

Mining Geophysical Data for Knowledge

**Edmond Mesrobian, Richard Muntz, Eddie Shek, Siliva Nittel,
Mark La Rouche, Marc Krigger, Carlos Mechoso, John Farrara, UCLA
Paul Stolorz, California Institute of Technology
Hisashi Nakamura, University of Tokyo**

EXPLORATORY DATA MINING AND analysis for scientific hypothesis testing or phenomenon detection is an iterative, successive-refinement process. Scientists apply a preliminary model on the data and then use the outcome of a series of experiments to refine the model and methodology. They repeat this process until they either drop the hypothesis or refine it into one that is consistent with the collected data.

For such a research approach to be practical, scientists need a powerful system that supports

- easy formulation and execution of powerful queries and discriminant functions against the database,
- a natural representation of the relationships of the scientific domain of interest (for example, in the natural domains of space and time but possibly in the frequency domain, as well), and
- efficient execution of these queries without requiring the scientists to be aware of the storage structures and processing strategies involved.

We are developing Oasis (*open architecture scientific information system*) to be such a system. In this article, we explain how scientists can use this flexible, extensible, and seamless computing environment for scien-

OASIS IS A FLEXIBLE, EXTENSIBLE, AND SEAMLESS ENVIRONMENT FOR SCIENTIFIC DATA ANALYSIS, KNOWLEDGE DISCOVERY, VISUALIZATION, AND COLLABORATION. THE AUTHORS DESCRIBE HOW OASIS CAN HELP EXPLORE DATA ANALYSIS AND DATA MINING OF SPATIO-TEMPORAL PHENOMENA FROM LARGE GEOPHYSICAL DATA SETS.

tific data analysis, knowledge discovery, visualization, and collaboration.¹

How Oasis can help

Consider a scientist charged with determining whether the observed change in an important atmospheric trace gas at a particular spatial location is due primarily to dynamic processes (that is, transports from place to place) or to chemical processes (for example, losses caused by a catalytic photochemical process). One way to quantify the effects of transport processes is to compare changes in the target gas (ozone) at the same spatial location (same pressure level, latitude, and longitude) with those in a long-lived trace gas such as nitrogen oxide. This study might require retrieving large volumes of

data from (possibly distributed) repositories; reformatting the data; locally managing that data; developing analysis algorithms; and storing, visualizing, and interpreting the results (over several iterations).

These subtasks typically require the scientist to battle with a plethora of computer systems, programs, protocols, and data formats. Heterogeneity is a fact of life in dealing with computers. Hardware vendors continue to develop diverse platforms. Operating systems, file systems, and network software continue to proliferate. And vendors and research institutions continue to develop numerous software tools, each of which satisfies part of what the users need. Ironically, the availability of such a range of hardware and software creates a major problem. More often than not, even a single scientific team uses a diverse set of

platforms and tools. This diversity usually translates into unanticipated time-consuming and costly integration problems. The problem is even more severe when teams try to share information or code.

The players might change and today's state-of-the-art systems might become tomorrow's legacy systems, but hardware and software heterogeneity are here to stay. That's why we're developing Oasis. This system will provide application developers and end users (scientists) the logical abstraction that the computing environment is simply a set of objects of various types (*classes*), as Figure 1 illustrates.

Central to the architecture are the scientific, spatio-temporal objects accessed by applications via a distributed object-management framework. An object class defines a type of object in terms of attributes (*variables*) and operations (*methods*). In Oasis, the scientist investigating the influence of dynamic processes on ozone changes would use the Oasis catalog browser to locate the ozone and nitrogen oxide data set objects (collections of multidimensional, cellularly gridded coverages). The reference found in this way (which can point to an object located somewhere on the Internet) can be used by a visualization package such as LinkWinds² or by a scientific data-mining tool such as the Conquest Parallel Query Execution System (see sidebar "Data-mining techniques").³ The selected objects encapsulate the code needed to retrieve subsets of data possibly stored in a file system or database-management system, or needed to perform operations on the data. Objects in the multidimensional, cellularly gridded data set class (for example, UCLA AGCM data) all have a common interface—that is, they support a common set of operations. A *handle*, or object reference, refers to the object and invokes these operations uniformly, regardless of the object's location or the implementation details. So the point here is that the scientist using Oasis does *not* have to worry about intricate yet meaningless information on data storage and handling, and can remain focused on the task at hand.

