LTER Databits

Information Management Newsletter of the Long Term Ecological Research Network

Spring 2009

The Spring 2009 Databits is here and is full of some great articles: Following the presentation of ProjectDB at the 2009 LTER Science Council Meeting Wade Sheldon provides a great introductory article for those interested in getting started with XQuery and the native-XML database eXist, the primary technologies used for that project's development. Complimenting this is an article by Margaret O'Brien on the lessons learned from the ProjectDB workshops. A contributing guest author, Sonja Palfner, shares her experiences doing collaborative work in e-infrastructure. There are 'Good Tools' articles for the rapid development of MySQL table-editing interfaces, software for managing sensor-network data, an auto-documentation tool for relation databases and an online site providing online courses in software development and management for scientists and engineers. In 'News Bits', read about some important Information Management workshops that occurred. And enjoy a great set of three reviews of articles sharing a common thread central to the work of LTER information managers. Enjoy!

Featured Articles

Commentary

News Bits

Good Tools And Programs

Good Reads

Calendar

Featured Articles

Representing Geographic Features

edit

- Robert Ischiel Petersen (PAL)

When faced with storing geographic information one must determine how to represent that information in the storage system. There are a number of factors to consider such as the precision of the location, the coordinate system, the data storage mechanism, the types of features and the degree to which the location data will be specified. This is by no means an exhaustive list but to my mind represents some of the key factors.

The precision of the location information is dependant on the data that is to be referenced. Some data sets are sufficiently represented by location information with precision on the order of a kilometer, while other data require precision down to the centimeter scale. Another consideration is the precision to which a location can be established. This may depend on the equipment used: a legacy dataset many have navigation determined by sextant; GPS suitable for recreation may give a location that is accurate to 10 meters; and differential GPS systems can, with a sufficiently long site occupation, give a location accurate to the nearest centimeter. A storage system for recording this location information must therefore be capable of preserving the precision of the measurement.

While a location can be specified without a coordinate system, e.g. ‘the end of the Scripps Institution of Oceanography pier’ or ‘in the refrigerator behind the ketchup’, the efficacy of such references is dubious. More common is a set of coordinates representing the location. For the coordinates to make any sense they must be associated with a well-defined coordinate system. There are a wide variety of coordinate systems available. Some span the globe such as WGS84 or UTM. Other coordinate systems are more locally focused such as the State Plane coordinate system. Still other coordinate systems are unique to a specific application or area of study such as the California Cooperative Oceanic Fisheries Investigations (CalCOFI) project sampling grid. Regardless of the system used a set of coordinates paired with a coordinate system specifies a well-defined point. As such one can convert coordinates between coordinate systems. The details of such transformations are frequently non-trivial; nonetheless, they are common and in many cases computer codes already exist to facilitate these transformations.

Given that sets of coordinates require a coordinate system to be meaningful, the representation of location must therefore include the coordinate system. This is sometimes not evident when a community uses the same reference system tacitly; that is, it is when faced with data recorded in two different coordinate systems that the differences become evident and can be resolved if the coordinate systems are clear. Some representations explicitly include the coordinate system in the location data and in other cases the coordinate system is fixed for a representation and that system is specified as metadata.

Up to this point I have used the term “location” without a strict specification. It is now time to rectify this oversight. A location can be represented by various geometries. I’ll restrict the discussion here to two-dimensional geometries, though one can readily imagine including additional spatial or temporal dimensions. The simplest geometry is a point. A latitude and longitude pair is an example of a point. While the physical reality of a point, something of infinitesimal extent, may be questionable a point may well be a suitable simplification. A polyline is a collection of many points and includes the intermediate points along a straight line between specified vertices. This type of geometry is suitable for representing features such as the course of a river or track of a vessel. A polygon is like a polyline with the added requirement that the end point connect to the beginning point. It is also possible to have collections of these geometries. These are but a few of the simplest geometries available and give an idea of possible representations.

With these general considerations in place, we may now move from the abstract to the concrete and look at a few of the available representations.

Custom Database Storage

Storing a location in a database is often a convenient way to georeference data that is already is the database. The design of such a database is dependant on all of the above considerations and it is up to the designer to determine how to accommodate the geographic data. When storing a number in the database one must consider the precision requirements. Is an IEEE float sufficient or is a character field a better storage mechanism. Storing all locations as points can be accommodated with two columns in the case of decimal latitude and longitude or if storing degrees, minutes and seconds, six columns. More sophisticated geometries can be accommodated by leveraging the relational features of a database. The specification of the coordinate system may be universal to the database, e.g. all points are WGS84 or may be stored with each location. The benefits of such a storage system are that one may make the system as simple or complex as is necessary. The proprietary nature of such a system may make interactions with other systems more difficult.

Simple Feature

The “Simple Feature” specification is a standard published by the OpenGIS group. The Simple Feature specification defines an object model for geometries and can accommodate all of the geometries listed above. The standard defines two representations, “Well Known Text” (WKT) and “Well Known Binary” (WKB). While the specification object model accounts for a coordinate system or “Spatial Reference System” the WKT and WKB representations exist without a explicate reference to their coordinate system. Representing the feature as either WKT or WKB in a CLOB or BLOB column respectively could accommodate storage in an RDBMS. Using the WKT representation provides for arbitrary precision while the WKB representation suffers from the IEEE floating point limitations that one would imagine. As with the custom database representation the coordinate system would need to be specified either universally for all features or on a feature-by-feature basis.

KML

KML is a file format defined by Google and used with Google Maps and Google Earth. The format is based on XML. KML can represent all of the geometries listed above. In addition to representing geographic features KML supports the inclusion of additional data such as images and URLs and one can specify the display parameters of the features. KML is frequently stored as stand-alone file and can be imported to Google Earth or displayed by Google Maps using a URL referencing the document. Because it is a text document points can be represented with arbitrary precision. KML only supports the WGS84 coordinate system.

ESRI Shapefile

The Shapefile is the standard format used by ESRI GIS products. The term “Shapefile” is a bit of a misnomer as it is actually composed of several files. Shapefiles can accommodate all of the geometry types listed above. Shapefiles can also contain attributes associated with each feature. Each Shapefile references the coordinate system used. There are several libraries available that allow one to create and modify Shapefiles.

There are several options for representing geographic features that range from do-it-yourself solution to full-featured representation that include additional data and display information. Of course there are more options than those presented here. The ultimate choice of representation will impact the systems ability to represent the information accurately and interface with other systems. Currently there is no one clear standard and designers need to weigh all factors before settling on a particular implementation.

Example Case: Two LTER Oceanographic Sites

With Ocean Informatics work in conjunction with PAL and CCE LTER sites, where the data types, tool types, and logistics combine to result in a decision not to venture into the full realm of GIS, my work has involved evaluating how best to associate geographic information with biological oceanography datasets. In evaluating our requirements it was found that though the geographic information was recorded in proprietary coordinate systems storing the points according by latitude and longitude is preferred. The classes of features required the use of most of the geometries listed above. An additional consideration was to insure that the data could smoothly transition to a full GIS solution at some point in the future if necessary. Simple Features stored as WKT satisfied these requirements, could be stored in the database as a single column and provided a sufficiently rich representation without the overhead of other representations.

Further Reading

The following links may be of use to readers interested in the more technical aspects of computer storage and georeferencing.

- **Floating Point Errors** - http://docs.sun.com/source/806-3568/ncg_goldberg.html
- **Simple Feature** - http://www.opengeospatial.org/standards/sfa
- **ESRI Shapefile** - http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf
- **KML Documentation** - http://code.google.com/apis/kml/documentation/

Vocabulary Development as a Tool for Community-building

edit

- Lynn Yarmey (CCE/PAL)

Whether as formal as dictionaries or as tacit as locally-meaningful conversational words, vocabularies are a fundamental element of both science and daily life. As communicators, we easily slip between disparate vocabularies depending upon our context when speaking. Everyday examples include using different languages in foreign countries and the changing expressions we use when speaking with colleagues versus with neighbors or children. Formalizing vocabularies through publishing thesauruses or other reference lists creates an identifiable social commons that has the potential to improve understanding and communication.

