REINAS: Real-Time Environmental Information Network and Analysis System: Phase IV.Final

EXPERIMENTATION[*]

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

UCSC-CRL-95-62
December 29, 1995

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. Current applications are in meteorology and oceanography; others envisioned include geophysics, hydrology, air pollution, etc.

The REINAS project has been funded by a Department of Defense University Research Initiative through the Office of Naval Research. Participating institutions include the Naval Postgraduate School (NPS), the Monterey Bay Aquarium Research Institute (MBARI) and the Baskin Center for Computer Engineering and Information Sciences of the University of California, Santa Cruz (UCSC).

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 previous phases a survey of available technologies was made, selections of the those 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.

This report documents the status of REINAS in Phase IV – the Experimentation and System Verification Phase.

Acknowledgements

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 work of Phase IV in 1994 and 1995:

- **UCSC Faculty and Staff:**
  Patrick E. Mantey, Daniel M. Fernandez, Harwood G. Kolsky, Glen G. Langdon, 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 during Phase IV:

- **System and DBMS Architecture:**
  Bruce R. Montague, Eric C. Rosen, Linda Begley, Jose Mendoza, Theodore R. Haining, Carles Pi-Sunyer.

- **Instrumentation:**
  Eric C. Rosen, Catherine Tornabene, Robert Sheehan, John Wiederhold, Sume Biak, Amelia Graf, Ted Goodman, Dan Harris, Mike Allen, Suzana Djurcilov.

- **Instrument Interfaces:**
  Michelle D. Abram, Jimmy Chan, Ted Dunn, Scott Lin, Chris Thomas, David K. Schreiber.

- **Networking Research:**
  Chane L. Fullmer, Jochen Behrens, Hans-Peter Dommel, Brian Levine, Dean Long, Shree Murthy, Clay Shields.

- **Visualization System:**
  Naim Alper, Jeffrey J. Furman, Tom H. Goodman, Elijah C. Saxon.

- **Modeling:**
  Cheng Tang.

- **Data Compression:**
  William W. Macy, Robert J. Antonucci, David Kulp, 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. REINAS has been funded by a DoD University Research Initiative (URI) through the Office of Naval Research. Participating institutions include the Naval Postgraduate School, the Monterey Bay Aquarium Research Institute (MBARI) and UCSC's Baskin Center for Computer Engineering and Information Sciences.

REINAS is a prototype Geographic Information System (GIS) providing data and tools for estimating the "state" of the environment spatially and temporally. Its architecture and database design makes REINAS relevant to a variety of environmental measurements. Current applications are in meteorology and oceanography; others envisioned include geophysics, hydrology, air pollution, etc.

REINAS is a distributed system connecting instruments to a relational database in real time via a variety of physical network links all employing TCP/IP (Internet) protocols. It includes advanced visualization support, plus interfaces to models and to standard visualization tools. Operational users, including scientists involved in instrument deployment or "steering", desire real-time data from the GIS. Capture of data over time, and thus creation of an historical "archive", is critical to support scientific investigations.

REINAS has many potential benefits. It creates synergy among many organizations and users by combining data sources and avoiding duplication. The data in the GIS are registered to a common set of spatial coordinates, and time stamps are also consistent; correlation of different measurements (e.g. ocean currents and surface winds) is thereby greatly facilitated. Interdisciplinary research is also enabled.

In creating REINAS we have met technical and cultural challenges. Connection of diverse instruments, decoding instrument data streams and building "load paths" to the database, use of instruments in real time, data validation and instrument calibration, capture of instrument "metadata", and registration of data both spatially and temporally are some of the technical challenges. Non-technical issues included inter-institutional cooperation, ownership of data, demand of real-time (operational) users vs. retrospective users, long term support for the REINAS operation, plus cultural problems in sciences where data sharing is not the norm.

This report documents Phase IV of REINAS focusing on Real-Time Experimentation and System Verification. An earlier report [MLGL$^+$94a] documented the first half of Phase IV. Parts of it have been subsumed in this report for completeness. The original REINAS System Design is described in the Phase III report [MLGL$^+$94b] which provides a detailed description of the REINAS architecture.

Current WORLD WIDE WEB pages for REINAS and related work are:

- REINAS: http://csl.cse.ucsc.edu/reinas.html
- Environmental Visualization: http://www.cse.ucsc.edu/research/slvg/envis.html
- Uncertainty Visualization: http://www.cse.ucsc.edu/research/slvg/unvis.html
- SlugVideo: http:/sapphire.cse.ucsc.edu/SlugVideo/dream-inn.research/slvg/envis.html

---

[1] This chapter by P.E. Mantey

1.1 Project Objectives:

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

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

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

- As a LABORATORY, REINAS supports
  - Meteorology and Oceanography (Real-Time and retrospective)
  - Systems Research (Performance, Networking, Distributed and Federated Databases)
  - Development of Visualization
  - Evaluation of Instruments / Measurement Techniques
- As a SYSTEM, REINAS
  - Is Portable and Extensible
  - Exploits "Off the Shelf" Database Systems
  - Works on a Variety of Computer Platforms
  - Networking uses TCP/IP (Internet) Protocols
  - Can be used with (and on board) Mobile Platforms

In the REINAS architecture, continuous real-time data are 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 is designed to support both single-user and collaborative scientific work in a distributed environment.

Figure 1.1 depicts the REINAS system and data flow, from instruments to users, in its ultimate implementation. 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. (In earlier versions of REINAS all of the paths did not go through the API.)

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, Table 1.2 the schedule for the first four phases, and Table 1.3 items planned for Phase V.

1.3 Major Tasks Completed during Phase IV:

1. A significant milestone was achieved in the spring of 1995 when REINAS went on-line with its first real-time source, the Long Marine Lab wind profiler. Following some testing and verification, REINAS began providing simple support for the work of oceanographic and meteorological users.
2. The REINAS database design now supports measurement data in a temporal / spatial organization, and also includes support for metadata (i.e. the data about the instruments, calibration, etc.) within the database. During this past year, load paths have been created (software) to automatically and continually transfer data into the data base from a variety of instruments.
3. Currently, there exist approximately eleven MET stations that are connected directly to REINAS. Of these, two are (or have been) deployed on moving vessels. One is currently aboard the vessel R/V Pt. Lobos and is Internet accessible via their existing network shipboard microwave connection. The other has recently been deployed aboard the NOAA research vessel R/V McArthur, as well as the R/V Pt. Sur, and was Internet accessible via a UHF radio link from UC Santa Cruz to the MET station aboard the vessel.
4. A real time link was established via REINAS between two wind profiler locations in the Monterey Bay area, one at Santa Cruz and the other at Point Sur. Wind profilers provide a measurement of the horizontal winds as a function of height to a maximum height of approximately 4 km.
5. Three CODAR SeaSondes are deployed in the Monterey Bay area. These instruments are connected to REINAS and the data they produce are used to compute maps of vector ocean surface currents within Monterey Bay.
6. Other instruments that are now part of the current data feed into REINAS include CTD's (Conductivity, Temperature, Depth), ADCP (Acoustic Doppler Current Profiler), thermistor chains, and assorted biochemical sensing instruments.
7. During the past year, an investigation into the development and implementation of remote Video streams as real-time REINAS instruments was initiated. A prototype steerable video platform was defined during the fourth quarter '94, built during the first quarter of '95, and initially deployed at Long Marine Lab (an existing REINAS instrument site) on March 30, 1995. In August it was moved to a more appropriate site (atop a 10 story beach-front hotel) offering a better view of Monterey Bay.

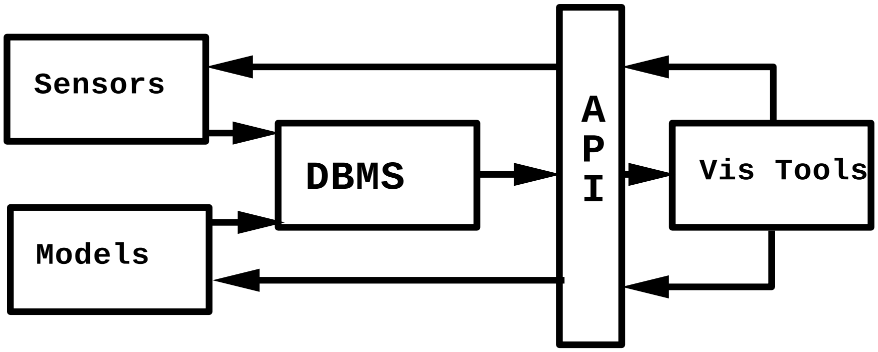

Figure 1.1: REINAS System Integration.

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"
- Database(s)
- Computer Networks Linking
  - Instruments - Databases
  - Users
- Real-Time Visualization
  - Visualization from Database(s)
  - Collaborative and Uncertainty Visualization
  - Fusion of Data from Measurements and Models
  - Retrospective Analysis
  - "Now-Casting"

Table 1.1: Key Components of the REINAS Project.

8. Network communication today among REINAS components and users is now being accomplished via a mixed-media networking infrastructure that encompasses new and existing telephone lines, Internet connections, and (point-to-point) radio links, as well as the networking software and hardware needed to control and manage the interconnection of REINAS sites.
9. Under REINAS support, we have developed new algorithms and protocols for channel access, routing, multicasting, and floor control of distributed applications. We have

REINAS SCHEDULE 1992-1995

- 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
- July 1994 – Begin Phase IV
  - Real-Time Experimentation / System Verification
  - Visualization Directly from parts of Database
  - Support Mobile (sea and land) Instrument Platforms
  - Connection of Additional (and advanced) Instruments
    - CODARs
    - Wind Profilers
  - Develop Support for Federated Dbases
  - Initiate Use of Collaborative Visualization
  - Provide REINAS as Real-Time Data to Pt. Lobos ROV operations, buoy support

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

been developing a new type of channel access discipline called Floor Acquisition Multiple Access (FAMA). FAMA consists of both carrier sensing and a collision-avoidance dialogue between a source and the intended receiver of a packet.

10. In the Visualization component, during the past year we have completed implementation of the major ideas behind Spray Rendering. Spray is an extensible framework for visualization. Its current implementation is divided into three modes: monitor, forecast, and analysis.
11. In 1995 a major visualization effort was focused on the redesign of Spray with the objective of creating a modeless system. The new version is nicknamed "PET slvg" and incorporates the same modularity as in Spray.
12. A new effort was undertaken on Uncertainty Visualization. Almost all data has some form of uncertainty. The uncertainty information must be presented together with the data to give users the ability to localize and distinguish regions where data quality is poor, and eventually make better analysis.
13. Results in Data Compression have impact in several dimensions of REINAS. Integration of compression transparent to the user is a part of the communications software for the instrument network, the database system design, and for Internet access to database for data of interest.

REINAS PLANS FOR PHASE V 1996-1997

- Begin Phase V
  - Refine / Extend System and Security
  - Incorporate Video Cameras as Digital Instruments
  - Provide Database Support for Automated Data Quality Monitoring
  - Support Dynamic Network Topologies
  - Continue 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 1.3: Schedule of REINAS Plans for 1996-1997

2. REINAS from a Science Perspective

2.1 Meteorological Applications

2.1.1 Collection of Observations

Cutting edge meteorology[1] 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.

2.1.2 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 combined with model renditions of the atmosphere and 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.

