Title

An Architecture Supporting Real-Time and Retrospective Environmental Data Management

Permalink

https://escholarship.org/uc/item/6qk1x8pv

Authors

Long, Darrell
Mantey, Patrick E
Haining, Theodore R
et al.

Publication Date

1994-06-21

An Architecture Supporting Real-Time and Retrospective Environmental Data Management

Darrell D. E. Long, Patrick E. Mantey, Theodore R. Haining, Bruce R. Montague, Eric C. Rosen*
University of California, Santa Cruz

Bruce R. Gritton
Monterey Bay Aquarium Research Institute

Proceedings of the First International Conference for Applications of Databases (ADB), June 1994.

Abstract

The Real-Time Environmental Information Network and Analysis System (REINAS) is a distributed database environment supporting both real-time and retrospective regional scale environmental science. Continuous real-time data is acquired from dispersed sensors and input to a logically integrated but physically distributed database. An integrated problem-solving environment supports visualization and modeling by users requiring insight into historical, current, and predicted oceanographic and meteorological conditions. REINAS supports both collaborative and single-user scientific work in a distributed environment. The goals and design of key data management aspects of REINAS are described.

Keywords: distributed database; heterogeneous database; real-time database; temporal database; scientific database; geographical information system; information modeling; environmental monitoring

1 Introduction

The Real-Time Environmental Information Network and Analysis System (REINAS) supports both real-time and retrospective regional-scale environmental science, monitoring, and forecasting. It is being developed by the University of California at Santa Cruz (UCSC), the Naval Postgraduate School (NPS), and the Monterey Bay Aquarium Research Institute (MBARI).

Users of REINAS observe, monitor, and analyze regional oceanographic and meteorological phenomena; the initial focus is the local Monterey Bay *sea breeze* air/ocean phenomena [8]. Unique to REINAS is its emphasis on regional-scale interactive real-time measurement and monitoring. The system and data management architecture are both designed to provide members of the oceanographic and meteorological communities with the ability to identify and visualize phenomena as they occur in real-time and to react to emerging phenomena and trends by reconfiguring instruments at sites of interest. Applying such capability to environmental and coastal science is currently an area of considerable scientific interest as demonstrated by the Norwegian *StormCast* project [10, 22].

Continuous real-time data is collected from a variety of dispersed sensors and stored in a logically integrated but physically distributed database. An integrated problem solving environment is being developed to support visualization and modeling by users requiring insight into historical, current, and predicted oceanographic and meteorological conditions. REINAS will support both single-user and collaborative scientific work in a distributed environment.

The visualization environment under construction will provide investigators with pictures of environmental features, trends, relationships, and dynamic behavior in a geographic context. Techniques are being developed to fuse data from sensors, the historical database, and models. Automatic methods of alerting users to interesting changes in the environment are being developed.

---

*Supported in part by the Office of Naval Research under Grant N00014–92–J–1807.

Instruments are connected to REINAS by both remote radio and land-line links. The system is designed so that new instruments can easily be added and assimilated by the data management and visualization subsystems. An interactive electronic log book tied to the database will populate and track instrument metadata used for calibration and control.

The data management structure is designed around a data architecture integrating data from multiple instrument technologies, classes (numeric, text, and video), institutions, levels of interpretation, representations, and technology generations. An information modeling process was applied to integrate both primary data and metadata into a stable data architecture.

The remainder of this paper is organized as follows: the scientific challenge is presented in §2, the REINAS system architecture in §3, and REINAS data management in §4. Current status is reviewed in §5 and related work discussed in §6. Conclusions are presented in §7.

2 The Scientific Challenge

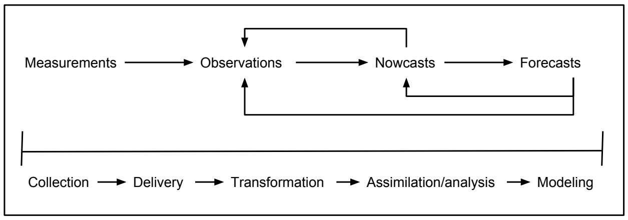

Figure 1: Environmental information life cycle – products and processes.

The primary scientific challenge addressed by REINAS is the conduct of real-time environmental science. In environmental monitoring and analysis, data collection, delivery, and transformation activities feed the intertwined and iterative processes of data analysis, modeling, and visualization. In this information life cycle, shown in Figure 1, instruments produce measurements which become observations by the application of calibration algorithms. Scientific databases must store all measurements, observations, and calibration parameters involved in this process. Data assimilation is the process of taking irregular (and often sparse) observations and creating an accurate and meaningful representation of the current state of the environmental phenomena being observed. Computer graphics ‘visualization’ is the usual mode of presenting the assimilated data. Models are often used in an iterative fashion to support data assimilation and are also used to generate forecasts.

