Title

Vega: A Flexible Data Model for Environmental Time Series Data

Authors

L. A. Winslow[1], B. J. Benson[1], K. E. Chiu[3], P. C. Hanson[1], T. K. Kratz[2]

[1]Center for Limnology, University of Wisconsin-Madison, 680 N. Park Street, Madison, WI 53706 USA; [2]Trout Lake Station, Center for Limnology, University of Wisconsin-Madison, 10810 County Highway N, Boulder Junction, WI 54568 USA;[3]State University of New York at Binghamton, P.O. Box 6000, Binghamton, NY 13902

Abstract

As large scale sensor networks grow, effective data curation of large data volumes is becoming important. Many sites have filled this need with site-specific database systems and software. Within the Global Lake Ecological Observatory Network (GLEON), a fundamental need for a data model allowing for growth and flexibility in sensing platforms and configurations requiring minimal or no data model changes was identified. The Vega data model is designed to fulfill that need. The Vega data model is a flexible, site agnostic data model optimized for high temporal resolution ecological sensor network data sampled at frequencies as high as a few seconds. Instead of storing data in a spreadsheet-like view with different variables denoted by columns, Vega stores observed values individually and describes them fully with linked, metadata containing tables. While being difficult to intuitively recognize, this more flexible and portable data model is beneficial at the individual institution level because it handles additional sensor deployments and configuration changes with no change in structure and at an inter-institution level because it represents a portable standard against which flexible and site agnostic software can be developed. Deployment and testing of this system has already begun within GLEON and has involved five different institutions.

Introduction

Modern ecological sensor networks are growing at a rapid rate (Porter et al. 2005). Several large scale projects such as NEON (National Ecological Observatory Network), WATERS (Water and Environmental Research Systems Network), and OOI (Ocean Observatories Initiative) propose deploying large scale environmental sensor networks across the US. Up to now, most groups with data curation systems have implemented site specific custom structures. While there are many different structures that effectively curate sensor network data, it is challenging to balance ease of use, query performance, and flexibility. Some groups have attempted to address the flexibility challenge by creating structures that store observations individually. One prominent example of an observation-based structure is the Observation Data Model (ODM) designed by CUAHSI (Consortium of Universities for Advancement of Hydrologic Science). In this paper, I describe a variant of the ODM called Vega. Vega was inspired by the ODM and has borrowed from ODM's terminology and concepts. The Vega data model is an observation-based data model for high-resolution time series data sampled at frequencies as high as a few seconds and is designed to optimize performance, flexibility, and simplicity.

Vega is currently being implemented in the Global Lake Ecological Observatory Network (GLEON; gleon.org). GLEON is an international, grassroots network of limnologists, ecologists, engineers, and information technology experts who have a common goal of building a scalable, persistent global network of lake ecology observatories. Data from these observatories will allow us to better understand key processes such as the effects of climate and land-use change on lake function, including carbon cycling in lakes, and the role of episodic events, such as major rainstorms and hurricanes/typhoons, in resetting lake dynamics. The current observatories consist of instrumented platforms on lakes around the world that are capable of sensing key limnological variables and moving the data in near-real time to web-accessible databases. Vega was developed after recognizing a fundamental need of GLEON for a data model that is flexible in both the number and configuration of instrumented sites and portable between institutions. Vega has been designed to meet the goals of GLEON but is applicable to any system with time series data.

Data Model Structure

The Vega data model stores data as individual observed values. All values are stored in a single large table and are directly linked to their supporting metadata. Through careful use of normalization and indices, redundancy is reduced to save space and query performance is optimized to improve retrieval times. Below is a description of the table, relationship, and index structure as well as conceptual issues in the representation of the data.

The Value

The fundamental record in Vega represents individual floating point values of discrete measurements. These values are all stored in a single table called ‘Values’. Because this table stores all observations within the system, it must be optimized to minimize storage space requirements and enhance query performance against it.

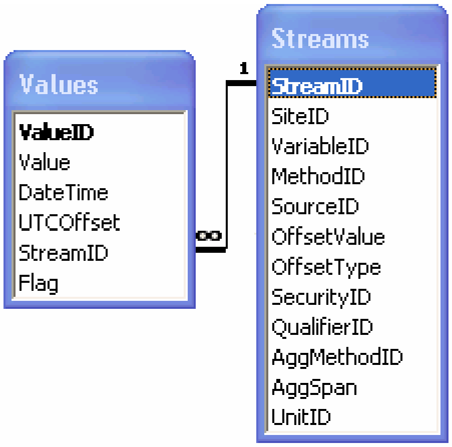

Figure 1 - 'Values' and ‘Streams’ tables at the core of Vega with the one-to-many relationship between them.