2.1.3 Forecasting

Operational meteorologists of all types require the ability to predict the future state of the atmosphere. 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 a numerical model that can be run in real-time to make short range predictions of small-scale circulations.

2.1.4 Meteorological Science Objectives

The overall meteorological 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 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 develop meteorologically relevant visualization tools that can be used to study circulations in the Monterey Bay region as well as more general meteorological applications;

---

[1] This section by Wendell Nuss, NPS

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.

2.1.5 Three Major Thrusts of REINAS

1. Data Collection: To develop a network of environmental observing systems capable of defining the meteorological and oceanographical scale of interest that occurs over a region.
   - Collection of data is done in "real-time" for all sites and instrument types.
   - Observations are sufficiently dense to define the circulations of interest.
   - Observations are used in conjunction with a numerical model to provide four dimensional views of the environment through data assimilation.
2. Data Management: To develop a real-time data management system capable of handling high-volume, high frequency, and dynamic data sets.
   - System loads a diverse suite of observational and model data.
   - System must be able to load and retrieve high-volume data in real-time.
   - Database is distributed across a network.
3. Data Visualization: To develop a display/visualization system that allows real-time viewing and analysis of observational and model generated data. Observations, satellite imagery, and model-generated fields viewed through a single, unified display.
   - Historical, current, and predicted fields.
   - Also allows for diagnostic exploration of the data.
   - In single-user and collaborative distributed environment.

2.1.6 Some Preliminary Meteorology 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.

2.2 Oceanographic Applications

2.2.1 Collection and Archival of Measurements

As in meteorology, state-of-the-art oceanography [2] requires the use of widely varied observing systems, such as CTD (conductivity, temperature, density), current meters, ocean current measurement devices, and satellite imagery. Integration of all of this data into one system for real time and retrospective applications has and will continue to save much time and effort in terms of directly visualizing events of interest as well as in accessing the data in general for basic analysis.

2.2.2 Ocean Science Objectives

Some ocean science goals of the REINAS project are:

1. To relate the forcing of the seabreeze to the advection of surface waters in Monterey Bay.
2. To develop data assimilation tools that will aid in the understanding and interpretation of oceanographic circulation phenomena within Monterey Bay.
3. To better understand the relationship between internal tidal forcing (M2) on the bathymetry of Monterey Bay and the signatures of these features on the near-surface currents.
4. To incorporate oceanographic measurements as initial conditions into an ocean model (such as the Mellor model) for real-time forecasts.
5. To further incorporate advanced visualization as a tool for analysis of oceanographic phenomena.

---

[2] Section by Dan Fernandez

6. To measure the near surface (uppermost meter) current shear within the Monterey Bay area with multi-frequency HF radar systems and better characterize previously observed rotations of the current vector with depth [FVT95] in conjunction with supporting in-situ meteorological and oceanographic measurements.
7. To understand the basic nature of high-frequency radar measurements of ocean surface currents and develop further the algorithms that use HF radar measurements to measure wind direction/wind speed as well as the directional ocean wave spectrum.
8. To examine the relationship between the near surface ocean currents, and the thermal/visual structure of the ocean surface within Monterey Bay as revealed by GOES-9 and remote video.

2.2.3 Some Preliminary Ocean Science Results

1. Strong diurnal variability has been detected within CODAR-derived ocean surface current vectors. This is believed to be associated with the diurnal cycle of the sea breeze.
2. The M2 tidal component ellipses computed from CODAR-derived surface current data appear to have an alignment with the local bathymetry. This is believed to be associated with an internal tide at the M2 frequency.
3. Radial data from CODAR ocean surface current radars appear to have a larger uncertainty than previously believed. Baseline comparisons between CODARs located across the Bay from each other appear to have an RMS difference in the radial currents they measure of 14-20 cm/s [Mel95], [Fer95]. This may be associated with antenna pattern problems, or else problems with the algorithm that accomplishes the inversion. Access to data from within REINAS has contributed to this study.
4. Daily (and longer) averages of CODAR data collected in Monterey Bay often exhibit the presence of an anti-cyclonic gyre located just outside the mouth of the Bay. The source of this feature and its effects on the circulation within the region are currently under investigation.

2.2.4 Operational Use of Real-time Data

In addition to the research applications described above, the real-time network of oceanographic instrumentation (especially the remote sensing instruments) provide essential data that could be used in search/rescue operations as well as in emergency spill response. Not only are the data vital in a real time sense, but retrospective analysis of such data collected (in conjunction with supporting in-situ meteorological data) can provide clues to trends associated with wind forcing, thus provide another means of forecasting the next state of the ocean based on the current state of the environment. This has direct implications, for instance, in predicting the location of floating objects.

3. REINAS Instrumentation

3.1 Introduction:

The REINAS[1] system has been designed to accept data from a variety of meteorological and oceanographic instruments. A significant milestone was achieved in the spring of 1995 when REINAS went on-line with its first real-time source, the Long Marine Lab wind profiler. One of the main accomplishments of REINAS during Phase IV was the testing of a large number of different instruments on line.

Sampling periods may be as short as five seconds (e.g. some surface Met stations) or as long as one hour, as in the case of some of the CODAR SeaSondes.

Despite differences in instrument specifics and manufacturer, 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.

In its early stages, REINAS focused upon managing data from three specific types of instruments: surface meteorological (MET) stations, Coastal Ocean Dynamics Application Radars (CODARs), and vertical wind-profilers. These core instruments provided a rich set of environmental data-streams. MET stations generate multiple parallel streams of time-series data (scalar and vector valued), profilers produce vertical arrays of vector and related scalar data, while CODARs generate two-dimensional arrays of vectors organized either radially or, when combined from multiple sites, on a regular grid bounded by the local coastline.

The current REINAS operational network consists of inputs from:

- Radio links (UHF and 915 spread spectrum)
- Leased lines
- Existing Internet links

The current suite of REINAS instruments or data feeds are:

- MET stations
- CODAR
- Wind Profilers
- CTD (Conductivity, Temperature, Depth) sensors
- Buoy Platform instruments, which include MET, CTD, thermistor chain, and ADCP (Acoustic Doppler Current Profiler)
- GOES satellite images
- UNIDATA
- Video
- OSCR, SAR samples

---

[1] Chapter by Dan Fernandez

3.2 MET Stations:

REINAS now has access to many different MET station feeds. They may be classified as real-time feeds, virtual feeds, and pending feeds. "Real-time" in this case means the data comes directly from the instruments. "Virtual" means the data has been stored in intermediate files and made available to REINAS by file transfer later, usually in summarized form.

The standard REINAS MET station consists of a number of individual sensors that include wind speed, wind direction, air temperature, barometric pressure, and humidity. Additional sensors that some sites are equipped with are rain gauge sensors and solar irradiance sensors. The outputs of these sensors are hooked to a data logger (Campbell Scientific CR-10/CR-21 data loggers are the standard ones used by REINAS).

The following is a list of REINAS MET Station Feeds:

- Real Time Feeds
    - UCSC (Ethernet)
    - Long Marine Lab (microwave link)
    - Lockheed, Bonny Doon (leased line)
    - Moss Landing (56 kB existing link)
    - Granite Canyon, includes CTD (UHF link)
    - R/V Pt. Lobos (ship to shore microwave)
    - Aquamet, when deployed (UHF link)
    - Portable land MET station, when deployed (UHF link)
    - Monterey Bay Aquarium (UHF link)
    - Pt. Sur Coast Guard Station (UHF link)
    - Point Sur Naval Station (UHF link)
    - Mount Umunhum (existing Internet connection and UHF link)
- Virtual Feeds
    - Fort Ord
    - M1 Atlas Buoy
    - M2 Atlas Buoy
    - California Department of Forestry
    - UNIDATA

3.3 CODAR

Coastal Ocean Dynamics Applications Radar (CODAR) is a version of high-frequency radar manufactured by CODAR Ocean Sensors, Ltd. Three CODAR SeaSonde instruments are currently deployed in the Monterey Bay area and provide data via REINAS. The three units are deployed at Long Marine Lab (Santa Cruz), Moss Landing, and Point Pinos, which is at the southern tip of Monterey Bay. These units provide measurements of radial ocean surface currents on an hourly basis. Data from two or more of these CODAR sites are combined hourly at UC Santa Cruz, which provides maps of ocean surface currents with a resolution of approximately 3 km. HF radar systems measure radial ocean surface currents by measuring the Doppler shift associated with radar energy that is scattered off of ocean gravity waves that are between 5 and 50 meters in length. The difference between the Doppler shift of the returned signal and the expected Doppler shift of the ocean gravity wave allows an estimate of the value of the advection of the waves by the ocean surface current over the patch of ocean that the radar observes. Data from the CODAR units is copied to the REINAS system hourly, where it is then combined to form the vectors. [BEW77].

Most recently we have made measurements both with CODAR and OSCR (Ocean Surface Current Radar), another type of high-frequency radar that is used to measure ocean currents. The first level of comparison has been to make baseline checks, or comparisons of radials as measured by each system along the line that joins the two systems. Figure 3.2 presents an example of such measurements for the OSCR system, where the errors along the baseline averaged approximately 5 cm/s, which is within the expected error tolerance for this system [FP96].

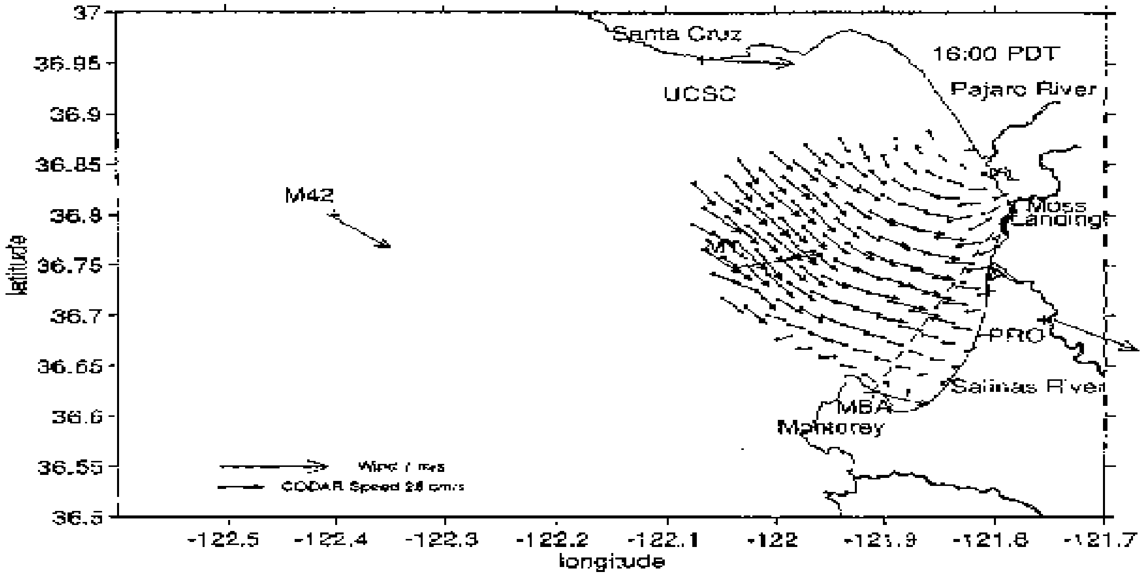

Figure 3.1: Example CODAR results: Mean CODAR-derived current and mean wind fields for 1600 PDT in September 1992.