Meteorological measurements and observations are produced from a variety of instrument sources. These sources include land based weather stations, ships, buoys, aircraft, radars, and satellites. Many of these data sources are not directly accessible to REINAS but can be assimilated using data exchange and format standards adopted by major national and international weather centers.

Oceanographic measurements are much more sparse than meteorological measurements. Ships and buoys provide some data while satellites obtain ocean surface data. REINAS will initially obtain real-time measurements from meteorological stations, wind profiler radars, CODARS (ocean surface current radars), acoustic Doppler current profilers, video cameras, and thermistor chains. Additional devices will be added in the future. Despite the differences in data sources there is broad commonality in the types of information created and used by meteorologists and oceanographers and both groups exhibit interest in air/sea interaction. Activities surrounding data collection, delivery, processing, analysis, modeling, and visualization functions have a common framework, especially from a data management perspective. The challenge for REINAS is to provide a common data storage and manipulation system for these data life cycle activities while supporting the specialized requirements of each discipline.

A problem common to both oceanography and meteorology is that while observations from many sources are provided by operational communication networks managed by military, academic, and private enterprises, this data is often not real-time and is often delivered in an un-integrated fashion involving manual operations. As a result, researchers frequently miss opportunities to coordinate data collection activities as interesting phenomena occur.

REINAS seeks to address these challenges by proving a unique real-time environment supporting interactive desktop experimentation by air/ocean researchers. Such a high-quality real-time environment will motivate the adoption of new methods of scientific collaboration and data exchange. Real-time desktop experimentation is potentially promising, and is applied to some degree in related sciences due to necessity [9, 2].

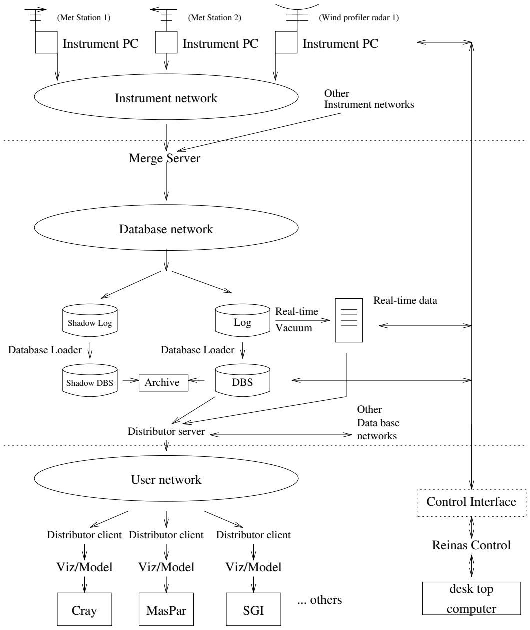

Figure 2: REINAS Logical Architecture

3 REINAS System Architecture

The REINAS system architecture was designed to address the needs of several different groups within the oceanographic and meteorological communities: Operational forecasters monitor current conditions, view standard data products, synthesize new data views, and issue forecasts and warnings. Modeling scientists visually analyze products of new data models and compare these products with the outputs of other models as well as with past and present conditions. Experimental scientists collaborate with other scientists on-line, observe individual data fields from specific sources as they are collected, and may modify data collection methods while an experiment is in progress. Finally, instrument engineers add new equipment to the system, access metadata describing individual devices, observe methods of calibration, study maintenance records, and profile sensor quality.

REINAS provides a system architecture integrating user group activity (both at the office and in the field) over the whole data life cycle. Central to this architecture is tracking the lineage of data elements and system resources and supporting a common information model which makes data accessible to the entire user community.

The REINAS architecture provides the following services:

- seamless access to real-time and retrospective data,
- the performance required to support the most frequently requested products and services,
- access to named resources and devices through a common data model while supporting rapid system configuration,

- dynamic control of system devices for real-time interactive scientific investigation,
- fault tolerant methods of data collection avoiding data loss due to communication link failures resulting from environmental factors, and
- security features restricting access and control privileges of users with respect to data and equipment.

3.1 System Organization

As shown in Figure 2, REINAS is organized into three subsystems, each corresponding to a specific part of the data life cycle. The instrumentation subsystem collects data and transforms it to a standard portable format. The database subsystem stores data both on-line and on archival media, as well as providing a framework for data manipulation within the system. The user subsystem provides users access to data in the instrument and database subsystems and supports end user visualization and modeling applications. Each subsystem has its own logical network which is defined by ownership and operational characteristics. Functionality is distributed throughput these networks, with any specific computation occurring at the most appropriate location.

The instrumentation subsystem consists of nodes containing measuring devices attached to microcomputers and connected to the Internet. Each microcomputer writes data to a log, converts it to a common format, and transmits the data to the rest of the system at a specified rate. Configuration and control commands for specific instruments and sensors are also executed at this level.

