ecological data

- Peter McCartney, Central Arizona Phoenix LTER



The Center for Environmental Studies, home of the Central Arizona Phoenix LTER (CAP) announces the release of a new website, the Southwest Environmental Information Network (http://seinet.asu.edu/) to serve as a gateway to distributed data resources of interest to the environmental research community. SEINet is more than just a web site - it is a suite of data access technologies and a distributed network of departments, museums and agencies that manage environmental information. SEINet will replace many existing features of the CAP LTER web site including the data catalog, bibliography application, and the protocol library. CAP LTER will be one of the primary contributors to SEINet, but this new design will enable the integration of CAP’s research data with other environmental data contributors in the central Arizona region and help foster a more open, collaborative data and research context among this developing partnership.

Interface

SEINet provides a unified search interface for a variety of environmental databases. Eight biological collections in the southwest publish their label catalog databases through SEINet. Taxonomic name information is mediated through a southwest-centric implementation of the Integrated Taxonomic Information System (IT IS), a database for managing names, synonyms and taxon systematics information. The output displays collection information, taxonomy, images, and spatial location of specimens.

The SEINet query system uses the hierarchical structure of ecological metadata to support simple or advanced searches across multiple inventories of different kinds of information. Searches for data resources defined by the various Ecological Metadata Language resource types can be done for one or all of them (dataset, literature, or protocols). Under current funding from NSF, we will add ecological models to the list of resources available.

Several applications for exploring data have been provided thus far. The one discussed here is the Data Analyst application, which handles all access to research data. Once users locate a dataset of interest, they can open the Data Analyst with that dataset active. The first step is to define a working data entity from the active dataset. Since datasets can contain multiple, related entities, the user is walked through a wizard interface which analyses the EML description for the dataset and presents a series of forms for selecting entities, selecting attributes, defining aggregation functions, defining joins, and setting filter expressions. Only the first step is required - the user can simply select an entire entity or pass through all forms to define a complicated query.

Once the desired data object is defined, the application offers three main choices - download the data object, view it online, or summarize it. The first choice builds a download package containing the datafile and an EML document describing it and returns a pointer to that package. The second opens the data in either a table or map viewer, depending on the object’s logical type (table, vector GIS, Raster). The summary option leads to wizards that let the user invoke a set of simple Exploratory Data Analysis (EDA) functions. The purpose of these is not to duplicate commonly used statistical packages online, but rather to allow users to evaluate the data prior to downloading.

Technology

Underlying this web interface is an application infrastructure that relies heavily on XML and rich, standardized metadata formats such as EML. The system is a four tier architecture typical of most web services applications now appearing on the internet (Figure 1). At the base are data sources stored in distributed archives. A services layer connects to these datasources using their native protocols and exposes them to the network using a common shared protocol, in this case SOAP messages based on shared schemas. The application layer communicates between the user’s client application (in this case a web browser) and the service layer, providing most of the business logic for the system. In SEINET, the application layer consists of Java Server Pages (JSP) and java beans. These manage the user's session, profile, login, and the input and output The query system is based on Xanthoria (McCartney 2002 Databits). Xanthoria uses connectors that convert native metadata storage into a federated xml schema, then executes an Xpath query across multiple targets that support that schema.

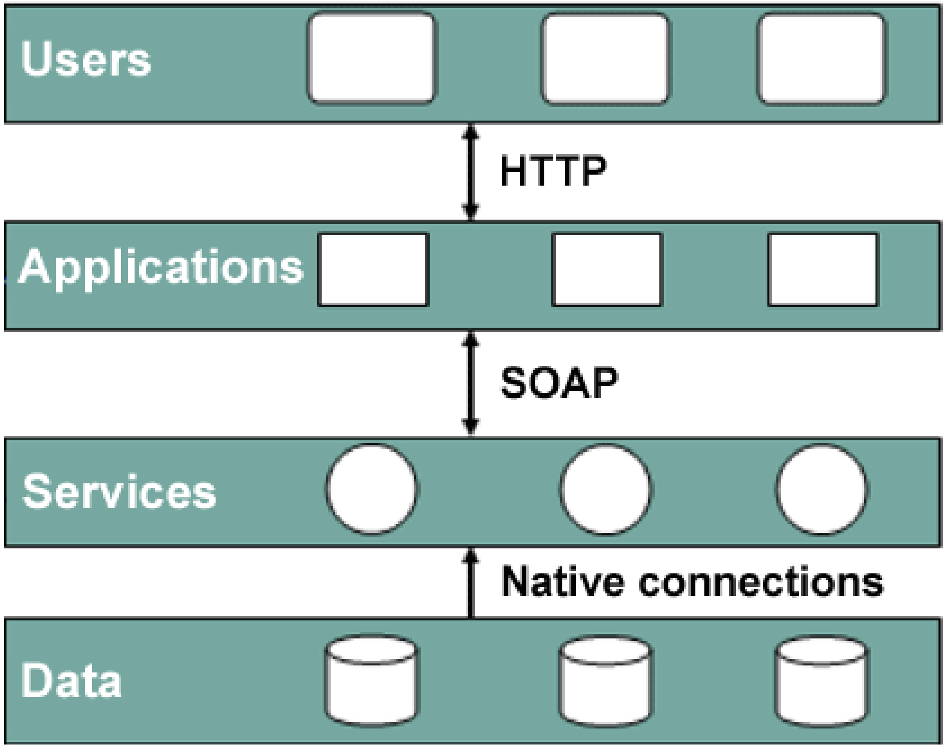


Figure 1. SEINet architecture.

