REINAS: Real-Time Environmental Information Network and Analysis System: Phase IV.1-EXPERIMENTATION[*]

P.E. Mantey, D.D.E. Long,J.J. Garcia-Luna,
A.T. Pang, H.G. Kolsky (UCSC),
B.R. Gritton (MBARI), W.A. Nuss (NPS)

UCSC-CRL-94-43
October 25, 1994

Baskin Center for
Computer Engineering and Information Sciences
University of California, Santa Cruz
Santa Cruz, CA 95064 U.S.A.

Keywords: Real-time, System Design, Environmental, Sensor, Data Management, Network, Visualization, Monterey Bay, Coastal, REINAS

---

[*] Supported by a Grant from the Office of Naval Research, No. N-00014-92-J-1807

Contents

Abstract

The Real-Time Environmental Information Network and Analysis System (REINAS) is a continuing engineering research and development program with the goal of designing, developing and testing an operational prototype system for data acquisition, data management, and visualization. This system is to support the real-time utilization of advanced instrumentation in environmental science. Advances in continuous time measurements and improved spatial resolution allow the monitoring and understanding environmental phenomena in much greater detail than has previously been possible. The system is also designed to support the retrospective use of integrated environmental data sets.

The project is a multi-year effort of the Baskin Center for Computer Engineering and Information Sciences of the University of California, Santa Cruz (UCSC), in cooperation with environmental scientists from the Naval Postgraduate School (NPS), and Monterey Bay Aquarium Research Institute (MBARI).

The REINAS system has been designed for regional real-time environmental monitoring and analysis. It is a modern system, integrated into 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. During the Spring 1994 final selections of the technologies to be used in the prototype system were made, and detailed architecture of REINAS and experimentation with subsystems for data collection, data management, processing, and visualization were started.

On September 13 and 14, 1994, a major site review was held at UCSC and NPS for our Office of Naval Research and DOD sponsors. This report documents the status of REINAS as it was presented at this review. It represents the mid-point of Phase IV – the Experimentation and System Verification Phase.

Acknowledgements

The Summer of 1994 represented a major effort in all aspects of the REINAS Project. We wish to acknowledge the dedicated work of all the UCSC faculty, students and staff, and especially our partners from MBARI, NPS and other interested groups. It is difficult to name all those who participated in the REINAS project, but the following contributed directly to the first half of Phase IV:

- **UCSC Faculty and Staff:**
  Patrick E. Mantey, Daniel M. Fernandez, Harwood G. Kolsky, Glen G. Langdon, Robert Levinson, Suresh Lodha, J.J. Garcia-Luna, Darrell D.E. Long, Andrew Muir, Alex T. Pang, Stephen Petersen, Craig M. Wittenbrink.
- **NPS Faculty and Staff:**
  Richard Lind, Paul Hirshberg, Wendell A. Nuss, Jeffrey Paduan.
- **MBARI Staff:**
  Bruce R. Gritton, Francisco Chavez.

Students who worked on the REINAS Project in 1994:

- **System and DBMS Architecture:**
  Bruce R. Montague, Eric C. Rosen, Linda Begley, Jose Mendoza, Theodore R. Haining, Carles Pi-Sunyer.
- **Instrumentation:**
  Catherine Tornabene, Robert Sheehan, John Wiederhold.
- **Instrument Interfaces:**
  Michelle D. Abram, Jimmy Chan, Scott Lin, Chris Thomas, David K. Schreiber.
- **Network Architecture:**
  Chane L. Fullmer, Dean Long, Shree Murthy, Jochen Behrens.
- **Visualization System:**
  Naim Alper, Hans-Peter Dommel, Jeffrey J. Furman, Tom H. Goodman, Elijah C. Saxon.
- **Modeling:**
  Cheng Tang.
- **Data Compression:**
  William W. Macy, Robert J. Antonucci, James Spring, Yi Zhou.

1. REINAS Introduction

The Real-Time[1] Environmental Information Network and Analysis System (REINAS) was started in 1992 as a multi-year project. This report documents the first half of Phase IV which mainly concerns Real-Time Experimentation and System Verification. Much of the work was done by students during the Summer 1994.

The REINAS System Design is described in the Phase III report [MLGL$^+$94] which provides a detailed description of the REINAS architecture, including the instrument support, data management and database design, load paths for data from instruments to storage and users, and the support for visualization. It also describes instruments, implementation details, and proposed uses.

1.1 Project Objectives:

The objectives of the REINAS project is to develop a prototype real-time system for

- Data Collection
- Data Assimilation
- Data Display
- Real-time Applications

of in-situ and/or remotely-sensed environmental data.

In the REINAS architecture, continuous real-time data is collected from a variety of dispersed instruments 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 upper part of Figure 1.1 depicts the REINAS system and data flow, from instruments to users, in its ultimate implementation. In the current prototype, which is an intermediate implementation, REINAS operates as in the lower figure. Full integration of all data sources into the database, and full development of the API (Application Program Interface) connecting visualization directly to the database, are essential components required for realization of the ultimate REINAS system.

1.2 Key Components:

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. Table 1.1 summarizes the REINAS key components and Table 1.2 the schedule for the first three phases.

---

[1] This chapter by P.E. Mantey

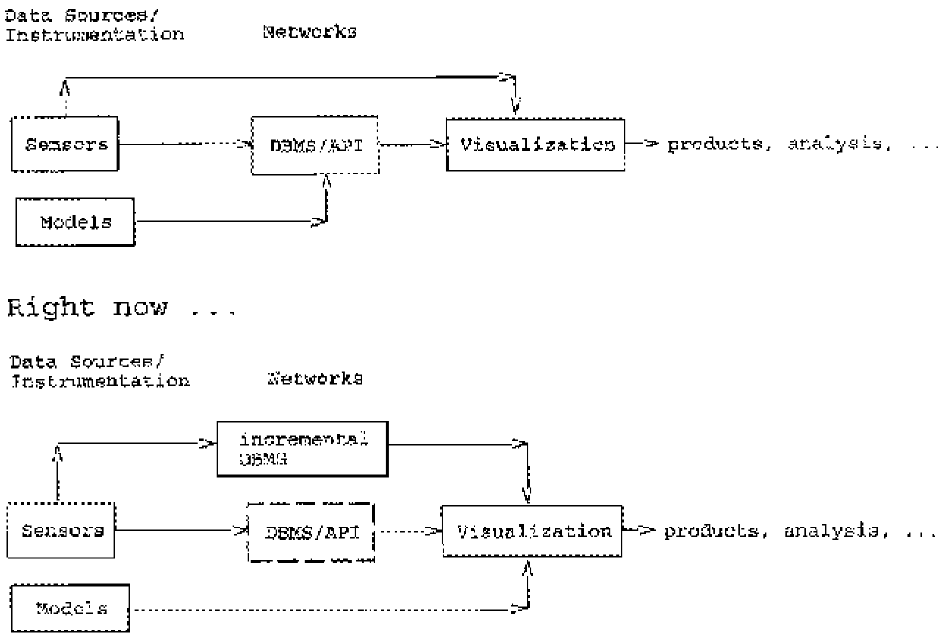

Figure 1.1: REINAS System Integration.

1.3 REINAS from a Meteorological Perspective

1.3.1 Collection of Observations

Cutting edge meteorology[2] requires the use of widely varied observing systems such as profilers, radars, high-frequency surface observations, as well as traditional surface and upper-air observations. Currently these systems are separate displays in operational settings and only brought together after considerable effort in research settings. REINAS solves this problem by providing a unified database for all types of observations.

1.3.2 Diagnosis of Current Atmospheric Structure

Operational meteorologists need tools to determine the current state of the atmosphere to make forecasts and research meteorologists need tools to determine the dynamic relationships for particular situations. Good diagnosis requires the display of all types of observations plus model renditions of the atmosphere in a convenient manner. REINAS does this by providing visualization tools that are capable of displaying observations, analyses of observations and models in a common display.

1.3.3 Forecasting

Operational meteorologists of all types require the ability to predict the state of the atmosphere in the future. Present numerical prediction models do an excellent job for the larger scales of motion in the atmosphere but are incapable of predicting details on the smaller scales. While good models exist that may be capable of such prediction, they have not been run in real-time. REINAS addresses this requirement by providing

---

[2] This section by Wendell Nuss, NPS

KEY COMPONENTS OF THE REINAS PROJECT

- Data Acquisition
  - Real-Time Instruments
  - Variable and High Data Rates
  - Automated Support for Instrument Calibration and Data Quality Checking
  - Support for Distributed and Diverse Sensors
  - Signal Processing
  - Data Compression
- Distributed Real-Time System Architecture
  - Modular
  - Extensible
  - Portable
  - Support Operational Users
  - Reliable (Fail-Soft) Data Integrity
- Support for Feedback Control of Sensors
  - Mobile Platforms
  - Variable Data Rates
  - Instrument "Steering"
- Computer Networks Linking
  - Instruments - Databases
  - Users
- Real-Time Visualization
  - "Now-Casting"
  - Visualization from Database(s)
  - Fusion of Data from Measurements and Models
  - Retrospective Analysis supporting Real-Time Uses

Table 1.1: Key Components of the REINAS Project.

a numerical model that can be run in real-time to make short range predictions of small-scale circulations.

1.4 Results from Year's 1 and 2 Studies

1.4.1 Naval Postgraduate School M.S. Theses as part of REINAS

1. Michael Foster - Evolution of Diurnal Surface Winds and Surface Currents for Monterey Bay
2. Robert Round - Climatology of Monterey Bay Sea-Breeze

REINAS SCHEDULE

- May 1992 – Start of REINAS Project – Phase I
  - Concept Design and Documentation
  - Characterize: Instruments, Data, Users and Uses of REINAS System
  - Create Project Plan
  - Assemble Staff
  - Evaluate System Technologies
  - Develop Preliminary Architecture
- January 1993 – Begin Phase II
  - Detailed Requirements Definition
  - Prototype Evaluations of Key Components
  - Refine Architecture
  - Develop Preliminary System Design
- July 1993 – Begin Phase III
  - Detailed System Design
  - Prototype Implementation
  - Development of REINAS Instrument Network
  - Connection of Real Instruments
  - Data Feeds from Other Instruments (MBARI, NPS, NOAA, etc.)
  - Addition of More Instruments
  - Database Design
  - Implementation of Data Load-Paths
  - Advanced Visualization

Table 1.2: **Schedule of the REINAS Project's First Two Years.**

3. Emil Petruncio - Characterization of Tidal Currents in Monterey Bay from Remote and In-Situ Measurements
4. Pat Cross - A Comparison of Modeled and Observed Ocean Mixed Layer Behavior in a Sea Breeze Influenced Coastal Region
5. Michael Knapp - Synoptic-Scale Influence on the Monterey Bay Sea-Breeze

1.4.2 Key Preliminary Science Results

1. Horizontal flow of air in the Monterey Bay often shows a convergent flow into the mouth of the bay (northwesterly to north and southwesterly to south). This flow then turns to flow primarily down the Salinas Valley.

2. Horizontal surface wind patterns have considerable diurnal variability.
3. Sea-breeze at Ft. Ord has 6 distinct patterns of diurnal variation. These range from little identifiable diurnal variation to abrupt onset of an intense sea-breeze.
4. Diurnal wind variations in the Monterey Bay have a measurable impact on the ocean circulation in the bay.
5. Diurnal variation of the winds in the Monterey Bay region is controlled by the large-scale pressure gradient and the diurnal variability of the boundary layer structure and stability. Large scale pressure gradient sets the maximum wind speed that can be achieved in afternoon. Boundary layer stability changes from night through the day to allow vertical momentum mixing to produce the observed afternoon surface sea-breeze.

1.5 Meteorological Science Objectives

The overall NPS science goals for the REINAS project are:

1. to better understand the interaction of the large-scale flow, the thermally driven sea-breeze, boundary layer processes, and the complex orography of Monterey Bay in producing the observed mesoscale atmospheric circulations in the Monterey Bay and adjacent coastal zone;
2. to develop in conjunction with UCSC an observing network of both surface and upper-air observing systems capable of defining the three dimensional mesoscale circulations over the Monterey Bay region;
3. to collaborate with UCSC in developing meteorologically relevant visualization tools that can be used to study circulations in the Monterey Bay region as well as more general meteorological applications;
4. to develop mesoscale modeling and data assimilation tools that can be used to help analyze and forecast small-scale structures by blending asynoptic, high frequency observations together with an appropriate numerical model.

Some specific scientific objectives are:

1. to characterize the nature of the mesoscale circulations in the Monterey Bay region under a variety of large-scale flow patterns in order to understand how the large-scale flow interacts with the local topography;
2. to understand the dependence of the observed diurnal variability of the surface winds on the larger scale thermal forcing from the California Central Valley and on the local thermal forcing across the immediate coastline (continent-ocean circulation versus mesoscale sea-breeze);
3. to understand what produces the convergent flow into the Monterey Bay region and how it is driven diurnally;
4. to describe and understand the variations of the boundary layer stability and inversion height as they relate to large-scale forcing and local circulations;
5. to understand how clouds impact/drive the diurnal circulation.

2. REINAS Instrumentation

The REINAS[1] system has been designed to accommodate data from a variety of data sources. It must be able to accept data from many diverse sources at greatly varying rates. Sampling periods may be as short as five second (e.g., surface Met stations) or as long as one hour (e.g., vertical wind profiler or CODAR in long-term averaging mode). Instrument interfacing hardware will also vary greatly. Some instruments, such as surface MET stations, will provide relatively primitive interfaces (e.g. a generic datalogger). Others may interface to REINAS using modern microcomputers.

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.

REINAS has experimented with sample periods as small as one second and generating averaged datasets for archival with temporal resolutions of one minute. At present we do not sample any instrument faster than 5 seconds. Physical limitations in some instrumentation makes this impossible. Typical sample rates for REINAS sites is 10 seconds, other sites, 1 minute to 1 hour.

2.1 MET Stations:

A typical REINAS MET Station is distributed by Campbell Scientific and consists of:

- R.M. Young Wind Monitor
- Vaisala Pressure Transducer
- Vaisala Temp./Humidity probe
- Tipping Bucket Rain Gauge

2.2 CODAR

Coastal Ocean Dynamics Applications Radar (CODAR) is a remote sensing system developed by the Wave Propagation Laboratory (WPL) now known as the Environmental Technology Laboratory (ETL), which is part of the National Oceanic and Atmospheric Administration (NOAA). The basic principle of CODAR is that radar energy in the high-frequency band (3 - 30 MHz) is resonantly backscattered from the ocean surface by surface waves that are half the radar wavelength [Cro55].

Because the resonance is due to reflection from surface waves of a known wavelength (one half the wavelength of the incident radar wave), the phase speed of the reflecting ocean wave is known precisely. Reflected energy received back at the radar site is Doppler-shifted in frequency by an amount equal to the contributions from waves traveling toward and away from the radar plus the contribution from the background current field upon which

---

[1] Chapter by Dan Fernandez

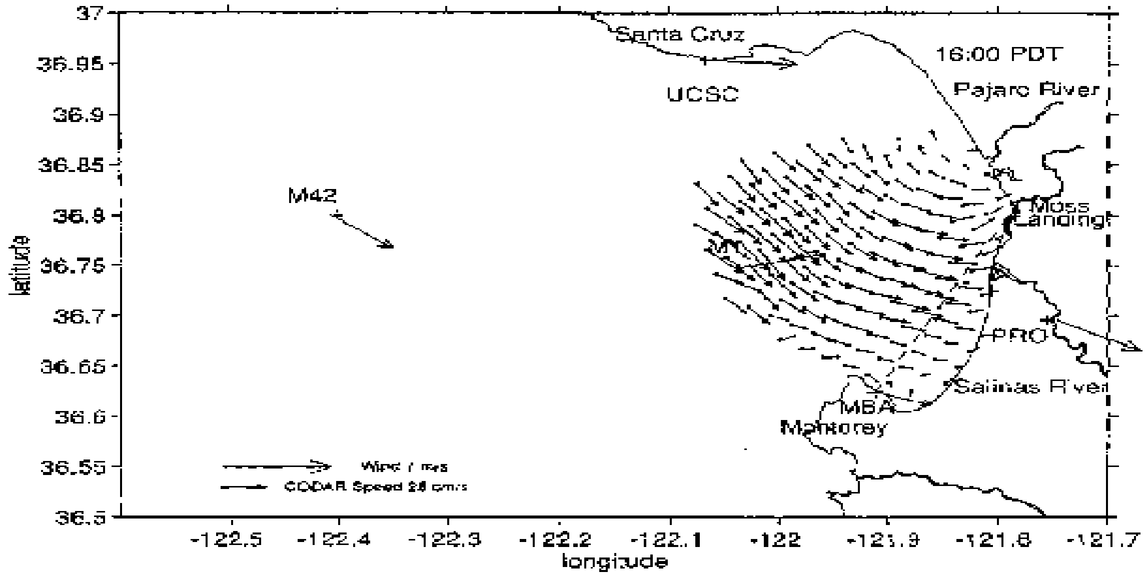

Figure 2.1: Example CODAR results: Mean CODAR-derived current and mean wind fields for 1600 PDT in September 1992.

the waves are traveling. Removal of the known wave contributions provides a remotely sensed measurement of the surface current [BEW77], [BLC85]. The depth extent of the measurement depends on the depth of current influence on the reflecting surface waves. The depth range is estimated to be confined to the upper 1m of the water column.