Data obtained by an instrument node is transmitted to the database subsystem. There the data is received by a *merge server* process and written to a database load log. A *database loader* process then reads the data from the log and writes it into one or more historical databases. Data is also read from the log and copied into memory resident real-time data structures which reflect current data values and simple statistics, such as moving averages. Data from both the historical databases and the real-time data structures is seamlessly accessed from end user applications by a process called the *distributor* which can provide applications with real-time state information at a given rate.

Due to the large amounts of data processed by the system, data may eventually be migrated from fast storage in the database to slower archival media such as tape or compact disc. An archiving process in the database subsystem accomplishes this task.

The user subsystem runs REINAS applications. This subsystem consists of general and application specific hardware and software which access the distributor via the Internet using an application software interface. This application interface provides a standard programming model for accessing and controlling both the database and instrument subsystems. User applications retrieve data by using the application interface to issue queries that are executed by the distributor. The user subsystem also contains the REINAS application visualization support routines.

3.2 Advantages of the Architecture

Given this layered architectural approach, resources can be duplicated throughout the system and subsystems can communicate with multiple instances of subsystems in higher and lower layers. For instance, instruments can provide input to multiple database subsystems, and a database subsystem can receive input from more than one instrument subsystem. Likewise, a single database subsystem can service multiple user subsystems and a single user application can request services from multiple database subsystems. Database replication can be implemented using multiple database subsystems.

The design of each layer of the system is optimized toward a specific task. Since microcomputers are used as nodes of the instrument subsystem, naming and access control of individual instruments can occur at a low level within the system, providing a framework which is both secure and easy to navigate. Dedication of data acquisition activity to distributed microcomputers permits the database subsystem to be optimized for the manipulation and storage of data through the use of the Andrew file system, large storage devices, and large database servers. Lastly, since user applications for visualization and modeling are computationally expensive, these activities can occur on specialized hardware and do not adversely impact machines involved in other tasks in the system.

Finally, the separation of concerns between layers makes the system extensible through the development of regular interfaces between machines on different layers. Since the different architectural layers all communicate through the Internet, any machine attached to the Internet can be added to the system through a protocol which conforms to the appropriate interface.

4 Data Management from Instrument to Historical Use

REINAS data management supports user needs throughout the data life-cycle. Scientific data management is not simply a repository problem. Scientific data management must also support a range of tasks starting with data collection and proceeding through interpretation and scientific collaboration. REINAS data management must also address systemic problems challenging current data management practices in ocean and atmospheric sciences: integrated data/metadata handling, data lineage, and quality assessment.

Data management occurs within all system elements. Operational users require immediate access to the instruments and to the state of the network, while retrospective researchers require access to historical data.

Instruments produce different classes of data (e.g. numeric, image, video, acoustic) and have different operational profiles. Computational and laboratory processes are also sources of data and are complicated by interpretation activities performed by scientists with differing vocabularies and research backgrounds. Data management functions must provide transport, real-time access, data storage, and retrospective data access. Equally important, each subsystem must support the capture, storage, and access of information documenting the context, content, structure, and representation of the primary data. This class of data is called *metadata*. This integrated end-to-end approach to scientific data management has implications for all subsystems.

The instrument subsystem is more than a transport mechanism. The reliability and integrity of data transport must be assured and a data log synchronized with database loading. Since instrument control protocols can dynamically configure instrumentation suites, instrument configuration history must be maintained.

The data management subsystem must support functional integration, data integration, and data pedigree and quality. Functional integration is provided by the specification of a consistent framework for the tasks of scientific information creation and use. This framework must:

- provide access to real-time, retrospective, and predicted future states through a consistent interface,
- integrate data acquisition, access, analysis, and visualization tasks into a common computing environment, and
- support both environmental scientists and engineers responsible for the development and maintenance of the system.

A stable database architecture is needed which can accommodate the multimedia, multidisciplinary, and multiformat data delivered from the data sources. This architecture must also cope with continuous addition of new metadata and new quality assessments of existing data.

Ultimately, data is analyzed using the visualization subsystem. This subsystem provides a standard user interface by which the user can query the database subsystem and control both the database and instrument subsystems. Also included within the visualization subsystem are tools for producing standard environmental science information products and 4-D visualization products. Visualization of routine monitoring activity, experiment history, and sampling event history is possible using temporal metadata.

4.1 Instrumentation

The REINAS instrumentation interface must be designed around two challenges associated with environmental data collection: the need for instruments to operate in an independent and isolated fashion, and the ability to support rapid deployment and configuration with minimal maintenance of REINAS software and hardware in the field. In addition, a broad class of instruments must be supported, and the instrumentation interface must be robust and flexible enough to easily allow new or previously unknown instruments to be connected.