The CODAR and SeaSonde feeds are:

- SeaSonde at Long Marine Lab (UHF link)
- SeaSonde at Point Pinos (UHF link)
- SeaSonde at Moss Landing (existing Internet, 56 kB link)
- Best fit of radials to vectors, done at UCSC

3.4 Radar Wind Profiler

Wind profilers provide a measurement of the horizontal winds as a function of height to a maximum height of approximately 4 km. A real time link was established via REINAS

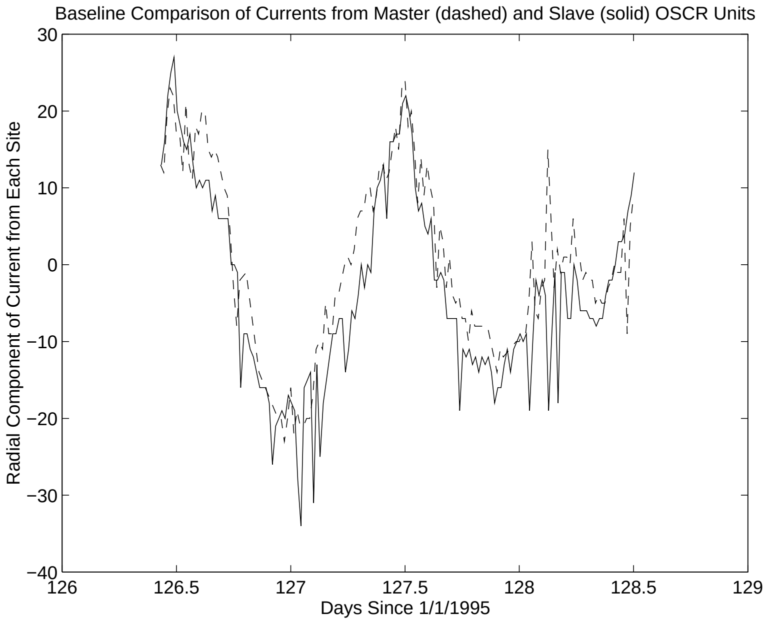

Figure 3.2: OSCR Baseline Comparison

between two wind profiler locations in the Monterey Bay area, one at Santa Cruz and the other at Point Sur. A real time link to a wind profiler results in moments (from which wind vectors may be computed) that are received approximately each minute. Receipt of the data in this fashion results in the capability to estimate the wind speed and direction vs height on a much more regular basis than the hourly data provided by the NOAA wind profilers. Two other wind profilers exist in the Monterey Bay area (both of which are at Fort Ord and are owned by the Navy Postgraduate School). 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. Real time data from these instruments will become available when a network link is established at Fort Ord.

- "Real Time" Wind profilers
  - Long Marine Lab, NOAA (Microwave link)
  - Previously, Point Sur, NOAA (UHF radio link)
- "Virtual" Wind profilers
  - Fort Ord 915 profiler (NPS)
  - Fort Ord 404 profiler (NPS)

3.5 Other Instruments

Other instruments that are part of the current data feed into REINAS include CTD (Conductivity, Temperature, Depth) sensors, ADCP (Acoustic Doppler Current Profiler), thermistor chains, and assorted biochemical sensing instruments, such as those that measure Chlorophyll-A or trace gases. Instruments that make these measurements exist both along the coast (CTD at Granite Canyon) and (more generally) offshore, such as on the MBARI buoys M1 and M2:

- MBARI M1 and M2 buoys (microwave link to MBARI)
  - Standard atlas MET instrumentation
  - Thermistor chains to 250 meter depth
  - CTD at surface
  - Various chemical/biological sensors (e.g. fluorescence)
  - M1 has ADCP
- CTD at Granite Canyon, NMFS (UHF link)

Experiments with satellite data have been done on:

- GOES (data collected from NWS, Monterey)
- Synthetic Aperture Radar (6 images from ERS-1 spring 1995)

3.6 Portable Meteorological Station

The development of "Port-a-Met" was completed late in June 1994 and was demonstrated during the ONR site visit on September 14, 1994. It is a portable battery-powered MET station and REINAS PC architecture which is linked to REINAS on the ethernet via half-duplex 9600 baud radio modems from Teledesign. "Port-a-Met" was built on a trailer and is easily deployable to a location of interest.

In addition, two MET stations have been deployed on ships. One is currently aboard the vessel R/V Pt. Lobos and is Internet accessible via their existing network shipboard microwave connection. The other has recently been deployed aboard the NOAA research vessel R/V McArthur as well as the R/V Point Sur and was Internet accessible via a UHF radio link from UC Santa Cruz to the MET station aboard the vessel.

3.7 Video

During the past year, an investigation into the development and implementation of remote video streams as real-time REINAS instruments was initiated. At a system level, this included defining a prototype video instrument, building or acquiring the necessary hardware and software, deploying the camera at an appropriate location, and commencing work on integrating the resulting video data into REINAS. The prototype steerable video platform was defined during the fourth quarter '94, built during the first quarter of '95, and initially deployed at Long Marine Lab (an existing REINAS instrument site) on March 30, 1995. In August it was moved to a more appropriate site (atop a 10 story beach-front hotel) offering a better view of Monterey Bay. The high data-rates achievable with a remote camera instrument (64 kbps to 8 Mbps depending on frame rate, content, and other factors) also motivated a re-examination of the radio modem technology used to network remote sites. A new option augmenting the existing 9 kbps technology was discovered integrated into the REINAS/PC design; this technology provides IP-level data rates exceeding 0.8 Mbps.

4. REINAS Visualization

4.1 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+94b]. 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.
- An extensible system to explore different ways of visualizing data.
- Support for collaborative visualization among geographically dispersed scientists and data sets.[Man94]

The development and testing has been done on Silicon Graphics (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.

4.1.1 Monitor Mode

In the monitor mode, users can watch the most current state of the environment. See Figure 4.1. 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.

4.1.2 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).

---

[1] This chapter by Alex Pang

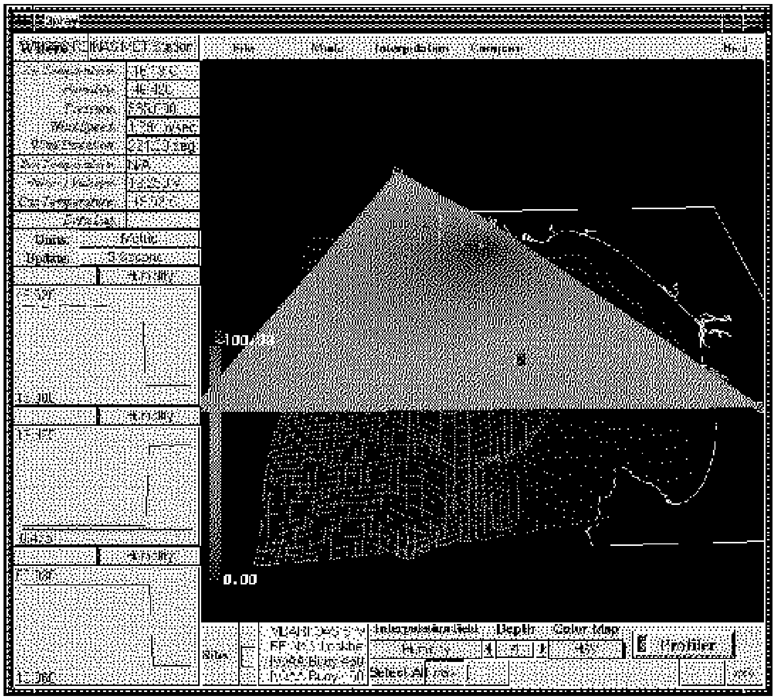

Figure 4.1: View of the Monterey Bay showing three sensor locations and an interpolated humidity field from a subset of these sensors.

Figure 4.2 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.

4.1.3 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 4.3 shows the interfaces available in analysis mode as well as illustrate some of the possible visualization methods.

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 (**sparts**) which are sent into the data space to seek out features of

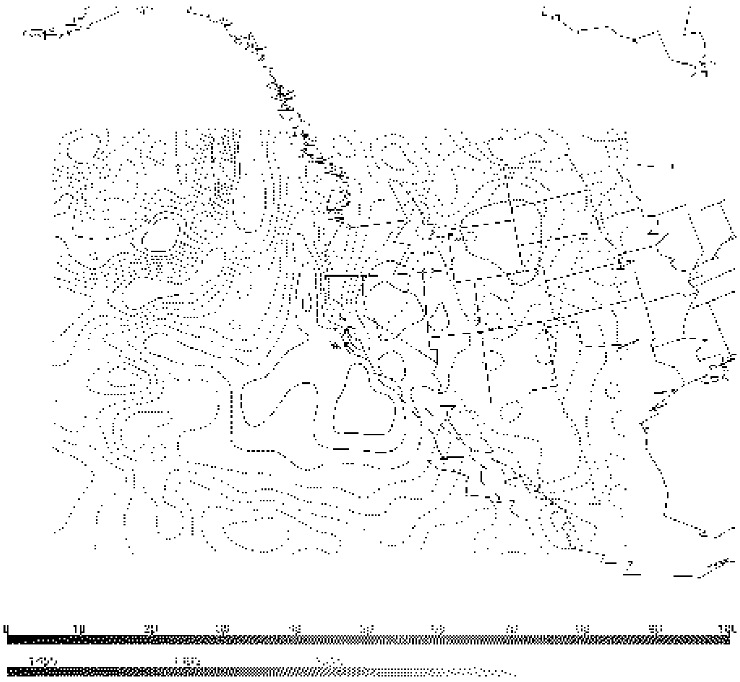

Figure 4.2: Sample Forecast Product

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 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.

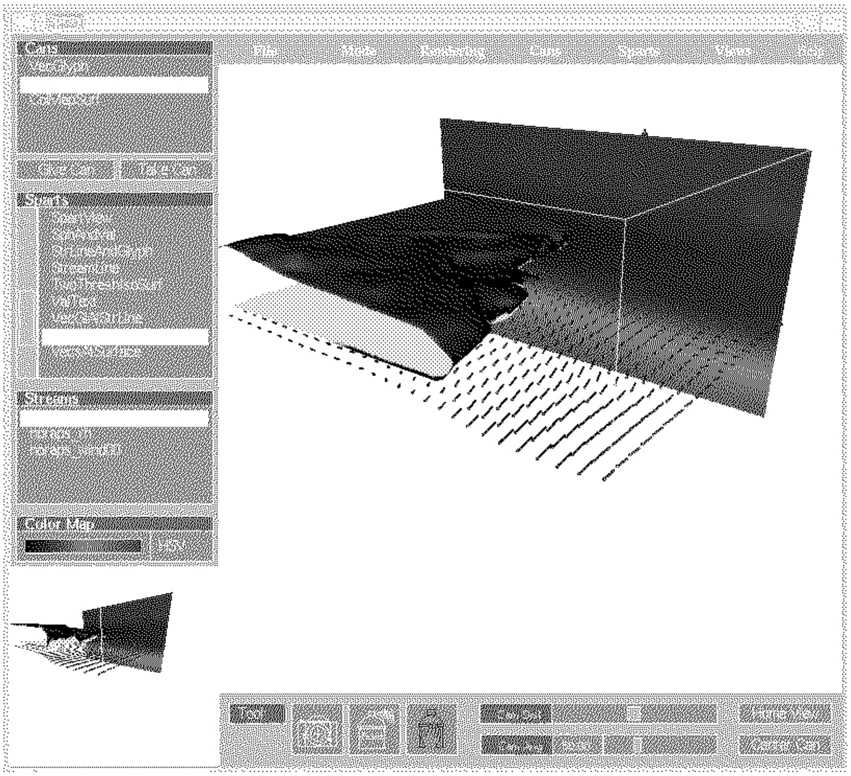

Figure 4.3: Analysis mode interface

- Multi-platform extension (OpenGL) and GUI issues.
- Data assimilation for integration of model and observation data.
- Parallelization and efficiency issues.
- Irregular grids.