Currently, newer versions of CODAR (called SeaSondes) are connected to the REINAS system from Long Marine Lab and Point Pinos. The older CODAR at Moss Landing will be connected by 12/1/94. The multiple CODAR sites in the Monterey Bay area permit excellent coverage of the ocean surface currents within the Bay and beyond the mouth of the Bay.

2.3 Radar Wind Profiler

There are two radar wind profilers owned by the Naval Postgraduate School that are located at Fort Ord. One operates at 404 MHz and the other operates at 915 MHz. Both systems broadcast three beams from which a vector is calculated to determine the actual speed and direction of the wind. The newer 915 MHz system offers vertical resolution as small as 60 meters and a maximum height of 3 to 5 km.

In addition to these wind profilers, NOAA leased three of their own to REINAS for spring-summer 1994. These profilers (also operated at 915 MHz) were located at Hollister, Fort Ord, and Long Marine Lab. The Long Marine Lab profiler will remain there until September, 1995. The Point Sur wind profiler and the Long Marine Lab wind profiler were connected to the REINAS system so that their data was accessible in real time.

2.4 Other Miscellaneous Data Sources

- Acoustic Doppler current profilers (ADCP)
- CTD

- Sea temperature from land sites
- Wave data from NOAA buoys
- LIDAR
- SODAR, sonic anemometer
- Satellite data: GOES, AVHRR, SEAWIFFS
- NEXRAD

2.5 Oceanographic Applications of REINAS

REINAS applications for oceanography in Monterey Bay may include: research applications, search/rescue applications, operational use of real time data, and the development and test of new forms of instrumentation.

An important factor in making data useful is the validation of instruments against each other. For example, we plan to compare data that has been collected or will be collected from the following instruments. More details may be found in the Phase III Report [MLGL$^+$94].

- CODAR Seasondes
- Marconi OSCR
- Stanford / U of M / ERIM HF Radar
- LIDAR - Wind Profiler

REINAS is already a useful platform for instrument verification, calibration, and comparison. A basic suite of MET stations, CODARs, and wind profilers has been connected in "real time". Rich data sets are just now becoming available to us and our partner institutions. New instruments will continue to be added to the basic network.

2.6 Portable Meteorological Station

The development of "Port-a-Met" was completed late in June 1994. This movable MET station has a radio link to UCSC, thus providing real-time access to remote, but portable instrumentation. It is a portable battery-powered MET station and REINAS PC architecture which is linked to the ethernet via half-duplex 9600 baud radio modems from Teledesign. Port-a-Met was demonstrated during the ONR site visit on September 14, 1994.

Plans are currently underway to deploy Port-a-MET on the R/V Pt. Sur, for an extended 10 day trip in the Monterey Bay sanctuary area. The trip begins on 11/7 and runs through 11/16. Another plan is to refurbish a weather station on the MBARI research vessel Pt. Lobos, and link it to the Internet via MBARI's existing microwave ship-to-shore link.

3. Advanced Visualization

3.1 Background

The visualization[1] component of REINAS is designed to meet the various needs of its users as identified in the REINAS Phase Reports [MLP+93],[MLGL+94]. The highlights of the visualization system include: an integrated interface for users to get to their three-dimensional time-dependent data; support for real-time monitoring and retrospective analyses of model and sensor data; extensible system to explore different ways of visualizing data; and support for collaborative visualization among geographically dispersed scientists and data sets.

The present status of the program was shown in a live demonstration at the September 13, 1994, ONR meeting and a video tape has been prepared to show visualization highlights. To date the program has taken approximately 5 student years of work. The code size is approximately 40K lines of C code. Development and testing is done on SGI platforms using IRIS GL.

The longer term plan is to port the visualization program to OpenGL, a window and platform independent version of IRIS GL. The graphics functions are left pretty intact and different library functions are provided for dealing with different window and mouse events. Several vendors have already signed on to develop and support OpenGL applications on their platforms. The list include SGI, IBM, DEC, and MicroSoft. Although some vendors have already released OpenGL, we are waiting for it to stabilize before doing our conversion.

REINAS Visualization is organized into three modes: Monitor, Forecast and Analysis corresponding to the needs of the three main groups of users. We now report on the features and current status of each mode.

3.1.1 Monitor Mode

In the monitor mode, users can watch the most current state of the environment. See Figure 3.1. Users have a bird's eye view of the region of interest. Environmental sensors are represented by simple icons. Users can select one or more icons to monitor the readings from those sensors. Selection of sites can be done by clicking on the icons or by clicking on the items in the pulldown menu. Users have the option of obtaining a qualitative (see interpolation below) or a quantitative view. For the latter, a popup window is provided for each selected site. Users can then select the field parameter they want plotted.

At present the system can get data and display the most recent sensor data from both the Xmet and Oracle servers. Present sensors include: fixed and portable Met stations, wind profilers, CODAR, and ADCP. Work is planned to include other sensor data such as ship tracks, seal tracks, LIDAR, and others. At present model data is not available together with observation data in this mode. Retrospective sensor data query and display is also not currently supported except for those available through files.

---

[1] This chapter by Alex Pang

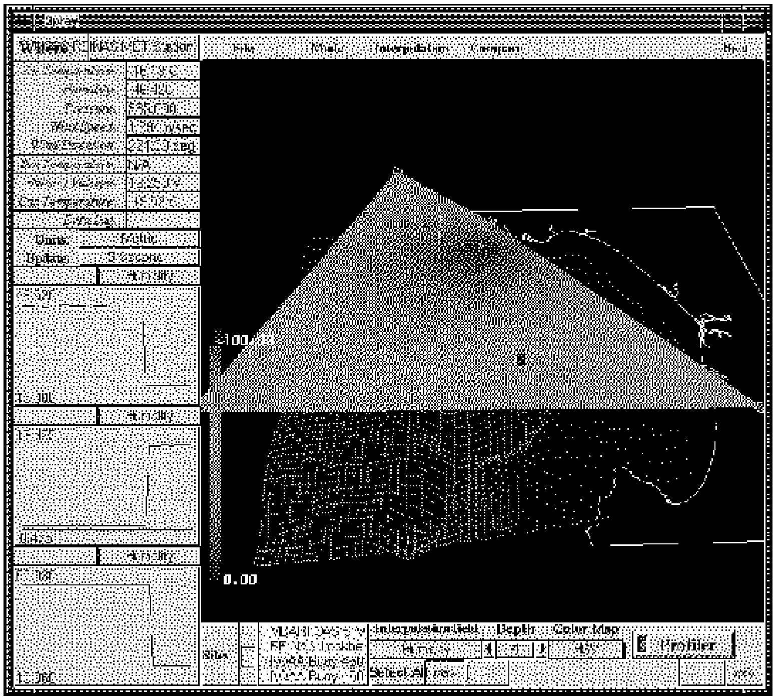

Figure 3.1: View of the Monterey Bay showing three sensor locations and an interpolated humidity field from a subset of these sensors.

3.1.2 Forecast Mode

Operational forecasters will want to look at standard products from forecast models, satellite observations, and generate standard products from model output e.g. height against vorticity, humidity, and temperature. They can also generate customized products e.g. different projections, different contour spacing, and heights. One can register and overlay observation data with products. (e.g. wind barbs, animated goes images).

Figure 3.2 shows a typical forecast product. Users can customize forecast products according to their needs. For example, user-specified contour spacing, user-specified pressure height, etc. These parameters can be specified textually or with sliders. One can watch sensors individually in textual form or time charts as data come in, and interpolate sparse sensor data.

3.1.3 Analysis Mode

Most of the visualization efforts have been concentrated in this mode. It allows users to explore large data sets interactively using different visualization techniques. It is also extensible and can easily grow with users' needs.

Spray Rendering: We provide users with the metaphor of spray painting their data sets as a means of visualizing them [PS93, PAFW93]. In its simplest form, data are painted or rendered visible by the color of the paint particles. By using different types of paint particles, data can be visualized in different ways. Figure 3.3 shows the interfaces available in analysis mode as well as illustrate some of the possible visualization methods.

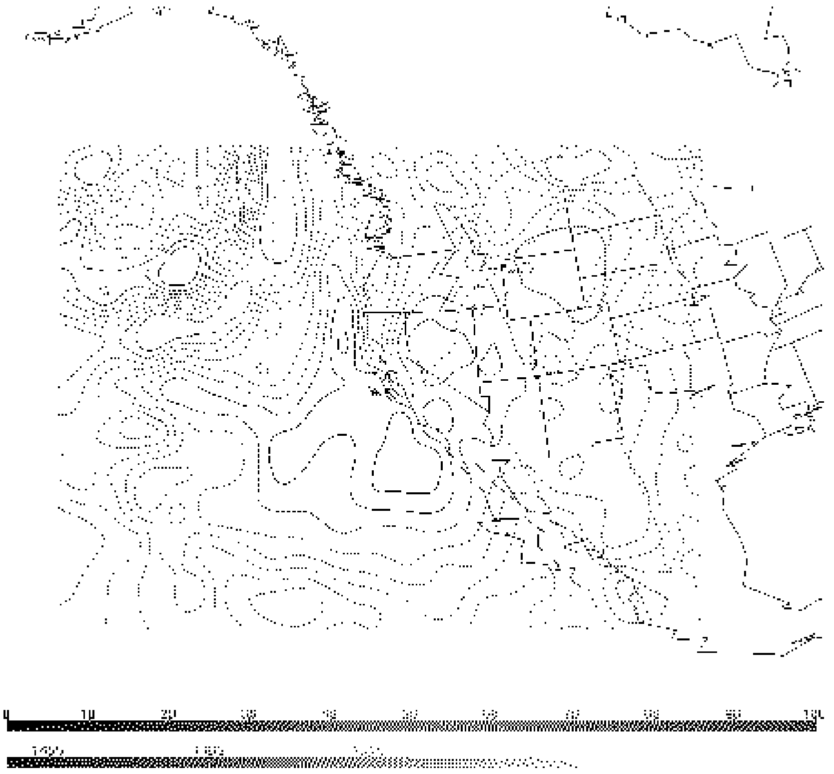

Figure 3.2: Sample forecast product

Spray is similar to other modular visualization environments (MVE) like AVS, Explorer, Data-explorer, in terms of extensibility, modularity and drag and click interface[PW95]. Spray differs in terms of execution flow (active agents vs data-flow) and finer granularity (making it more flexible). Spray is a research system that is continuously evolving. Currently, it works with rectilinear grids only and does not utilize database API calls yet. It is still unoptimized in that it does not exploit parallelism.

The key component of spray rendering is how the paint particles are defined. They are essentially smart particles (or sparts) which are sent into the data space to seek out features of interest and highlight them. Among the advantages of this visualization framework are: grid independence (sparts operate in a local subset of the data space and do not care whether data is regularly or irregularly gridded), ability to handle large data sets (sparts can be "large" and provide a lower resolution view of the data set or they can be "small" and provide a detailed view of an area of interest), extensible (it is easy to design new sparts). Sparts can also travel through time-dependent data sets.

Region Selector: Originally, the design of the visualization component assumed that the physical scale of study would be comparable to the Monterey Bay. This has since been expanded, at the request of some of our users, to a larger area. We have added two mechanisms to allow users to navigate through the larger space. The first method allows the user to zoom in/out and pan around using a combination of mouse and button selections. This is desirable for looking at regions close to the current area of study. The second method provides a tool for selecting a region of interest from a 3D globe. This method is preferable when the user wants to jump around and look at geographically distant data sets. It allows REINAS visualization to examine registered data sets from different localities. It

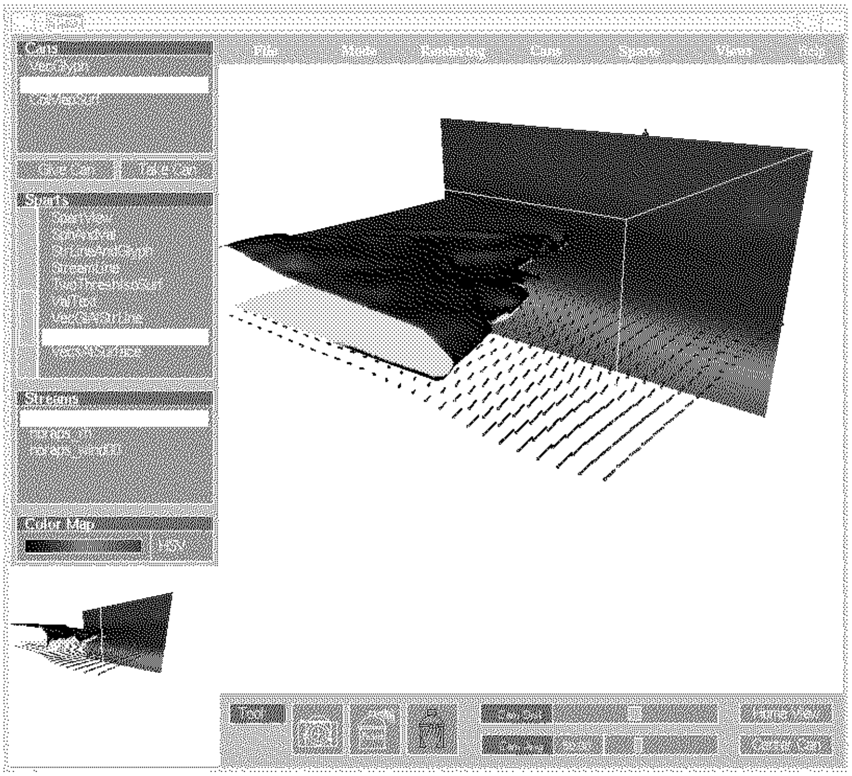

Figure 3.3: Analysis mode interface

is user definable preset regions of interest. One can select projection method and vertical exaggeration.

Related Research Areas:

- Uncertainty visualization.
- Collaborative visualization.
- Virtual reality interface for scientific collaboration.
- Integration of visualization and database e.g. to handle very large data sets; spatial/ temporal queries.
- Multi-platform extension (OpenGL) and GUI issues.
- Data assimilation for integration of model and observation data.
- Parallelization and efficiency issues.
- Irregular grids.

3.2 Visualization Programming and Spray Rendering

Mix and Match[2] provides users with the ability to graphically create new sparts by mixing and matching different spart components [PA94],[PA95]. The components are organized into four categories that reflect the stages of the particle nature of the spray rendering model used to achieve the visualization techniques.

---

[2] This section by Naim Alper

1. *Target* behavior functions are feature detection components. They usually test to see if a condition is satisfied at the current location of the spart.
2. *Visual* behavior functions are the key visualization components. They are responsible for the conditional output of the visualization objects.
3. *Position* functions update the current position of the spart. These can be absolute or dependent on the data as in vector fields.
4. *Death* functions determine when the spart should die. There is also a birth function in this category that spawns new sparts.

A spart composition is the specification of the components that make up the spart and the connections between them. Users can select components from a browser, drop it onto a canvas and graphically connect them. This composition defines how the spart behaves at the current location. The compositions and the components are usually quite simple. However, complex visualizations can be obtained by multiple uses of multiple sparts.

The exploratory aspect of the visualization has been emphasized by using the spray can metaphor of spray rendering to launch the sparts and interact with the data set. Visualization is incremental and the result of direct interactions with the data set. By controlling how and how often the sparts are launched and when the visualization objects are updated many different interactions can be achieved. For instance, a spart could be used in a probe like interaction where the visualization objects and the scene are updated after each delivery. This allows interactive exploration of the data. The same spart can be used in flood mode where the visualization objects are generated at grid nodes. This is useful when the visualization objects are continuous as in an iso-surface.

The system is extensible so that new components can be added to it by writing C code. The integration of this component into the system is facilitated by a configuration manager.

3.3 Collaborative Visualization

In the collaborative mode[3], a number of participants can contribute in the creation of a visualization product over the network[PWG95]. There are several components that are needed to make this feasible: session manager, sharing data/cans, floor control, multiple window, audio/video support, different collaboration/compression levels.