Each instrument interfaces with REINAS through a microcomputer at each node, which autonomously collects data from the instrument and immediately stores it in a local log. In the event of a network failure which isolates the node, the microcomputer continues to operate autonomously, using its considerable local storage to avoid any loss of data. Once connectivity with the merge server has been reestablished, the contents of the local log are used to bring the server up to date.

The microcomputer also provides flexibility in the design and manipulation of the instrument interface. The use of a standard computer configuration to connect *all* instruments to REINAS creates a common environment in which standardized system and device interfaces for every class of device supported by the system can be developed. All software can be developed and updated remotely. When necessary, direct communication with the attached instruments can be established through the REINAS microcomputer, allowing scientists and REINAS engineers to directly manipulate instrument parameters. These abilities combine to reduce and simplify the labor required to deploy and initially connect a new instrument to the system, and allow a more proactive approach to be taken toward maintenance, resulting in a more reliable and useful instrument.

The typical REINAS instrument is a surface weather station or radar such as the CODAR or vertical wind-profiler. Despite differences in instrument specifics and manufacturer, these instruments can be and are usually configured to output data and accept commands through a generic serial interface. Typically, this interface is connected to an automated storage device or dial-up modem, but by connecting the instrument instead to a local REINAS microcomputer which itself is networked in some fashion to the Internet, a generic and flexible connection that enhances the utility of these remote instruments is created.

4.2 Storage and Access for Long Term Use

Effective end-to-end scientific data management requires support for long-term data use as well as support for the activities of collection through interpretation. Information modeling techniques were applied to the meteorological and oceanographic science domains, to typical operational scenarios, and to high profile requirements. The MBARI Scientific Information Model served as a starting point and was extended to support more diverse representations of observational data in the environmental enterprise [3, 4].

The primary information modeling goals are:

- to identify classes of information created and used by scientists and engineers working in all areas of oceanography and meteorology,
- to analyze objects in each class to identify broad class generalizations and critical specializations,
- to synthesize a data architecture at the appropriate level of abstraction which is unlikely to change in basic structure as technology and science evolve, and
- to identify where the architecture will extend and evolve over time.

The rationale behind these goals is that scientific domain, technology, and system changes can be reflected by changes in database content and not in database structure.

Data paths were followed from source to sink to identify and describe concepts and objects important to each type of data source. End user information needs were analyzed via their data management query forms to identify important objects and concepts which spanned data source types. The result of this analysis was a set of generalized concepts which mapped well to specific objects and concepts while identifying areas where the architecture will evolve.

As an example, this data flow analysis identified the following objects and attributes for an ocean temperature sensor:

- sensor (model, best accuracy, best resolution, range, serial number, operator ID number),
- calibration history (date of calibration, coefficients, date range of applicability, person who performed the calibration),
- data acquisition instrument hosting the sensor (model, configuration options, configuration history),

- processing history (baseline algorithms, software programs implementing algorithms, quality control procedures, person who performed the processing), and
- instrument platform hosting the data acquisition instrument (type, name, responsible person/organization, log of events and malfunctions),
- and data management log (processing events, reprocessing events, archive tracking).

An analysis of user information needs was performed by studying typical user queries which were source independent. For example, the query *"return all collocated temperature and nitrate observations taken at site H3 during the winter season where the temperature is greater than 13 deg-C"* indicates the need for named locality descriptions in addition to a precise space/time tag for each observation to determine collocation. The query *"return all temperature and salinity observations taken from the Pt. Lobos research vessel where the data collection runs are marked as questionable"* indicates the need to track the platform (in this case a ship) which collected the data and the requirement to associate quality assessments with individual and aggregate sets of observations.

The architecture consists of major information groups called realms which contain enough substructure to fully capture the semantics of the major types and subtypes of the realm. These realms include: systems, processes, parameters, localities, data generation activities, descriptions, quality assessments, and measurements/observations. Objects in each realm will participate in intra-realm and inter-realm relationships.

The system realm contains generalized and specialized attributes of major classes of systems which occur in the environmental enterprise. Examples include instrument platforms (ships, aircraft, satellites, remotely operated vehicles, buoys, or land meteorological stations), instruments, instrument platform subsystems (winchs or cranes), sensors (temperature sensors or wind speed/direction sensors), and computers.

Process realm objects include those items which document automated or manual procedures intended to accomplish a specific purpose. Examples include calibration algorithms for environmental sensors and laboratory procedures for performing sample analysis.

Objects in the parameter realm are used to define the types of environmental properties which may be represented in the database and the logical and physical form of their representation. This realm supports the requirement to store and reconcile data representing the same concept in different formats.