Xylopia is a collection of components linked together by a common messaging framework. Applications submit requests for data to the message router, which breaks the job into individual component requests that are managed serially (Figure 2). Requests are expressed as SOAP messages containing one or more processing requests- each conforming to the XML schema defined for each service component. Typically, the first component executed in a pipeline is a Data Retriever request. This accepts an emlQuery message that defines the desired data object. The message includes (or specifies the location of) the eml-dataset documentation of the data, and includes references - within the query expression elements - to the identifiers of the entity and attributes in that document. The data retriever service parses the message, reads the relevant eml-dataset descriptions, substitutes the actual physical entity and attribute information, and builds the appropriate query syntax according the connection and storage type (SQL for most relational databases, SDE for Spatial Database Engine, and file-based access for shapefiles, ASCII files, etc.). The data object is retrieved and placed in a drop-box for subsequent actions and a pointer to the data (along with a new eml-dataset description of the object) is returned as the response message.

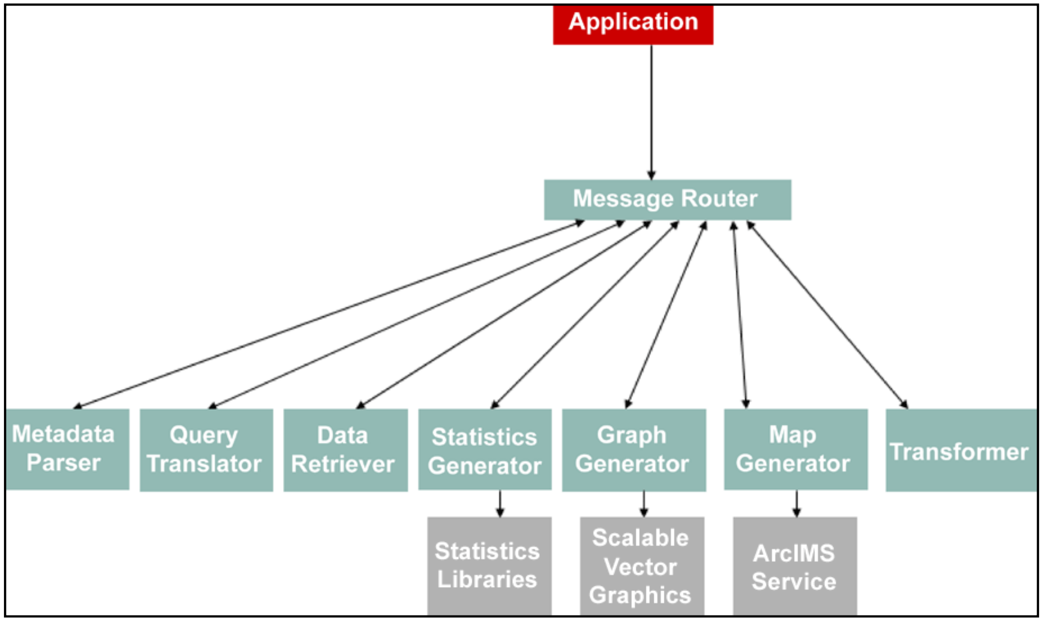


Figure 2. Xylopia architecture.

The significance of this design is that it divorces the physical details of the query request from its logical meaning. The query statement is, in essence, “I'll have attribute id’s 3,5, and 9 from entity 4 in dataset 34 in the CES system”. Users need not see, or understand, the physical data format, location, or actual column names in order to define a query or request a visualization option: they need only refer to the logical data description which can be parsed and presented to them in a user-friendly wizard interface.

Other services are invoked to perform subsequent operations with the data object, including a recursive call to the data retriever to further subset the data. These transformation processes provide the linkages that establish the compatibility between the output of one process (such as the data retriever) and a subsequent process. For now, these are restricted to common reductive tasks such as binning data for passing to a histogram display or cross- tabulating data to summaries counts of nominal categories. Still other services produce statistical or graphic output such as generating the ArcIMS message (AXL file) for displaying a GIS object in a Map service or Scalable Vector Graphics (SVG file) for rendering data in a chart.

Future directions

For now, the SEINet interface is relatively simple, seeking merely to expand existing modes of internet discovery, visualization and access to data. However, its service- based architecture can be extended to more advanced integration in subsequent development projects, such as a recent ITR award for “Networking Urban Ecological Models using Web Services”.

Acknowledgements: This material is based on supported by the National Science Foundation under Grant number 9983132. Any opinions, findings or conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

A brief overview of Ecological Metadata Language

- Matthew B. Jones, jones@nceas.ucsb.edu, National Center for Ecological Analysis and Synthesis, University of California Santa Barbara

Ecological Metadata Language (EML) is an open metadata specification that facilitates data sharing, data preservation, and data interpretation across institutional and research boundaries. This article is a brief introduction to EML for software developers and data managers that want to use EML 2.0.0 (recently released by the contributors to EML) in their systems.