A vocabulary reflects the perspectives of a community's participants; it may take different forms depending on community focus. For example, site scientists within the LTER may have a vocabulary list for different sampling locations and another for instrumentation types. Using these as a starting point, information managers (IM) can create separate vocabularies in the form of database field names and metadata content lists to best capture, describe and make available this local scientific information. For data to be shared, those diverse scientific and IM vocabularies must be brought together. This process is typically not easy and is frequently time-consuming, however the comparisons and discussions involved create an authentic foundation for cross-community understanding. The design and development of vocabularies at all scales serve to highlight and synthesize language in use within a community. As an articulation of common understandings, a vocabulary and the negotiations by which it is formalized can provide an important framework for social convergence and community development.

While technically straightforward, bringing two or more vocabularies together is an underestimated social challenge. Given that vocabularies can serve multiple purposes, the seemingly simple vocabulary convergence task becomes even more difficult. Keywords, for instance, are a type of vocabulary with a variety of roles ranging across scales and perspectives. At a given site, keywords are often drawn from locally-familiar terminology for the purpose of local data findability, and perhaps even preliminary data categorization. However, keywords also have multiple cross-community purposes such as use for public data discovery and creating relations to other organizations through mapped terms. For example, at CCE and PAL we are working on mapping our local geographical keyword list to authoritative sources including the Getty Thesaurus of Geographic Names (http://www.getty.edu/research/conducting_research/vocabularies/tgn/), and the Encarta World Atlas (http://encarta.msn.com/encnet/features/mapcenter/map.aspx). Use of synonyms, different levels of language formality, different organizational mandates, group histories and interests are a few elements of many that create vocabulary differences and thereby complicate convergence. The process of comparing and contrasting vocabularies certainly underscores the diversity of communities and their perspectives!

Vocabularies frequently capture the ways a community changes over time and can play an important role in achieving community agreement for future trajectories. Because vocabularies can be a version of an ideal context rather than being limited to existing practices, their development and maintenance represent an opportunity to lay out an infrastructural foundation for the future. An example would be a scientific community developing a (or adopting/adapting an existing) sampling vocabulary that includes not just current technical terminology but also terms from instrumentation and practices that are prospective additions to the group's sampling schemes. In this way, vocabularies can aid in solidifying a community's identity and goals while keeping a record of legacy practices as well.

As multifaceted and ever-changing representations of numerous social, organizational and technical community perspectives, vocabularies and their formalization offer an opportunity to identify, appreciate and negotiate the complexity and diversity within and across communities.

Cyberinfrastructure Travels: Sharing & Shaping Time, Space and Data

edit

- Sonja Palfner (Technische Universität Darmstadt)

The picture of the nomad drawn by Deleuze and Guattari makes an important point: The life of the nomad is the in-between, but he/she has a territory.

The nomad seems to fit perfectly, at first glance, as a representation of the life of today's scientist: hopping from one conference to another, from one (not always funded) project to the next, moving from one city/country/continent to the next locus... BUT: What is the "territory" of a scientist, what paths do exist and what paths do we follow?

"The nomad has a territory; he follows customary paths; he goes from one point to another; he is not ignorant of points (water points, dwelling points, assembly points, etc.). But the question is what in nomad life is a principle and what is only a consequence. To begin with, although the points determine paths, they are strictly subordinated to the paths they determine, the reverse happens with the sedentary. The water point is reached only in order to be left behind; every point is a relay and exists only as a relay. A path is always between two points, but the in-between has taken on all the consistency and enjoys both an autonomy and a direction of its own. The life of the nomad is the intermezzo."
(*Deleuze and Guattari* 1987: 380)

A scientist's life today seems to have much in common with the floating data-streams which are produced in an increasing way.

Floating people – floating data?

I have no answer. But sharing data, time and space (beyond "hand-shaking events") seem to be *conditions of possibilities* for following scientific paths that prevent ending up with a nervous breakdown in an uncomfortable hotelroom somewhere on this planet – somehow lost in space and time.

This month (June 2009) I have/had the great opportunity to take the time to visit US colleagues who are working on/in scientific cyberinfrastructures. I started with a visit with Geoffrey C. Bowker and Susan Leigh Star (authors of "Sorting Things Out: Classification and Its Consequences") at the Center for Science Technology and Society in Santa Clara. Now I have the exciting chance to stay a few days at Scripps Institution of Oceanography meeting Karen Baker and her Ocean Informatics team in San Diego. I also will go to UCLA for one day and my last stop will be the School of Information in Ann Arbor meeting with Paul N. Edwards and other colleagues.

So what am I doing?

I work on e-infrastructure (cyberinfrastructure) developments in Germany.

Sidenote: to say "in Germay" as I did it above, of course does not mean that the ongoing scientific and technological developments just stop at a countries border. Transnational as well as national growing cyberinfrastructures require more collaborative and comparative work: we are confronted with international, european and national e-infrastructure developments. What are the challenges and problems of such parallel and seperated but also overlapping endeavors which cannot be localised in national contexts? This is also a methodological question: how to follow these developments?

For one and a half year I've been in contact with the the German High Performance Computing Centre for Climate-and Earth System Research (DKRZ) in Hamburg/Germany. I conducted an initial participant observation at this Center in March, collected empirical materials (documents) and had several meetings with individuals in Germany (mainly at the DKRZ) who are involved in national and transnational cyberinfrastructure-projects (C3Grid, IS-ENES, EGEE). As you see, my interest is to get a better picture on the micro-level of the daily practices in working with technologies which are going on in science.

How does a new e-infrastructure grow and possibly change scientific culture?

I carry out document analysis as well as participant observation and interviews. Related to the research design, a further important question is how to „engage" in growing e-infrastructures as a social scientist: what could engagement mean dealing with both becoming part of ones object of study (e.g. the e-infrastructure for climate research) and maintaining a critical distance?

What is C3Grid?

C3Grid is a "Collaborative Climate Community Data and Processing Grid" on the national level. The aim of the C3Grid is to create a grid-based working environment for earth system research. This is a subproject of the D-Grid Initiative (www.d-grid.de) for the german earth system research "community" and has been running since 2005. Like the D-Grid, this project is supported by the German Federal Ministry of Education and Research (BMBF).

For more information look at www.c3grid.de/index.php?id=44&L=1 As you can imagine, in this short time C3Grid is more a prototype than a well established infrastructure for climate research.

The project ended officially recently. Like a lot of such short term funded infrastructure-projects, the C3Grid is applying for a second phase of funding.

Otherwise on the European level the FP7-project IS-ENES (Infrastructure for the European Network for Earth System Modeling) was launched a few month ago in France. So IS-ENES is an infrastructure for the European Network for Earth System Modelling (ENES).

For more information visit www.enes.org/IS-ENES.429.0.html

*** How to build up a sustainable (transnational) infrastructure with short term (national) funding?

When I started study on a micro-level at e-infrastructure in climate research by visiting the German High Performance Computing Centre for Climate- and Earth System Research, I "stumbled across" the history of the Center. It was built up 1987 as a "service unit" for German climate research/modelling. It provides not only a fascinating story about the becoming of complex models and the related growing amount of data in research. It is also a story about the shaping and re-shaping of climate research in relation to technological innovations and about the extremely problematic division between "service" and "science". And of course it is also a story about an upcoming national climate policy in the 1980 in Germany.

You may think that it is only interesting to collect and archive scientific data – but from a historical perspective (and also from a social science point of view), it is incredibly important to archive the institutional, organizational and social history of such institutions/projects.

This is what I hope to carry out this year: a little history of the German High Performance Computing Centre for Climate- and Earth System Research in Hamburg.

My view is that this work would be valuable if carried out as part of a larger project on e-infrastructure in science. Currently I applied for a grant from the German Federal Ministry of Education and Research (BMBF).

Research project: "Governance von Wissenschaft durch E-Infrastruktur. Fachspezifische Governance-Funktionen von E-Infrastrukturen und ihre Effekte/ Governance of science through (cyber)infrastructure".