The session manager is a piece of software that maintains a list of ongoing sessions and the participants in each session. A session consists of a group of participants working on a common theme or problem. Participants may join or leave the session at any time. Thus, the session manager needs to inform the application programs of any changes so that traffic delays are minimized and also late comers may easily catch up with what is going on.

Users can collaborate at different levels. Sharing can occur at the image (visualization product) level, spray can level (abstract visualization objects – AVOs) or data stream level (e.g. files). At the image level, participants can see what the other participants see and may perhaps be able to change view points. At the can level, participants have access to a list of public spray cans put up by other participants. Using these public cans will generate AVOs from the remote hosts and distributed to other participants. Users may also give permission to other participants to have direct access to data streams and replicate those on local machines for faster response times.

---

[3] This section by Tom Goodman

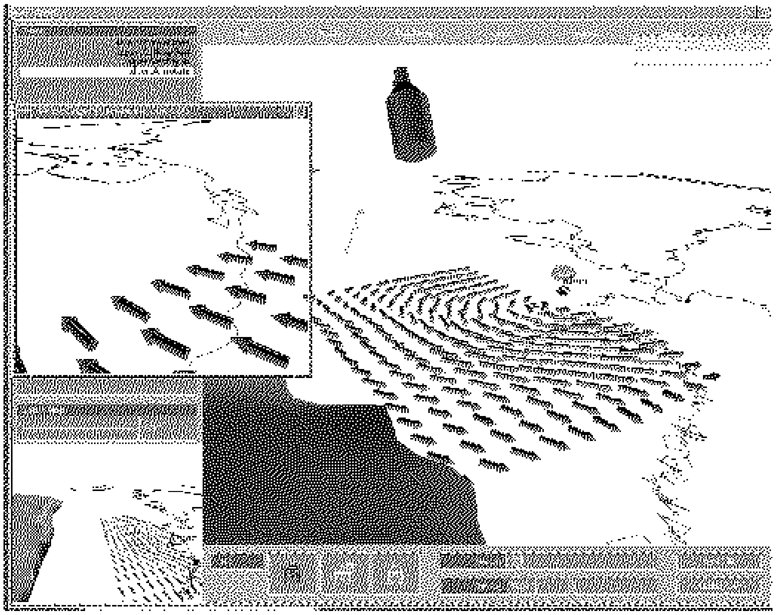

Figure 3.4: Collaborative visualization. The larger view shows what the local user is viewing. The small views show what the collaborator is looking at. The "eyeballs" show the locations of the viewers.

In single user mode, users can create multiple cans but can control only one can at a time (limited by the number of input device – mouse). With multiple users and sharing of spray cans, it is possible that more than one user want to use a particular spray can. Floor control software regulates the use of spray cans. Figure 3.4 shows how the two views are presented on the screen.

Just as users can have local and public spray cans, they can also have local and public windows. Users work in their local window and may once in a while look at the public window to see what others are doing. The public window is also where one might do a broadcast as in during a briefing mode to show other users an item of interest.

Since participants are assumed to be geographically distributed, it may be difficult to get a point across, or try to get the attention of other participants by simply moving the cursor around. It is therefore necessary to include audio/video tools to help facilitate communication.

The different levels of collaboration also implies different requirements for compression. Tradeoffs will have to be made between graphics workstation capabilities, network bandwidth and compression levels. Objects that need to be transmitted can either be images, AVOs (together with can parameters and other transformation matrices), or files.

3.4 Visualization of Uncertainty

Measured[4] environmental data have inherent uncertainty which is often ignored in visualization. Radar, light, and sound are used to remotely sense physical phenomena, but because of instrument limitations the measurements are approximate. We have developed

---

[4] This section by Craig Wittenbrink

new vector glyphs, or icons, to visualize uncertain winds and ocean currents[WSF$^+$94]. Our approach is to include uncertainty in direction and magnitude, as well as the median direction and length, in vector glyph plots. We examine three data sources: meteorological stations, doppler radar wind profilers, and doppler radar ocean surface current radars, and compare our glyphs to traditional ones.

Meteorological stations measure wind with an anemometer and vane, and the accuracy is good, but to compare winds from many sites, winds are often averaged over minutes or hours. The variation during an hour is an uncertainty in time. Often the sparsely located sites are interpolated, which adds a derived uncertainty in space. A similar processing method is used with the radars. Each radar–wind profiler and ocean current radar–take a volume sample which averages the returns. The time series data has a wealth of information, which may be examined in detail, but is not used in the vector visualizations. Wind profilers have weak scattering from dry air, and measurements are also influenced by airplanes and migrating birds [WoRSR$^+$94]. Ocean surface current radars have varying performance depending on the ocean conditions. Current methods of display simply threshold or ignore uncertain vector component measurements.

We illustrate how visualizing vector deviations allows users to more accurately interpret their data when uncertainty increases with distance from a measurement. We show several meteorological station sites (cones and cylinders) and regularly interpolate a vector field from those sites' winds, without uncertainty, Figure 3.5, and with uncertainty, Figure 3.6. We use both qualitative and quantitative methods to evaluate our glyphs. We plan to perform comparison tests with experts–meteorologists and oceanographers–and see which graphics are most effective in discerning patterns in the data. Trends in the uncertainty will be tested by user evaluation, using control patterns and measured data. We also plan to evaluate the data ink maximization, where the information density is compared. We hope to show that visualizing uncertainty enhances understanding of the continuous range of data quality[PFN94].

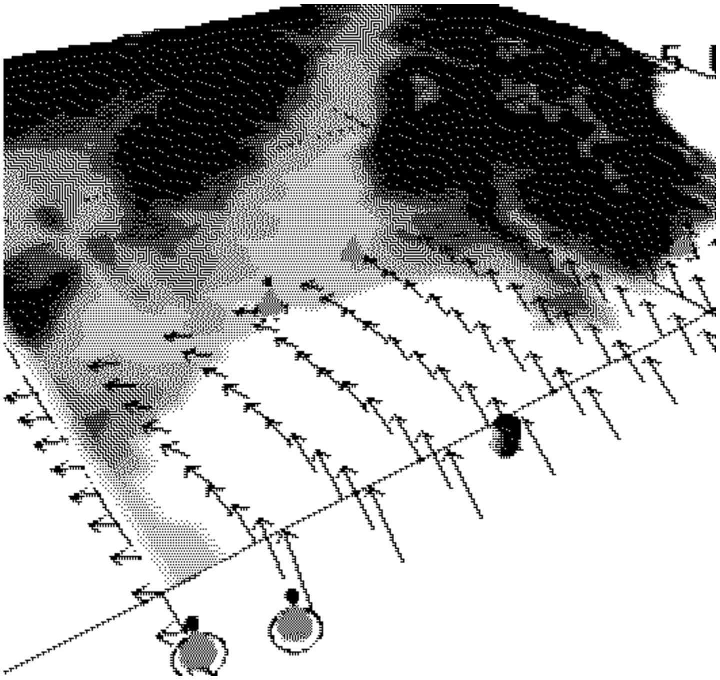

Figure 3.5: Interpolated winds over the Monterey Bay region looking south-east near Santa Cruz, on a regular grid using vector glyphs.

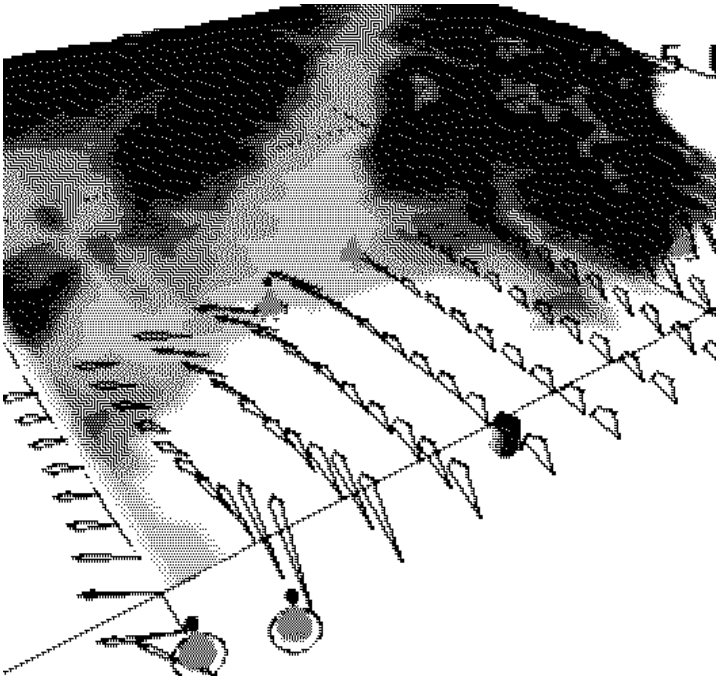

Figure 3.6: Interpolated winds over the Monterey Bay region, on a regular grid with uncertainty angle glyphs. Uncertainty grows with distance from meteorological stations giving a much different impression from the traditional vector glyphs.

4. Systems, Networks, Databases and Schemas

4.1 Project Challenges

A major goal of REINAS [1] is to apply techniques from Computer Science and Engineering to support *real-time* research in Environmental Science. The system must be extensible and applicable to many problem domains. It must be able to store and retrieve large quantities of data. It must provide easy access to data for visualization, reduction and analysis. It should allow users to direct sensors, to control access, adjust parameters, and follow developing events.

In this project we have had to address difficult computer engineering problems that are seldom seen in academic research, i.e. Building large heterogeneous systems. Having built the first prototype, we see many areas of research that need to be addressed, especially in performance and security.

User Issues:

Data quality and pedigree are major issues for users. Multiple versions of the data should be kept to ensure quality. Controlled access to data and sensors is necessary. The scientist in charge should be able to decide who gets access, and be able to steer it when feasible. System must operate in several modes: Real-time, historical, and retrospective.

Design Principles:

"Protocols define the system" (Cheriton) [Che88].

The design must provide Extensibility. Instruments should be *plug-and-play.*

The design must provide Scalability. Adding new sensors, more storage, or users should be easy.

The design must provide Resiliency. The failure of a node or the network should not affect the entire system.

Distributed Systems:

Benefits of distributed systems include: Increased processing capacity, Highly fault-tolerant (k-resilient) [Svo84]., better *scalability,* and *extensibility,* [CL91]. and reduced incremental costs.

However added complexities that must be addressed: consistency, naming, security and a single system view.

Instrument Sites: Most non-trivial distributed systems span one building, but REINAS spans an entire region in central California. See Figure 4.1. 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. Using a small personal computer, with a Unix operating system and with a standardized interface for attaching instruments, and by attaching this PC to the network using standard networking software, each instrument becomes an intelligent device on the REINAS network. An interactive electronic log book tied to the database will populate and track instrument metadata used for calibration and control.

---

[1] This chapter by Darrell Long, UCSC

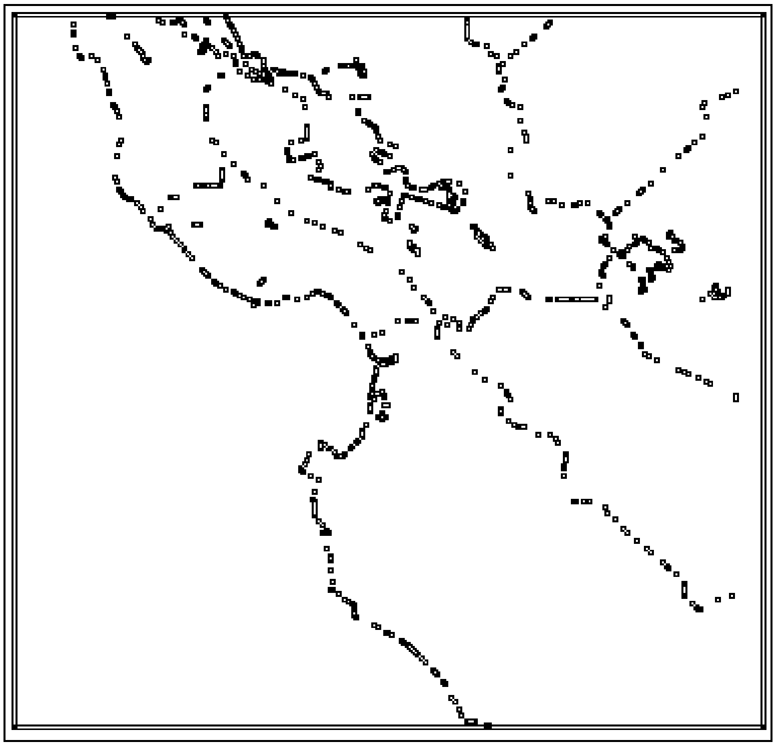

Figure 4.1: REINAS Fully Connected Instrument Sites in Monterey Bay Area, September 1994

4.2 Architectural Overview

Networks

REINAS consists of three logical networks: instrument, database, and user networks. See Figure 4.2. Duplicate resources are possible in all networks. The Instrument nodes are Unix PCs. Normally they are continuously connected but are able to operate autonomously. The database schema is tailored to environmental data. It provides memory resident data for real-time support. Data is organized into system-wide time-series.

Instrument Nodes

The instrument nodes define the interface of the instrument to the system (*plug-and-play*). They provide fault tolerance through logging even when disconnected. They provide local processing for data compression, image registration, and first-order quality control. They can also provide network instruments remote steering, and a means to adjust parameters.

Core Instrumentation

Weather or MET Stations, provide time-series of scalar measured physical quantities (temperature, wind speed, direction, humidity, pressure), with *real-time* sampling periods of 1 to 10 seconds.

CODAR SeaSondes measure ocean surface-current fields from fixed coastal radar sites. Each radar produces radial current fields. Two or more sites combine to produce surface-current vector field.

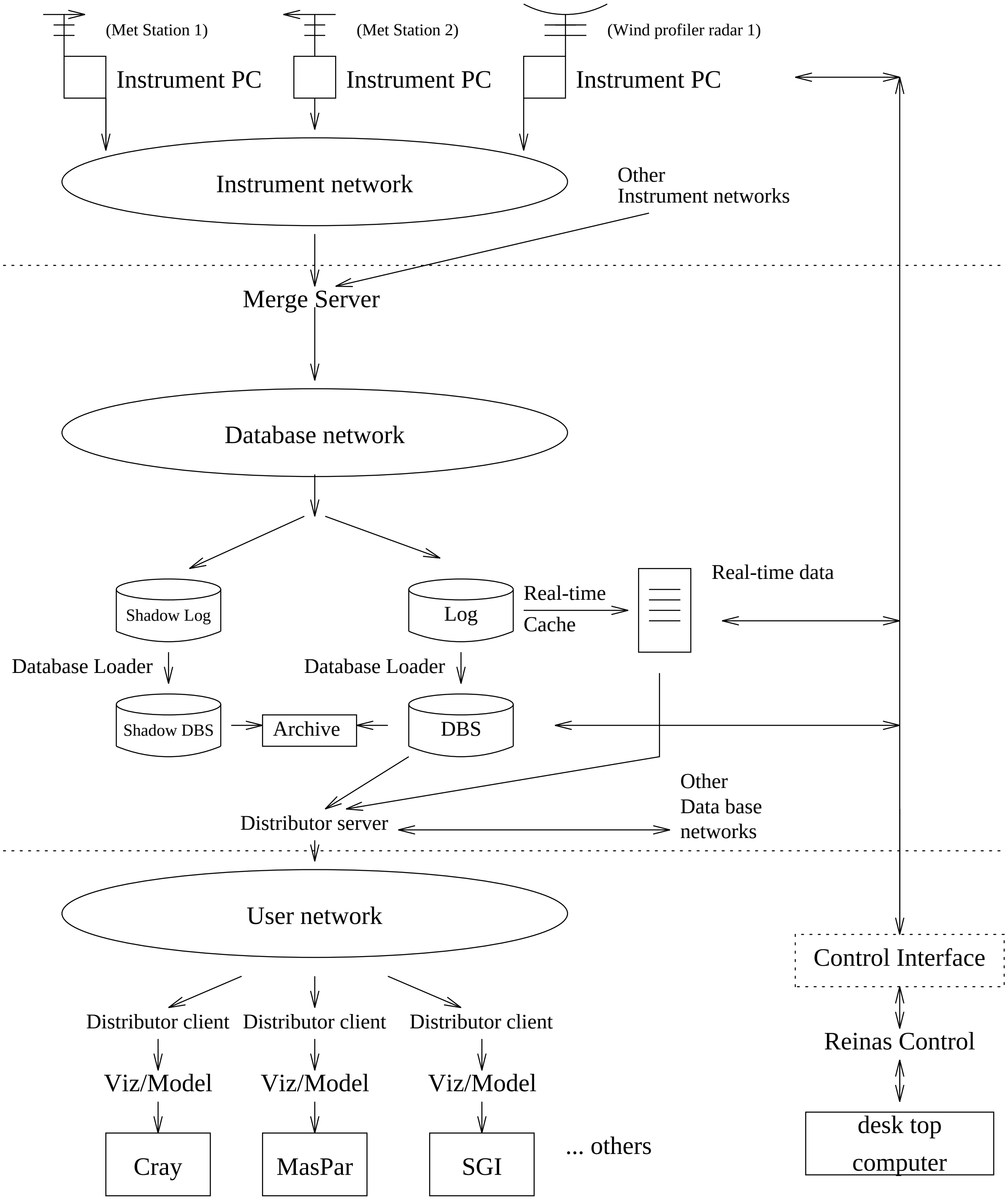

Figure 4.2: Overview of REINAS Logical Structure

Vertical Wind Profilers provide vertical atmospheric profiles. Using radar. they produce wind speed and direction estimates.