Ecologically relevant data span a number of distinct domains of science, including biodiversity studies, ecosystem function, meteorology, soil science, hydrology, remote sensing, and many others. EML allows scientists to provide standardized descriptions of data from these various disciplines in an openly accessible text format. The descriptions in EML can be exceedingly terse (Listing 1), or incredibly rich (e.g., full descriptions of data syntax, schema, and semantics). Thus, EML can be used in a variety of research applications, from simple data discovery to advanced data processing.

EML was designed to be:

- *open* to allow human readability and facilitate long-term data archiving
- *modular* to promote re-use of metadata sections and structures
- *extensible* to allow additional metadata that is not part of EML to be included
- *structured* to allow machine processing for analytical applications and other software applications
- *easy to implement* by minimizing required metadata

A simple EML document (Listing 1) demonstrates that EML is a human-readable and yet structured language. EML has long-term archive value because it is a simple text file that can be read by scientists and data managers, and it has broad utility in software applications because it is in a structured format that can be machine-parsed. The metadata in Listing 1 includes only the title, creator, and contact for a set of data, without any further details about how to obtain the data or interpret it. This level of metadata provision is useful for simple data discovery across the ecological research community. One example of a simple data registry that utilizes EML can be found in the Organization of Biological Field Stations Data Registry.

Listing 1: A minimal EML document

```
<?xml version="1.0"?>
<eml:eml
    packageId="eml.1.1" system="http://knb.ecoinformatics.org"
    xmlns:eml="eml://ecoinformatics.org/eml-2.0.0">

  <dataset>
    <title>Biodiversity surveys for Lesser Tree Frogs at
             Barro Colorado Island (BCI) from 1994 to 1999</title>
    <creator id="23445" scope="document">
      <individualName>
        <givenName>Jane</givenName>
        <surName>Smith</surName>
      </individualName>
      <electronicMailAddress>jane@data.org</electronicMailAddress>
    </creator>
    <contact>
      <references>23445</references>
    </contact>
  </dataset>
</eml:eml>
```

Note that Listing 1 does not actually contain any of the data, nor does it have a description of the methods used to collect the data, nor does it contain a description of the structure and schema of the data. All of this information and more can be provided in richer EML documents, and for almost all purposes providing this more extensive information is desirable. An example of a data catalog that utilizes this more extensive metadata can be found at the Knowledge Network for Biocomplexity. However, EML's authors recognized that generation of metadata can be resource-intensive, and thus ensured that contributors could create simple data descriptions with minimal effort.

Overview of EML modules

An EML document focuses on providing metadata for a single resource, which can be a set of data, a literature citation, a scientific protocol, or a piece of software. For a given focal resource that is being described, a wide variety of other metadata types, including other resources, can be used. For example, when describing a `dataset` resource, it would be typical to reference one or more `literature` citations as well as descriptions of the data tables (entities) and the variables in those tables (attributes). Table 1 provides an overview of the 23 modules in EML 2.0.0.

Table 1: Brief descriptions of EML 2.0.0 modules

General modules	
eml	Overall wrapper containing EML metadata and additional metadata via extension
eml-resource	General description of a resource, containing fields common to all resources
eml-dataset	Fields for describing data resources, including tables, spatial images, and other entity types
eml-literature	Fields for describing literature resources in terms of their bibliographic citation
eml-protocol	Fields for describing a scientific protocol resource
eml-software	Fields for describing a software resource
Entities and Attributes	

eml-entity	General fields that describe all entity types such as tables and spatial images
eml-attribute	Fields for describing an attribute (variable) within an entity
eml-constraint	Fields for defining (mainly relational) constraints within and among entities
eml-physical	Fields for describing the physical format of data files and data streams
eml-dataTable	Specialized fields for describing tabular data
eml-spatialRaster	Specialized fields for describing spatial raster data
eml-spatialReference	Specialized fields for describing spatial data
eml-spatialVector	Specialized fields for describing spatial vector data
eml-storedProcedure	Specialized fields for describing stored procedures found in a data management system
eml-view	Specialized fields for describing views found in a data management system
Research Context	
eml-party	Fields for identifying and providing contact information for a person, organization, or role
eml-coverage	Fields for describing taxonomic, spatial, and temporal extent
eml-methods	Fields for describing scientific methods
eml-project	Fields for describing a research project that is associated with a data resource or other type of resource
Miscellaneous	
eml-access	Fields specifying access constraints for metadata and data
eml-text	A type for providing formatted text passages in various other modules in EML
stmml	A language for defining custom measurement units and the conversion factors relating units

Because EML is intended to be easy to implement, most of the modules in Table 1 are optional. Providing minimal metadata in an EML document (listing 1) is not considered best practice from a data management perspective. In an ideal world, ecologically relevant data would be described using the full suite of applicable EML modules. In a resource constrained world, scientists can more likely only provide the overall descriptions of their data sets (e.g., title, keywords, abstract, creators) for use in data discovery, descriptions of the data entities (e.g., tables), the variables within those tables (attributes), and the physical format of the data. This relatively minimal metadata, when provided in EML format, provides an excellent base for building shared data services, data processing services, analytical services, web-based data visualization services, and much more.