System architecture

The major design goals of Oasis that contribute to its usefulness as an environment for data mining (especially in a large-scale geo-scientific information system) are to

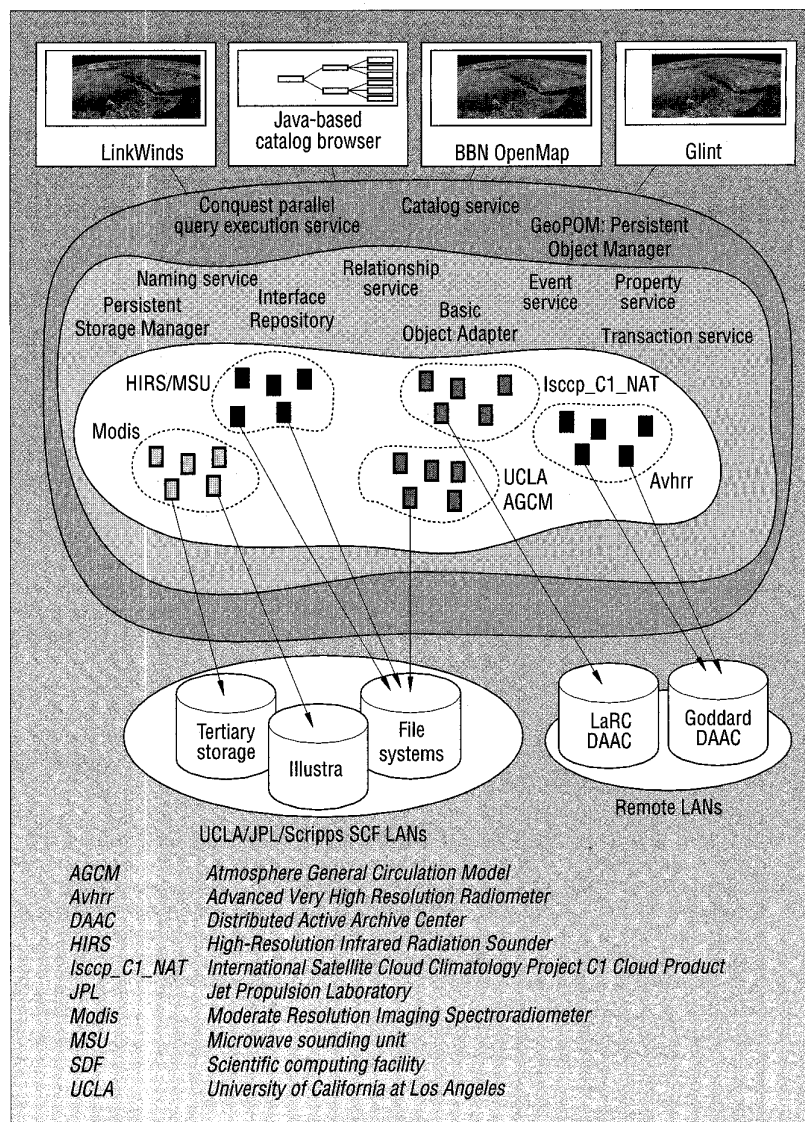


Figure 1. Conceptual architecture of Oasis. Datasets such as UCLA AGCM are modeled as collections of objects whose storage formats and locations are hidden from applications (for example, LinkWinds) accessing the objects' contents. Surrounding these objects are layers of services provided by distributed-object-management systems based on CORBA and by Oasis.