The intended project will look at two growing e-infrastuctures in different scientific cultures: the climate science and the humanities. This study should lead to a comparative analysis of these two cases.

What is my theoretical interest?

My assumption is that e-infrastructure must be understood as a new "instrument of governance" for science. Governance – in a broad meaning – are all arranging-practices which create social order. Taking the notion of governance (which is mainly used for non-scientific and non-technological loci) and applying it to e-infrastructure, I strive to connect Political Science and Science and Technology Studies (STS). I try to make a shift from a study of the governance of science (e.g. through funding programmes) to a study of a governance of science through new (cyber)infrastructures.

My main research questions are threefold:

1. What instruments of governance are delegated to e-infrastructure (and thereby become invisible)?
2. How does this happen?
3. What are the effects on scientific work and their objects?

These questions may seem at first to be outside the realm of information management. But they seem to me, at least, to be related to the question of the role information managers play in scientific cyberinfrastructures. In fact, I am not familiar with this role in the US context. In the case I am studying, the role of an information manager is not defined as part of the whole project. But the work must be done. So, that means the work has to be done by someone, this work that seems to be mostly invisible and thereby unacknowledged. These situations are often related to institutional framings; and indeed the power of institutions in distributing work and defining work roles should not be underestimated.

Short Curriculum Vitae

2003: Diploma at the Otto Suhr Institute for Political Science, Freie Universität Berlin.

2005-2007: PhD Student, DFG Research Training Group „*Gender as a Category of Knowledge*" at Humboldt Universität zu Berlin.

Dissertation „Gen-Passagen. Eine Studie zu molekularbiologischen und medizinischen Praktiken im Gefüge der Brustkrebs-Gene", Otto Suhr Institute for Political Science, Freie Universität.

Since October 2008: Postdoctoral Researcher, DFG Research Training Group „*Topology of Technology*" at Technische Universität Darmstadt.

Research Interests: Science, Technology and Society, Gene-Technology and Predictive Medicine, Scientific E-Infrastructures, Interdisciplinary and Comparative Approaches and Qualitative Research Methodologies

Contact:

Dr. Sonja Palfner
Technische Universität Darmstadt
FB 2 - Graduiertenkolleg „Topologie der Technik"
Karolinenplatz 5
64289 Darmstadt

Germany
E-Mail: spalfner@gmx.de

Getting started with eXist and XQuery

edit

-Wade Sheldon (GCE)

Introduction

Two recent LTER workshops were convened to plan and develop ProjectDB, a cross-site research project description language and database (Walsh and Downing, 2008). During the first workshop participants agreed to use eXist, an open source native XML database (http://exist.sourceforge.net/), as the back-end system for storing and retrieving research project documents. This database was primarily chosen to leverage ongoing software development work at CAP LTER that uses eXist, but excellent query performance, built-in support for both REST and SOAP web service interfaces, and simplicity of configuration and administration were also influential factors.

The combination of eXist and XQuery (the XML query language used by eXist) proved to be extremely effective for ProjectDB, exceeding all expectations. A working group of six Information Managers and a CAP software developer designed and implemented a complete system for storing, querying, summarizing and displaying research project documents in just a few days, including RESTful web services to support all the primary use cases identified during the planning workshop (Gries et al.). The rapid success of this development effort has sparked interest in eXist and XQuery across the LTER IM community, and this article presents an overview and brief guidelines on how to get started using this new XML technology.

eXist Database

Many options are currently available for storing and searching XML documents. Recent versions of major relational database management systems support XML storage and retrieval natively, and software packages like the KNB Metacat (http://knb.ecoinformatics.org/software/metacat/) XML-enable conventional relational databases by decomposing nodes into text strings and providing XML-centric document models and supporting technology. In contrast, eXist represents a new breed of native XML databases (NXD) that use XML documents as logical units of storage rather than rows in a table, and employ XML-based specifications and technology for all operations (e.g. XQuery, XPath, XML Infoset, XSLT). Other NXDs include Apache Xindice (http://xml.apache.org/xindice/), BaseX (http://basex.org/) and OZONE (http://www.ozone-db.org/), but eXist stands out due to its excellent XQuery implementation, built-in support for a broad range of web technology standards, and active developer community.

The eXist database is implemented entirely in Java, so it can be run on most computer platforms. The only dependency is that a complete JDK (Java Platform) 1.4.2 or higher must be present on the system prior to running the installer .jar file. A simple Java web server (Jetty, http://www.mortbay.org/jetty/) is installed by default to provide web-based access and REST/SOAP support, but eXist can also be integrated into an existing Apache Tomcat installation using a provided .war file. Once installed, the database can be administered using XQuery-based web forms or a Java client application.

Fine-grained access control is supported, and users and groups can be defined and managed internally by eXist or optionally retrieved from an LDAP server. In addition to the supplied interfaces, eXist can be configured as a data source in the oXygen XML editor (see http://www.oxygenxml.com/demo/eXist/eXist.html), which proved to be invaluable during the ProjectDB development workshop.

Storing and retrieving XML data using eXist is extremely easy and fast compared to relational databases, because documents are kept intact.

XML files are simply uploaded or downloaded using the management interfaces, HTTP GET or PUT commands, or by clicking on files in the oXygen database connection view. Documents can be stored in a single root directory or organized into hierarchical “collections”, similar to directories in a file system. Besides organizing documents, collections are useful for restricting or “scoping” queries to include a specific subset of documents and for providing different levels of access for specific users or groups. XML documents must be well formed to be stored, but are not validated by default; however, both implicit validation (rejection of invalid documents) and explicit validation (validation report after insertion) can be configured, if desired. Non-XML documents can also be stored in eXist, including XQuery files, XSL and CSS style sheets, HTML and JavaScript files, and binary files like images, making eXist ideally suited as a back-end for web applications and content management systems.

As already mentioned, one of eXist’s strengths is built-in support for many web technologies, including XQuery, XSLT (1.0 and 2.0), REST, SOAP, XMLRPC and WebDAV. For ProjectDB, we focused on eXist’s REST interface which allows documents to be stored and retrieved, XQueries run, and XSLT transforms performed using parameterized URLs (i.e. HTTP GET requests) or other standard HTTP commands (i.e. POST, PUT, DELETE) and HTTP authentication. Collections are automatically mapped as virtual directories below the base directory (i.e. /exist/rest/db/), providing logical access to any files stored in the system. For example, the following URL retrieves a GCE research project document from the LNO eXist database using the REST API:

http://amble.lternet.edu:8080/exist/rest/db/projects/data/gce/gce_hammocks_project.xml

XQuery Language

As mentioned above, XQuery is the primary language used to search and manipulate data stored in eXist and other native XML databases. The XQuery 1.0 specification was accepted as a formal W3C recommendation in January 2007, along with 7 related specifications including XQueryX 1.0, XSLT 2.0 and XPath 2.0. XQuery provides many of the same capabilities for working with XML documents and collections that the SQL Data Modeling Language provides for relational databases, including searching, filtering, joining and re-factoring data dynamically. XQuery 1.0 leverages the document model, document navigation syntax and function library of XPath 2.0, as does XSLT 2.0.

Unlike XSLT, however, XQuery is not itself an XML dialect so queries are simpler to read and write and special characters do not need to be encoded.

Simple queries can be performed using XPath syntax alone (i.e. called path queries), but most XQueries are written as FLWOR (pronounced flower) expressions to provide more control over the data returned.

FLWOR is an acronym for For, Let, Where, Order by and Return, which are the keywords used for statements in the query definition. The For statement is used to define a set of nodes to iterate through when evaluating the expression (i.e. collection, document, or specified part of a document or collection of documents). The Let statement is used to bind XML document nodes or other data to a variable, for programming convenience and more efficient evaluation. Note that multiple For and Let statements can be used in a single query, and that Let statements can precede For statements unless they reference nodes in the iteration. The Where and Order by statements restrict nodes based on specified criteria and control ordering of the results, resp., similarly to their SQL counterparts. The Return statement, which is the only statement that is actually required in any XQuery, specifies the structure of the output document. XQueries can return an XML fragment, XML document, HTML, or plain text depending on the query implementation. Specific examples from the ProjectDB development effort are described below, and many more examples are provided in the XQuery Wikibook online (http://en.wikibooks.org/wiki/XQuery) and various XQuery references (Walmsley, 2007).