Benefits of EML Validation

One challenge of writing generic metadata-driven software applications is the difficulty in easily determining if metadata that is critical for the application is provided for a given data set. Most metadata that are available for ecological data are written in an ad-hoc syntax that prevents applications from quickly determining whether critical metadata are provided, and whether the format of the metadata file is a valid format that the application can parse.

EML solves this problem because EML documents are written in a standard syntax (Extensible Markup Language (XML)) and the required metadata content is specified using a standard schema description language (XML Schema). Thus, there are readily available software tools that application developers can use to process EML documents. These tools immediately indicate if there are errors in the format of a given EML document, if required metadata is missing, and whether the metadata that is provided is of the right type.

In addition to the EML rules that are expressed in the EML schemas, there are a few additional rules expressed only in the EML specification. The EML 2.0.0 distribution includes an open-source validation tool that can determine if any given EML document is valid, and if not where the problems are. This tool can be run from the command line, incorporated into existing software, or accessed online. Application developers and information managers that are creating EML exports from their data systems can use these tools to easily validate that the EML that they create and consume are valid, which will tremendously improve software interoperability and therefore make it easier for software developed by one institution to be used at others.

By making it simple to determine whether EML documents are compliant with the EML specification, the job has been simplified for software application developers to write metadata-driven software. Any document that is EML compliant has a known, predictable structure, and it is simple to validate this structure before trying to utilize the metadata found in the document. Consequently, the best-practice is for metadata providers to validate their EML documents before making them publicly accessible.

Data types and measurement unit definitions

For a scientist or a software application to understand a set of data, they must have detailed knowledge of the attributes (i.e., variables) that are contained in those data. At a minimum, they must know the number of attributes, their measurement domains, and their measurement units. EML provides a structured means for providing this information for each attribute (i.e., variable) in a data table.

The measurement unit is one of the most critical pieces of metadata needed to understand an attribute, and yet is also one of the most difficult to provide because a tremendous number of complex units are used in ecology. EML provides a solution by providing an extensible unit dictionary. This is basically a large set of formally defined measurement units (e.g., meters, kilometers, joules) along with the information needed to convert each unit to a fundamental SI unit. The current EML

2.0.0 release ships with definitions for over 190 units that are commonly used in ecology. In addition, the metadata for any given data set can define custom units that are derived from units in the EML unit dictionary or from fundamental SI dimensions. As EML development continues, we expect to provide more unit definitions and to make it easier to add units to the unit dictionary in a future EML release. This common set of measurement units represents an incredible resource for data integration and synthesis.

EML extensibility

Multiple metadata specifications are relevant to ecological data, and so EML was designed to allow extensible inclusion of these other metadata types in an EML document without affecting the validity of the EML document. Consequently, site-specific metadata, spatial metadata such as the Federal Geographic Data Committee's Content Standard for Digital Geospatial Metadata (CSDGM), and Dublin Core metadata, among others, can easily be inserted into an EML document. This extensibility also allows for inclusion of specialized metadata that might only pertain to one subdiscipline of ecology. Finally, EML metadata can also be transformed into many of the other metadata standards that might be needed by various disciplines, and contributors to the EML Project are working on standard tools to provide these conversion services. One common conversion will be to take an EML document and produce metadata for the NBII that conforms to the NBII Biological Metadata Standard.

Utilizing EML-documented data

As our community begins to provide more extensive sets of data described with EML metadata, we'll find it increasingly powerful to build metadata-driven software applications that streamline and automate the analytical process. Some existing work has started already in this area. For example, both Monarch and Xylopia are scientific workflow applications designed to automate and document complex analyses on arbitrary data sets. In addition, data catalogs are a common use for metadata, and they are even easier to construct using structured metadata such as EML. Some current examples of data catalogs constructed using EML include the Metacat system (see the Knowledge Network for Biocomplexity, the UC Natural Reserve System Data Registry, the Organization of Biological Field Stations Data Registry, and the Multi-Agency Rocky Intertidal Network Data System) and the Xanthoria system.

Current research for members of the EML team and others has begun to focus on the Science Environment for Ecological Knowledge (SEEK), a multi-institutional collaboration that is building a unified framework for data management and analysis for ecology. Many of the components of SEEK, such as the EcoGrid, will make extensive use of EML-described data.

The current release of Ecological Metadata Language 2.0.0 is available on the KNB web site. I encourage data managers and developers to download the release and tools for use in their systems, and I encourage scientists to find and use software tools that interoperate using EML.

This material is based upon work supported by the National Science Foundation under Grants No. DBI99-04777 and No. DEB99-80154. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author (s) and do not necessarily reflect the views of the National Science Foundation (NSF).

Information modeling: concepts, tools & techniques

- David Blankman, LTERNET

'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe:
All mimsy were the borogroves,
And the mome raths outgrabe.
-Lewis Carroll, excerpt from The Jabberwocky

'Twas brillig and the slithy toves: that is sometimes how it feels when trying to get a handle on the messy world of ecoinformatics. Initially I was asked to write about ER Studio, a database modeling tool. As I sat down to write, it struck me that it might make more sense to place ER Studio in the larger context of information modeling. If there is interest, I will follow up this introduction in future issues with a focus on one or more of the modeling tools.

The play's the thing...