- Develop an object hierarchy to serve as a common foundation on which to build next-generation scientific applications and information repositories while still embracing legacy systems.
- Provide a query facility to efficiently process complex scientific queries (data-mining algorithms) involving computationally expensive calculations on large data sets (tens of gigabytes to terabytes).
- Provide the basis for efficiently, systematically, and uniformly accessing a wide variety of data repositories, including file systems, relational database systems, and object-oriented databases.
- Provide seamless integration of data

analysis and visualization with data management.

- Enable the reuse of commercial and public-domain applications that provide valuable analysis and visualization services.
- Provide a system easily customized to the needs of a particular scientist or group of scientists.

To meet these goals, we are exploiting the emerging distributed-object-management system technology as promoted by the Object Management Group's (OMG) Common Object Request Broker Architecture standard.⁴ CORBA-compliant software is now available from many vendors. Moreover,

Data-mining techniques

Although data mining has more and more often been touted as a technique to extract additional value out of existing information systems, a precise definition does not exist. We define data mining as a process to extract content information from large (and often unstructured) data sets to compactly summarize each data set. Content information comes in many forms, and we can apply it in many ways. For example, content-based indexing techniques take advantage of feature extraction algorithms to succinctly and compactly summarize the characteristics of large data sets so that interesting subsets can be easily located and retrieved based on high-level descriptions of user interests. On the other hand, we can use unsupervised and supervised learning techniques (or knowledge discovery) to interpret the information contained in large data sets. An unsupervised learning algorithm aims to classify and identify interesting patterns in data, using techniques such as singular value decomposition (SVD) or projection pursuit, when ground truth exemplars (input-output pairs) are not available. Alternatively, we can use supervised learning algorithms when robust training sets, characterizing the behavior of a particular system, are available. In both cases, we can use feature analysis and clustering techniques to reduce the parameter-space dimensionality.

Phenomena-based sequence data mining and correlation

A data-mining technique that has attracted a lot of interest, especially from large corporations, is sequential pattern extraction.⁴ Instead of addressing data mining from a traditional knowledge-discovery perspective by attempting to discover interesting patterns in sequences of events, sequential pattern extraction aims to efficiently sieve through large volumes of data and locate sets of events exhibiting some predefined correlation relationship. For example, financial analysts perform stock market trend analysis by correlating the price movements of selected stocks.

Geoscientists face a conceptually similar problem when they want to extract patterns representing natural phenomena that evolve over time. For instance, consider a geoscientific phenomenon such as a cyclonic storm (usually labeled "L" in new paper weather maps) and the storm's trajectory (the cyclone track). A cyclone's center can be defined as the location of a deep local minimum at sea-level pressure. To obtain a cyclone track, geoscientists must link together a time series of locations with local minima in a sequence of sea-level pressure maps (possibly based on the cyclone's expected velocity) to form a higher-level and more informative temporal pattern.

The common ground between sequential pattern extraction in business applications and geoscientific applications is that they are both correlation problems where a heuristic links isolated events (stock price movement, sea-level pressure minima) to form more sophisticated and meaningful features or phenomena (stock market trends, cyclone tracks). Some properties of geophysical phenomena, however, make geoscientific sequence correlation a much more challenging problem. In a business application, the events from which patterns are extracted are generally simple alphanumeric data items. In a geoscientific application, on the other hand, the events often derive from more basic data sets through a computationally expensive algorithm. This significantly increases the need for high performance in geoscientific sequence-mining applications. Furthermore, although the correlation rules (descriptions of the sequential patterns) for business applications are generally simple patterns formed by lists of sets of alphanumeric data items, those for feature extraction in geoscientific applications often depend on the semantics and definition of the phenomenon of interest. In addition, scientists often do not agree on the best descriptors of a natural phenomenon. Two examples that illustrate these properties of geoscientific feature extraction applications are cyclone tracking and the detection of events of enhanced propagation of wave energy from the lowest regions in the atmosphere (the troposphere) to the layer immediately above it (the stratosphere).