The locality realm contains objects which represent spatial features of interest in their own right or as spatial identifiers for other database objects. Locality features may be points, two or three dimensional regions, linear networks, or names with no specific boundary definition. Regular sampling/monitoring sites, the spatial extent of a data collection activity, or the spatial extent of an observation data aggregate may be defined.

The data generation realm contains objects defining those things which can be part of the data generation process or document the process. A few important generalizations in this realm include expeditions, projects, experiments, data collection runs, and sampling plans.

The measurement/observation realm contains the primary data of interest to the environmental scientist. Direct sensor outputs, derived observations of environmental properties, and ancillary information which may be tagged with each individual measurement/observation are included. In addition, aggregations of individual observations may be identified and tracked. For example, an image may be seen as an aggregation of the individual pixels comprised of separate, distinct, and accessible environmental observations. Other typical aggregate types include time series, vertical profiles, and spatial/temporal grids.

Quality assessment realm objects document multiple assessments of the quality of individual observations or aggregates of those observations. These assessments may include both quantitative and descriptive assessments by data users.

The descriptive realm contains objects which are used to document the environmental science enterprise and the database system itself. General object types such as person, remark, and calculated summary parameter may be associated with any other object in the database. This is the realm where logical, physical and other special data formats may be described.

Beyond the identification of these generalized realms, another challenge remained: to determine the appropriate logical/physical packaging of measurement/observation data. Two primary alternatives were considered, a state vector representation and a data stream representation.

A state vector representation packages all observations of all types from all collocated sensors into an observation group with a single space/time tag. This form of organization provides the most effective support for queries of the type *"What is the complete state of the environment at point x,y,z?"*. Alternatively, in the data stream model all observations of the same type from all sources in the region are packaged together and physically sorted by time. This model best serves queries which assess the state of a small number of environmental variables over a large viewing area, for instance, *"What is the state of sea surface temperature and wind velocity fields around the Monterey Bay area?"*.

This issue is resolved based on the answer to two questions: how are the data most often requested? and how are the data most effectively managed in terms of data/metadata linkage? The answer to both of these questions indicate a strong preference for the data stream information model.

4.2.1 The Schema

A REINAS schema system is defined as a collection of hardware, software, and procedural components that work in cooperation and perform a specific function or produce a specific product. Systems have configuration, malfunction, and maintenance histories. A system hierarchy exists, as does a REINAS name space reflecting this hierarchy. A system belongs to a parent system and may have child systems. An important system attribute is its type. System types include platform, instrument, sensor, and computer.

The schema defines a REINAS process as an automated or manual procedure initiated to accomplish a specific purpose. Every process is an instantiation of a process type. As with systems, all processes exist in a process hierarchy. Processes may invoke, or be invoked by, other processes. An important process attribute is its type. A process type can be either procedure or algorithm. A procedure is an operationally defined activity that produces a known result. A procedure often reflects a human activity performed according to a prescribed sequence of steps.

An algorithm is defined by the schema as a known calculation or mathematical transform. An algorithm often corresponds to a published scientific data processing technique. Algorithms are used in many places in the REINAS scientific database to transform measurements into observations.

A program is defined as a process subtype corresponding to an executable computer program. A program may implement one or more algorithms and any number of programs may use the same algorithm. The database keeps track of the transformations performed on data by keeping track of the algorithms and programs that have transformed the data.

An environmental property is a measurable quantity such as temperature, humidity, or wind speed. An environmental property may be represented by multiple parameter types. For example, temperature might be represented by Kelvin, Fahrenheit, or Centigrade.

The scientific data types used by REINAS are referred to as parameters. Parameters define the primitive formats containing the scientific data in the database. Parameters have types and are associated with the systems and processes that can instantiate elements of that parameter type. Typically, a parameter type is a representation of some environmental property. A value domain is a description of legal data values. Primary units (Kelvin, Fahrenheit, *etc.*), unit modifiers, and legal value ranges may be specified by a value domain. An instantiated parameter is called a data element or (sometimes) a value, field, or item.

The scientific data stored in a REINAS database is organized into an activity hierarchy as shown in Figure 3. At the top of the hierarchy is an expedition. An expedition describes a operational activity with a well defined mission. Ship cruises, aircraft flights, satellite orbits, buoy deployments, met station networks, and remotely operated vehicle dives are all examples of expeditions. An expedition can be associated with one or more projects or experiments.

A data run occurs when collocated systems sample the environment for a continuous period of operation using a fixed configuration of data sources. A data run may be a short activity or may last a considerable period (a one month buoy data run). A data run has an associated locality which represents the extent of the spatial coverage of the data run.

Measurements and observations from the same data run, and of the same type, source, and processing lineage are combined into data streams. Data streams corresponding logically to the parameter types produced by the data sources during the data run. Individual point observations, profile aggregates, and 2-D or 3-D fields may all be defined as single elements within a stream. A stream element, or group of elements, may also have associated spatial extents. These stream element localities will fall within the locality associated with the data run in which the data stream was produced.