Cast of Characters

- I.M. Overwhelm (IM): Our hero, perhaps superhero to be more accurate, a very bright, well trained expert in ecoinformatics, who though often under appreciated, is like Hercules in the Augean stables, asked to clean up some very messy situations.
- R.D.B. Meister (RDB): The champion of all the relational database workers. DB is mature, well liked, familiar, but seems to be growing some unusual appendages.
- X.M.Slover aka XML Schema Lover (XML):
- J. P. aka The Programmer aka The Whiz (JP): Cocky, a little arrogant, with a lot of powerful weapons (think Tom Cruise in Top Gun).
- Eko Informatics (Eko): Our guide, a wise, grey bearded one, with a magic wand, but can be a pedantic.

The setting

Like all good metalogs this takes place in a mythical land that bears a strong resemblance to your own workplace. The action zooms in and out of cyberspace. A few uncredited cameo appearances by well known and loved stars can be expected. There may also be a few relative unknowns making there debut to mass audiences.

And now our play...

IM: "I need help. I have legacy text metadata to convert, data to be analyzed, people and literature citations to track. My researchers seem to want more access to data. I have the feeling that this upcoming Decade of Synthesis is more like a decade of doom."

RDB, XML, JP (shouting randomly using different metaphors, different languages. Applying a universal translator the cacophony can be interpreted as): "I can help. I can solve all your problems. Pick me. Pick me. Pick me."

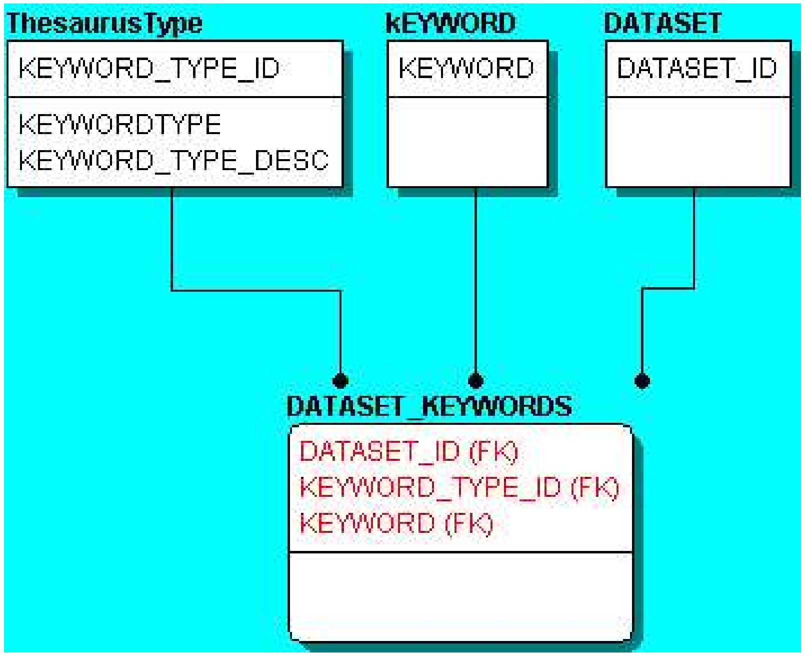
IM: "This is too much. I can't absorb all of this at once."

Eko: "Let me see if I can help. It seems to me as if you have several issues: information modeling, information processing and information sharing. Let's start with the first: information modeling."

Modeling Types

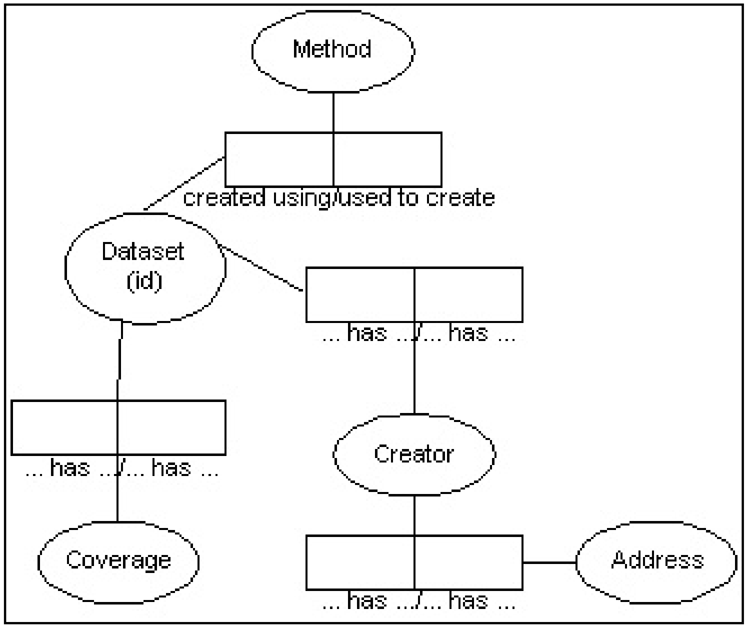
There are four basic types of modeling that you might have to do: Object Role Modeling, Entity Relationship Modeling (ER diagrams, database modeling), Object Modeling Language (UML modeling), and XML Schema modeling.