ProjectDB Examples

A number of XQuery files were written during and after the ProjectDB development workshop to support the targeted web services. These files are stored in the LNO eXist database and archived in the LNO SVN server, and are all available for review and customization (see http://intranet.lternet.edu/im/project/LTERProjectDatabase/userguide for more information). These queries range from simple path queries to complex multi-parameter searches with derived return documents, and all contain code documentation and comments.

For example, getProjectById.xql searches all research project documents and retrieves the document with an lter:researchProject id attribute matching a specified id parameter, using the FLWOR expression below (abbreviated for illustration):

```
xquery version "1.0"; let $id:= request:get-parameter("id",“”) for $researchProject in
collection('/db/projects/data')/lter:researchProject[@id = $id] return $researchProject
```

In this query, the `let $id…` statement binds the variable $id to the query input parameter `$id`, defaulting to “” (empty string) if no id parameter is provided as input. Note that `request:get-parameter` is an eXist function used to interact with the web server session to retrieve parameters from the HTTP request. The `for $researchProject …` statement iterates through the collection of documents specified by the path statement, binding nodes from each iteration to the variable `$researchProject`. The XPath `collection` function is used in this query to search across all XML documents in the corresponding collection in the eXist database. Note that in this query, `[@id = $id]` defines a restriction in the XPath, so that only documents with an id attribute in `lter:researchProject` matching the input “id” will be returned. The `return …` statement simply references the `$researchProject` variable, resulting in the entire node set (research project document in this case) being returned without modification as output.

A more complex example is provided by getProjectsByKeyword.xql, which returns a derived XML document summarizing content from all research project documents matching site and keyword search parameters:

```
xquery version "1.0"; (: get input parameters from http request :) let $keyword := request:get-parameter("keyword","") let $keywordSet
:= request:get-parameter("keywordSet","") let $siteId := request:get-parameter("siteId", "") return if (string-length($keyword) > 0)
then ( <projects> {   for $p in collection(concat('/db/projects/data/',   lower-case($siteId)))/lter:researchProject     let $title :=
$p/title/text()     let $idstr := $p/@id     let $time := $p/coverage/temporalCoverage     let $kw := if(string-length($keywordSet)>0)
    then $p/keywordSet[@name=$keywordSet]     else $p/keywordSet     where matches($kw/keyword,$keyword,'i')     order by $idstr
    return     <project id="{$idstr}">     <title>{$title}</title>     {for $c in $p/creator     let $individual := $c/individualName
    let $userid := $c/userId     return     <creator>{$individual}{$userid}</creator>}     {for $ap in $p/associatedParty     let
$ap_name := $ap/individualName     let $ap_id := $ap/userId     let $role := $ap/role     return     <associatedParty>{$ap_name}
{$ap_id}{$role}     </associatedParty>}     <keywordSet>{for $k in $p/keywordSet/keyword     let $kwd := $k/text()     return
    <keyword>{$kwd}</keyword>}     </keywordSet>     {$time}     </project> } </projects>) else (<projects/>)
```

In this query, three input parameters are supported (“site”, ”keywordSet” and ”keyword”), but only the “keyword” parameter is required as dictated by the if-then-else logic and empty string test. The “site” parameter is used in the XPath to restrict the `for $p in …` statement to a specific site collection in the LNO eXist database, by concatenating the `$site` string to the base collection path (i.e. because a separate collection in `/db/projects/data` was created for each participating site during the workshop, e.g. `/db/projects/data/cap`). If the “site” parameter is omitted or blank a double slash appears in the path, effectively removing the site restriction so that all documents in `/db/projects/data` are searched (i.e. because in XPath syntax, a double slash denotes a child node at any depth level below the parent node).

Following the `for $p in …` statement that sets up the main iteration, several let statements are defined to bind nodes to variables for use in generating the output document (`$title`, `$idstr`, `$time`). The statement

```
let $kw := if(string-length($keywordSet)>0) then $p/keywordSet[@name=$keywordSet] else $p/keywordSet
```

employs if-then-else logic to bind `$kw` to a different node-set depending on whether a `keywordSet` was specified as input or not (i.e. based on non-zero string length of `$keywordSet`). If a `keywordSet` was specified as input then a restriction is included in the path, otherwise no restriction is used so `keywords` in any `keywordSet` will be searched.

The statement `where matches($kw/keyword,$keyword,'i')` applies the main keyword match restriction. Only documents with keyword text nodes matching the input parameter `keyword`, based on a case-insensitive regular expression match (i.e. indicated by the `i` flag), will be returned from the query. The `order by $idstr` statement then specifies that documents should be returned ordered by `researchProject` id.

The code in the return statement defines the structure for the query results, in this case an XML document that contains a subset of content from each matched project document in a separate `<project>` element, nested within a `<projects>` root element. Variables from the `for` and `let` statements in the XQuery are denoted in the return schema by curly braces, and these variables are combined with direct XML element constructors (i.e. literal XML tags) to generate the desired output. Note that additional XQueries are embedded in the return section, also denoted by curly braces, to generate nested elements containing a specific subset of the elements in the original documents (e.g. portions of the creator and associatedParty nodes). In contrast, other nodes (e.g. `coverage/temporalCoverage` denoted by `{$time}`) are copied intact to the output document.

The syntax for running these two example XQueries in the LNO eXist database using the REST API is as follows:

```
getProjectById.xql: http://amble.lternet.edu:8080/exist/rest/db/projects/util/xquery/getProjectById.xql?id=knb-lter-gce.p1.1
getProjectsByKeyword.xql: http://amble.lternet.edu:8080/exist/rest/db/projects/util/xquery/getProjectsByKeyword.xql? keyword=Temperature
```

Note that the queries stored in eXist are configured to return results using an XHTML doctype without an XML declaration for broadest web browser compatibility, but viewing the page source will display the XML output. Also note that query results can be styled on the server if an XSL URL is specified using the “_xsl” parameter, as follows:

```
http://amble.lternet.edu:8080/exist/rest/db/projects/util/xquery/getProjectsByKeyword.xql?
keyword=Temperature&_xsl=/db/projects/util/xslt/lterProjectsListText.xsl
```

When the XSL file is stored in an collection in the same eXist database as the XQuery, as in this case, the relative database path to the XSL can be used; however, a fully-qualified URL for an external XSL file can also be specified if XSL files are stored on a web server and not in eXist. More complex XSL stylesheets (e.g. containing import directives, parameters, and other advanced options) can also be referenced to display query results in a complete web site template, as in the following GCE example:

```
http://amble.lternet.edu:8080/exist/rest/db/projects/util/xquery/getProjectsByKeyword.xql?
keyword=Temperature&_xsl=/db/projects/util/xslt/gceProjectsListText.xsl
```

Concluding Remarks

The eXist XML database and XQuery language are powerful tools for working with XML, but they are also relatively easy to use making them well suited to rapid application development and deployment. The simple installation, low resource requirements, and low administrative overhead of eXist also makes it a very practical tool for use in local LTER site information systems. In informal performance testing, queries of thousands of complex research project and EML documents in eXist took only seconds to run, indicating that this database should scale well for production use.

Given these strengths, the question of how eXist compares to Metacat naturally arises. As mentioned in the introduction, Metacat is a hybrid XML-RDBMS system, making it inherently more complex to install and administer than eXist and requiring far more system resources. The back-end RDBMS (e.g. Oracle, Postgres, etc.), a Java Servlet engine (Apache Tomcat) and the Metacat Java software need to be installed, configured and administered separately. In addition, the pathquery syntax used to query the database is unique to Metacat (see http://knb.ecoinformatics.org/software/metacat/metacatquery.html), and considerably less flexible than XQuery. The overall structure of the return document is also fairly rigid, generally requiring an extra XSL transformation step to produce useful output. Lastly, Metacat relies on versioned document ids for managing files and does not support collections or other organizational structures, requiring much more planning and discipline when storing documents and limiting options for restricting searches.