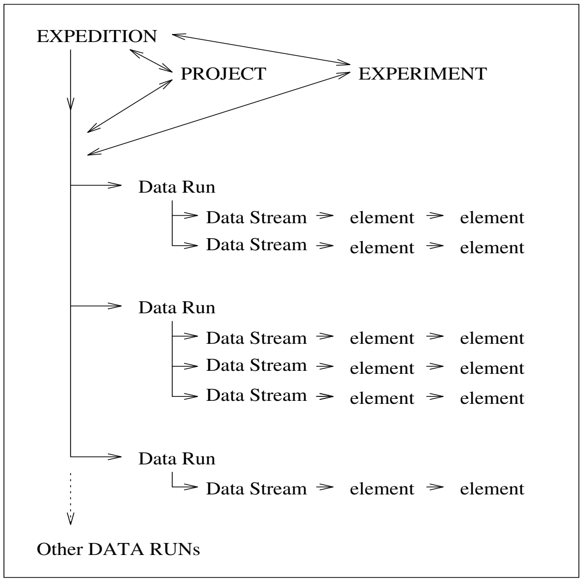

Figure 3: Database Organization

The elements in a data stream are either aggregated or non-aggregated. A non-aggregated element is instantiated as a single parameter value. The parameter can be scalar or complex. An aggregated element is an array of parameters. Aggregated parameter have internal structure (the dimensionality of the array) of which the database is aware. A stream element for an aggregated parameter has the same format as for any other stream element.

A typical aggregated parameter is a wind profile vector. In this case, a profile consists of multiple observations that have all been collected at the same time. Profiles are linear. Complex calculations do not need to be applied to a profile to obtain the locations of each of the elements of the profile. This is different from a satellite image where the relationship between pixels in the image is non-linear and unique to every image.

Data streams are classified into a common type if:

- they originate from the same system type and process type,
- they contain the same parameter type, and
- they have identical physical representations.

All primary scientific data is stored in containers. Containers are designed to host time-ordered stream elements from compatible data streams. Such data streams contain elements with logically consistent parameter types and physical representations. Elements from different but compatible data stream types can be stored in the same container.

This schema provides an extensible framework for managing oceanographic and meteorological scientific data. The schema describes the central items that must be tracked to support oceanography and meteorology research. Users need not develop custom data handling solutions as particular data needs can be supported by simply changing database content rather then the schema definition.

To further illustrate this schema, Figure 4 shows objects that need to be populated to describe a data stream.

4.3 Visualization

Data visualization is accomplished by end-user computer graphics applications. Some applications use commercial 3-D packages such as OpenGL to display data, and use the REINAS application interface to provide data of