Acoustic Doppler Current Profilers may be used later.

Instrument Node Software

The instrument nodes include the following software:

1. Collector – a master controller and log manager.
2. Reader – transmits logged data to the database.
3. Device Managers – log real-time device data using RVM. [SMK$^+$93]
4. Device Interface Library – Standard interfaces, for instance: serial line I/O and data parsing.

Instrument Research Issues Major research issues in this are:

- Security – authenticated access to instruments for control applications.
- Fault tolerance – logging and metadata techniques.
- Mobility – network protocols, location, caching.
- Data compression.

4.2.1 The REINAS Database

Queries to the REINAS database consist primarily of request for time sequences of data. Archival data are used by projects other than the one that collected it. Different versions of data will exist as data quality assessments change. Data will be owned by different people and organizations, and will be stored at different physical sites. Data access for applications provided through an Application Programmer Interface (API) written in C [KR78].

Database Nodes

Database nodes provide content-based retrieval of data. They enforce consistency of the data, provide access control, and bind metadata closely to measurements. Automatic entry ensures consistency.

Database Node Software The database nodes contain the following software:

- Merge server – Logs data from instrument nodes.
- Database loader – Loads data from log into database.
- Real-time cache – Keeps a copy of the most current data.
- Shadow – Forwards data from one log to another.
- Distributor – Supports client application API.

Issues:

- Multidatabase Interoperability: Data may be stored by science partners using different database managers with independently developed schemas [CS93].
- Query Optimization: The real-time nature of the system requires that queries must be fast [CS93].
- Data Management: Storage of scientific data and metadata in an extended relational database system [CS93].

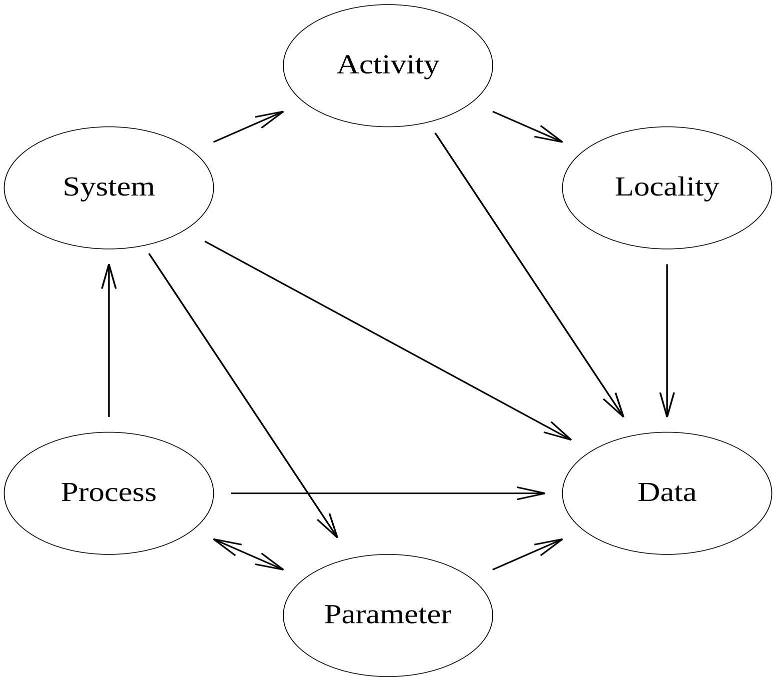

Figure 4.3: Example Realms

4.2.2 Realms:

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.

4.2.3 Schema Organization

To manage the large amount of metadata that REINAS will generate, the REINAS database system schema is divided into several realms. Each realm contains metadata about a specific part of the REINAS system. The division of metadata into realms also allows version, quality assessment, and collection activity data can be compactly maintained.

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.

4.2.4 The Data Stream Model

REINAS uses a data stream model in which data is stored in an extended relational database system and optimized for temporal accesses. Data are separated into source dependent and independent parts and all data of the same type is physically stored in streams.

Database Problems:

Centralized databases have poor performance. Data from all science partner organizations must be managed by a single group. Science partners may be separated by network links of variable quality. Data is accessed through a single schema, making the integration of data from existing databases difficult. There is incompatibility of commercial DBMS [CS93].

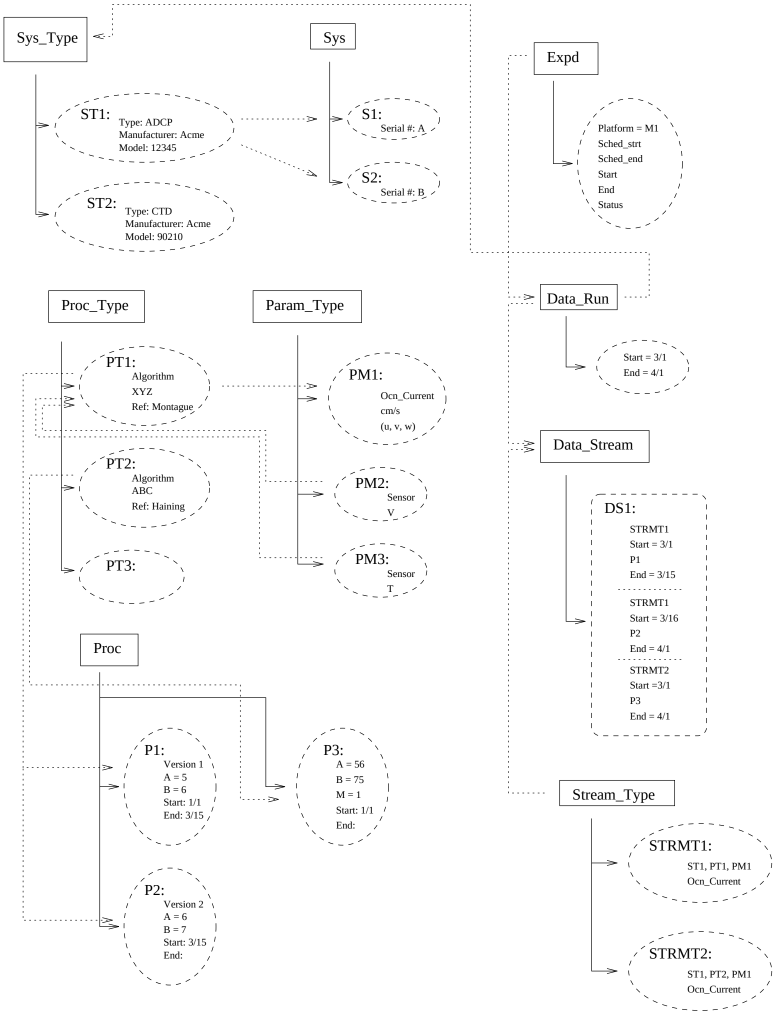

Figure 4.4: An Example Schema

4.2.5 Proposed Multidatabase

The advantages of a multidatabase architecture are: [CS93].

- Performance is not limited by the throughput of a connection to a single machine.
- Data gathered by each partner organizations can be stored at different sites.
- Data can be organized by different schema, allowing easier integration of existing archives.
- Locality is maintained, making frequent queries to local data much faster.

Major Research Issues for multidatabases are:

- Query response time: Size estimation [KS92], network link optimization, result caching, I/O scheduling.
- Access to real-time and very recent data: etensible hashing schemes [LNS93].
- Schema Resolution: Each database may employ different local schema, Schema just adjust to local convention to generate a REINAS schema overlay [SL90].

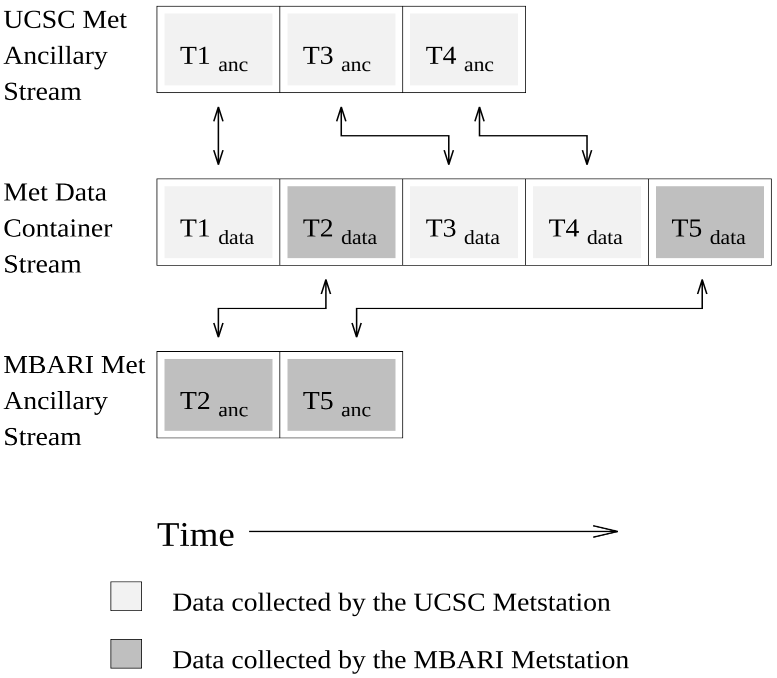

Figure 4.5: The Data Stream Model. Showing how data is handled between two Collection Stations.

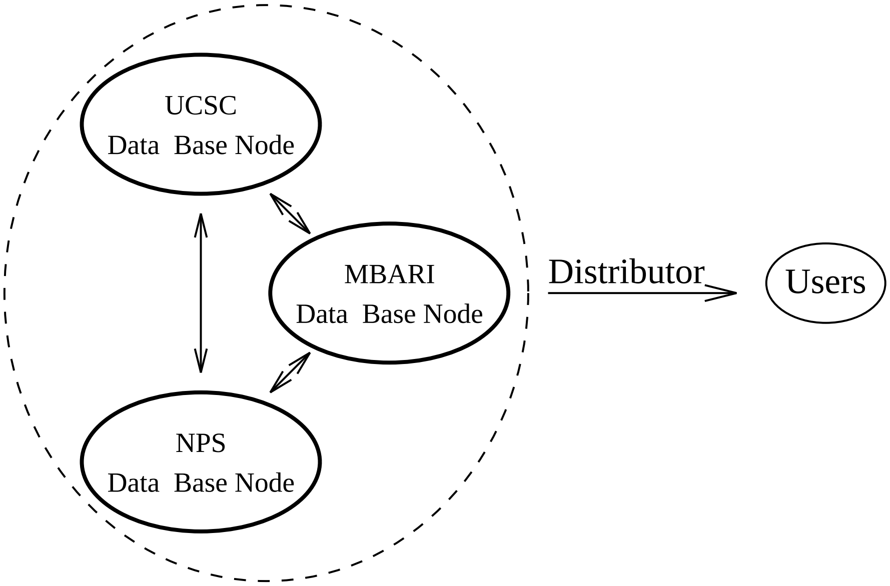

Figure 4.6: Database Subsystem Design. Because several science partners independently collect data and retain ownership of it, the REINAS database subsystem is designed to operate as a distributed, multidatabase that is accessible through a single distributor process.

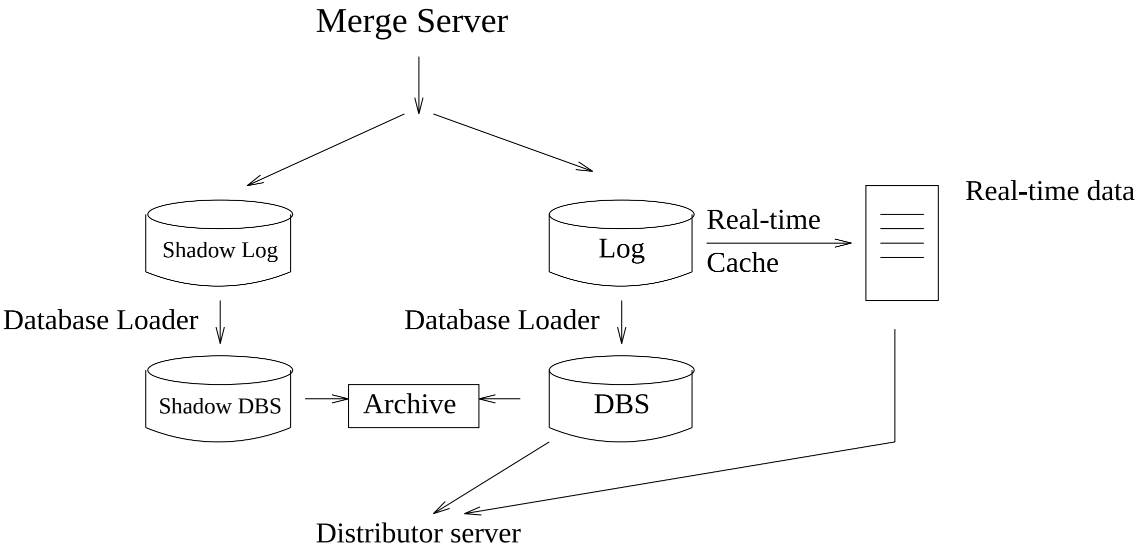

Figure 4.7: Current Database Structure

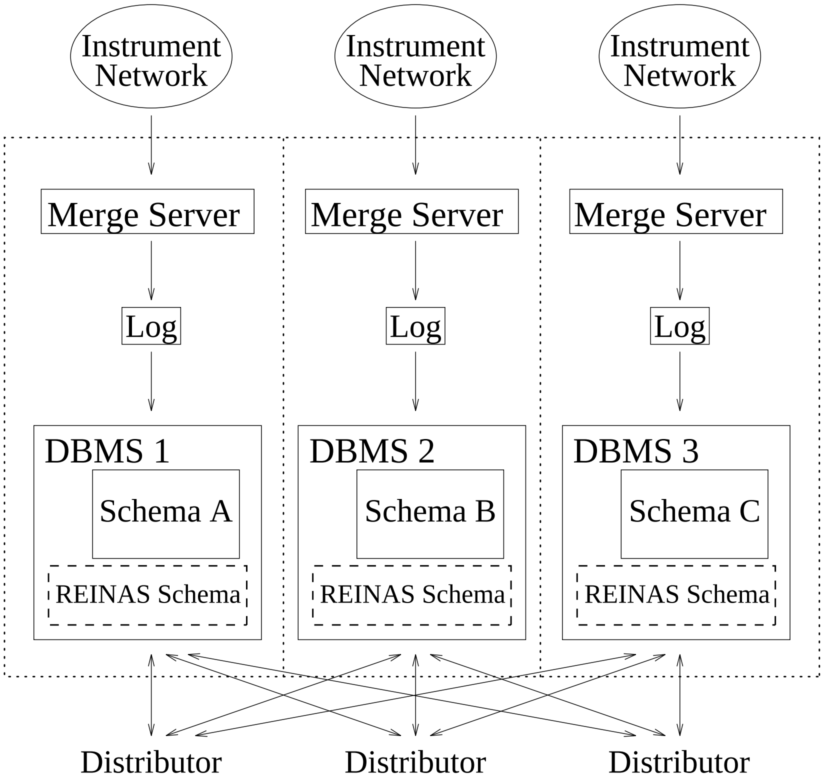

Figure 4.8: Proposed REINAS Multidatabase

- Schema translation: Reconcile schema using translation rules and methods. Objects may also be used to handle schematic differences [Ber91].
- Write-mostly queries: REINAS uses write-mostly queries. Current databases handle this *poorly.*
- Large data products: Data products generated will be too large to be generated on-the-fly. It is necessary to generate data products incrementally, amortizing processing time and resource usage [Rou91].

4.3 Data Management Design Philosophy

4.3.1 REINAS Specific Problems

Controllable Instruments: [2] Controlling instruments enables rapid changes in operation of sensors. This means less latency in performance and quality assessment.