I believe that these characteristics limit the utility of Metacat as a general repository for XML and other files needed to build web applications and web services, and make it less suitable for collaborative application development than eXist (e.g. the LTER Research Projects database). However, Metacat provides very important benefits over eXist for archiving XML documents, including versioning, strong validation, replication, interoperability with other relevant EcoInformatics software (EcoGrid, Morpho, Kepler, SRB) and services (MapServer, LSID), plus support for automated document harvesting. Consequently, Metacat is a superior database for long-term storage of EML metadata and other authoritative XML documents in LTER and other environmental science networks.

For information about all the XQuery-based web services deployed for the ProjectDB database, please visit the LTER Research Project Database page on the LTER Information Management website (Gries et al.).

References

Walsh, J. and Downing, J. 2008. PROJECTDB – Planning and Development of a Collaborative Programming Effort. LTER Databits - Information Management Newsletter of the Long Term Ecological Research Network. Fall 2008 Issue. (http://databits.lternet.edu/node/33)

Gries, C., O'Brien, M., Porter, J., Sheldon, W., Walsh, J. and Bohm, S. Documentation for the LTER Research Project Database. LTER Information Management Website. (http://intranet.lternet.edu/im/project/LTERProjectDatabase/documentation)

Walmsley, P. 2007. XQuery. O'Reilly Media, Inc. Sebastopol, California. 491pp.

Commentary

Lessons learned from the projectDB workshops

edit

- Margaret O'Brien (SBC)

The 2008-2009 workshops to plan and construct the Projects Database ("projectDB") employed a different model of collaborative development for the Information Managers Committee. In these workshops information managers dedicated time to developing a common solution for a common need rather than proceeding individually or using occasional meetings to advance a collaborative project. Participation was entirely voluntary; a total of ten sites were represented in the planning workshop, and six of these in the coding workshop. I believe that everyone involved agrees that using a workshop format was a good choice, particularly in three areas: speed of output, the general scope that a network-wide workshop accommodates, and that workshops both take advantage or our diverse skills and promote the development of new ones.

1. **Speed.** Most obviously, it proved efficient for a group to dedicate a block of time to developing code. As expected, workshops resulted in much faster output than the model in which information managers try to advance a collaborative project with occasional meetings. This is not intended to detract from our use of VTCs, but to efficiently create code, there is no better way than to remove local distractions.
2. **Scope.** Many site IMs had expressed the need for a database for their sites' research projects, and some were considering development. At the first workshop, nearly half the sites were represented. This broad scope generated many use cases: a container for research products (citations, datasets, or images), a record of visitors and/or permits at a field station, or consolidated text for creating funding agency reports. All of these uses were accommodated by the schema. In considering the range of needs and required code, we also gained knowledge about the amount of work which could be reasonably accomplished in one week.
3. **Skills.** By collectively designing a modular project, each participant was able to focus in a specific area. This meant that we could take advantage of our combined expertise in XML, EML, XSLT and various web development strategies. By choosing the EML-project module, the workshop participants gained in-depth knowledge about other components of that schema, and the IMC is actively contributing to EML's further development. Additionally, the timing of the projectDB workshops, i.e., following the 2008 training workshop on XML technologies, means that most sites are now equipped to make use of the new database with little additional training (e.g., see the article by Wade Sheldon in this issue).

Coding workshops represent a different model of collaborative development to address our common needs, as opposed to relying on the usual 1 IM:1 site model in which one person covers all bases– a situation that will become increasingly difficult to maintain as internet technology grows. The participation of site information managers in the planning of network-level databases avoids the establishment of central, top-down repositories and their mandated contributions in favor of databases which both address the needs of the members and promote the network "look-and-feel". Working collectively on a modular project allows each participant to further his/her expertise in a chosen area. This collaborative model could help to define standards and practices for developing network-level cyberinfrastructure.

News Bits

Pacific Coast Zooplankton Working Group: Data and Information Infrastructure

edit

- Karen Baker (PAL/CCE)

A Pacific Coast Ocean Observing System (PACOOS; http://pacoos.org) zooplankton workshop was held 9-10 June 2009 at Scripps Institution of Oceanography at University of California, San Diego. PACOOS is a large marine ecosystem (LME) program for the California Current so attendees represented ongoing oceanographic sampling programs reaching geographically from Baja, Mexico to Canada. Objectives included considering scientific interests, policy needs and data arrangements for formalizing a zooplankton community as well as developing dialogue, vocabulary and agreements regarding data sampling, analysis, and sharing among data providers. In addition, benefits, roles and responsibilities of data management participation were considered. The workshop effort may be considered an exploration of and a prototype for how to support collaborative biological group efforts and their data. One goal of the meeting was the identification and formalization of a zooplankton community comprised of both scientists and data management participants. Zooplankton researchers from along the Pacific coast of the U.S. have a history of informal collaboration; this meeting served as a forum for considering and reaching agreement on the potential benefits of formalizing some aspects of the collaboration. A community goal is to coordinate development of products through knowledge sharing and management of zooplankton data. Participants from both the Integrated Ocean Observing System program (IOOS) and from the National Coastal Development Data Center (NCDDC) expressed interest in coordinating with the workshop biological data efforts.

Co-conveners of the workshop came from three distinct but interrelated scientific arenas: Jonathan Phinney, a PACOOS Project Manager at Southwest Fisheries Center; Karen Baker, an LTER Information Manager; and Sharon Mesick, an Ecosystem Program Manager from NCDDC. Members of the Ocean Informatics team (James Conners, Mason Kortz, and Lynn Yarmey) supported the working groups as rapporteurs. The workshop focus was 'data and information infrastructure' with particular attention to the standards-making process, joint activities, and community-building. Each co-convener paired with an Ocean Informatics team member to lead discussion on one of three questions: (1) Community issues: infrastructure requirements for a PACOOS network, e.g. what are zooplankton standards? (2) Data policy issues: zooplankton information sharing goals, e.g. what are data sharing issues and responsibilities? (3) Data documentation issues: infrastructure methods, models and the development process, e.g. what are representative data types, characteristics, and metadata?

Three sub-groups representing a mix of participants - zooplankton biologists, data managers, and national interests - and of geographies were formed for working group discussions. Throughout the workshop combination of presentations, demonstrations, and working groups, an emphasis was placed upon the concept of 'horizontal integration' needed within and across communities as well as internal to the National Ocean and Atmospheric Administration (NOAA) in alignment with their broadening interest in biological data work.

Biological data are complex and difficult to convert into well-described data objects given today's nascent standards that have been developed largely for physical data. A project funded to prototype creation of a queriable application across two zooplankton project datasets at different institutions was reported (http://oceaninformatics.ucsd.edu/zooplankton/) as a case study carried out by researchers responsible for data collection (from sampling plan together with sample collection and analysis) in collaboration with the Ocean Informatics team at SIO. Throughout the working group, care was taken with management of expectations and joint understandings in terms of what is involved in negotiating the boundary between large-scale and local-scale situated solutions. Focus was on recognizing the multiple types and scales of data work at differing levels requiring both mediation work and changes in data practices. Bringing together zooplankton researchers and information managers as partners from different contexts and scales is critical for collaborative data efforts and information infrastructure-building as well as for further development of our limited understanding of data differences. The workshop itself was an example of what can be accomplished with scientists, data managers, and program managers working together.

Dynamically Plotting the Future: An International Information Management Workshop for Forest Dynamic Plot Databases

edit

- John Porter (VCR) & Meei-ru Jeng (TERN)

The cicadas were singing loudly in the trees at the Leihuachih Research Center of the Taiwan Forestry Research Institute (TFRI) in Nantou Taiwan, when an international group of forest researchers and information managers started work on developing new systems and methodologies for the study of dynamic forest plot databases in mid-June 2009. Their goal was to build on a series of workshops focused on IM Training in the East-Asia Pacific ILTER Region, by providing concrete, scientifically valuable products that exploit advanced information management techniques. The workshop was hosted by Chau Chin Lin and the Information Management Team of TFRI with funding from the Council of Agriculture. The forty participants included representatives from the East-Asia Pacific region including Japan, Malaysia and Taiwan and four U.S. LTER Information Managers.