4.2 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.

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.

---

[2] This section by Naim Alper

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.

4.3 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.

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 4.4 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.

---

[3] This section by Tom Goodman

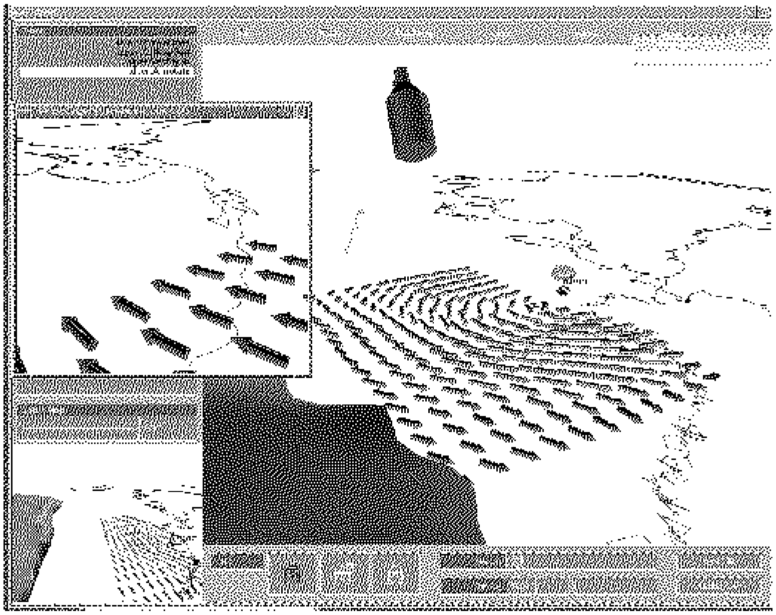

Figure 4.4: Collaborative visualization. The larger view shows what the local user is viewing. The small views show what the collaborator is looking at. The "eyeballs" show the locations of the viewers.

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.

4.4 Floor Control Protocols

Floor control [DGLA96] [4], together with session control as an infrastructure for online meetings, allows to mitigate race-conditions in shared resource usage within networked multimedia applications for multipoint, multimedia collaboration across the REINAS network. Examples are multiparty control of remote instruments, turn-taking on voice channels, or selective and fine-grained control of visualization objects. Provision of floor control builds on top of reliable or unreliable multicast and can be applied in both asynchronous and synchronous collaboration. Although a decade old, the concept of floor control has been implemented only in few collaborative systems, whose architecture is generally centralized and whose session links and nodes are assumed to be failure-free.

The methodology behind floor control relates to distributed mutual exclusion and, for database-oriented multimedia applications, concurrency control. Opposite to concurrency

---

[4] This section by Hans-Peter Dommel

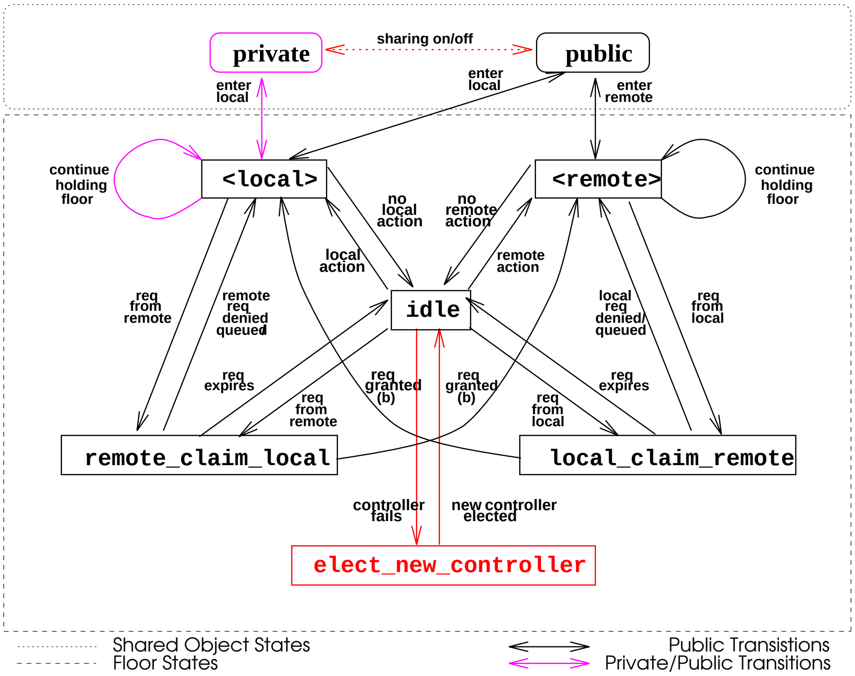

Figure 4.5: State Diagram for Floor Control Protocol (FCP)

control, resources are of various types, and since group work is highly interactive, the transaction notion of serialization, commits, or rollbacks does not translate into the multimedia context. Important criteria for implementing floor control are:

1. Scalability: adjusting to varying number of users.
2. Performance: keeping control information traffic low, serving requests with high responsiveness.
3. Resilience: fault-tolerance 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 etc.).

A distributed, scalable, and generic (adapting to different resource types) Floor Control Protocol (FCP) has been designed (see Figure 4.5). It is hierarchical with respect to collaborators (groups/subgroups) and resources (granularity). It has primary states (private vs. public), secondary states (for floors), and a recovery state for session failures. FCP performs contention-avoidance on shared resources without predefined token-scheduling and allows for selection of different floor service disciplines ("policies"). The distributed mutual exclusion algorithm behind FCP, called FACE allows for single or multiple entries into the critical section of shared work for certain media (voice, video). There are feedback floors for temporary backchannels.

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

4.5 Visualization of Uncertainty

Measured[5] 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 new vector glyphs, or icons, to visualize uncertain winds and ocean currents[WSF+95]. 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, with uncertainty, Figure 4.6. 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].

---

[5] This section by Craig Wittenbrink

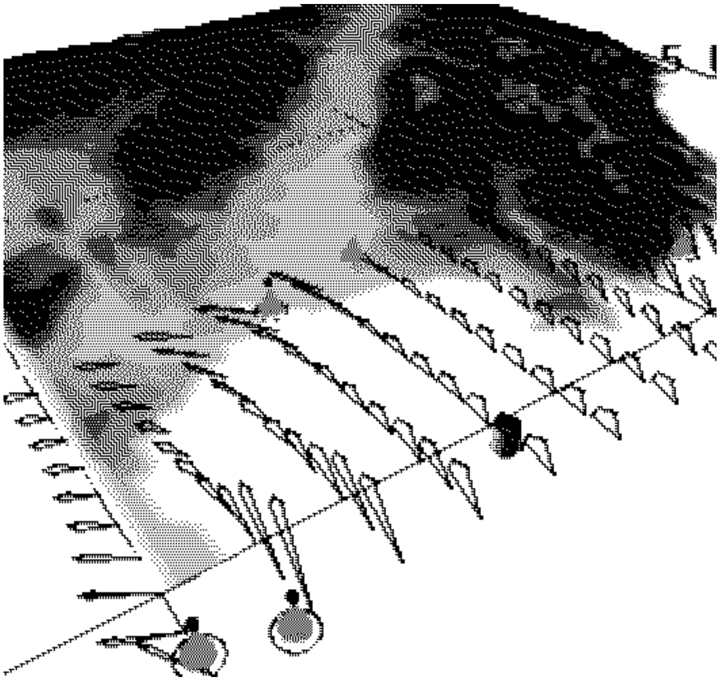

Figure 4.6: Interpolated winds over the Monterey Bay region, on a regular grid with uncertainty angle glyphs. Uncertainty grows with distance from meteorological stations giving a much different impression from the traditional vector glyphs.

5. Systems, Networks, Databases and Schemas

5.1 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. [LMW+95]

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. 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

5.2 Architectural Overview

Networks

REINAS consists of three logical networks: instrument, database, and user networks. See Figure 5.1. 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 are 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

1. 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.
2. 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.
3. Vertical Wind Profilers provide vertical atmospheric profiles. Using radar they produce wind speed and direction estimates.
4. Acoustic Doppler Current Profilers may be used later.

Instrument Node 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, e.g. serial line I/O and data parsing.

Instrument Research Issues

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

5.2.1 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].