This implies a need for:

"Active Logs" for tracking and control. The ability to monitor and playback instrument performance, and the need for isolated plus regional Comparison Quality Checking.

Real-time and Retrospective Analysis: Applications require continuous updates, retrieval-for regional monitoring, retrieval for source specific monitoring, and ad hoc retrieval (using broad based selection criteria).

This implies need for:

Bimodal "physical organization", one organized for fast update, and another for adequate retrieval of broadly indexed attributes of geophysical observations.

One "physical organization" supports regional monitoring of small number of parameters (geophysical observations of type x OR status of all sensors of type y) Another "physical organization"supports monitoring of sensor observations and/or status for a specific instrument.

4.3.2 Systemic Data Management Problems for the Sciences

Isolated "Islands" of Data: Data can be unavailable because it is partitioned by: data class, discipline, metadata .vs. data, structured data .vs. unstructured data, data aggregates, data representations, structures, syntax and semantics, and data sources.

Non-integrated data: Data can be separated by acquisition, delivery, management, analysis, and presentation. This implies separate toolsets, different platforms, and no lineage.

Non-extensible architectures: Old systems can prevent the use of new sensor technology, new data management technology, new data items, representations, structures, and higher granularity (more selective query).

If the system is to be extensible and stable this implies separate load paths within a common information architecture based on appropriate generalizations and specializations.

Non-integrated management of multiple information classes: These include: measurements, observations, derived Observations, synthetic Observations (Nowcasts, Forecasts), interpretations (features, phenomena, etc.), and documentation (unstructured to structured).

---

[2] This section by Bruce Gritton, MBARI

REINAS DATA MANAGEMENT APPROACH

- Life Cycle View of Scientific Data and Information: Collection implies Long Term Usefulness.
- Commercial off the Shelf Data Management Platforms: Relational implies with Object Views, Extended Relational implies with Objects.
- Eliminate Metadata/Data Division: Metadata includes Content, Representation, and Structure. Data are the measurements and observations. Results in a scientific information architecture that is stable via technology independence.
- Data Integration:
  - data class (numbers, sets of numbers, text, images, video, sound)
  - discipline (meteorological/oceanographic/and beyond)
  - metadata .vs. data (integrated information model)
  - structured data .vs. unstructured data (documentation evolution)
  - data aggregates - point data, profile data, field data, gridded data, image data
  - data representations, structures, syntax and semantic
  - data sources (in-situ and remotely sensed; measured and derived observations; models)
- Function Integration:
  - Data acquisition plus dynamic quality control implies via load path.
  - Data delivery implies network configuration and monitoring.
  - Data management implies multiple layers of access (plus batch loading/unloading plus tables, tuples, attributes, relationships - SQL plus object based interface).
  - Data presentation implies standard products and visualization.

Table 4.1: Elements of REINAS Data Management Approach

4.3.3 REINAS Data Management Approach

4.4 Requirements Analysis and Information Architecture

As part of this phase of requirements analysis, the potential users of a REINAS system were identified. In the prototype implementation we may not be able to work directly with users from all of the categories so defined, but we will attempt to provide functionality relevant to each of the categories.

The initial user community served by REINAS will consist of the REINAS science partners located at NPS and MBARI. The developers themselves will serve as the target population for the engineering aspects of the system, e.g., instrumentation engineers and system developers.

The user profiles defined are listed in the following sections.

4.4.1 Operational Users:

- *Operational Forecaster*: Needs current situation visualization, nowcasting, and short-range forecasting. Traditional meteorology map product displays are required. An easy to use suite of "canned" products must be available. Easy, fast access to previous environmental situations with a similar signature would provide a new, significant capability.

- *Operational Policy Maker/Planner*: Needs integrated "birds eye" views of complete geographical area and the ability to focus (zoom in) on smaller, specified areas of interest. The ability to set up scenarios (models) and view results is desirable. A user-friendly interface similar to that of operational forecaster is required.
- *Disaster Control*: Needs immediate view in the form of current observations, nowcasts, and local climatologies. Requirements similar to operational planner, but requires additional map data and products, i.e., environmental Sensitivity Index maps. Also will require the capability to plug-in models specific to the hazardous response activity.
- *Students and Casual "browsers"*: Need canned products and rapid visualization, i.e., access to precomputed visualization results. An integrated visualization and "dry-lab" modeling capability makes REINAS a significant educational asset. It should likewise provide a base for educational research projects.

4.4.2 Scientific Users:

- *Retrospective Researcher*: Needs synoptic views and historical analysis. Requires data quality information, ability to readily construct complex database queries, and ability to write C programs which access REINAS data and functionality through well documented APIs. Should also be able to make SQL queries. These custom programs are to become part of the individual researchers REINAS environment.
- *Experimental Researcher*: Needs include those of both operational forecasters and retrospective researchers, as well as ability to monitor instrument status, control instrument settings, and view "data streams" in real-time.
- *Sensor Scientist*: Needs detailed control of instrument and detailed view of instrument status. Requires simple mechanisms for interfacing and low-level real-time control.

4.4.3 Developers/Intrumentation Engineers:

- *Instrumentation Engineer*: Needs a "cookbook" approach for adding new instruments to REINAS. Support tools should be such that REINAS is the environment of choice for a new instrument development project. REINAS should provide interfaces to instruments, e.g., for calibration or other parameter changes.
- *Network Engineer*: Requires tools to support an enterprise scale/style network using Internet.
- *System Developer*: Requires well documented APIs providing access to all REINAS functionality, ability to readily integrate and test new REINAS applications, ability to run a REINAS node in "development" mode, and debugging and timing tools.
- *Database Administrator*: Need traditional capabilities, e.g., ability to tune database, ability to visualize its utilization and access patterns, and ability to define database organization at both physical and logical levels. Automating these functions is a research topic.

- *System Manager/operator*: Need traditional capabilities, e.g., ability to control user's access/capabilities, ability to view system status and utilization, and ability to tune system on-the-fly.

4.4.4 Special Applications:

Some users of REINAS services will be automated procedures which require timely and regular access to data and sensors. We can identify two primary categories.

1. Data Mining Applications: These applications can run continuously in the background, to detect and classify significant patterns that may exist in the observation data across multiple scales in space and time. The system must be able to gracefully distribute the loads placed on it by these applications to times of minimal impact on primary operations. In addition, the system should support the capability to easily extend the number and type of these applications.
2. Standard and Customized Product Generation: Applications which produce scientific or engineering information products. The products may be produced on a regular schedule, based on the occurrence of an event, or based on input data availability. The system must able to maintain a description of and requirements for such products and produce them accordingly. Standard products are those needed by a large cross section of REINAS users or by high priority users. In addition, the system should provide a capability for users to build customized product profiles for automatic production by the system.

4.5 System Data Flow Architecture

4.5.1 Writing Device Managers

The REINAS [3] system requires that a broad class of instruments 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 device manager 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.

A REINAS device manager is a process that provides the REINAS abstraction of the hardware device. This abstraction supports standard control and data functions. Device managers are not written from scratch, rather they follow a framework which provides a very specific proscribed template. Device managers are not REINAS end-user applications, specifically they do not use the REINAS User API. Rather, they are software components which are indirectly part of the implementation of this API.

Each unique device type within REINAS has its own device manager that has been custom built for the specific device. A device manager consists of two sections: a generic section and a custom section. The generic section handles the standard interface to the

---

[3] This section by Bruce Montague

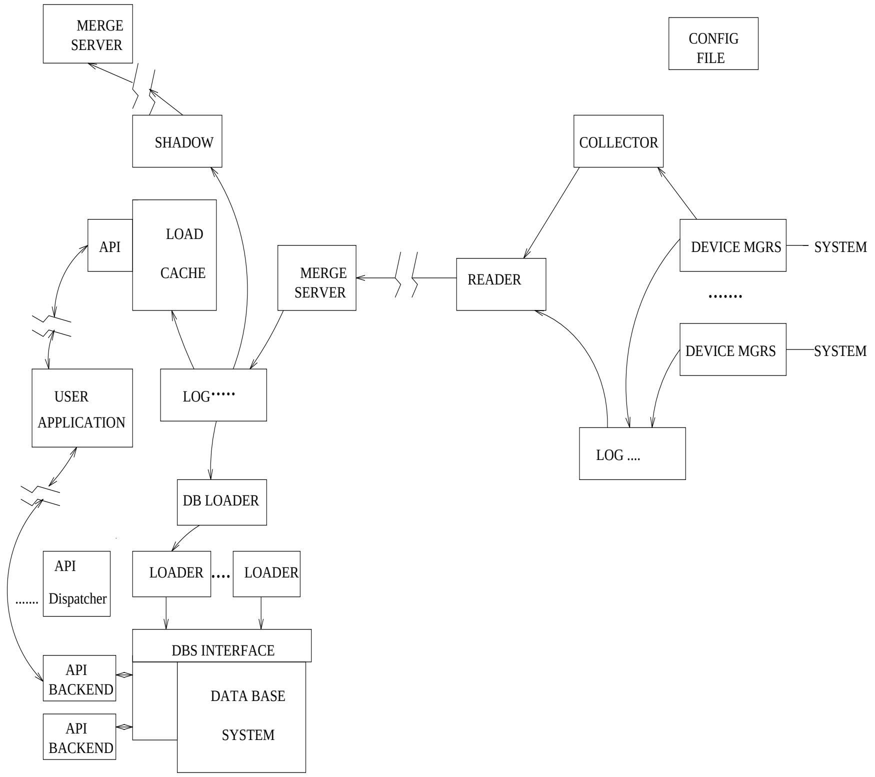

Figure 4.9: REINAS Data Flow, Showing the Role of Device Managers and the Application Program Interface

REINAS system and provides a framework into which the custom routines written by an instrumentation engineer are placed. Instrumentation engineers do not need to be aware of the generic device manager's internals.

The custom section of a device manager consists of a set of standard functions required of all device managers. These seven functions define the *REINAS instrument interface.* These routines are written by the instrumentation engineer. The arguments, return values, and responsibilities of each of these functions is standard and clearly specified.

To actually manage hardware and convert data to a standard REINAS format, instrumentation engineers use a *REINAS hardware support library.* An example of the types of routines in this library would be routines that are useful in reading data from a serial port. This library will grow as additional devices are added to REINAS.

The required instrumentation interface routines are: *Inst_open(), Inst_close(),* Access and deaccess a scientific instrument.

Inst_get_data(), Obtain instrument data.

Inst_get_attributes, Inst set_attributes. Control dynamic attributes of the instrument, for instance, sampling rate.

Inst_Suspend, Inst_Resume. Suspend and reactivate the device manager.

Functions use a standard support library for interaction with the rest of the system.

Device Managers have been written for: NOAA Wind Profiler, CODAR, Campbell Met station, and a Virtual device mod for testing. Their sizes range from 10K Lines of Code for the CODAR to 2K for the Campbell Met station.

DATABASES USED DURING THE DEVELOPMENT OF REINAS

Postgres - customizable, code available
Miro/Montage/Illustra - easy operations, rich names
Ingres - fast, integrity checks
Oracle - potentially very fast, feature rich

BUT –

Postgres - Slow, unreliable.
Miro/Montage/Illustra - Few utilities, mediocre performance.
Ingres - 24 character name restrictions, most restrictions of the systems used.
Oracle - requires close operational attention.

PORTABILITY

The system code in REINAS must be written as portable code. One motivation for this is that the following must be supported:

BSDI - Instrument PCs
HP - MBARI machines and MBARI ship bourne node.
SUN - Miro, development environment.
SGI (4 and 5) - Visualization and Oracle.
IBM - Ingres.
DEC - UCSC development environment.

Portability is nontrivial
BUT... IEEE Floating Point works.

Table 4.2: Databases and Portability

The programs to support Data Flow are referred to as “Plumbing”. All the plumbing code to support the PC, DB, API, and tests/applications comes to about 50K lines of code.

Such programs are sometimes called “Middleware”. They provide for a common software environment that spans operating systems, networks and databases. In the case of REINAS they provide for common normalization and eliminate files. Some middleware features and observations in REINAS are:

- Common application API provided on different Databases, Networks, and OSes.
- Middleware is system software. Developing it in the midst of operations is difficult. An NxN problem exists between file formats and application programs.
- Database/architecture independent SQL middleware layer is a useful goal.
- “Database drivers” must be written that “glue” diverse databases into the system.
- SQL as intermediate query language. The mapping between OO and relations is not necessarily natural.
- High level user interface tools such as TCL work, but are no panacea.
- Tools - A database system without good utilities is like an OS without an editor.

4.5.2 Device Managers

Device Managers are potentially stand-alone. An example would be a standard TCP datalogger. External networks such as Unidata, rain gauge network are a problem. It is tempting to treat them as one complex device, but maintaining accurate metadata requires treating them as communication media. Even with a framework it will take a few weeks to write a device manager (Similar in terms of time to the requirements for writing a device driver). Device managers require up-front time. All data fields must be considered. This time should be considered the first step in data analysis (normalization). One should not have to reverse engineer data formats. Since one never knows about all missed states. Time formats are also numerous and require normalization.

4.5.3 Database Systems

The time series storage model is natural. It has similarities to multi-media data streams. A scientific DBS may require explicit support for a file Explode/Materialize concept. The current data stream model handled all input, but we may need to tune it. Our long term goal is to support end-to-end Real Time by eliminating on-the-fly parsing and loading the database directly in binary.

There is a need to merge device managers and real device driver/data acquisition software. Another problem is data quality testing - how do you know data has become polluted?

An observation: SQL isn't standard. Date formats differ, for instance some support a time of “now”, others do not.

We still need lots of procedural code - SQL is insufficient by itself.

4.6 Data Compression in REINAS

Data compression [4] is just one resource managed within the REINAS system to achieve higher efficiency. It is most useful when it is under the covers of the system so that the average user does not have to be concerned with its details. It fits between all the major components of REINAS – instruments, computer network, data management, visualization, and simulation.

Data compression is being investigated for use in REINAS to solve several important problems:

1. Data archiving requires keeping data arriving from a large number of sensors, users, and imported processes.
2. Network bandwidths available to a rapidly increasing number of users will be limited.
3. Exporting of REINAS products requires careful resource usage so that the widest number of users will be able to effectively use the data and products generated by REINAS.

Compression helps to fulfill these three goals as efficiently as possible. Because compression is to be used for a variety of reasons there are multiple compression techniques that have been and are being investigated for REINAS.

---

[4] This section by Glen Langdon

4.6.1 Compression for Data Archives

Compression has been investigated to help reduce the amount of storage necessary to archive data. Recent data (100 Mbytes to 200 Mbytes) can be archived without compression. Data older than two weeks may be compressed to secondary storage, and the primary storage reused for incoming data.

Once the data are acquired, averaged, and rounded to the precision specified by the meteorologists, lossless waveforms compression is applied. This technique, developed for gray scale coding of satellite images under a NASA Summer Faculty Fellowship [LAML94], can be a common part to any compression technique. Predictive coding produces the prediction error, and a quantization of the previous error is the context. The binary arithmetic coding for encoding context-dependent prediction errors, described in [LM93], is intended for the compression of satellite images.

4.6.2 Data Compression for Scientific Visualization

REINAS data will be used by many simultaneous users. Because REINAS is using the Internet, and shared NPS, UCSC networks, the number of users will be limited by the bandwidth. Sufficient communication bandwidth is needed. Visualization users get their data over a communication link, and compression at the source offers bandwidth savings. Moreover, if the screen is broadcast to several collaborating users, the bandwidth savings are multiplied. Each user's workstation will have a "decompress and display" capability.

If the viewing station has the power of a graphics rendering engine, then a third opportunity for possibly even greater compression exists. The visualization objects are compressed and transmitted, and the viewing station renders them locally. Rendering power allows user freedom to change the viewpoint independent of other users, and subsequent visualization objects can be incrementally added to the list of objects to render. The amount of data transmitted over the collaborative session is greatly reduced. For example, a user is sent the higher-level graphics language commands and data to render If the next screen is a slightly rotated view of the same data the VQ or JPEG compression schemes require the compression and decompression of a completely new image. Using the high level commands all that is transmitted is the new rotated view point.

5. Communication Protocols for Wireless Networks

5.1 Mixed Media

Communication[1] among REINAS components and users is being accomplished via a mixed-media networking infrastructure that encompasses new and existing telephone lines, Internet connections, radio links, as well as the networking software and hardware needed to control and manage the interconnection of REINAS sites.