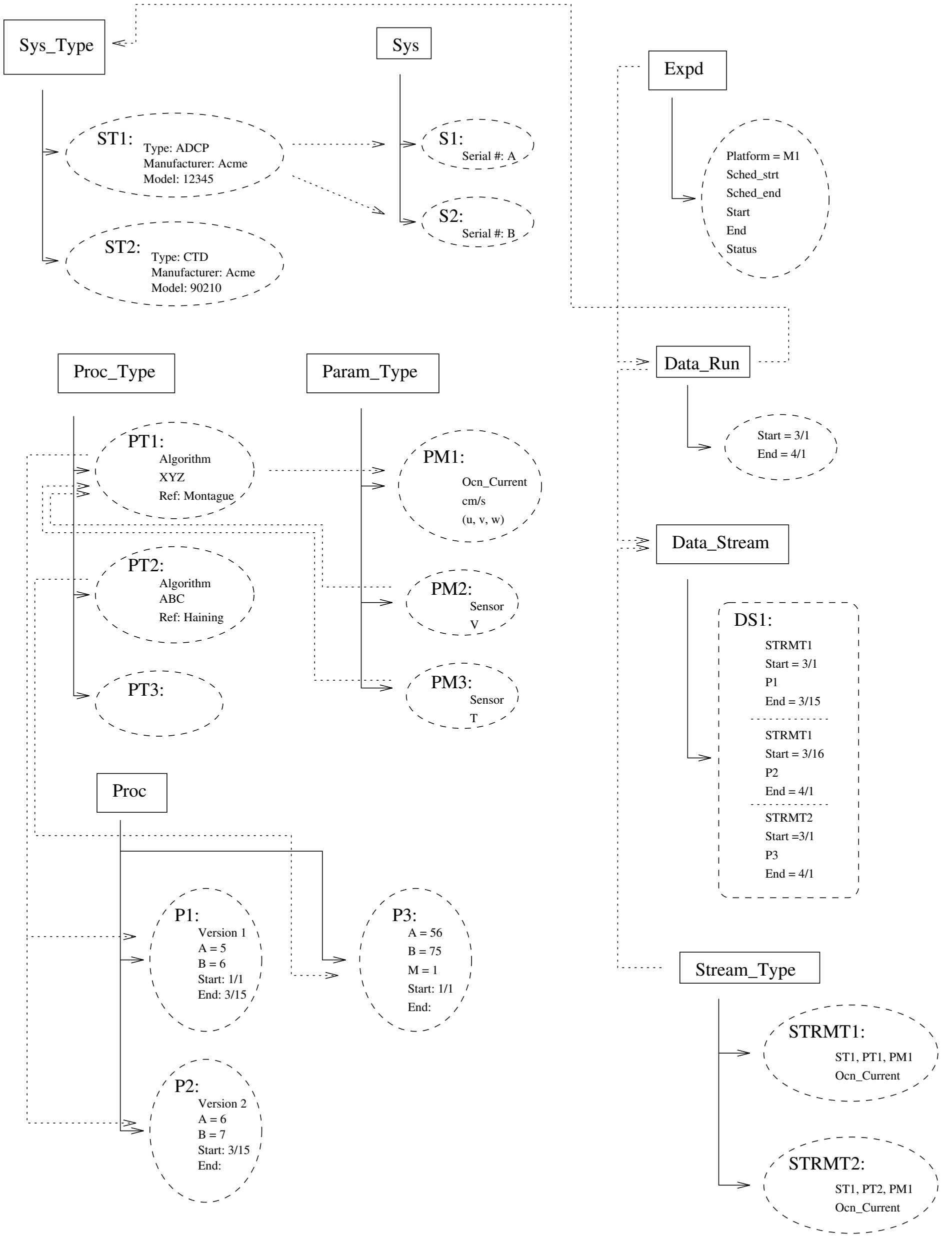

Figure 4: An example schema

interest to the visualization software. Once data is obtained, users can freely explore the data using a novel REINAS visualization technique based on *sparts*, or “smart particles” [13, 12]. In this approach users employee a virtual can of spray paint to render data and highlight areas of interest. This creates a coherent framework for surface, volume, and flow visualization of both regular and irregular grids of data.

Sparts are effectively independent threads of activity or agents which follow trajectories through a 3-D space populated by the data, visibly rendering selected data elements. Sparts can easily be extended and combined to form sparts with new behaviors. The spart technique addresses the multistep scientific query problem in which selective progressive refinement is used to explore relationships in the data [5].

5 Status

At the present time, the nodes in the instrument network are conventional low-cost 486 PCs running BSDi UNIX and communicating using SLIP (Serial Link IP) or PPP (Point-to-Point Protocol) over leased lines. Some PCs are connected by RF modems. The instrument node log is implemented using Recoverable Virtual Memory [14]. The database network layer consists of a Sparcstation/2 and an IBM RS/6000 workstation, both running extended Relational database software. Both Montage and OpenIngres are currently under evaluation [19]. Visualization activities are being performed on Silicon Graphics Indigo and Hewlett Packard workstations. Security is to be implemented using Kerberos [11]. The Andrew File System [6] will be used to provided a single file storage hierarchy for the system.

6 Related Work

The REINAS system is a large effort incorporating many research disciplines. Distributed computer technology is being used by projects such as the SEQUOIA 2000 to support large environmental databases [18, 16]. REINAS differs from SEQUOIA 2000 due to its emphasis on real-time desk-top experimentation and its regional focus.

Environmental and GIS systems which focus on coastal or regional air/ocean science and apply geographical information systems (GIS) and visualization techniques are being built REINAS expands on such approaches by providing a common integrated data model to enhance the value of the scientific data collected.

Developing database schemas which track the metadata associated with a given scientific enterprise or scientific discipline is an ongoing activity [21]. REINAS will develop a model that supports the oceanographic and meteorological communities and has broad environmental application.

The application of computer science techniques to the conduct of natural science is being widely investigated, especially in areas such as visualization and the application of object-oriented databases [1]. The REINAS project is developing a unique visualization technology based on smart particles and is investigating object-oriented approaches to the common data model.

REINAS applies and extends many traditional computer science techniques. REINAS applies distributed system techniques in the areas of security, naming, and client-server application design [6, 15]. REINAS uses an object-oriented database design to encapsulate data within the problem domain [7, 20]. The design will support a multidimensional structure tailored to analytical as opposed to structural queries [17].

7 Conclusions

The REINAS system has been designed for regional real-time environmental monitoring and analysis. REINAS is a modern system, compatible with the Internet, for conducting interactive real-time coastal air/ocean science. The database design of REINAS is independent of specific database technology and is designed to support operational scientific needs throughput the entire scientific data life-cycle.