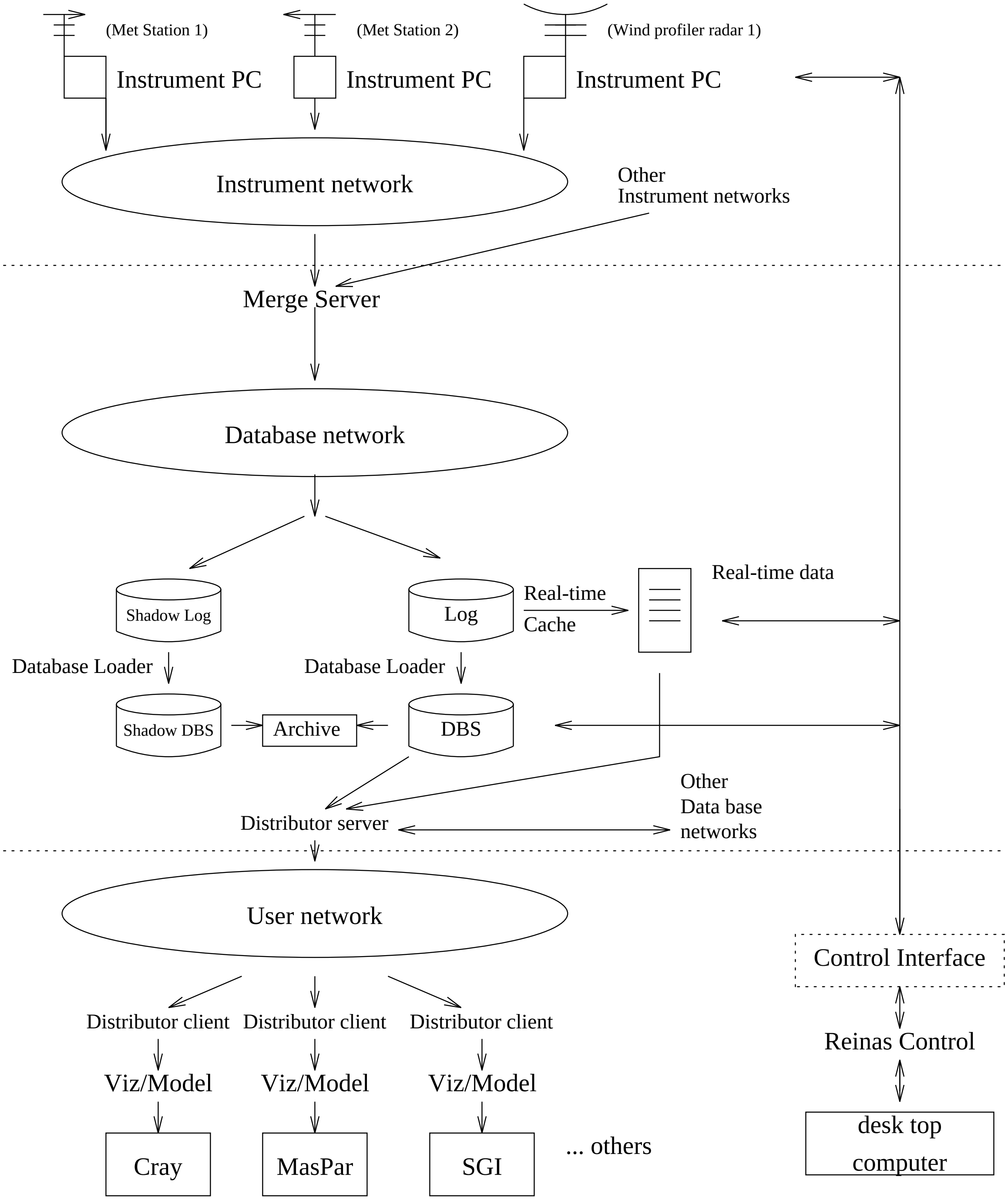

Figure 5.1: Overview of REINAS Logical Structure

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.
- Data Management: Storage of scientific data and metadata in an extended relational database system.

5.2.2 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.

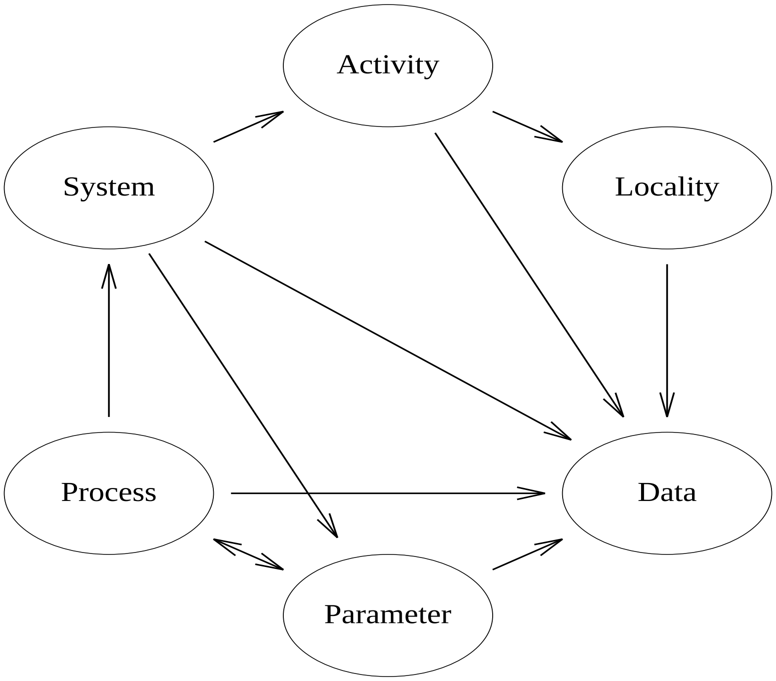

Figure 5.2: Example Realms

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.

5.2.3 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 are 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.

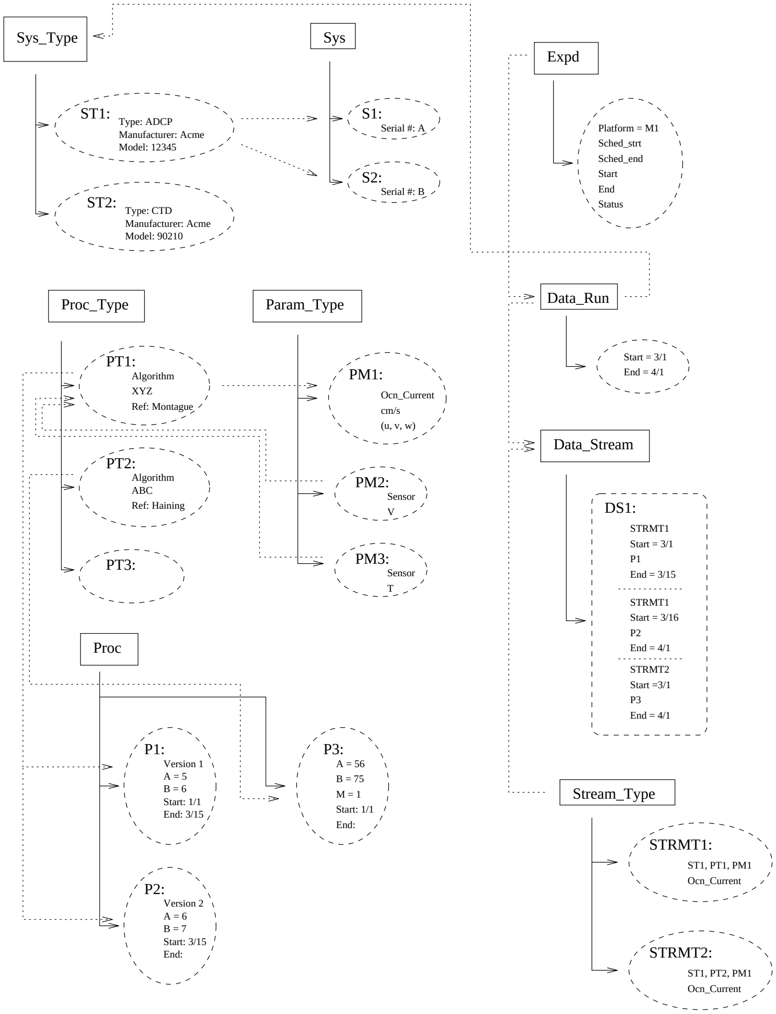

Figure 5.3: An Example Schema

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.

5.2.4 The Data Stream Model

REINAS uses a data stream model in which data are 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.

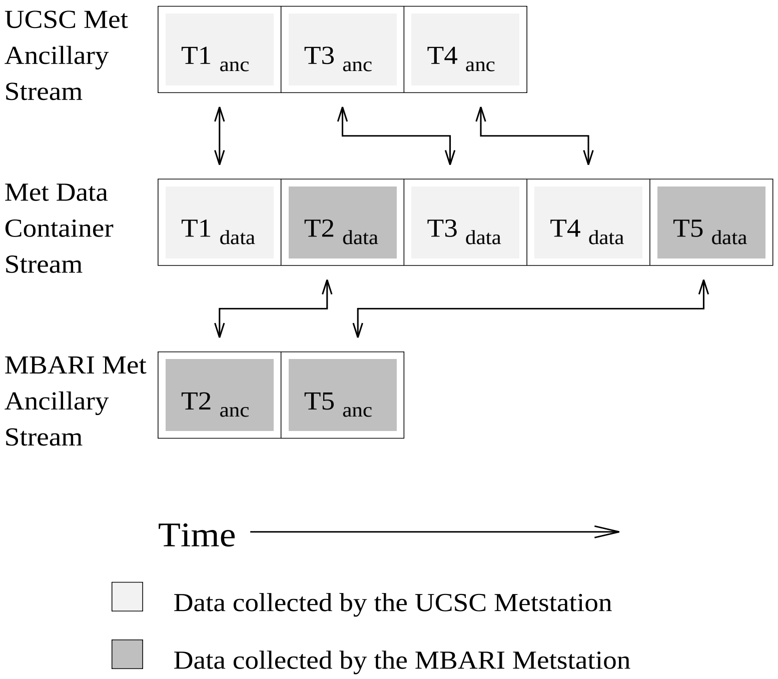

Figure 5.4: The Data Stream Model. Showing how data are handled between two Collection Stations.

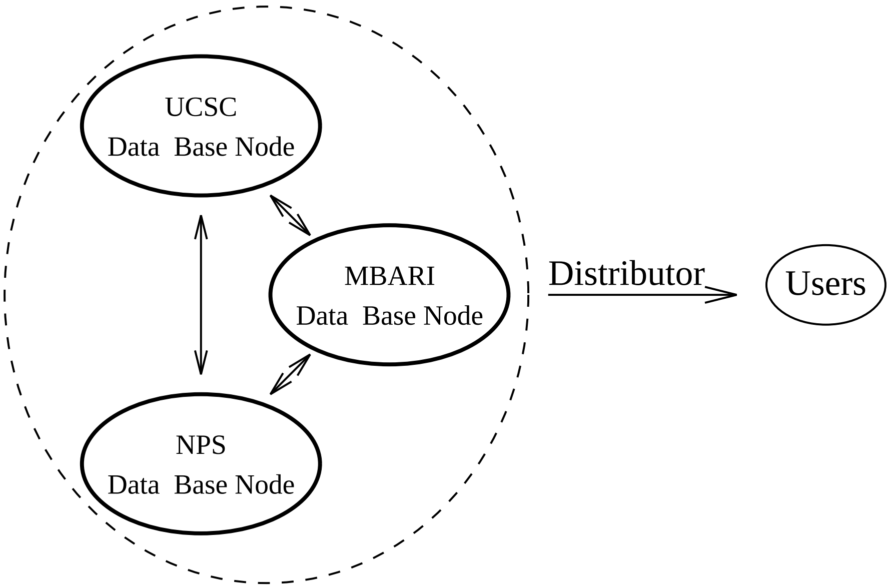

Figure 5.5: Database Subsystem Design. Because several science partners independently collect data and retain ownership of it, the REINAS database subsystem is designed to operate as a distributed, multidatabase that is accessible through a single distributor process.

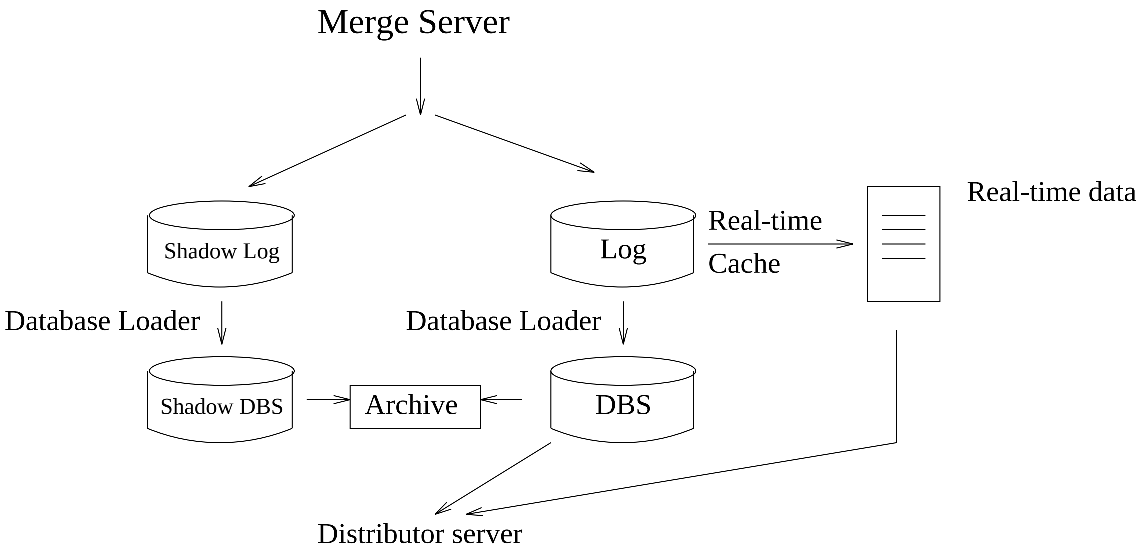

Figure 5.6: Current Database Structure

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 are accessed through a single schema, making the integration of data from existing databases difficult. There is incompatibility of commercial DBMS [CS93].