The kind of modeling that is most common is *entity relationship* (ER) *modeling*. An ER diagram looks like:

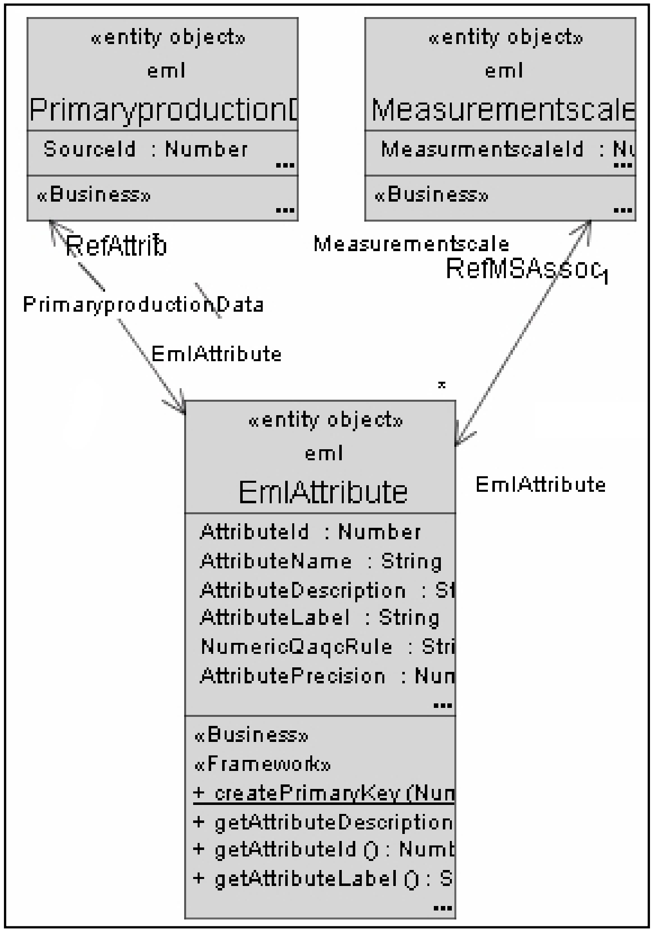


A more basic kind of modeling is represented by an approach called *Object Role Modeling (ORM).*

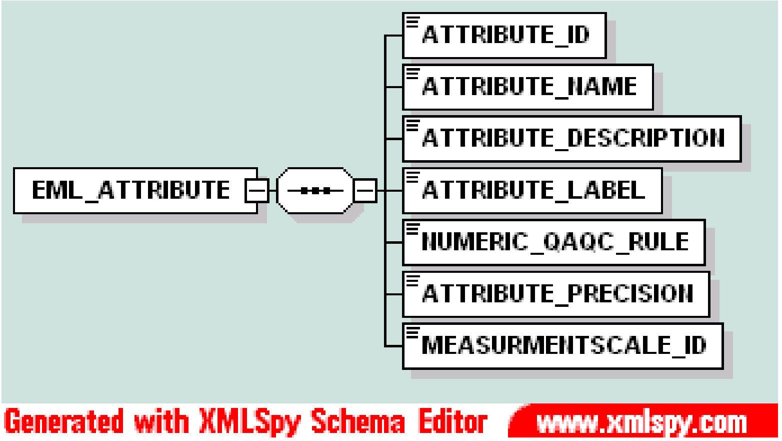
Object Role Modeling (ORM) is a powerful method for designing and querying database models at the conceptual level, where the application is described in terms easily understood by non-technical users. In practice, ORM data models often capture more business rules, and are easier to validate and evolve than data models in other approaches. (www.orm.net)
An ORM diagram might look like:

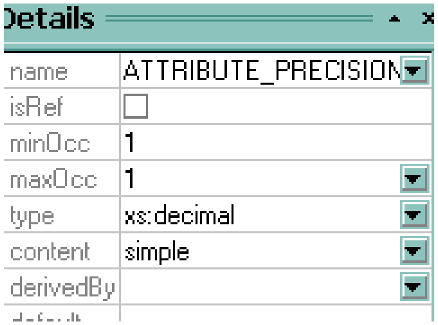


Next we'll consider *UML Modeling.* The OMG's Unified Modeling Language (UML) helps you specify, visualize, and document models of software systems, including their structure and design, in a way that meets all of these requirements. (You can use UML for business modeling and modeling of other non-software systems too.) Using any one of the large number of UML-based tools on the market, you can analyze your future application's requirements and design a solution that meets them, representing the results using UML's twelve standard diagram types. See http://www.omg.org/uml/. For example:



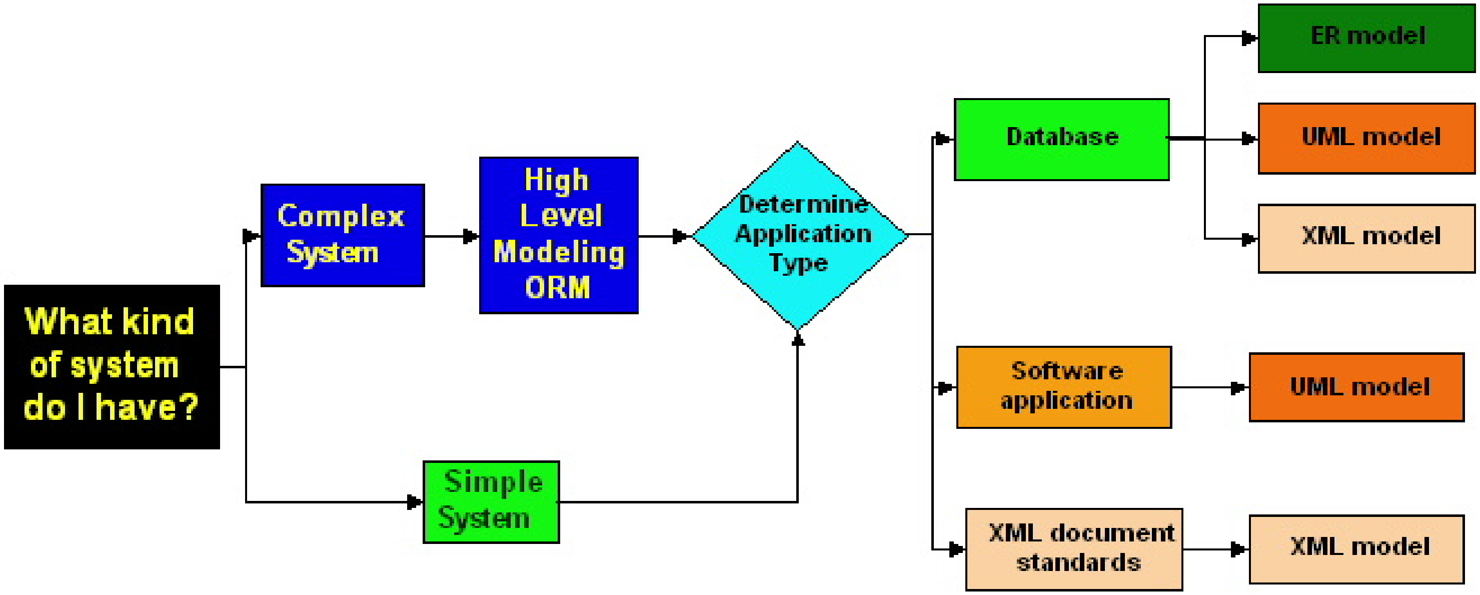
People are beginning to use *XML Schema* to model database schemas. XMLSpy can be used to generate a database based on a schema; however, care must be taken, because the hierarchical structure of XML does not translate into database schemas. For instance:





name	ATTRIBUTE_PRECISION
isRef	□
minOcc	1
maxOcc	1
type	xs:decimal
content	simple
derivedBy	

It is also important to understand the relationships among the four modeling types, briefly represented as follows:



Modeling uses and tools

Object Role Modeling

Use: ORM is the approach of choice when you are modeling a complex system, especially when you need to work with domain experts who are unfamiliar with database concepts. For experienced database modelers ORM can help prevent missing entities and can help the modeler see the basic patterns in a systems, thus avoiding missing the forest for the trees. ORM can be a helpful first step.

Tools: Microsoft Visio Professional can produce static ORM diagrams for planning and presentation purposes. Microsoft Visual Studio .NET Enterprise Architect allows ORM models to generate ER models based on the ORM model. This is a powerful set of tools from modeling to web application development to software development.

Entity Relationship Modeling

Use: Relational Database Modeling.

Tools: Visio Professional can reverse engineer existing databases. .NET Enterprise architect can reverse and forward engineer multiple database platforms (Oracle, SQL Server, Access, DB2 and others). ER Studio can reverse and forward engineer multiple database platforms (Oracle, SQL Server, Access, DB2 and others)

UML Modeling

Use: Database and web/software application modeling.

Tools: Visio Professional can create static diagrams. .NET Enterprise architect can generate code in multiple programming languages from UML Models. Oracle JDeveloper can generate code from UML Models. JDeveloper is primarily a java development environment

XML Schema Modeling

Use: Database modeling: can model tables but not relationships.

Tools: XMLSpy

Comment: While I have not worked with .NET Enterprise Architect, for educational institutions, it is impossible to match the price/performance ratio. From what I can tell, it will do everything that ER Studio can do plus a lot more. The MSDN subscription price for educational users is about $900. This is less than half the cost of ER Studio.

Conclusion

We've only barely scratched the surface of information modeling, but perhaps some of the issues are becoming clearer. Left for other discussions are the general topics of information processing and information sharing, as well as specific treatment of individual modeling tools. If you're still feeling a little brillig, you're probably not alone. Hopefully, though, developing a clear vision of the underlying issues will help to rescue your data...and keep it away from the slithy toves!

News Bits

A whirlwind tour of collaborative practice

- Karen Baker and Helena Karasti, Palmer Station

The Computer Supported Cooperative Work (CSCW) is a computer systems' research and development community that brings together the social and technical aspects for supporting collaboration. Karen Baker and Helena Karasti attended the biannual CSCW Conference in November 2002 (http://www.acm.org/cscw2002/). CSCW is sponsored by the Association of Computing Machinery (ACM). The ACM is an organization founded in 1947 (http://www.acm.org) to advance the skills of information technology (IT) professionals and students worldwide and it houses several special interest groups (SIGs) such as Groupware (SIGGROUP) and Management of Data (SIGMOD).