Information Management Workshop for
Forest Dynamic Plot Database
Nantou, TAIWAN 15-19 June
2009

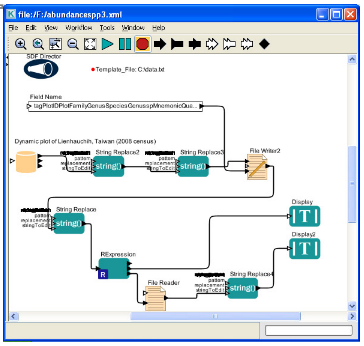
The workshop focused on the analysis of data from large dynamic forest plots, many, but not all of which, are associated with the Center for Tropical Forest Science (CTFS). It began with scientific presentations by I-Fang Sun (Taiwan), Abdul Rahman bin Kassim (Malaysia), Kaoru Niiyama (Japan), that addressed the science issues related to large forest plots. Yu-Yun Chen (Taiwan) and Eda Melendez (US) presented information on the existing data structures used by CTFS for forest dynamics plots.

Participants then used a brainstorming session to identify scientific needs associated with the analysis of plot data, the analysis of ancillary data (e.g., mammals on forest plots), quality control and assurance and visualization. They then consolidated their vision to focus on particular analysis tools and how they would be used. Each regional group also included information managers, who then worked on defining a system architecture that could be used to create the needed tools, while the forest plot researchers worked on further defining and prioritizing the functionality they needed. The system they proposed focuses on exploiting the capabilities in Metacat, an Ecological Metadata Language (EML) database, linked to the Kepler scientific workflow system, which, in turn, exploits an existing CTFS library of R statistical language scripts designed specifically for forest plot data.

During the workshop, new EML documents were prepared for forest plot data at several locations in Malaysia, Japan, Taiwan and the Luquillo LTER site using the Morpho editor. In the meantime, the IM-Team of the TFRI and the Taiwan Ecological Research Network (TERN), in particular, Chi-Wen Hsiao and Cheng-Tsai Chou, provided support and training in the use of R and Kepler to scientists and visiting IM's alike. With their help, participants prepared several prototype workflows, including some that for the first time provided comparative analyses of these forest plots. There are plans for additional workshops in coming years to build on these successes, with the aim of advancing scientific progress in the East-Asia Pacific Region.

Participants also got a chance to tour the Forest Dynamics Plot at the Leihuachih Research Center and to tour nearby Sun-Moon Lake, Checheng ecological village, an organic tea farm and historic villages. U.S. LTER Information Managers participating in the meeting were Kristin Vanderbilt (chair of the ILTER IM committee), Don Henshaw, Eda Melendez and John Porter.

Good Tools And Programs


file:/F:/abundancespp3.xml
File Edit View Workflow Tools Window Help
SDF Director
Template_File: C:\data.txt
Field Name
tagPlotIDPlotFamilyGenusSpeciesGenusspMnemonicQua...
Dynamic plot of Lienhauchih, Taiwan (2008 census)
String Replace2
String Replace3
File Writer2
String Replace
Display
Display2
RExpression
R
File Reader
String Replace4
pattern
replacement
stringToEdit
string()
T

LEADUM Software DBScribe

edit

- Wade Sheldon (GCE)

LTER Information Managers have been promoting the importance of documenting environmental data and other research products since the earliest days of the network, but less attention has been paid to the need to fully document the information systems used to archive and deliver these products. At GCE, all custom software code is well documented and we've been using SVN for several years to track changes to software and website code. However, writing good documentation for relational database designs has proven challenging. EML can be used to document database objects and relationships, and SQL scripts (source code) can be generated from many RDBMS systems, but neither of these options produces very readable output for communicating a database design. To date, we've primarily archived graphical images of entity-relationship diagrams and relied on stored SQL scripts in our SVN (and database backups) to archive the complete structure.

Recently, I was asked to design a database for the National Park Service to manage metadata about water quality monitoring programs operating in the Southeast. The database design was one of the deliverables for this project, so I decided to look for a better database documentation solution. While reviewing various options, from shareware programs to major commercial packages, I discovered DBScribe from LEADUM Software (http://www.leadum.com/products.php?prod_type=DBScribe). This software analyzes every detail of the database and generates comprehensive documentation in MS Word, HTML or Microsoft Help format, complete with cross-references, lists of dependencies and navigation links. Any attribute and table annotations are automatically used as descriptions, and additional descriptions can be added using an object browser interface after the analysis phase is complete. Details to include in the report can be customized at several levels, and then saved in XML format to re-us

e when regenerating documentation after changes to the database structure. Sensitive details (e.g. logins, database configuration, etc.) can be excluded from the report for security reasons. The MS Word report format includes a table of contents and document links, which are automatically converted to hyperlinks and bookmarks when converted to PDF format by Adobe PDF Writer.

DBScribe proved very useful for our NPS project, generating hundreds of pages of very readable (and attractive) documentation in a few minutes, so we're now using it to document all GCE production databases. Versions are available for all major RDBMS systems, priced from $39-$99 depending on database platform. A fully-functional 30-day trial is also available for evaluation.

Software Carpentry

edit

- Wade Sheldon (GCE)

Software Carpentry (http://www.swc.scipy.org/) is an online course designed to teach scientists and engineers how to design, build, maintain and share computer programs efficiently. The authors' premise is that many scientists and engineers spend much of the lives programming, but few have ever been taught how to do this well; consequently, they spend most of their time wrestling with software instead of doing research and solving problems, and they have no idea how reliable or efficient the programs they write are. Lectures are included on using version control systems (e.g. SVN), automating building and testing software, basic scripting, debugging, quality assurance, server monitoring, network monitoring and a host of other topics.

The authors make a case for using high-level, interpreted languages for most scientific programming (e.g. Python, MATLAB and others), arguing that programming efficiency and code readability (i.e. human time) have a greater impact on scientist productivity than raw software performance (i.e. machine time), particularly on modern computing hardware. Most examples are therefore written in Python, and additional material is being developed for MATLAB. Not surprisingly, both the Python Software Foundation and The MathWorks (publisher of MATLAB) have signed on as sponsors of this course. However, many of the lectures are general enough to be applicable to any software development technology.

I've learned a lot from these lectures, and they are now required reading for students in our ecological modeling group in Marine Sciences at UGA. All of the material is provided under an open source license, and can be freely copied for use in classes.

'Vista Data Vision' Data Analysis Software Review

edit

- Jason Downing (BNZ)

Like many research programs, our LTER site is continually struggling to manage numerous remote sensor stations and their data . We are constantly looking to improve the quality and availability of the data they are collecting while using technician time and resources effectively. One option on the market to manage sensor network data is Vista Data Vision (VDV) and after some initial testing, we decided to take the plunge and have not yet been disappointed.

You must be thinking, "What can this program do for me?". The answer is everything; well just about. It comes prepackaged with everything you need to take data from a field datalogger, pass it through some standard QA/QC filters, deposit it in a MySQL database, and make it fully viewable and downloadable on the internet. And all of this can be done without any custom coding or database configuration; in fact you could practically do it without ever having created a web page or even knowing anything about MySQL.

The system architecture is designed to function through three main application components. The first, db.robot.c, monitors the data files in your library and copies new data into the MySQL database. While the specific design and configuration of this database could be completely ignored, more advanced users can use any number of MySQL GUI tools to interact with this instance as they would any other; allowing even more options to extract and manipulate the contents. The second application, db.data.browser, is a configurable data browser that allows users to configure setup options and display data from a variety of stations and sensors in graphical format. With this the user can create pages with up to six graphs, each having up to six variables, that can then be browsed through time or edited if need be. This component also functions to create reports for entry back into the db.robot.c or produce advanced calculations, such as wind roses or wind energy calculations. Lastly, there is db.web.browser which allows the web user to do much of the db.data.browser operations (excluding setup and graphical edits) and includes everything needed to initiate and deploy an accessible web service. This allows users outside the local network to access data and download the data as text files for use with other applications to analyze or archive.