5.2.5 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].
- 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*.

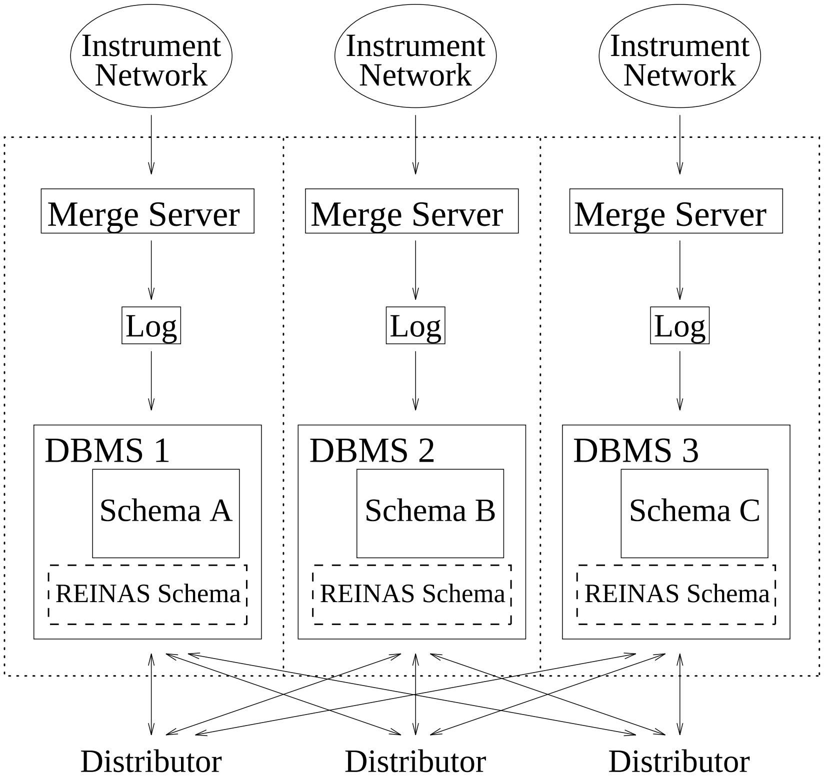

Figure 5.7: Proposed REINAS Multidatabase

- 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].

5.3 Data Management Design Philosophy

5.3.1 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 mode organized for fast update, and another for adequate retrieval of broadly indexed attributes of geophysical observations.

---

[2] This section by Bruce Gritton, MBARI

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.

5.3.2 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).

5.3.3 REINAS Data Management Approach

5.4 Requirements Analysis and Information Architecture

The initial user community served by REINAS consists 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.

5.4.1 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.

5.4.2 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.

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 5.1: Elements of REINAS Data Management Approach

- *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.

5.4.3 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.

5.4.4 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.

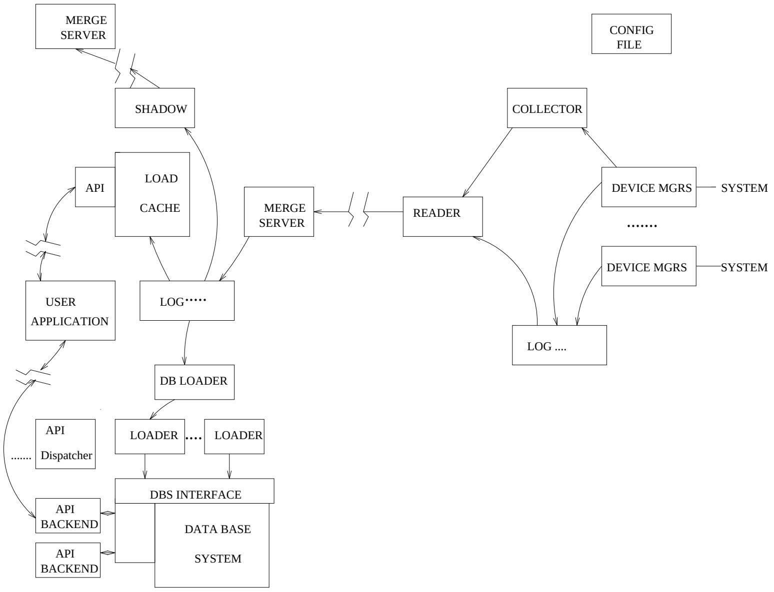

Figure 5.8: REINAS Data Flow, Showing the Role of Device Managers and the Application Program Interface (API)

5.5 System Data Flow Architecture

5.5.1 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 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.

---

[3] This section by Bruce Montague

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.

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.

5.5.2 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.

5.5.3 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 does one know if data has become polluted?

An observation: SQL is not 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.

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 5.2: Databases and Portability

6. Data Compression in REINAS

6.1 Introduction

Data compression [1] 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 following major components of REINAS:

[LPW+95].

1. Compression of raw data into database
    - Data originating at sites connected by radio
    - Satellite data
2. Compression of data produced by scientific visualization
    - AVO (Abstract Visualization Objects)
    - Image of Screen Display
3. Compression of modeling
    - Large amount of data involved
4. Compression of database queries
    - Results of SQL query experiment

In dealing with data compression, the typical scenario is that a compression algorithm is given a data file to compress, and the algorithm delivers a smaller output file containing the same information, or if not the same, a reasonable approximation to the original information. See Figure 6.1. The REINAS project is a prototype, and relative to data compression, the scenario is not always as simple. Part of the learning experience concerns how to integrate compression into the database, which requires an understanding of the uses of the data.

Some of our exploratory work has converted algorithms such as LZSS that compress an input file to an output file (FTF) into a compression algorithm employing an MTM (memory-to-memory) paradigm. The MTM paradigm can be used within an executable, where the compression part is an object module.

Another thrust being considered for the GOES (satellite images) feed, which uses features of the Unix environment, is a compression algorithm that accepts data from *standard input* (stdin) and delivers data to *standard output* (stdout).

Many objects in the database are *blobs* (binary large objects). Sometimes the motive for compression is to reduce the bandwidth required over communications lines, and other times the motivation is to save storage space. The objects subject to compression are varied, so no single compression algorithm can be considered to be the best.

The goal of compression is efficient end-to-end storage and transfer of information among different REINAS components and users. Tradeoffs to consider are the added time for doing compression and decompression, potential space savings, and resulting savings on transmission of compressed information.

---

[1] This chapter by Glen Langdon

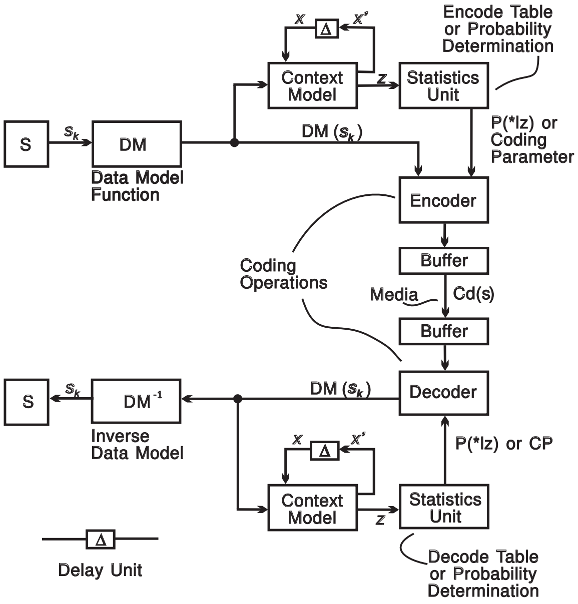

Figure 6.1: Typical Compression Model

The common design goal in employing compression in REINAS is to make it transparent to the users. That is, compression is performed automatically if it is beneficial. There are several areas where compression can potentially benefit the REINAS system. These are discussed in the following sections.

6.2 Compression for Data Archives

Data come from various sources including in-situ instruments, remote sensors, numerical model outputs, etc. Instruments come in a wide variety including simple MET stations which measure wind vector, temperature, rainfall, etc.; wind profilers which measure wind vectors at different heights; CODARs which measure ocean surface current vectors; ADCPs which measure ocean current, salinity, pressure, etc, at different depths; and AVHRR and GOES satellite images that provide periodic snapshots of a larger region. Available computer models of interest to REINAS scientists include the NORAPS [Hod87] model for weather and the Mellor model for ocean [BM87].

REINAS scientists not only access data, but also the *metadata* that determines data lineage and quality. In data assimilation research, the output from weather models are validated against measured data and are also used to drive the model.

With all the variety of data sources and their corresponding metadata, the volume of information that REINAS deals with quickly becomes large. The plan for distributing met data is in "time-series containers" of about 600 KB each. Researchers can request data from particular instruments at a given time granularity. The intent is to send this information in compressed form. The approach for time series is predictive coding. One approach to lossless compression [LH95] has its genesis with one of the authors (Langdon) while on a NASA Summer Faculty Fellowship. The approach is applicable to most one-dimensional digitized waveforms, as well as to two-dimensional waveforms. Moreover, nothing in the approach precludes its future employment at the met stations themselves.

6.3 Instrument Networks

REINAS instruments range from continuous sampling of meteorological data to periodic snapshots. The transmission of data from instruments to the REINAS load paths can also range from continuous to periodic, and in small chunks or large chunks. A specific instance includes instruments mounted on buoys that continually log data to local disk, but can only afford to do transmission at fixed time intervals or on demand due to constraints of battery supply. New routing protocols are being developed for use in the REINAS distributed instrument network. The use of compression in networked communication is especially important because of the low bandwidth available to the packet radio modem technology.

6.4 Multispectral Images from Satellites

The current REINAS plan is to store GOES satellite images covering Monterey Bay. Since the data is so voluminous, instead of storing the images themselves in the data base, REINAS plans to store the names and pointers identifying the location of the compressed images. We are using primarily GOES-West.

GOES images have five bands of interest. From the data source, the GOES images are passed through an archive path that compresses the GOES image in its raw state, losslessly. The current algorithm is *gzip*. However, to be useful, the images must be registered to the latitude-longitude-time scheme used by the data from other sensors feeding the database. Thus, a second part of the task is to "warp" the raw data to the grid, and store the result in HDF format [LAML94], [NCS95].

For satellite images that are multispectral, Hotelling's method of principal components, or EOFs (elementary orthogonal functions), or discrete Karhunen-Loeve transform, is being investigated (G. Ubiergo).

The compression approach for registered images can have some loss since registration is already lossy. However, the original data is archived. The current idea is to segment the multispectral image, and identify the region boundaries. The boundary information may indicate the cloud cover, via segments that hide natural boundaries. We also plan to employ the transform on the regions themselves, and have basis matrices on a region basis.