Each value is stored as a floating point double, is time stamped by a date time field, and is linked to its metadata by its stream identification (Figure 1), to be described in more detail later. The ‘ValueID’ field is included for convenience when programming and manipulating individual values. ‘Flag’ is included to allow for QA/QC descriptive data and to maintain backwards compatibility with systems that use data flagging as an indicator of potential data quality or other metadata.

Duplicate data are prevented at the table level. A unique index is defined for the ‘Values’ table on the ‘DateTime’ and ‘StreamID’ columns. No two values can have both the same stream and timestamp.

The Stream

The data stream is an entity designed to fully describe data that only vary through time, or in other words, a unique time series. Each stream is described by attributes stored in the ‘Streams’ table and can be thought of as a unique combination of attributes. For example, air temperature sampled at a particular meteorological station through time would be a unique stream. Soil temperature at that same station would be a different data stream.

Each stream has required attributes necessary to form a unique description and optional attributes necessary when those required are insufficient to uniquely describe the stream or when additional metadata are desired. Each stream is assigned a unique integer identifier, ‘StreamID’, forming the one-to-many relationship back to the ‘Values’ table. Duplication of streams is prevented by a unique index spanning all columns except ‘StreamID’. Most of these attributes are stored as foreign keys, linked to other supporting tables and not directly in the ‘Streams’ table with only ‘OffsetValue’ and ‘AggSpan’ being exceptions.

Uniqueness

Inadvertent insertion of duplicate data is prevented by table-level unique indexing applied to the ‘Streams’ and ‘Values’ tables. Programs inserting data do not need to know what data have been entered into the database as all potential duplicate inserts are prevented.

Supporting Attributes and Entities

The stream attributes map onto logical and useful supporting entities. Most of these are relatively straightforward, like the concepts of *site* and *variable*, but some are more abstract, like *aggregation* and *method*. The definitions are restricted to maintain usefulness and reduce ambiguity but are kept flexible to support different uses and unique requirements. For example, defining site too rigidly as specific latitude/longitude coordinates could make simple queries unwieldy.

Data aggregation is described by the *aggregation method* and *aggregation span* fields. Aggregation method is the method by which the data are aggregated, e.g., the mean, maximum, or standard deviation of a signal over a certain time. Aggregation span is the timeframe over which data are aggregated, e.g., one hour, one day. This allows simple types of temporal aggregation to be represented in the database but does not attempt to other potential aggregation types (e.g., spatial). If the data are sampled instantaneously and not aggregated, the aggregation method is defined as instantaneous and the span is zero. Aggregation types are stored in the ‘AggTypes’ table.

Sites in Vega are generally 2d locations in space. For some purposes, a looser definition of site may be useful. For example, in limnology, it may be helpful to define a whole lake as one site. Sites are stored in the ‘Sites’ table and can have a name, latitude and longitude, elevation, and country. Only the name is required. A third dimension is available through the ‘OffsetType’ and ‘OffsetValue’ fields.

An offset can be used to describe many different situations where a simple 2d site doesn’t adequately or uniquely describe a value. Offset value can be any double floating point value and offset type can be of the users choosing. Common examples include depth, height, and distance along a transect. Offset types are stored as name/id pairs in the ‘OffsetTypes’ table.

The ‘VariableID’ is linked to the ‘Variables’ table and describes what type of measurement or observation the value describes. The ‘Variables’ table stores name/id pairs.

‘Method’ within Vega is used to associate each stream with either a unique sensor or laboratory method attribute. This stream attribute is optional and is not required to fully describe the data, but can be useful for sites with a large number of sensors and complex calibration requirements.

Each value’s unit is included and is linked to by the ‘UnitID’ field. Units are included in defining stream uniqueness as it is possible that the same variable is stored with differing units.

Discussion

The Vega data model is a flexible system for storing environmental time series data and can handle changes to deployments and configuration without structural change or database level manual intervention. It is currently in use at five GLEON member sites and centrally as an intersite repository of shared data. Many interesting implications of using an observation level data model and Vega specifically have arisen over this time.