The Vista Data Vision program monitors your .DAT data files, collected with CS's Loggernet or Canary System's Multilogger; and as they are updated with new data, it is imported into a MySQL database (installed and configured by VDV) where it is immediately available to view or download on the web. Historical data files as well can be simply imported and there are file converters available if your data is in .CSV files.

There are also numerous access control options so you can have a host of users that would each have customized access to specific stations or sensors. The controls can also moderate which features are available to each user so you can control who can edit, download, or simply view the data.

Additional features that are available include an alarm monitoring function to send messages via email or text message when stations or sensors are either reporting data outside of acceptable limits or if the stations are failing to report altogether. There is also a toolkit to produce automatic reports in Word or text format on variable time scales (daily, weekly, monthly, quarterly, annually or simply on demand) and then email these to desired contacts. There is also a notation feature so a maintenance logs can be easily maintained along side the actual data. The validation toolkit provides the functionality for automatic removal of out of range data as well as unexpected noise or data spikes. Users can also create virtual variables, which are calculated values based on mathematical functions using data from the datalogger, to produce rescaled values or calculated results that were not performed by the datalogger. And lastly, in has a module to easily include web camera images along side the data so these images are available on the web and connected to the data if that is what you desire.

The cost for this program ranges, based on the chosen version, from $995 to $4,995 (Version Comparison: http://www.vistadatavision.com/index.php?page=version-comparison). This is a significant investment for sites operating with increasingly tight budgets but our analysis determined that for our needs, even the full featured PRO version with the highest price tag would be worth the investment when we factored in the ease of installation, quality of support, and time saved developing these tools ourselves. The PRO version will allow us to quickly implement systematic data quality procedures, easily get our current streaming data available and viewable through a web interface, give us room to grow over time as we add stations, and make this service available to some of our affiliated research groups that are also running telemetry stations located in our research areas. We initially deployed the free 30-day Trial Version that is offered on their website and initiated a dialog with an application specialist at Vista Engineering. Even though they are based in Iceland and multiple time zones from Alaska they were quick and helpful in addressing our questions. They were also extremely helpful in helping us to customize and troubleshoot the installation for our site and actually included a few code changes and improvements to the program specific to our requests. These customizations will eventually be included as standard features in future releases of the product, hopefully making the program even stronger for other users as well.

Unfortunately for some, this program is only available to run on Windows operating systems (Server2000/2003, XP Home/Pro, Vista, and 2000/ 2000 Pro). However, this program has been designed especially with Campbell Scientific (CS) dataloggers in mind; which also requires a Windows environment in some way to operate the remote telemetry connection software. Therefore, it is simple to have just one computer that runs all of the CS and VDV software concurrently. There are ways to host the web components through a Linux/Apache server running PHP if you desire but we opted to stick with the Windows IIS option for simplicity and ease of installation. With that, we were able to imbed the VDV pages within our existing web template so the newly generated data pages have the same look and feel of our existing web infrastructure.

The VDV (http://www.vistadatavision.com/) website is full of useful product information, educational materials, case studies and user testimony. The CS (http://www.campbellsci.com/) website also has additional information about the product and its interaction with other Campbell software. They also provide some additional user profiles and reviews.

The end result is a powerful and user friendly 'out of the box' solution to effectively mange almost any sensor network. The initial cost may be on the steep side but the simplicity to use and quickness to get the system deployed has saved enormous amounts of development time and technician training. Feel free to check out the Bonanza Creek site (http://www.lter.uaf.edu/bnz_vdv.cfm) to see what we have done or contact me (jpdowning@alaska.edu) for additional information.

"Form"ulating a Simple Solution to Editing Chores

edit

- John Porter (VCR)

As recent workshops and Databits articles have demonstrated, Xforms is a powerful system for creating new web forms. However, sometimes there is the need for simple one-off forms for database tables or the need to enhance existing forms. I recently came across two exceptionally easy-to-use tools that can help.

The first tool is phpMyEdit (http://www.phpMyEdit.org) is billed as an "Instant MySQL Table Editor and PHP Code Generator". The "instant" is maybe overstated, but it's not too far off. It comes as two PHP program files (and some associated directories) that can be placed into your web space. The first, phpMyEditSetup.php, when accessed by your web browser, provides a web form for specifying the MySQL table which you'd like to develop an editing interface to. It then generates PHP code for editing that MySQL table and displays the resulting PHP code in your browser, for pasting into a text file (hereafter called the PHP Form) for upload to the web server. The second file phpMyEdit.class.php is called by the new PHP Form to provide the underlying functionality.

The generated PHP Form file contains a wealth of configuration options that allow you to specify what access (View, Add, Change, Copy, Delete) a form user will be granted. Individual fields can also be customized for access control.

The documentation of phpMyEdit is reasonably good, with one real drawback. The options are set by assigning values, or arrays of values, to specific variables. However, the syntax used in the manual won't work as shown because in the PHP Form file the options tend to be nested (often with arrays of arrays), but the examples are just for setting single options. However, with a reasonably small amount of playing around, I was able to figure out that adding `'options'='LAFV',` to the array of options associated with a field was what it took to set List, Add, Filter and View capabilities to that field.

The second tool addresses a different problem - upgrading existing TEXTAREA tags in HTML forms with decent editing capabilities. The trick here, is that for my forms I want an editor that will allow users to easily edit text, but I don't want them to be using a large number of formatting commands that might be incompatible with subsequent uses (e.g., conversion to DOCBOOK for use in EML documents).

The "Loki-editor" (http://code.google.com/p/loki-editor/) is contained in a JAVASCRIPT file. You simply add two script tags to your web form (one to give the web address of the loki.js file, the other to fire off Loki-editor with the appropriate options), and when the form is viewed, all (or selected - depending on the options) text input boxes are replaced with an attractive and functional (but not overly fancy) text editor. The editor allows text formatting (e.g., **bold** and *italics*), but not adding tables etc. that might later prove to be a problem. The editor also adds paragraph markers whenever a user double-spaces between lines - a very handy feature!

Good Reads

Informatics and the Electronic Geophysical Year

edit

- Karen Baker (PAL/CCE) and Matthew Bietz (UWashington)

The International Geophysical Year (IGY) occurred in 1957-58, a time following postwar experience with large-scale science such as the U.S. Manhattan project and amidst initiation of international activities such as scientific committees with members from around the globe. It would take a large-scale history of science to help us discern how the events of such scope, involving the law-like understanding of physics in particular and the development of scientific collaboration in general, set the stage for a conceptualization of the earth as a whole. Suffice it to say, there emerged in the 1950’s within the geosciences the idea of capturing a scientific snapshot of the earth. Data taking covering the globe over the period of a single year was recognized as both possible and as worthwhile.

In celebrating the IGY, this article recalls and reviews a remarkable past milestone in terms of a geosciences understanding of its object of study, the earth as a whole. Fifty years later, in declaring the Electronic Geophysical Year (IGY, 2007-2008), organizers are both recognizing an important past event and declaring a new focus, a re-envisioning of the earth in a digital era. In particular, the authors call for the development of virtual observatories as a kind of "informatics common." Interestingly, the authors define virtual observatories as "a system of interoperable, distributed data repositories made accessible through grid and Web services." This characterization shifts scientific focus from the act of making an observation (manually or through distributed sensor networks) to the organization, aggregation, and curation of data.

This shift highlights informatics as integral to scientific work today, and the authors visualize informatics as a fourth pillar supporting science, along with experiment and observation, theory, and computation and analysis. Three emerging subdisciplines of informatics are identified: cyberinformatics, core informatics, and informatics as applied to a particular domain of science. This description of informatics practices, technologies, and disciplinary concerns may seem overly simplistic to practitioners. For example, the status of approaches to data handling is not delineated explicitly. An introductory statement that "there remains great reluctance among research scientists and others to invest time in good data management practices" because "research scientists are rewarded only for doing research" does not make it clear whether the authors are saying that scientists are not willing to conduct research into how to do good data management or whether scientists are resisting following through with an already established set of data management practices.