We plan to use knowledge of the terrain as well. For example, knowing the location of a region, and estimating the cloud cover, it may be possible to develop a library of basis matrices. This would save the need to store them with the image itself. An alternative being considered is gathering expected values for Monterey Bay for a predictive coding approach.

6.5 Video Camera

Another type of instrument being employed in REINAS is the video camera. A submarine is available for video capture of underwater viewing of Monterey Bay. There is also a video camera installed on the roof of a beach front hotel in Santa Cruz, to provide a panoramic view of the Monterey Bay. Since the robot-controlled video camera captures weather phenomena as it happens. The idea applies to watching weather phenomena as it develops, and observe the values of the instruments at the same time, whether the observer is watching it live, or watching a coordinated and realistic "playback" ten years hence.

The idea of a permanent location for the camera suggests building a large image of what the camera is expected to see in a certain direction under various weather conditions. Thus the compression could include a predictive component.

6.6 Instruments in remote locations

Several of the instruments are located where only packet radio is available for the "first hop" on the way to the REINAS data base. Compression algorithms are being investigated for this application. However, the meteorological data itself, on two minute averages, does not seem to present a bandwidth problem at this time [LM93].

6.7 Model Output

This class of data are more predictable and can benefit greatly from compression. Forecast models are typically run daily and produce predictions for different time periods for the following day. The output formats and statistical properties of the data are usually stable. For instance, the NORAPS model runs on a supercomputer and has its output available daily. The model output consists of predictions for the temperature, pressure, wind, and other fields every several hours for the next 24 hours. Currently, files containing the model output are transferred to the REINAS system manually (compress + ftp). Ideally, the feed from the model output into the REINAS database should be automatic and transparent to the user as well as to the person maintaining REINAS data.

6.8 Data Compression for Scientific Visualization

On the Internet each byte transmitted has a cost that ultimately is borne by the user. Thus, compressed data is a desirable option for all users. In addition to reducing the connect fees, effective compression can also reduce latency for low bandwidth clients. In the remainder of this section, we detail several applications of compression for accessing REINAS data over the Internet.

6.9 Collaborative Visualization

The visualization component allows multiple users to access data stored within the databases, allowing simultaneous collaboration among several geographically distributed scientists. Collaboration over the Internet can happen at several levels depending on the willingness of the collaborators to share data, on the processing powers of the workstations, and on the available network bandwidth. Compression at the host offers bandwidth savings, assuming each workstation can compress and decompress. Moreover, if the screen is broadcast to several collaborating users, the bandwidth savings are multiplied.

Visualization options considered are:

1. Transfer the scientific data to be rendered to each graphics workstation, assuming each has a renderer. Changes in the viewing position may involve very little Internet traffic, during a collaboration. However, broadcasting the raw data can be overly expensive.
2. Transfer the graphics primitives, or visualization primitives, to the users and allow them to render the image. Collaborators that have a rendering engine but who cannot directly access the raw data on their own machines can still make requests to the owner's machine to generate visualization primitives for display on their own machines.
3. Transfer a copy of the rendered image. An observer may only have an image decompression program. If the viewing station has the power of a graphics rendering engine, then an 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.

One of the simplest and fastest compression algorithms for images, in terms of the decoder, is Vector Quantization (VQ). An investigation into the theory and practice of VQ was done, and an algorithm was obtained and modified for experimental use in REINAS. Other decompressors studied include JPEG, a simple lossless algorithm called FELICS of Howard and Vitter [HV93], and a combination VQ-based and BTC algorithm called VPIC (Visual Pattern Image Coding) of Alan Bovik and his students [CB92].

As part of REINAS we are also developing part of the database as an image library. Users can obtain a small picture of the image before acquiring all of the data using progressive transmission. There is an experimental client/server program under Xwindows that first transmits the mean value of each 16x16 block of the image, and progresses from there. If the user decides not to continue, a mouse click terminates the transmission.

6.10 Accessing Time-series Containers and other Database Queries

A study was performed on the feasibility of compressing the outcomes of database queries (Pi-Sunyer). The SQL returns a record at a time, so to get statistical correlations, the study adapted the LZSS algorithm provided by Mark Nelson [Nel91]. The algorithm is an LZ77 class algorithm, and maintains a rotating history buffer of the previous bytes sent to the client. Since the history buffer may be larger than each record, both the encoder and decoder remember the state from the previous records sent and received. Thus, a session consisting of a sequence of queries between the same client and server maintains a state that persists between queries. In the simple experiment, the compressed data transferred amounted to about 38 percent of that which would have been transferred over the network in the absence of compression.

7. Communication Protocols for Wireless Networks

7.1 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, and 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 presented the following 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

7.2 Channel Access in REINAS

Carrier Sense Multiple Access (CSMA) and ALOHA protocols have been widely used in the past in the context of packet-radio networks. However, these protocols degrade in a multi-hop network, because of the effect of hidden terminals. Furthermore, they cannot provide any performance guarantees, because each packet contends for the channel.

---

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

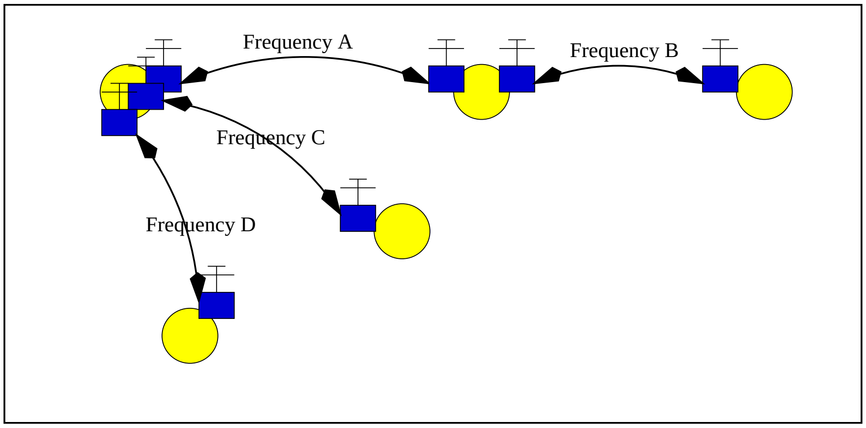

Figure 7.1: Current REINAS Wireless Link Architecture

We have developed a new type of channel access discipline called Floor Acquisition Multiple Access (FAMA) [FGLA95]. FAMA consists of both carrier sensing and a collision-avoidance dialogue between a source and the intended receiver of a packet. Control of the channel (the floor) is assigned dynamically to at most one station at any given time, and this station is guaranteed to be able to transmit one or more data packets to different destinations with no collision with transmissions from other stations. As it is well known, CSMA degrades substantially when the sender cannot sense collisions. Our results indicate that variants of FAMA can avoid this problem and achieve much higher throughput than CSMA with hidden terminals, and comparable or higher throughput otherwise. Higher throughputs can be achieved by increasing the packets sent for each successful floor acquisition.

Figures 7.2 and 7.3 compare the throughput of FAMA protocols with that of non-persistent CSMA in both a low speed network (9600 b/s) and a high speed network (1 Mb/s) using data packets of 400 bytes. We assume a network with a maximum diameter of 10 miles, which gives us a propagation delay of approximately 54μs. FAMA provides similar throughput to the ideal case of CSMA in which a virtual channel is used to send CSMA acknowledgments.

At present, basic FAMA is implemented on the BSDi REINAS system. Implementation is complete and currently being tested and verified.

7.3 Routing Protocols for REINAS

A critical element in the provision of fault tolerance and the ability of a network to scale is the choice of the routing protocol used.[2] 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

---

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

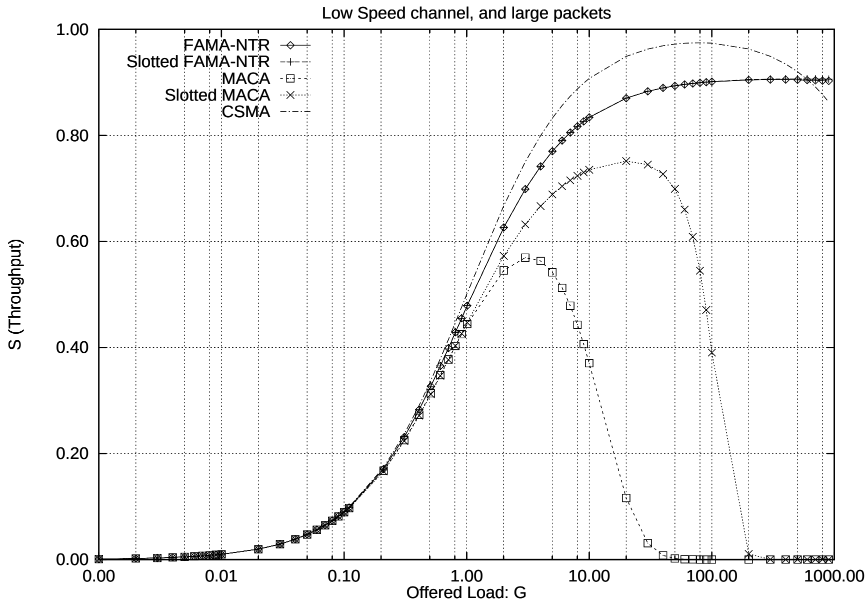

Figure 7.2: Comparison for Low Speed Channel and Large Packets

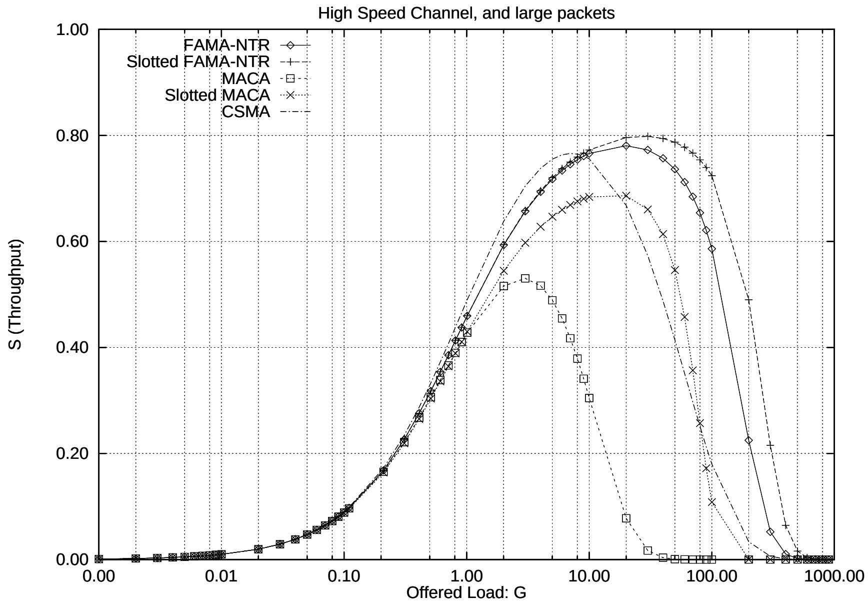

Figure 7.3: Comparison for High Speed Channel and Large Packets

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].