El Niño and cyclone storm

Let us focus on the problem of finding links between the El Niño/Southern Oscillation phenomenon (a quasi-periodic warming of the tropical Pacific Ocean that typically occurs around Christmas along the coasts of Ecuador and Peru) and changes in weather patterns around the world, represented by the frequency or strength of cyclones and their tracks. To address this problem, a scientist must design algorithms for recognizing occurrences of ENSO and cyclones in data archives. These archives can provide 4D (longitude, latitude, height, and time) multidimensional cell-by-cell gridded data containing, for each point on the grid, the value of various geophysical quantities (such as temperature and humidity). However, some archives (such as satellite retrievals) provide data that are often in irregularly sized grids, have missing and noisy values, and so on. (Methods for addressing issues of data quality control are beyond the scope of this article.)

Figure A presents a dataflow description of an algorithm for detecting cyclones. First, at each time instance during the period of interest, the algorithm examines the array of sea-level pressure values around the globe. (A scientist can work with a higher-resolution data set by fitting a bicubic spline function to produce a smoother spatio-temporal variation in

the *on-the-wire* protocol standard for inter-ORB communication, Internet Inter-ORB Protocol (IIOP), supports interoperability between different vendors' implementations of CORBA on heterogeneous platforms. Basically, CORBA-compliant systems provide a framework for distributed object programming in which an object's location, platform, and implementation are hidden from applications.

At the heart of the system lies our object hierarchy, which acts as the conceptual and semantic glue linking our applications with scientific data. Over the past two years, we have been actively involved with the OpenGIS Consortium's effort to define an Open Geodata Interoperability Specification for GIS data.⁵ The core of this specification is

the Open Geodata Model (OGM), a common means for representing the Earth and Earth phenomena mathematically and conceptually.

Surrounding this core is OMG's Object Request Broker (ORB), Basic Object Adaptor (BOA), and Interface Repository (IFR). The next layer contains a host of Common Object Services (our Oasis development effort is based on SunSoft's CORBA-compliant NEO software). The latter two layers provide the physical machinery and services necessary to develop a distributed, object-oriented computing environment. The final layer contains the query, storage, and catalog services to support data-mining activities. Applications operating atop this distributed computing substrate include a Java-based catalog browser, Glint, OpenMap, and Link-

Winds. Glint is a data-analysis tool developed at the Scripps Oceanographic Institute, and LinkWinds is a data-visualization tool developed at the Jet Propulsion Laboratory. OpenMap is a networked GIS visualization tool developed by BBN.

Conquest parallel query execution service.

Scientists want a system that's easy to use and fast—doesn't everyone? We can describe what this means to the scientists with whom we've worked, as follows. Taking the conceptually simplest requirement first, *fast* really means interactive at some reasonable scale. Because a system often must handle tens of gigabytes of data, scientists don't expect response times in the subsecond range. On the other hand, more than an hour

the locations of pressure minima.) Deep local minima in the sea-level pressure exist at each time step. For example, to retain only salient minima, the magnitude of a given sea-level pressure value at a particular spatial location must be sufficiently lower (for instance, 4 millibars) than pressure values in adjacent areas. The algorithm then uses gradients in the interpolated surface to locate the centers of the minima extracted in the first step. Next, the algorithm groups minima at successive time steps into tracks according to a heuristic method. In Oasis, our algorithm uses wind velocity and direction in the lower troposphere (for instance, corresponding to a pressure of 500 millibars) to approximate a cyclone's trajectory.

During the query formulation stage, the scientists supply values for the minima basin strength, the pressure level at which wind information will be used, the minimum lifetime for a cyclone, and so on. Next, they execute the query on a candidate data set and compare the results produced by the algorithm with those that they would have manually produced. This comparison might lead to query refinements, by either providing new values for operator arguments or introducing additional processing steps (operators) to the query expression. This iterative process of converging on an algorithm that worked well emphasizes the need for a system that allows quick and easy specification of the algorithm and, when the algorithm is settled, the capability of reasonably fast execution for massive data sets.

Upward propagation of planetary waves

Many data-mining applications start with variables or variable tuples

that represent discrete attributes of interesting entities. Locating phenomena is often a necessary first step in geoscientific studies; it reduces the number of variables that must be analyzed during the classification step. It also helps detect phenomena such as cyclones and determine the attributes of these phenomena, and this information is semantically richer than raw parameters such as wind velocity.