Today's REINAS network consists of existing Internet connectivity with a few additional telephone and point-to-point radio links. However, in the future, REINAS needs to include several mobile sites (e.g., boats and trailers with MET stations) and applications that require the transport of large amounts of information, specially for remote visualizations, where one minute animation can require in the order of 160 Mbps. The information exchanged in REINAS will include multiple media (text, voice, images, graphics and animation, and even video), and such information has to be distributed in real time (e.g., during a multimedia conference among multiple sites) over different types of transmission media, including radio links and high-speed lines. Furthermore, the networking infrastructure of REINAS should allow a very large number of sensors to be incorporated into the system. Accordingly, we see six major networking requirements in REINAS:

1. The ability to transport multimedia data in real time.
2. Scalability to a large number of geographically-dispersed sensors.
3. Mobility of sites.
4. Fault tolerance.
5. Efficient use of multiple transmission media.
6. Connectivity to the Internet.

The marked differences between REINAS networking needs and traditional networking technology indicated the need for new multimedia networking solutions to REINAS's unique characteristics and to allow REINAS system engineers to better manage and monitor communication resources in support of data management, data visualization, and user communication.

Mixed-Media Networking for REINAS - Research Problems:

- Quick recovery after failures
- Scaling to hundreds or thousands of nodes (sensors, workstations, routers)
- Supporting multimedia traffic over mixed transmission media
- Supporting mobility of nodes (boats, aircraft, terrestrial vehicles, hand-held devices)
- Conserving power and bandwidth (mobile and static nodes)
- Supporting multipoint communication with reliable and unreliable data delivery
- Manage mixed transmission media intelligently

---

[1] This chapter by J.J. Garcia-Luna

5.2 Channel Access in REINAS

In extending wireless LANs to mobile MET stations and portable devices in the REINAS network architecture, we relax the distinction between links, networks, and internets used to provide connectivity, so that the various transmission systems are treated in a uniform manner, leading to a fault-tolerant multimedia system. The provision for type-of-service routing and load-sharing via multiple types of transmission media is a key component of our design. A communication security architecture could be integrated into such a communication system. This design is based on earlier work by Mathis and Garcia-Luna on survivable multimedia networks [MGLA87].

The challenges to Channel Access in REINAS are:

- Using distributed control
- Supporting multimedia information
- Conserving power in portable devices or remote sensors
- Dealing with hidden- terminal problems

The limitations of Existing Approaches are: Current schemes are contention-based and contention-free protocols. Most efficient contention-based protocol (CSMA/CD) cannot be applied to packet radio. CSMA is approach commonly used in multihop packet radio nets but suffers from the hidden-terminal problem. Contention-based protocols require a node to contend for channel for every packet. Contention-free protocols use either a separate control channel, centralized control by a base station, slotting, or a fixed number of reservation slots.

5.2.1 Floor Acquisition Multiple Access (FAMA)

FAMA[2] is a new method developed for REINAS based on the AppleTalk Link Access Protocol.

In FAMA a packet radio acquires the "floor" (right to use channel) before sending data packets. The Floor is acquired dynamically by sending a request-to-send (RTS) packet to intended destination or multicast group [DGLA95]. The Destination or group representative replies with a clear-to-send (CTS) packet. The Packet radio uses carrier sensing to avoid transmission if channel is sensed busy.

FAMA ensures that no data packet collides with RTS or CTS packets, and that each packet radio eventually "gets the floor". FAMA supports packet bursts and performs much better than CSMA.

At present basic FAMA is implemented on the BSDi REINAS system. Implementation is complete and currently being tested and verified. Deployment of basic FAMA to REINAS field network is to begin in the fall 1994.

Some FAMA Open Issues remaining are:

- Accommodating priorities
- Building distributed queues
- Adding collision detection
- Eliminating hidden-terminal problems

---

[2] This section by Chane Fullmer

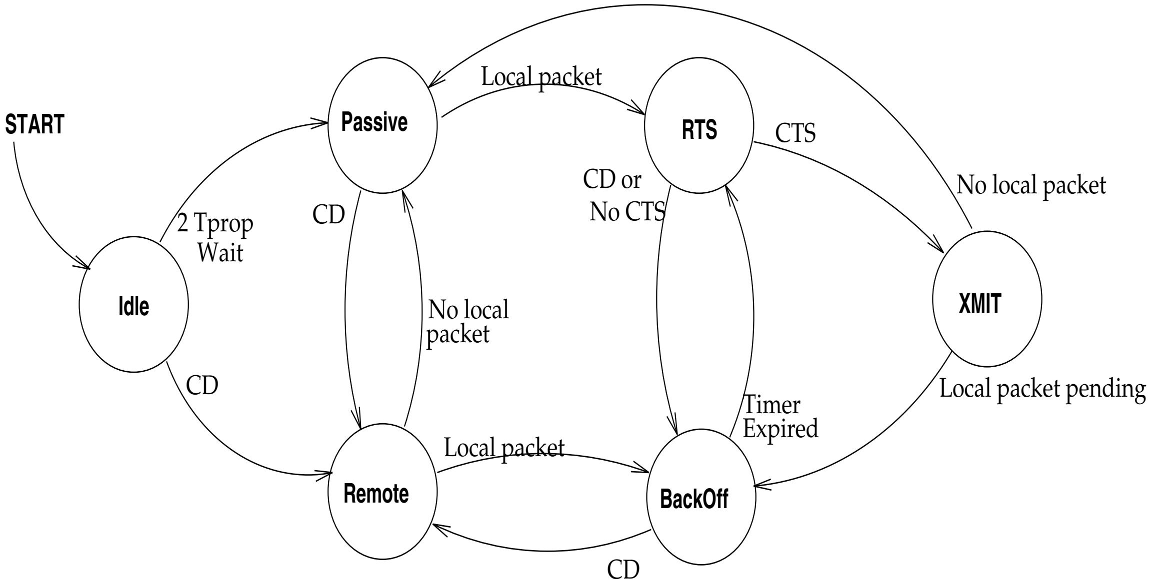

Figure 5.1: State Diagram for FAMA - Floor Acquisition Multiple Access Protocol

- Supporting multicasting
- Using transmission agility to handle low-power nodes

5.3 Routing Protocols for REINAS

5.3.1 Internet Routing

A critical element[3] in the provision of fault tolerance and the ability of a network to scale is the choice of the routing protocol used. For the purposes of routing protocols, an internet can be viewed as consisting of a collection of interconnected domains, where each domain is a collection of such resources as networks, routers, and hosts, under the control of a single administration. Current work in interdomain routing has proceeded in two main directions: protocols based on distance-vector algorithms (DVA), which we call distance-vector protocols, characterized by BGP [LR91] and IDRP [ISO91], and protocols based on link-state algorithms (LSA), which we call link-state protocols, characterized by the inter-domain policy routing (IDPR) architecture [Ste92]. The same two basic approaches have been used in the Internet for intradomain routing (e.g., RIP [Hed88] and Cisco's IGRP [Bos92] are based on distance vectors, and ISO IS-IS [ISO89] and OSPF [Col89] are based on link states). We view REINAS as a single domain, and focus on intra-domain routing protocols.

The key advantage of DVAs, and the distance-vector protocols that use them, is that they scale well for a given combination of services. Because route computation is done distributedly, DVAs are ideal to support the aggregation of destinations to reduce communication, processing, or storage overhead [GLA88].

---

[3] This section by J.J. Garcia-Luna

5.3.2 Updating Routing Tables

Today's routing protocols are: (1) Distance vector: Its limitation is the cost of the preferred path through each neighbor to all destinations. (2) Link state: Information about the entire network topology is flooded.

Many distance-vector protocols are based on distributed Bellman Ford algorithm (DBF). The drawbacks of DBF are bouncing effect and counting to infinity. Flooding the network topology or organizing the network into areas are not feasible approaches in mobile networks. Sending complete path information in updates is wasteful in a mobile network. Approaches adopted in RIP and its derivatives, OSPF, IS-IS, BGP/IDRP, or even EIGRP are not applicable to REINAS

5.3.3 New Routing Algorithms for REINAS

Two new algorithms have been designed, verified, and analyzed for REINAS. All previous routing protocols based on two main algorithm types:

1. Distance-vector algorithms: They exchange path information. Their overhead grows with number of combinations for multiple service types / policies. Protocols based on Distributed Bellman Ford algorithm behave badly with resource failures.
2. Link-state algorithms: They exchange link characteristics. They use flooding of complete topology information. The aggregation of topology information is difficult.

The new algorithms are Path-Finding Algorithms and Link-Vector Algorithms:

1. Path-Finding Algorithms (LPA, PFA): Eliminate counting to infinity problem and temporary loops using predecessor information, shortest-distance information, and a single-hop internodal synchronization mechanism.
2. Link-Vector Algorithms (LVA): Eliminate counting to infinity by distributing topology information selectively based on distributed computation of preferred paths

5.3.4 Path-Finding Algorithms

Each node maintains three tables:

1. Distance Table: Predecessor and distance for each destination reported by each neighbor.
2. Routing Table: Shortest distance, successor, and predecessor for each destination.
3. Link-Cost Table: Cost information about a node's adjacent links.

A node can be in two states: Passive - in which it has a feasible successor, and Active - in which it is waiting for a feasible successor. There are three message types: Update, Query, or Reply. Predecessor information is used to infer whether a loop exists. A query reports an infinite distance to block potential loop. Node receiving a query must reply to it immediately.

5.3.5 Link-Vector Algorithm

The Link-Vector Algorithm has partial topology at each node: links adjacent to node and links reported by neighbors. Uses local path selection algorithm to select preferred paths; all nodes use identical path selection algorithm. Distributes information only about links used in preferred paths. It asks neighbors to erase information about links no longer used in preferred paths. The Node can determine whether an update message is valid (e.g., by sequence numbering). The Link level protocol detects changes in link status. Reliable transmission of messages is guaranteed.

5.3.6 Results to Date on Routing

We have shown that both LVA and LPA/PFA can be applied to REINAS. Both algorithms are very efficient. LPA and LVA shown to perform better than ideal topology broadcast algorithm, DBF, and previous loop-free distance-vector algorithms. LPA/PFA is the fastest distance-vector algorithm designed to date. LVA is the first algorithm based on link-state information applicable to radio networks. Temporary loops are not expected to be a major problem in LPA/PFA or LVA.

Other important results of the REINAS project have been student support and publications. Together with AASERT, four Ph.D. students are being supported. There have been two submitted journal articles, four published conference papers, and five submitted conference papers.

5.3.7 Future Work on Routing

We are currently implementing PFA in BSDi using RIP Version 2 packet formats. Work is underway on extensions to handle broadcast links and mixed media efficiently. We plan to apply previous results on loop-free routing to packet forwarding over multiple paths (shortest multipath routing). We hope to extend LVA and LPA/PFA to account for aggregation of information and Integrate hop by hop congestion control with routing. We expect to Develop distributed algorithms for multicast trees and graphs and to integrate GPS with routing information.

5.4 Protocols for Floor Control

Networked Multimedia Applications [4] may be accomplished by Protocols for multimedia, multipoint dialogues across the REINAS internet. They provides Floor control for networked multimedia applications and reliable and unreliable multicast.

Floor control can provide concurrency control for distributed shared multimedia resources along with session control. Most collaborative software provides little or no floor control. Difference to concurrency control: user-resource-user paradigm with multiple resources, but in general no rollbacks.

Criteria for a Floor Control Protocol are:

1. Scalability: adjusting to varying number of users.
2. Performance: keeping control information traffic low, serving requests ASAP.

---

[4] This section by Hans-Peter Dommel

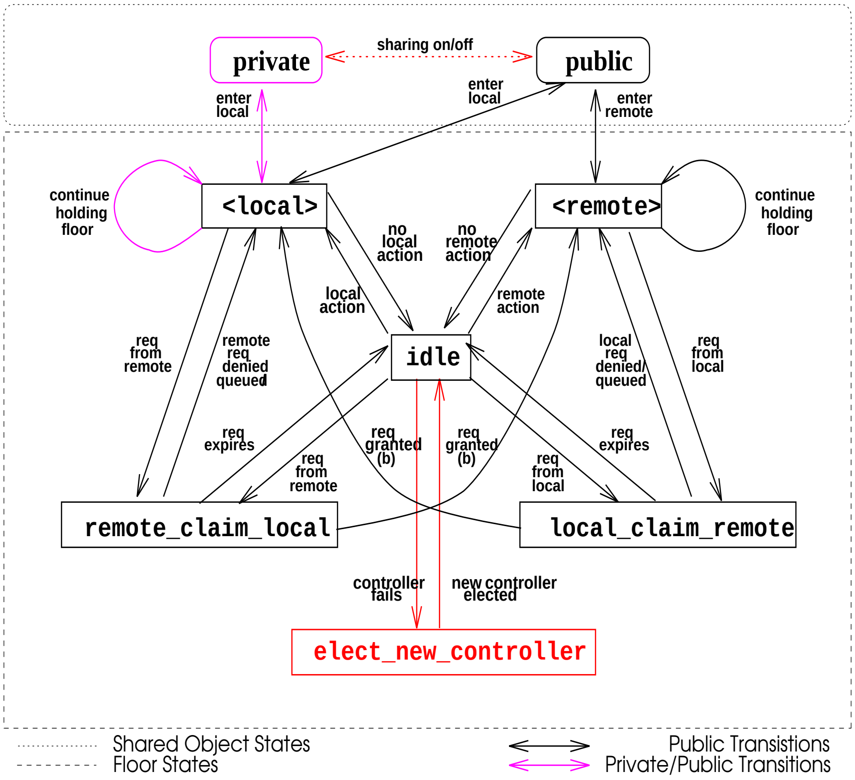

Figure 5.2: State Diagram for FACE, an Algorithm for Floor Assignment in Collaborative Environments.

3. Resilience: graceful degradation via distributed and replicated floor control information storage
4. Correctness: deadlock freedom, fairness (preemption, no starvations).
5. Flexibility: service different media with different policies (text, voice, video...).

A program has been developed to experiment with the protocols. It is called "Floor Assignment in Collaborative Environments" (FACE). Its specifications are: It works distributedly, scalable, and generic (adapts to different resources). It is hierarchical with respect to collaborators (groups/subgroups) and resources (granularity). It has primary states (private vs. public), secondary states, recovery state for fault-tolerance. It has contention-avoidance and no predefined token-scheduling. It provides resource-reservation via first-come-first-served, recovery via election algorithm. Multiple floors allowed for certain media (voice, video). There are feedback floors for temporary backchannels (See Figure 5.2).

Work is underway to compare FACE with other floor control algorithms by simulation, to implement FACE in support of collaborative visualization, and to support other REINAS collaborative applications.

6. REINAS Observations and Models

6.1 Observational Network from a Meteorological Perspective

The primary aim of the REINAS observing network[1] is to provide high temporal and spatial resolution measurements from "research quality" surface and upper- air stations with special emphasis on "timely" communication of data into REINAS.

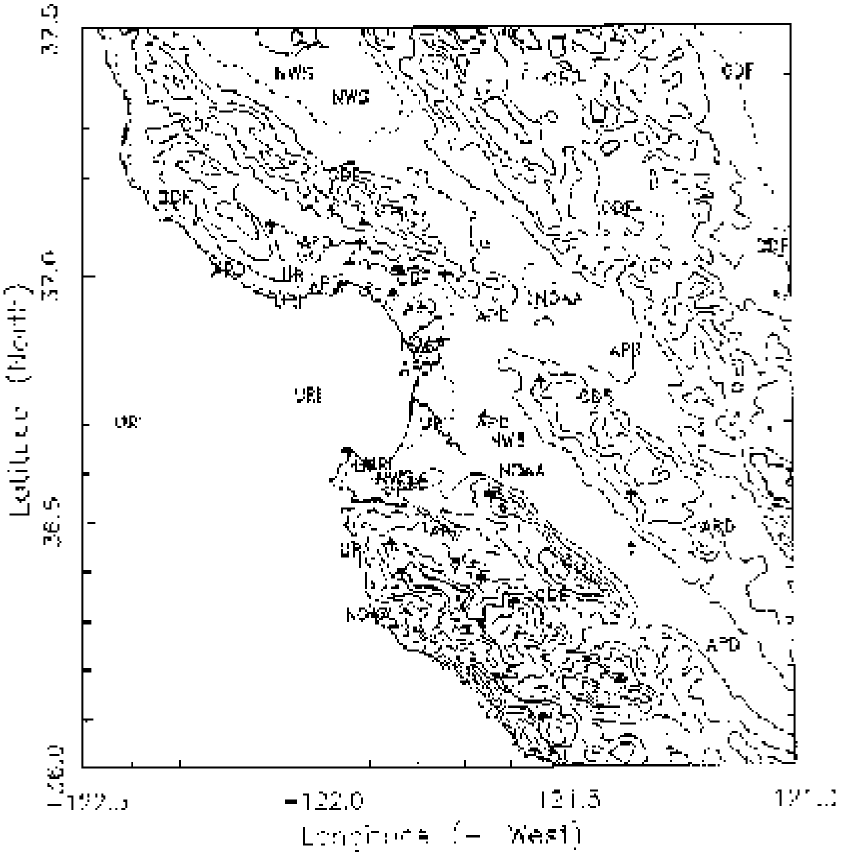

Figure 6.1: Observation Sites in the Vicinity of Monterey Bay. URI = UCSC and MBARI Sites. NWS = National Weather Service. CDF = California Dept. of Forestry. NOAA = Environmental Technology Lab. APD = Air Pollution Control Districts.