We have developed a new type of routing algorithms, called link-vector algorithms (LVA). An LVA diffuses link-state information selectively based on the distributed computation of preferred paths. We have analyzed the performance of LVA by simulation; our results show that, on the average, an LVA is faster and requires less traffic overhead than the distributed Bellman-Ford algorithm (used in RIP, for example) and any link-state algorithm based on flooding (which is the type of algorithm used in OSPF and IS-IS).

We have also developed the first known loop-free routing algorithm based on vectors of distances and second-to-last hops to destinations. We call this algorithm the loop-free path-finding algorithm (LPA). We have analyzed the performance of LPA by simulation; our results show that LPA has better performance than any link-state algorithm based on flooding (which is the type of algorithm used in OSPF and IS-IS) and the most efficient loop-free routing algorithm previously known (the Diffusing Update Algorithm).

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.

8. REINAS Observations and Models

8.1 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.

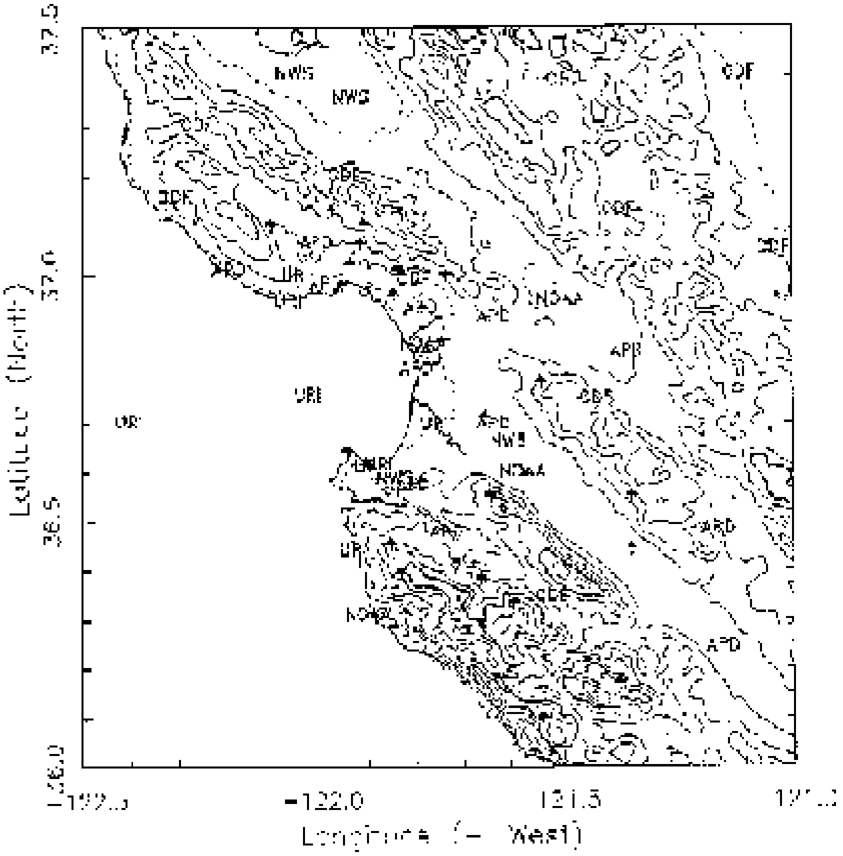

Figure 8.1: Observation Sites in the Vicinity of Monterey Bay. URI = UCSC and MBARI Sites. NWS = National Weather Service. CDF = California Dept. of Forestry. NOAA = Environmental Technology Lab. APD = Air Pollution Control Districts.

- Desired measured surface meteorological fields include:
  - Air Temperature
  - 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

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 contracted with NOAA Environmental Technology Laboratory for deployment of three 915 MHz radar wind profilers and two 2-axis monostatic SODARs (See Chapter 3).

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.

8.2 REINAS Atmospheric Numerical Modeling System

8.2.1 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:

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.

---

[2] Atmospheric Modeling section by Paul Hirschberg, NPS

6. More in-depth scientific investigation e.g. sensitivity studies) of various physical phenomena affecting the REINAS region of interest. [HNS96]

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), when it becomes available.

8.2.2 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 SGIs 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.

8.2.3 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. SGIs
- 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.

8.2.4 Modeling Plans

- 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.I. or multiquadric analysis especially over the inner domain.
- More thorough verification.
- Begin development and testing of 4DDA system.

For the more distant future 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.

8.3 Ocean Modeling

As part of REINAS[3] there is the 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.

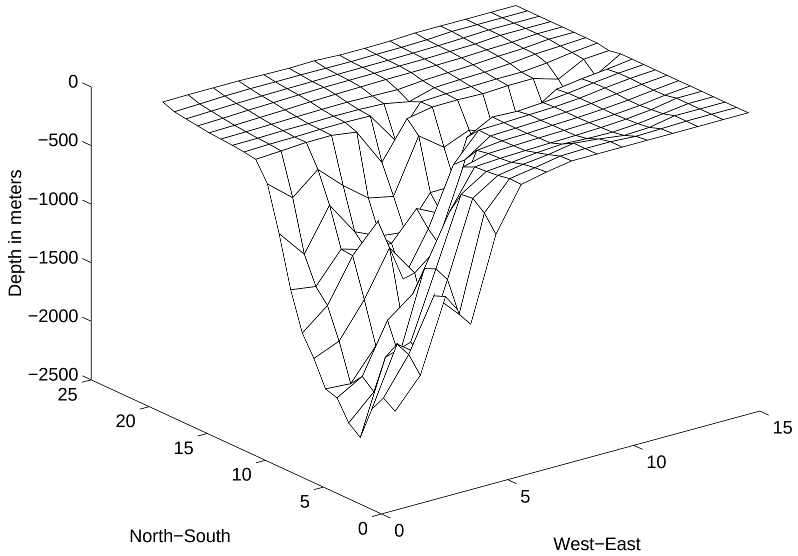

Figure 8.2: Monterey Bay Ocean Model Grid (2km x 2km Spacing)

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 (POM), sometimes 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. The principal attributes of the model are:

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.

---

[3] Ocean Modeling section by Harwood Kolsky, UCSC

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.

After obtaining a copy of the code and documentation from NOAA/GFDL (ftp gfdl.gov), we experimented with it in conjunction with the IBM Data Explorer visualization system and inserted the Monterey Bay bathymetry as boundary conditions, using a 2 km x 2 km grid over the Bay region only. See Figure 8.2. The Monterey Bay canyon causes difficulties in the model because of the very steep canyon walls.

An active local "POM Users Group" has been formed which meets regularly at NPS to carry on research for the West Coast.

8.4 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 have been installed in REINAS to generate fields for visualization. However, simple interpolation does not account for the dynamic balances inherent in the atmosphere.

8.4.1 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.

8.4.2 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.

8.4.3 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.

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.

[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.

[CB92] D. Chen and A. Bovik. Hierarchical visual pattern image coding. *IEEE Transactions on Communications*, 40(4):671–675, April 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, August 1989.

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

[DGLA96] H.-P. Dommel and J.J. Garcia-Luna-Aceves. Floor control for multimedia conferencing and collaboration. *Multimedia Systems (ACM/Springer)*, page (to appear), 1996.

[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.

[Fer95] Daniel M. Fernandez. Remote sensing ocean currents in Monterey Bay with OSCR and CODAR. In *Eastern Pacific Oceans Conference, Stanford Sierra Camp*, September 1995.

[FGLA95] Chane Fullmer and J.J. Garcia-Luna-Aceves. Floor acquisition multiple access for packet-radio networks. In *ACM SIGCOMM 95, Cambridge, MA*, September 1995.

[FP96] Daniel M. Fernandez and Jeffrey D. Paduan. Simultaneous CODAR and OSCR measurements of ocean surface currents in Monterey Bay. In *IGARSS'96 Conference, Lincoln, NE*, 1996.

[FVT95] D. M. Fernandez, J. F. Vesecky, and C. C. Teague. Measurements of upper-ocean surface currents with high-frequency radar. In *Proceedings of the IEEE Fifth Working Conference on Current Measurement, St. Petersburg, FL*, February 1995.

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

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

[HNS96] P.A. Hirschberg, W.A. Nuss, and P. Sirayanone. Numerical simulation sensitivity of the 9-11 June 1994 California southerly surge to initial conditions. In *AMS Conference on Coastal Oceanic and Atmospheric Prediction, Atlanta, GA*, June 1996.

[Hod87] R. M. Hodur. Evaluation of a regional model with an update cycle. *Monthly Weather Review*, 115(11):2707–2718, November 1987.

[HV93] P. Howard and J. Vitter. Fast and efficient lossless image compression. In *DCC '93. Data Compression Conference, IEEE Computer Society, J. Storer and M. Cohn, editors*, pages 351–360, 1993.

[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.

[LH95] G. Langdon Jr. and C. Haidinyak. Experiments with lossless and virtually lossless image compression algorithms. In *Proceedings SPIE, Still Image Compression, M. Rabbani et al., editors*, volume 2418, pages 21–27, 1995.

[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.

[LMW$^+$95] D.D.E. Long, P.E. Mantey, C. M. Wittenbrink, T.R. Haining, and B.R. Montague. REINAS the real-time environmental information network and analysis system. In *Proceedings of COMPCON, San Francisco, CA*, pages 482–487, March 1995.

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

[LPW$^+$95] Glen G. Langdon, Alex Pang, Craig M. Wittenbrink, Eric Rosen, William Macy, Bruce Montague, Carles Pi-Sunyer, James Spring, David Kulp, Dean Long, and Patrick Mantey. Compression research on the reinas project. In *Workshop on Science Information Management and Data Compression*, NASA Goddard Space Flight Center, Greenbelt, MD, October 1995.

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

[Man94] P.E. Mantey. A view of Visualization: Its origins, development and new directions. In *SPIE Proceedings on Visual Data Exploration and Analysis*, volume 2178, pages 2–11. SPIE, February 1994.

[Mel95] D.C. Melton. Remote Sensing and Validation of surface currents from HF radar. Technical report, Naval Postgraduate School, M.S. Thesis, 1995.

[MLGL$^+$94a] P.E. Mantey, D.D.E. Long, J.J. Garcia-Luna, A.T. Pang, H.G. Kolsky, et al. REINAS: Phase 4.1 - Experimentation. Report UCSC-CRL-94-43, Univ of California, Santa Cruz, Santa Cruz, CA 95064, October 1994.

[MLGL$^+$94b] 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.

[NCS95] NCSA. *Hierarchical data format (hdf) specification.* ftp.ncsa.uiuc.edu, 1995.

[Nel91] M. Nelson. *The data compression book : featuring fast, efficient data compression techniques in C.* M&T Books, Redwood City, CA, 1991.

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

[PA94] A. Pang and N. Alper. Mix and Match: A construction kit for visualization. In *Visualization'94 Proceedings*, pages 302–309, 1994.

[PA95] A. Pang and N. Alper. Bump mapped vector fields. In *SPIE Visual Data Exploration and Analysis*, pages 78–86, 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*, volume 29(2), pages 33–36, 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*, pages 317–326, February 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$^+$95] C. Wittenbrink, E. Saxon, J. Furman, A. Pang, and S. Lodha. Glyphs for visualizing uncertainty in environmental vector fields. In *SPIE Conference: Visual Data Exploration and Analysis*, February 1995.

Index