At the same time, this article is written for an audience of scientists and policy makers, and it should be read in that spirit. To change the name of that which you aim to celebrate is no minor act – it plants a stake for the important role informatics (and informatics professionals) serve in scientific practice. The article on the Electronic Geophysical Year leads us into a timely consideration of data taking and the concept of an information commons pertinent to global science, data taking, and data sharing.

Data at Work: Supporting Sharing in Science and Engineering

edit

- Mason Kortz (CCE/PAL)

Birnholtz, Jeremy and Matthew Bietz, Data at Work: Supporting Sharing in Science and Engineering. Conference on Supporting Group Work. (2003) 339-348

As information managers, we are all acutely aware that there are social barriers to data sharing. In this paper, authors Birnholtz and Bietz discuss the origins of some of these social barriers, and provide suggestions on how those barriers can be addressed from a Computer Supported Cooperative Work (CSCW) system design standpoint. To do so, they first explore the various ways in which data are valued in scientific communities. From there, they discuss the ways in which this value may be enhanced or compromised by data sharing. They end with a set of considerations for developing collaborative data systems, and a brief summary of data sharing issues in need a further detailed research.

The methodology for the research leading to this paper is a set of ethnographic studies across three disciplines – HIV/AIDS research, earthquake engineering, and space physics. This interview process allows the authors to identify both common and divergent themes in data sharing. It also means that the viewpoints on the value of data presented as described by researchers themselves. Two broad roles are identified for data: as scientific evidence and as a social construct in the community. The latter role is explored more extensively, and is further broken into sub-roles. This discussion includes the value of data in defining communities of practice, in establishing relationships within communities, and as an indicator of status. The differences in how data are valued in various communities are pointed out and drawn back to qualities of those communities including task uncertainty, feasibility of single-lab science, and academic tradition.

Having established the ways in which data are valued by researchers, the authors address the impact on data sharing practices. Data are described as objects that have the potential to generate various revenues – status, publication, funding, etc. By sharing data, scientists have the potential to gain revenues by entering collaborations that exceed the scope of what can be achieved in one lab. However, this exposure also presents the risk of data being misused, of mistakes in the data being made public, and in the data provider getting 'scooped' on a publication. Also discussed is the need for the context of the data to be shared. While metadata is acknowledged as an important part of context, it is also recognized that metadata is rarely complete and never easy to generate.

The authors close with some brief recommendations on the design of CSCW systems as well as future research into data sharing practices. Design suggestions include building support community-specific social constructs into collaborative systems, recognizing the multiple roles of data and supporting them appropriately, and not relying solely on metadata to enable sharing, but also to support sharing of broader contextual information. Further studies would include research on the role of data abstractions – specifically on how they can maximize the benefits of data exposure while minimizing risks – and on the further development of metadata or other contextual information.

While the themes present in this paper are not unknown to our information management community, it is helpful to see them presented in a broader framework. For participants who are new to the practice of information management or who do not have a background in scientific research, this paper makes very clear the value of being aware of data as more than simply a tool for publication and as part of a knowledge-making process. Understanding the values perspectives of a community is essential to the design of systems that will support that community, and thus essential to the role of the information manager.

Motivating Data Publication

edit

- John Porter (VCR)

Mark J. Costello, in his paper in the May 2009 issue of BioScience addresses "Motivating Online Publication of Data" outlines the challenges and opportunities associated with sharing environmental data in an international context (BioScience 59:418-427. http://www.bioone.org/doi/abs/10.1525/bio.2009.59.5.9).

The article addresses the benefits and costs of data publication for the scientist, and the value of publication of data for science in general. It includes a full-page box detailing (with responses) reasons scientists have given for not making data available, that should be very useful for LTER Information Managers (who have been known to hear similar reasoning from some local sources). A central thesis is that many of the reasons given for not publishing data are equally valid for conventional publications (you can't control the use someone makes of ideas you put forth in a paper, any more than you can control what users will do with published data). As he puts it "The only valid reasons for scientists not to publish their data online are the same as for not publishing in print media – namely, that the data are of such poor quality that they could have no useful purpose, scientists lack the competence or time-management skills required to prepare data for publication, or publication is not a priority in the scientists' work or career."

His prescription for fostering data publication includes the need to for peer review, for journals to play a role in requiring data publication, development of citation standards and the inclusion of citations in publication metrics, and the role of funders in requiring data publication. He notes that the culture of science is being increasingly changed by the Internet (interestingly, BioScience now includes DOI's directly in the title of downloadable citations), and that publication of data is an idea whose time has come!

Calendar

Events

edit

Event: MULTICONF-09

Location: Orlando, Florida, USA
Dates: July 13-16, 2009
Web: http://www.promoteresearch.org/

The 2009 multi-conference in computer science, information technology, computer engineering, computational science, control and automation technology.

Event: BIOCOMP'09

Location: Las Vegas, Nevada, USA
Dates: July 13-16, 2009
Web: http://www.world-academy-of-science.org/worldcomp09/ws/conferences/bioco...

The 2009 International Conference on Bioinformatics & Computational Biology is held simultaneously (ie, same location and dates: July 13-16, 2009, Las Vegas, USA) with a number of other joint conferences as part of WORLDCOMP'09 (The 2009 World Congress in Computer Science, Computer Engineering, and Applied Computing). WORLDCOMP'09 is the largest annual gathering of researchers in computer science, computer engineering and applied computing. Many of the joint conferences in WORLDCOMP are the premier conferences for presentation of advances in their respective fields.

Event: Ecological Society of America Meeting

Location: Albuquerque, NM, USA
Dates: August 2-7, 2009
Web: http://www.esa.org/albuquerque

Oral Session with LTER participants: OOS 49 - Standards, Protocols, and Tools for Sharing Ecological Information: Data Interoperability on a Global Scale; http://eco.confex.com/eco/2009/techprogram/S4181.HTM co-organizers: Annie Simpson, Elizabeth Sellers, Viv Hutchinson

Event: European Conference on Computer Supported Cooperative Work

Location: Vienna University of Technology, Austria
Dates: September 7-11, 2009
Web: http://www.ecscw09.org/

Special Working Group with LTER participation: Collaborative Infrastructuring – Conceptualizing Emergence and Design of Information Infrastructures co-organizers Karen Baker, Pelle Ehn, Sara Eriksen, Helena Karasti, Volkmar Pipek and Michael Twidale

Event: LTER Information Management Committee Meeting

Location: Estes Park, Colorado, USA
Dates: September 13, 2009
Web: http://intranet2.lternet.edu/node/3/

Event: LTER All Scientists Meeting

Location: Estes Park, Colorado, USA
Dates: September 14-16, 2009
Web: http://lternet.edu/asm

Event: 2009 Society for American Foresters National Convention

Location: Orlando, Florida, USA
Dates: September 30- October 4, 2009
Web: http://www.safnet.org/natcon-09/

Event: American Society for Information Science and Technology

Location: Vancouver, Canada
Dates: November 6-11, 2009
Web: http://www.asis.org/conferences.html

Event: 5th International Digital Curation Conference

Location: London, England
Dates: December 2-4, 2009
Web: http://www.dcc.ac.uk/events

Event: American Geophysical Union Fall Annual Meeting

Location: San Francisco, California, USA
Dates: December 14-18, 2009
Web: http://www.agu.org/meetings/fm09/

Session with LTER participation: Strategies for Improved Marine and Synergistic Data Access and Interoperability co-organizers: Cyndy Chandler, Karen Baker, and John Graybeal

Event: Hawaii International Conference on System Science

Location: Kauai, Hawaii, USA
Dates: January 5-8, 2010
Web: http://www.hicss.hawaii.edu/hicss_43/apahome43.htm