- Desired measured surface meteorological fields include:
    - Airemperature
    - Moisture content
    - Barometric Pressure
    - Winds
    - Solar and Infrared Radiation

---

[1] Observation section by Richard Lind, NPS

  - Precipitation
- All sensors are “off the shelf” models
- Each site utilizes a datalogger for interrogation of sensors
- Surface station siting criteria are based on EPA and WMO siting quidelines.
- Quality control is maintained by continuous comparison with meteorological fields. Auditing of each site carried out at six-month intervals.

The approach has been to utilize existing surface station data as much as possible, supplemented by the special REINAS sites. To date, the following agencies have been identified as willing to cooperate with the REINAS concept:

- Monterey Bay Unified Air Pollution Control District
- Monterey County Water Resources Board
- Bay Area Air Pollution Control Districts

Standard meteorological data feeds also include hourly and supplemental special observations from:

- National Weather Service
- Federal Aviation Administration
- California Department of Forestry

Over the next few months, work to incorporate these data sources into the REINAS database will be completed. The desired measured upper-level meteorological fields include:

- Temperature
- Moisture content
- Winds

Twice-daily rawinsondes (Oakland, Vandenberg AFB) and hourly data from NPS 404 MHz and 915 MHz Doppler radar wind profilers are the only routine upper- air measurement sources in region. To fulfill the desire for higher temporal and spatial upper-air measurements, NPS has contracted with NOAA Environmental Technology Laboratory for deployment of three 915 MHz radar wind profilers and two 2-axis monostatic SODARs (May-Sep 1994).

In addition to these upper-air sites, surface stations have also been deployed at each of these locations by NOAA. Strong cooperative development between REINAS and NOAA has led to real-time data links to REINAS for some of these sites. NOAA plans to leave wind profilers at Long Marine Laboratory and Point Sur Naval Station at least over the winter season.

6.2 REINAS Atmospheric Numerical Modeling System

6.2.1 Properties of an Ideal Model

REINAS[2] is a system that will enable real-time and retrospective regional-scale environmental science, including nowcasting of current conditions and forecasting. A critical component of REINAS is a sophisticated numerical model that provides the dynamic backbone for:

---

[2] Atmospheric Modeling section by Paul Hirschberg, NPS

1. Objective quality control of special data
2. Near real-time data assimilation of all observations
3. Nowcasting
4. Short-term forecasting
5. The generation of dynamically-consistent, temporally- coherent products from which high quality visualization of the current and future evolution of the flow can be made.
6. More in-depth scientific investigation g. sensitivity studies) of various physical phenomena affecting the REINAS region of interest.

Ideally the modeling system will be used in two modes:

1. Analysis (4DDA) mode: Intermittent (update cycle) or continuous (data-nudging) data assimilation. In this mode the model is influenced by incoming conventional and special data. The model is run for short periods of time optimally designed to not lag or outrun real time to any great extent.

   Products obtained in this mode could be used for quality control for incoming data. Problem observations could be either corrected or discarded, and assimilated fields used for visualization and animation.
2. Forecast mode: longer duration forecast initialized with current 4DDA analysis. The model is used to make predictions of the flow at a later time. It can aid in the decision of whether to initiate any special data collection activities.

An optimal model choice would have the following properties. It would

1. be able to simulate diverse meso-scale phenomena in any geographic region. In the present case this means land and sea breeze circulations associated with complex topography.
2. contain sophisticated physics packages.
3. contain multiple nesting capability.
4. be nonhydrostatic
5. contain coupled atmosphere / ocean components
6. be easily networked or run locally.

For example: The Navy's Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS). But it is not yet available.

6.2.2 Current (Prototype) Configuration

We have installed and are running a version of the Navy Operational Regional Atmospheric Prediction System (NORAPS) on the NPS CRAY Y-MP EL98. It has the following properties:

- Short and long-wave radiation (Harshvardhan et al. 1987)
- Cumulus heating and moistening (modified Kuo 1974)
- Vertical turbulence parameterized using a 1.5 order closure.
- Terrain from the U.S. Navy 10 minute data base
- Single nest
- 45 km horizontal resolution (109 X 82) centered over Monterey Bay

- 24 sigma levels (surface to 10 mb; six levels below 900 mb)
- 135 second time step

The model is initialized with 0000 UTC NOGAPS 2.5 deg by 2.5 deg analyses on 16 mandatory pressure levels. 12-hourly horizontal boundary conditions are obtained from 0000 UTC NOGAPS forecast run. A 24-h forecast is generated once daily. 6-hourly model output is brought back to SGls for display and also made available for transfer to UCSC.

Using the model we have observed some interesting mesoscale features associated with the California coastal zone in the model simulations. For example: the June 9-11 Catalina eddy, southerly surge case that occurred during Monterey Area Ship Tracks (MAST) experiment.

6.2.3 The Time Line for an Automated NPS NORAPS Run

- 0200 - 1000 UTC: Full suite of NOGAPS analysis and forecast grids transmitted by Fleet Numerical to the NPS Met. Dept. SGls
- 1000 - 1 100 UTC: Multiquadric analysis [NT94] is performed to interpolate 2.5 deg by 2.5 deg NOGAPS analyses and forecasts to NORAPS 45-km grid centered over Monterey Bay.
- 1 100 UTC: NORAPS analyses and boundary conditions are transferred to the NPS CRAY.
- 1100- 1230 UTC: 24-h NORAPS simulation is performed.
- 1230 UTC: 6-hourly output is transferred back to Met. Dept for display and also to a TAR file that UCSC can grab.
- 1230 - 1400 UTC: Selection of standard products output at NPS.
- Cycle complete by 1400 UTC.

6.2.4 Modeling Plans for Near Future (FY 95)

- Obtain and install new nested version of NORAPS
- Operationally run 3 nests: 45 km, 15 km and 5 km depending on CRAY limitations.
- Obtain 45-km operational NORAPS analyses and forecasts from Fleet to serve as initial and 6-hourly boundary conditions.
- Begin blending special REINAS observations into analysis with a regional O.l. or multiquadric analysis especially over the inner domain.
- More thorough verification.
- Begin development and testing of 4DDA system.

For the more distant future (FY 96) we plan to:

- Adapt COAMPS (sigma-z) when it becomes available.
- Nonhydrostatic, go down to 1 km or less horizontal resolution.
- Implement 4DDA system.
- Run in coupled mode with ocean model.

6.3 Ocean Modeling

As part of REINAS[3] we have a need to find a method for practical interpolation and state-estimation in oceanography to make the real-time measurements useful to users. A method that provides "added value" is also desirable. Linear interpolation between sparse measurements provides smoothing, but this may not conserve physical properties such as mass and momentum.

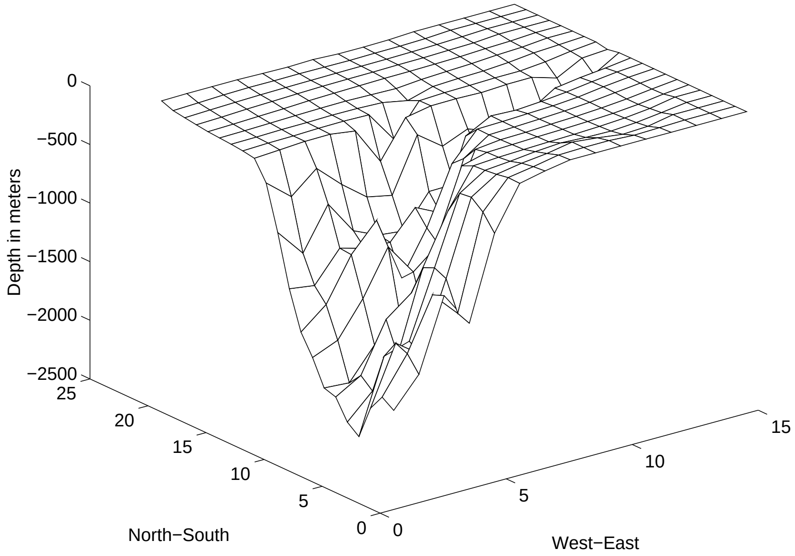

Figure 6.2: Monterey Bay Ocean Model Grid (2km x 2km Spacing)

Another approach is to use some version of stream function and vorticity fields to interpolate between measurements. This has a long history in fluid dynamics, particularly in meteorology. After some experimentation with this we went to a more complete physical approach of using an numerical ocean model.

The model selected was the public domain Princeton Ocean Model, usually called the "Mellor Model". It was originally developed by George L. Mellor and Alan F. Blumberg [BM87] and has since been developed and applied to many oceanographic problems within the Atmospheric and Oceanic Sciences Program of Princeton University, NOAA's Geophysical Fluid Dynamics Laboratory, and Dynalysis of Princeton [EKM92].

Princeton Ocean Model has the advantages of being small enough to run on a workstation, has good physical approximations, is easy to modify for boundary conditions, and is freely available. We obtained a copy of the code and documentation from NOAA/GFDL by file transfer (ftp gfdl.gov). Since then we have experimented with it in conjunction with the IBM Data Explorer and have inserted the Monterey Bay bathymetry as boundary conditions.

We have used a 2 km x 2 km grid over the Bay region only. See Figure 6.2. The Monterey Bay canyon causes difficulties in the model because of the very steep canyon walls.

The principal attributes of the model are:

---

[3] Ocean Modeling section by Harwood Kolsky, UCSC

1. It contains an imbedded sub-model to provide vertical mixing coefficients. This produces realistic bottom boundary layers.
2. It uses sigma coordinates in which the vertical coordinate is scaled on the water column depth.
3. The horizontal finite difference scheme is staggered, usually called the "Arakawa C" differencing scheme. The model uses curvilinear orthogonal coordinates, which can be a rectilinear or a spherical coordinate system as special cases.
4. The horizontal time differencing is explicit whereas the vertical differencing is implicit. The latter eliminates time constraints for the vertical coordinate and permits the use of fine vertical resolution in the surface and bottom boundary layers.
5. The model uses a free surface and a split time step. The external mode (i.e. surface) is two-dimensional and uses a short time step based on Courant, Friedrichs, Lewy stability conditions and the external wave speed. The internal mode is three-dimensional and uses a longer time step.
6. Complete thermodynamics have been implemented.

6.4 Data Assimilation Plans for Atmospheric Modeling in REINAS

Data assimilation [4] is required to make full use of the diverse and asynoptic nature of the REINAS observations. Simple interpolation schemes has been installed in REINAS to generate fields for visualization. However, simple interpolation does not account for the dynamic balances inherent in the atmosphere.

6.4.1 Requirements for REINAS Data Assimilation

- Requires system that can produce fields with proper scales represented in the various nests of the model domain.
  - 1-5 km scales for the dense observation network over Monterey Bay.
  - 30-100km scales for the West Coast domain of the outer model nest where only routine observations are available.
- System must be able to blend observations with model first guess in a dynamically balanced manner
- Scale differences require proper "weighting" functions for dense versus sparse observation regions.
- System must be efficient to run in a short period of time to produce analyses in near real-time.

6.4.2 Proposed System

- Multiquadric interpolation as a basis for the data assimilation.
  - Multiquadric (MQ) is similar mathematically to Optimum Interpolation but uses hyperboloid basis functions.
  - Temporal interpolation can be included if needed as well.
- MQ interpolation has advantages over other data assimilation methods.

---

[4] This section by Wendell Nuss, NPS

  - MQ can easily account for data density variations which are important in the coastal region.
  - OI functions used in large scale modeling are not appropriate for small mesoscale problems, consequently they must be derived for new mesoscale situations each time.
  - MQ uses a set basis function that performs superior to the OI basis functions in side by side tests.
  - Other methods such as adjoint or Kalman filtering techniques are too computationally expensive.
- Blend model first guess with observations using model dynamic equations as a constraint on the MQ matrix solution.

6.4.3 Status and Plans

- MQ code is running routinely at NPS to produce surface analyses for West Coast.
  - Present operational version does not use model first guess.
  - Use of model first guess has been solved and extensively tested on several cases.
- Need to extend to 3- and 4- dimensions.
  - Proof of concept tests have been done and published by Nuss (1994).
  - Construction of working system part of work by NRC post-doc, Supachai (Pom) Sirayanone.
- Need to add dynamic constraints and make system operationally viable for real-time applications.

7. REINAS Long-Term Goals

7.1 Goals of The REINAS System Research and Development

7.1.1 Initial Goals

The goals of the first two years of the project have been to analyze, develop, and test the elements of a prototype system. REINAS has demonstrated the feasibility of real time data collection from sources and the dissemination of regional data to support environmental visualization. The goals for REINAS have been refined because of the experiences, and changes in demands on system. We quickly summarize the goals from the prior progress report, and discuss the directions for our near-future and long-term research and development goals. REINAS is designed for regional scale, as opposed to global scale, environmental science. The three modes of data access are real-time, retrospective, and nowcasting access. In the immediate term, retrospective access is essential for our collaborating environmental (oceanographic and meteorological) scientists, while the real-time and nowcasting are active research goals. An example of the real-time use is the automatic alert when a given phenomenon or condition arises in the data.

REINAS is also designed as a physically distributed system, in terms of the instruments, database, and users. The instrument network has also proven to be very challenging in implementation and integration. Many new instruments have been made available, but are difficult to incorporate into a prototype system. Important work has been done in the development of an instrument network using radio modems, and routing data with Internet protocols. Several radio links and land line links have been operational for the majority of 1994. Also, longer term research has been completed on protocols and schemes for dynamic wireless networks. Improved network infrastructure in dynamic, intermittent, and wireless networks remains a high priority goal for our research.

7.1.2 Distributed Database

The database system is also to be distributed, and a current research goal is also to make the database federated. A federated database provides linking and access to different database schemas which need not be identical. This will be a significant contribution of REINAS, both because of the regional data management schemas available, REINAS, MBARI, NPS, and FNOC, but also because it is a current commercial and research development challenge. A continuing goal for the REINAS data schema is the handling of technology and representation generations of data. This long-term approach to data management is necessary for maximizing the return from the large investments required in data collection, as the important work is done through retrospective studies and data dissemination [GM93].

7.1.3 Visualization

Verity visualization (the true visualization) is an important aspect of both the data model and the visualization. We have made a concerted effort to quantify the data uncertainty in each of the measurement and modeling processes, and to include these uncertainties with the information system.

The REINAS information model attempts to do away with the practice of separating data and metadata, because the conditions under which any scientific observations are made (contained in the metadata) are important to analysis and interpretation. The legacy of the data requires improved information models, which we have developed, but the model requires in depth analysis and evaluation to measure our success. Operational use will also challenge and evaluate the success of the data model.

Because the users are also to be distributed, we have collaborative visualization software, or CSpray discussed in the report. We also have single user visualization applications (Spray, Mosaic MET) running now. The success of the Internet and distributed processing systems research has made this initial milestone possible. Support for future real-time (and not best effort as used in the Internet) and nowcasting applications will require research into more parallel and distributed applications and libraries to meet the throughput requirements. An exciting near term-goal is the migration of the REINAS collaborative applications onto the regional ATM (Asynchronous Transfer Mode) real-time network. Continuing improvements in user interface and functionality drive the work in our environmental visualization.

The primary products of REINAS are quantitative outputs, and graphic visualizations. The work in visualization methods is fundamental, and includes usability, multi-media, networking, and visualization methods. We are also actively working on new methods for visualizing the uncertainty associated with data and with the visualization process. The ability to provide the uncertainty directly in the visualization is exciting, for it gives our users an additional means of evaluating the data validity. Other developmental goals include the porting of the visualization software onto non-Silicon Graphics platforms. OpenGL and Motif are planned in the reworking of both Spray and CSpray which has already started. Much of the code in Spray is not GUI related, and the porting should be quite feasible.