References

[1] J. B. Cushing, D. Hansen, D. Maier, and C. Pu. Connecting scientific programs and data using object databases. *Bulletin of the Technical Committee on Data Engineering*, 16(1):9–13, Mar. 1993.

[2] W. Griethe and J. Burfeindt. CIS communication facilities, data transmission and computer-aided experiment evaluation for users of the MIR space station. *ESA Journal*, 16(4):455–462, 1992.

[3] B. R. Gritton, D. Badal, D. Davis, K. Lashkari, G. Morris, A. Pearce, and H. Wright. Data management at MBARI. In *Proceedings of the 1989 The Global Ocean, Oceans*, pp. 1681–1685. IEEE Publications, 1989.

[4] B. R. Gritton and C. Baxter. Video database systems in the marine sciences. *MTS Journal*, 26(4):59–72, 1992.

[5] M. Hapgood. Multistep queries: The need for a correlation environment. In A. Heck and F. Murtagh, editors, *Intelligent Information Retrieval: The Case of Astronomy and Related Space Sciences*, pp. 153–171. Kluwer Academic Publishers, 1993.

[6] J. H. Howard, M. L. Kazar, S. G. Menees, D. A. Nichols, M. Satyanarayanan, R. N. Sidebotham, and M. J. Wes. Scale and Performance in a Distributed File System. *ACM Transactions on Computer Systems*, 6(1):51–81, Feb. 1988.

[7] A. R. Hurson, S. H. Pakzad, and J.-B. Cheng. Object-oriented database management systems: evolution and performance issues. *IEEE Computer*, 26(2):48–58, Feb. 1993.

[8] J. M. Intrieri, C. G. Little, W. J. Shaw, R. M. Banta, P. A. Durkee, and R. M. Hardesty. The land/sea breeze experiment (LASBEX). *Bulletin of the American Meteorological Society*, 71(5):656, May 1990.

[9] Y. Ioannidis and M. Livny. Conceptual schemas: Multi-faceted tools for desktop scientific experiment management. *International Journal of Intelligent & Cooperative Information Systems*, 1(3–4):451–474, Dec. 1992.

[10] D. Johansen. StormCast: Yet another exercise in distributed computing. *Distributed Open Systems in Perspective*, Feb. 1993.

[11] B. C. Neumann, J. G. Steiner, and J. I. Schiller. Kerberos: An authentication service for open network systems. In *Proceedings of the Winter 1988 USENIX Technical Conference*, pp. 191–201, Dallas, TX, 1988.

[12] A. Pang, N. Alper, J. Furman, and J. Wang. Design issues of spray rendering. In *Proceedings of the 1993 Compugraphics*, pp. 58–67, 1993.

[13] A. Pang and K. Smith. Spray rendering: Visualization with smart particles. In *Proceedings of the 4th Conference on Visualization (VIS '93)*, pp. 283–290, 1993.

[14] M. Satyanarayanan, H. Mashburn, P. Kumar, D. Steere, and J. Kistler. Lightweight recoverable virtual memory. In *Proceedings of the 14th ACM Symposium on Operating Systems Principles (SOSP '93)*, pp. 146–160, Dec. 1993.

[15] M. Shapiro. Structure and encapsulation in distributed systems: the proxy principle. In *Proceedings of the Sixth International Conference on Distributed Computing Systems (ICDCS '86)*, pp. 198–204, May 1986.

[16] T. R. Smith, J. Su, D. Agrawal, and A. E. Abbadi. Database and modeling systems for the earth sciences. *Bulletin of the Technical Committee on Data Engineering*, 16(1):33–37, Mar. 1993.

[17] J. P. Stamen. Structuring databases for analysis. *IEEE Spectrum*, pp. 55–58, Oct. 1993.

[18] M. Stonebraker. An overview of the SEQUOIA 2000 project. In *Proceedings of the 37th IEEE Computer Society International Conference Digest of Papers: COMPCON Spring 1992*. IEEE Computer Society Press, Feb. 1992.

[19] M. Stonebraker. The Miro DBMS. *SIGMOD Record*, 22(2):439, 1993.

[20] M. Stonebraker and J. Dozier. Large capacity object servers to support global change research. Technical Report 91–1, Berkeley, July 1991.

[21] S. N. Walker, H. S. C. Alleyne, L. J. C. Wooliscroft, D. L. Giaretta, M. A. Hapgood, D. R. Lepine, B. J. Read, M. A. Albrecht, A. Ciarlo, H. Stokke, and P. Torrente. A space-physics query language for the european space information system. *ESA Journal*, 16(4):463–476, 1992.

[22] P. Wilde. First workshop on the Yellow Sea experiment (YESEX-1). *ONR ASIA Scientific Information Bulletin*, 17(3), July 1992.