The Vega data model has advantages over more traditional archival models. Individual sensor deployments can be added and removed easily and without requiring manual intervention in the database system. We have used this system to store data from not only long-term sensor deployments, but also short-term datasets generated by individual experiments. Tools developed for Vega will always work against the same set of tables, regardless of what data are contained. Tools need not be changed or updated when datasets change. Vega also offers flexibility in discovering data. For example, it is very straightforward to select a site and determine variables measured at that site, or select a variable and determine the sites where that variable has been measured. Potential exists for even more pathways to discovering data, like retrieving all data sampled by a single sensor that was deployed to multiple different sites to get sensor history and quality assurance statistics.

Vega’s flexibility not only proves to be an important advantage in single institution systems but also presents a great opportunity within the entire community. When flexible standards are adopted, all systems and software developed for those standards can be adopted by other institutions, reducing the burden of site-specific software development. Just as any browser that understands the HTML standard is immediately compatible with sites developed in HTML, using a common data storage standard across institutions like Vega would allow tools developed portability and broad compatibility between all systems using that standard.

The Vega data model also has some limitations. Each value is individually time stamped and indexed. This means that observations don’t share timestamps or unique indexes as is typical for a flat table structure and requires more storage space comparatively. Data are not inherently in the form most users are used to seeing, and tools developed to expose data stored in Vega must format them in a form readily consumed by the user. Tools can, depending on the circumstances, be more difficult to develop. The data model is inherently more complex than a simple flat table view of the data as it contains a number of table relationships and more abstract table definitions. This results in a structure that may not be intuitively recognizable to the average user and requires further documentation and explanation.

Additionally, because the values are individually time stamped, data collected at the same time from the same platform must be transformed to matrix format if one wants a spreadsheet-like view of multiple variables. The division between streams and values also makes editing data somewhat more complicated. Tools editing values or altering metadata may be required to alter individual values, move values from one stream to another, or alter stream information independently. For example, changing the depth for a series of values may create a conflicting stream, in which case the ‘StreamID’ for values would need to be altered instead of simply changing the offset value.

Vega has not yet undergone rigorous experimental testing, but is the subject of an ongoing case study through its use in GLEON. During this time a few performance characteristics of interest have come to light. Simple data retrieval query times depend more on the number of values retrieved and less on the number of values stored in the ‘Values’ table. Well formed queries retrieving less than 100,000 values typically execute in less than one second. Inserting data typically takes, per value, longer than retrieval but still approaches tens of thousands of values per minute, very reasonable for most purposes. Storage requirements have been very reasonable. While naturally requiring more space than storage in raw text files, Vega has reliably required only about 100MB per million values stored, which represents about 19 variables sampled every ten minutes for one year.

As discussed, because of the more complex nature of Vega’s data structure, it is important that tools are developed to aid users in data importation, editing, and querying.

Several tools have already been developed and are being improved to fulfill this role. For discovering and retrieving stored data, an ASP.net web application called dbBadger (http://dbbadger.gleonrcn.org) was developed. For parsing and importation of data, GLEONDN (http://gleon.org/index.php?pr=Products) was developed. Both of these applications have been in use in one form or another for several months, are open source, and are freely available. Additional tools are either being planned or are currently in development. These tools include dynamic figure and data output systems for use in public web sites, data management and editing tools for manual QA/QC, and programmatic query tools for dynamic access to data by models and statistical tools.

Up to this point, Vega has been developed on a MySQL 5.0 backend. The data model in SQL form can be downloaded from the GLEON website (http://gleon.org/index.php?pr=Products). In the future, the Vega development group hopes to expand and interact with a broad range of groups inside and outside GLEON. The data model has already benefited from the expertise and input of GLEON network. This input has driven Vega and especially the tools built around it in the direction they are headed today. Throughout this, Vega will change and be updated to meet new requirements and match emerging standards.

Conclusion

By storing observed values individually and describing individual time series as streams, we created a flexible data model that fulfills the needs of GLEON and potentially other groups with similar curation requirements. Despite being somewhat less intuitive and having increased software development overhead, Vega's advantages in flexibility and portability make it a competitive alternative to traditional site-specific data curation systems.

Acknowledgments

This work was supported by awards from the National Science Foundation grants DEB-0217533, DBI-0639229, and DBI-0446017 and the Gordon and Betty Moore Foundation. We thank the member GLEON sites for their cooperation and encouragement.

References

Porter, J., P. Arzberger, H. Braun, P. Bryant, S. Gage, T. Hansen, P. Hanson, F. Lin, C. Lin, T. K. Kratz, W. Michener, S. Shapiro, and T. Williams. 2005. Wireless sensor networks for ecology. Bioscience 55:561-572.