7.1.4 Science Goals

An early goal of the project was to have an evaluation phase, using the Sea-Breeze phenomena as an atmospheric event to be studied. The prototype system did not have a large enough variety of instruments, nor sufficient and useable applications to support our science goals in 1994. Because the prototype cannot support operational users easily until a large variety of instruments becomes available, it was decided to focus on utilities to make the system easier to use–closer to "plug and play." We do have much of the data collected from our set of windprofilers and MET stations during the Monterey Area Ship Tracks (MAST) and coastal meteorology experiment available to retrospectively load into the REINAS database. A support utility has been developed to create load paths for new instruments. This should facilitate the addition of new data sources into the information model. The data load paths are a considerable challenge, and further refinements of these support tools will be a significant contribution of our REINAS research and development. An evaluation is planned where students at NPS will experiment with adding load paths into an instance of the REINAS information system.

Other science goals in the Phase III report [MLGL$^+$94] were to provide more standard products, or products that meteorologists routinely look at, to enhance the atmospheric modeling abilities, and to enhance the REINAS station network by adding new stations. The standard product features have been added to our Spray visualization software, and allow examination of data from the NORAPS atmospheric model, and of GOES satellite imagery with an easy-to-use interface. An ancillary goal for the standard products was to use them as a means of REINAS access to support other operational users.

There is limited access to meteorological station data through Xmet and Mosaic Met, but the standard products in Spray are not yet accessible. When REINAS becomes operational, there will be opportunity to evaluate our standard product applications and delivery to the Monterey Bay and San Francisco Air Pollution Control Districts, the National Weather Service Monterey Forecast Office, NOAA Fisheries and others who have expressed an interest. The products will be an excellent source for collaborative learning and research. A better evaluation by scientists will be possible as more data sources are made available.

The atmospheric modeling and data assimilation research goals directly affect the usefulness of REINAS as a regional tool. Work is ongoing to reduce the grid spacings in the NORAPS model so that its output is of more utility in a smaller region. The time required for output to be made available needs to be reduced. This requires better techniques for data assimilation in order to account for the instabilities introduced by the abrupt changes caused by assimilation of new measurements.

Reducing the regional scale and the time step is critical for nowcasting, an ongoing goal for the REINAS system. The challenges in regional forecasting include the lack of small scale observations and limited knowledge of how to dynamically constrain solutions on small scales. We expect to do this, and to display uncertainty about the product, which will be something unique and very useful. The ability to view, in real-time, the updates of the nowcasting model also drives the visualization algorithm and software development. An iterated development and evaluation of the visualization tools has provided for a close collaboration between computer scientists and environmental scientists.

7.1.5 New Stations

Station enhancements, include adding sensors where additional information will appreciably help the model assimilation process, is dependent upon the quality of modeling and interpolation developed. The possibility of using REINAS as a real-time system is compelling when one realizes that steerable instruments could be placed based on the conditions required, which saves money on the number and density of stations used, and yields better output results. A closed loop of data gathering, loading, querying, nowcasting, and decision making can provide features that would give scientists more for their money, and make real time another necessity for environmental research. Extensions of the present portable meteorological station (Port-a-Met), and construction of additional portable platforms are included in our plan. NEXRAD (Next Generation Radar) that is capable of a large range of environmental measurements has been installed on Mount Umunhum and should be available to us in the future.

7.1.6 User Specific Goals and Tasks

1. Develop the load paths for the additional data sets to allow development of data assimilation and visualization tools.
    - Load the UNIDATA surface land and ship data.
    - Load the UNIDATA rawinsonde data.
    - Load CDF data on hourly basis.
    - Load MBUAPCD and BAAQMD data once a day until more is available.
    - Develop decoder and load aircraft reports on 10 minute basis.

  - Load the UNIDATA large-scale model fields.
  - Load the GOES imagery.
  - Load AVHRR imagery.
  - Add video-camera sensors to the real-time REINAS network.

2. Develop the federated data base capability across the network to NPS and MBARI. NPS will need support in the installation of a REINAS data base to manage the NORAPS model fields and other data native to NPS.
3. Fine tune the standard visualization products and develop animation capability for display of large scale model fields as well as NORAPS fields. (This is the most user approachable part of Spray and is most likely to sell it to other users. Contour labeling is definitely desired.)
4. Develop the capability of working with observations, models and satellite images within the "analysis mode." (Analysis means diagnosis of the atmosphere from all pieces of information that represent it.)

   There is a need to develop capabilities to indicate data uncertainty, and the ability of Spray to correctly treat multiple vertical coordinates such as height, pressure or sigma coordinates. This will require some definition of horizontal surfaces when they are desired. We expect that different models will require different techniques.
5. Develop analysis capability using REINAS observation network for data gathered in summer 1994. The key here is to develop a method for initializing the NORAPS model from observations in real time. A working data base and API will help and save redundant code writing.
6. Develop data assimilation that is continuous (or at least as frequent) as the data can bear. Techniques to be explored or extended include the use of multiquadric interpolation in the NORAPS model as a constraint [NT94]. Kalman filtering should be explored also using simplified models for state estimation.

7.2 Detailed Goals from the Systems Perspective

1. To make REINAS into a long-term, retrospective scientific analysis tool collecting data in real time. This implies that the schema and API meet their objectives, which require the incorporation of different data streams, demonstrating changes of calibration, and quality checking the data. There must be a long-term strategy for bringing parts of data bases, network, and server components down, while maintaining the integrity of the system. This requires that processes, deployment, etc., and uncertainty features in the schema be debugged, and evaluated. It also requires that serious science users use REINAS for retrospective analysis, and to achieve their science goals.
2. To have REINAS show novel use of closed loop control for guided measurements of sparse data to reduce uncertainty optimally. This requires:
   - The Real Time Cache and API work quickly in practice.
   - The existence of a working guided instrument, and/or a test set up with sensors and a guided instrument. This task could be shown with a simple test set up, a simulator/emulator, a nowcasting model, and a reasonably populated data base.

- The characterization of the delays in the system: load delay, network delay, query delay, model delay, be characterized, and the resulting sum of delays be analyzed as to what the maximum delay is for different types of real-time control.
- End-to-end performance as the priority. This implies trade-offs. All of the planned features (e.g. the retrospective, and data lineage features.) may not be implementable and evaluated for the present study.

3. To have REINAS facilitate the rapid application development for scientific inquiry on a large body of data. This implies a robust API at a high level that scientific programmers can use and an electronic library managing a hierarchy of storage devices. The API requires:

   - The object API definition and implementation to be refined for ease of use, and extensibility.
   - Cross platform development, and likely, a set of test and example applications illustrating various features of the API, perhaps even an application builder that would incorporate the appropriate API calls.
   - Generalization of concepts so the API can handle more than metstations, wind profilers, and related instruments.
   - More examination of the language and application building issues, GUI independence, and the incorporation of real time and best effort streams.
   - Follow up on the attractive avenue of research for the collaborative visualization and the ATM.

   The electronic library requires means for managing and accessing the indices or catalogs of data stored, and the management of the data collections, including:

   - Support for data migration up and down the storage hierarchy.
   - Performance modeling and optimization of data migration.
   - Performance prediction, to provide users with meaningful estimates of the time required by the system for delivery of requested data.

4. To have REINAS provide combined measurement/model data for nowcasting and novel visualization synthesis, as well as data assimilation. This requires:

   - Model data within the database.
   - API supporting model data,
   - Graphics API solving irregular gridding,
   - New nowcasting models, and assimilation methods.
   - Visualization software to work with disparate scales and samplings of data.

5. REINAS tools to allow rapid incorporation of environmental instruments into a scalable, relocatable system. This requires:

   - Excellent utilities that automate incorporating new instruments into the schema.
   - Utilities for capturing metadata.
   - Visualization software to be general, allowing different regions to be used with any type of sensor.

- Utilities for the rapid deployment of instruments and REINAS system for conducting experiments.
- Scalability to be tested, including the network loads, data base loads, and instruments loads.
- A usability study to demonstrate incorporating a new instrument, and new instances of instruments easily into a version of the system.
- Ultimately a true relocation experiment will be necessary. For example, connecting to a network of instruments within the Gulf of Mexico, and getting the important regional/coastline data, and instrument data quickly loaded, debugged, and optimized.

  (This goal is probably the easiest to envision, but the hardest to achieve.)

REINAS FUTURE SCHEDULE

- o July 1994 – Begin Phase IV
  - Real-Time Experimentation / System Verification
  - Visualization Directly from Database
  - Support Mobile (sea and land) Instrument Platforms
  - Connection of Additional (and advanced) Instruments
    - CODARs
    - Wind Profilers
  - Support Federated Databases
  - Initiate Use of Collaborative Visualization
  - Provide REINAS as Real-Time Data to Pt. Lobos ROV operations, buoy support
- o July 1995 – Begin Phase V
  - Refine / Extend System and Security
  - Video Cameras as Digital Instruments
  - Database Support for Automated Data Quality Monitoring
  - Dynamic Network Topologies
  - Evaluation of New Technologies
  - Extend REINAS to Additional and New Instruments (e.g. NEXRAD)
  - Use Combination of Models and Measurements for State Estimation
  - Support Collaborative Visualization over (high speed) network
  - Support Operational Users (e.g. air pollution monitoring, forest fire, fisheries)
  - Develop REINAS System for port to other locales
- April 1997 – End of URI Funding

Table 7.1: Schedule of REINAS Future Plans

References

[Ber91] Elisa Bertino. Integration of heterogeneous data repositories by using object-oriented views. In *IMS '91 Proceedings*, pages 22–29. IEEE Comput. Soc. Press, 1991.

[BEW77] D.E. Barrick, M.W. Evans, and B.L. Weber. Ocean surface currents mapped by radar. *Science*, 198:138–144, 1977.

[BLC85] D.E. Barrick, B.J. Lipa, and R.D. Crissman. Mapping surface currents with codar. *Sea Technology*, October 1985.

[BM87] A. F. Blumberg and G. L. Mellor. A description of a three-dimensional coastal ocean circulation model. In *Three-Dimensional Coastal Ocean Models*, volume 4, page 208. American Geophysical Union, Wash. DC, 1987.

[Bos92] L. Bosack. Method and apparatus for routing communications among computer networks. *U.S. Patent assigned to Cisco Systems, Inc.*, 1992.

[Che88] David R. Cheriton. The V distributed system. *CACM*, 31(3):314–333, March 1988.

[CL91] Luis-Felipe Cabrera and Darrell D.E. Long. Swift: Using distributed disk striping to provide high I/O data rates. *Computing Systems*, 4(4):405–436, Fall 1991.

[Col89] R. Coltun. OSPF: An internet routing protocol. *ConneXions*, 3(8):19–25, 1989.

[Cro55] D.D. Crombie. Doppler spectrum of sea echo at 13.56 mc/s. *Nature*, 175:681–682, 1955.

[CS93] Dimitris N. Chorafas and Heinrich Steinmann. *Solutions for Networked Databases: How to move from Heterogeneous Structures to Federated Comcepts.* Academic Pres, 1993.

[DGLA95] H.P. Dommel and J.J. Garcia-Luna-Aceves. A taxonomy of floor control for networked multimedia applications. In *ACM SIGCHI 95*, page (submitted), 1995.

[EKM92] T. Ezer, D. S. Ko, and G. L. Mellor. Modeling and forecasting the gulf stream. *Marine Tech. Soc. Journal*, 26(2):5–14, 1992.

[GLA88] J.J. Garcia-Luna-Aceves. Routing management in very large-scale networks. *Future Generation Computing Systems*, 4(2):81–93, 1988.

[GM93] Nahum D. Gershon and C. Grant Miller. Dealing with the data deluge, special report: Environment, part 2. *IEEE Spectrum*, 30(7):28–32, July 1993.

[Hed88] C. Hedrick. Routing information protocol. RFC 1058, Stanford Research Institute, SRI International, Menlo Park, CA, June 1988.

[ISO89] ISO. Intra-domain is-is routing protocol. *ISO/IEC*, JCT1/SC6(WG2 N323), September 1989.

[ISO91] ISO. Protocol for exchange of inter-domain roueing information among intermediate systems to support forwarding of iso 8473 pdus. Technical report, International Standards Organization, 1991.

[KR78] B. W. Kernighan and D. M. Ritchie. *The C programming language.* PH, 1978.

[KS92] Ravi Krishnamurthy and Ming-Chien Shan. Very large data bases. VLDB '92. In *Proc of 18th Intl. Conference on Very Large Data Bases*, pages 277–291. Morgan Kaufmann, 1992.

[LAML94] Glen G. Langdon, Robert J. Antonucci, William W. Macy, and Dean Long. On compression of data from a meteorological stations. In *Data Compression Conference, IEEE Computer Society*, Snowbird UT, March 1994.

[LM93] Glen G. Langdon and Mareboyana Manohar. Centering of context-dependent components of prediction error distributions. In *Proc SPIE: Applications of Digital Image Processing XVI*, volume 2028, pages 26–31, San Diego, CA, July 1993.

[LNS93] W. Litwin, M. A. Neimat, and D. A. Schneider. LH* – linear hashing for distributed files. *SIGMOD Record*, 22(2):327–336, June 1993.

[LR91] K. Lougheed and Y. Rekhter. Border gateway protocol 3 (bgp-3). RFC 1267, Stanford Research Institute, SRI International, Menlo Park, October 1991.

[MGLA87] J. Mathis and J.J. Garcia-Luna-Aceves. Survivable multiband networking. In *IEEE MILCOM '87*, Washington, DC, October 1987.

[MLGL$^+$94] P.E. Mantey, D.D.E. Long, J.J. Garcia-Luna, A.T. Pang, H.G. Kolsky, et al. Reinas: Phase three - systems design. Report UCSC-CRL-94-08, Univ of California, Santa Cruz, Santa Cruz, CA 95064, March 1994.

[MLP$^+$93] P.E. Mantey, D.D.E. Long, A.T. Pang, H.G. Kolsky, et al. Reinas: Phase two - requirements definition. Report UCSC-CRL-93-34, Univ of California, Santa Cruz, Santa Cruz, CA 95064, July 1993.

[NT94] W.A. Nuss and D.W. Titley. Use of multiquadric interpolation for meteorological objective analysis. *Monthly Weather Review*, 122:(to appear), 1994.

[PA94] A. Pang and N. Alper. Mix and Match: A construction kit for visualization. pages 302–309, 1994.

[PA95] A. Pang and N. Alper. Bump mapped vector fields. In *SPIE Visual Data Exploration and Analysis*, page (accepted), 1995.

[PAFW93] A. Pang, N. Alper, Jeff Furman, and Jiahua Wang. Design issues of spray rendering. In *Compugraphics'93*, pages 58–67, 1993.

[PFN94] A. Pang, J. Furman, and W. Nuss. Data quality issues in visualization. In *SPIE Visual Data Exploration and Analysis*, volume 2178, pages 12–23, 1994.

[PS93] A. Pang and K. Smith. Spray rendering: Visualization with smart particles. In *Visualization'93 Proceedings*, pages 283–290, 1993.

[PW95] A. Pang and C. Wittenbrink. Spray rendering: A modular visualization environment. In *SIGGRAPH report on Modular Visualization Environment*, page (submitted), 1995.

[PWG95] A. Pang, C. Wittenbrink, and T. Goodman. CSpray: A collaborative scientific visualization application. In *SPIE Conference on High- Speed Networking and Multimedia Computing*, page (accepted), 1995.

[Rou91] N. Roussopoulos. An incremental access method for ViewCache: concept, algorithms, and cost analysis. *ACM Transactions on Database Systems*, 16(3):535–563, Sept 1991.

[SL90] A. P. Sheth and J. A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *Computing Surveys*, 22(3):183–236, Sept 1990.

[SMK+93] M. Satyanarayanan, Henry H. Mashburn, Puneet Kumar, David C. Steere, and James J. Kistler. Lightweight recoverable virtual memory. *Proceedings of the 14th ACM Symposium on Operating Systems Principles*, (5):146–160, December 1993.

[Ste92] M. Steenstrup. Inter-domain policy routing protocol specification: Version 1. Internet Draft, May 1992.

[Svo84] Liba Svobodova. Resilient distributed computing. *IEEE Transactions on Software Engineering*, SE–10(3):257–268, May 1984.

[WoRSR+94] J.M. Wilczak, others R.G. Strauch, F.M. Ralph, B.L. Weber, D.A. Merritt, J.R. Jordan, D.E. Wolfe, L.K. Lewis, D.B. Wuertz, J.E. Gaynor, S.A. McLaughlin, R.R. Rogers, A.C. Riddle, and T.S. Dye. Contamination of wind profiler data by migrating birds: Characteristics of corrupted data and potential solutions. Technical report, Environmental Technology Laboratory–NOAA, 1994.

[WSF+94] C. Wittenbrink, E. Saxon, J. Furman, A. Pang, and S. Lodha. Glyphs for visualizing uncertainty in environmental vector fields. In *SPIE Visual Data Exploration and Analysis*, page (accepted), 1994.

Index