Karen and Helena participated at CSCW02 in a metadata-related workshop entitled "Storytelling and Collaborative Activities". A paper derived from our presentation will be published this year in the SIGGROUP Bulletin. In addition, we were able to attend several tutorials including "A Whirlwind Tour of CSCW Research", "Understanding Collaborative Activities and Applications: Methods for Studying Usefulness, Usability and Use of CSCW Systems", and "Collaboration Technology in Teams, Organizations, and Communities".

Helena has returned to the Oulu University in Finland, with her UCSD and LTER ties continuing. As a professor in the Department of Information Processing Science, she recently placed an ad (http://www.tol.oulu.fi/lter.html) for master's students to work with her on LTER materials collected during her year at UCSD. From the diverse ethnographic materials, final transcriptions of more than 50 interviews are being completed just this month.

In collaboration with Karen Baker and Geof Bowker, she has proposed to lead a workshop at the European Computer Supported Cooperative Work conference to be held in Helsinki, Finland this fall (http://ecscw2003.oulu.fi/). The workshop, titled "Computer Supported Scientific Collaboration", aims to bring together for the first time those interested in use of CSCW views and methods within the scientific arena. The workshop proposal is in recognition that much of the CSCW community work focuses on the business, medical, and education fields whereas the challenges that scientific collaborations pose for CSCW may be somewhat different.

The alignment of opportunities was most fortunate to support a one year study with the LTER community combining an ethnographic focus on technologically mediated work practices with participatory design (e.g. Information Systems Research in Scandinavia, http://iris.informatik.gu.se/). A report and paper are in preparation to serve as a continuation of the dialogue initiated with the LTER IM community in February 2002 at the IM Executive Committee meeting in San Diego and at the LTER IM Committee meeting in July in Orlando. Having introduced the concepts of sociotechnical systems and participatory design, plans are developing for Karen to visit Oulu in relation to attending ECSCW03 and for Helena to revisit San Diego. The visits allow continuation of analysis and co-writing, work with the students, and plans for future collaboration as well as of investigations into local floras, faunas and saunas.

International LTER workshop prioritizes information management

- Tim Bergsma, W.K. Kellogg Biological Station

In November 2003, scientists from Taiwan, P.R. China, Korea, and the U.S. met in Taiwan for an international workshop on LTER in Agriculture. Hosted by the Taiwan Ecological Research Network (TERN), the meeting considered factors related to the successful establishment of agricultural LTER sites. Workshop planners specifically requested that the delegation from KBS (the U.S. agricultural LTER) include an information manager -- illustrating the growing visibility of the role of information management in LTER. Tim Bergsma attended as information manager, accompanying station manager Mike Klug and project manager Andrew Corbin. Tim's talk drew on material from *Ecological data: design, management, and processing* (Michener and Brunt, eds., 2000), analyzed some KBS case studies, and included a brief description of EML. Copies of *Ecological data* were distributed to the other delegations.

Good Reads

Information Ecology

- Karen Baker, Palmer Station

Davenport, T.H. Information Ecology: Mastering the Information and Knowledge Environment. Oxford University Press, 1997.

Davenport in Chapter 1 makes clear his views on information management through presentation of a pair of lists that invite the reader to compare and contrast. He lists four beliefs of those looking to technology to solve our information challenges:

- information is easily stored in computers - as 'data';
- modeling computer databases is the only way to master information complexity;
- information must be common throughout an organization;
- technology change will improve the information environment

and four beliefs of those taking a more ecological approach to information management:

- information is not easily stored on computers-and is not 'data'
- the more complex an information model, the less useful it will be;
- information can take on many meanings in an organization;
- technology is only one component of the information environment

The brief lists are an effective method to highlight differences between technological and sociotechnical approaches. Be wary browsing this author on the bookstore shelf as there is a Thomas O. Davenport who has written a book "Human Capital: What It Is and Why people Invest in It" (1999) that details a popular contemporary management philosophy. It's interesting but distinct from Thomas H. Davenport's "Information Ecology".

The Invisible Present

- Karen Baker, Palmer Station

Magnuson, John, "The Invisible Present" in Ecological Time Series. T.M. Powell and J.H. Steele (eds), 1995.

A classic LTER article providing insight into the multitude of views derived from varying temporal time scales in ecological science. A series of graphs, showing the changing view resulting from opening up of a time series, illustrates a point fundamental to the philosophy of LTER. Although published earlier in BioScience as one of a trio of LTER articles

(1990), this adaptation of the article becomes part of a broader context when it appears within an ecological time series book that spans the land, ocean and human health domains.

◆ Calendar

May 5-6, 2003 NIS Advisory Committee Meeting, KBS, Hickory Corners, MI.

May 7-8, 2003 LTER Coordinating Committee Meeting, KBS, Hickory Corners, MI.

June 9-10, 2003 Core EML Workshop, Sevilleta, NM.

July 14, 2003 BDI grant proposal deadline.

August 12-14 NSF Workshop on Cyber-Infrastructure for Sensor Networks, San Diego, CA.

September 5-6, 2003 ILTER Annual Meeting, Beijing, P.R. China.

September 14-18, 2003 ECSCW 2003, the Eighth European Conference of Computer-Supported Cooperative, Helsinki, Finland.

September 18-21, 2003 LTER All Scientists Meeting, Seattle, WA.

September 22, 2003 LTER Information Manager's Meeting, Seattle, WA.