Scientists can use this algorithmic approach, for example, to detect episodes of enhanced upward propagation of planetary-scale waves from the troposphere to the stratosphere, which can strongly affect the structure of the circulation in that important region in the atmosphere. The propagation characteristics depend on the distribution of the wind in latitude and height, as well as the amplitude and phase of the waves in the troposphere. Occasionally, the rapid growth and upward propagation of waves during winter in the Northern Hemisphere can lead to a reversal of the high-latitude stratospheric wind from westerly (west to east) to easterly. The weaker upward propagation of the planetary waves in the Southern Hemisphere leads to stronger westerly winds than in the Northern Hemisphere. This contributes to the formation of suitable conditions for the development of the ozone hole, each spring, over Antarctica, but no such ozone hole develops over the Arctic.

To detect upward propagation of wave energy into the stratosphere, we might first measure the phase difference, at a given latitude, of a particular Fourier component of the tropospheric waves (for example, Zonal Wave Number 1, the wave with the longest wavelength) between two pressure levels in the upper troposphere (for example, 250-millibar

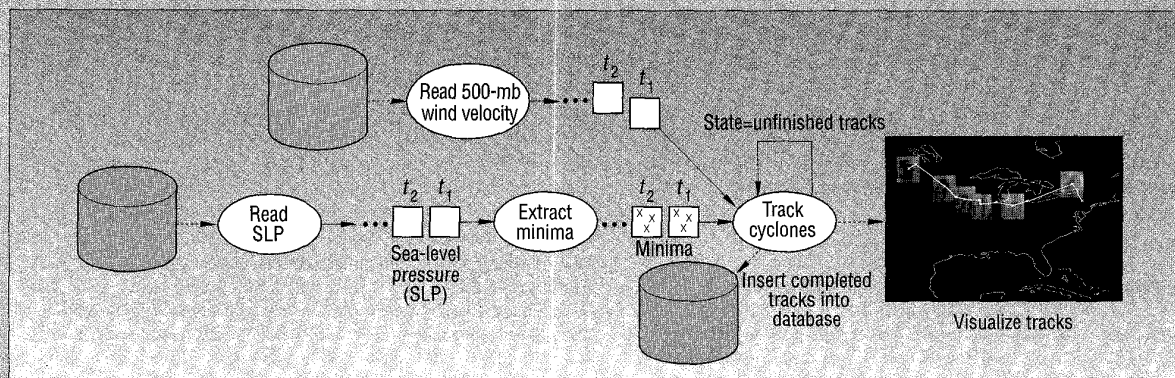


Figure A. A cyclone-tracking algorithm.

or two is often not acceptable. Part of the requirement for performance comes from the need to pose a question (in the form of a query), get the response, examine that response (for example, visually), and then refine and rerun the query. Such interaction with the scientist during the process is a hallmark of this application area and makes high-performance demands on the system.

Although technology is producing mass-storage systems with higher processing and bandwidth capacities at lower costs, the appetite for high performance is never satiated. For interactive response to the scientists' queries, parallelism is still desirable to adequately reduce waiting time. The Conquest Parallel Query Processing System, Oasis's query service, exploits parallelism across het-

erogeneous environments consisting of workstation farms and massively parallel processors to efficiently process complex scientific queries involving computationally expensive calculations on large data sets. The system achieves this by supporting automatic query optimization and parallelization, as well as various inter- and intra-operator parallelisms (for example, pipelining, partitioning, and multicasting), in the query execution environment. In addition, Conquest provides an extensible framework where scientists can easily implement and combine data processing and analysis operators to form complex geoscientific queries.

Conquest field data model and algebraic language. We have developed a data model and

an algebraic language designed to handle geoscientific data that differ significantly from traditional data models and their languages. Scientific data often comes in the form of data fields where data values are at points of a multidimensional spatio-temporal index space. Generally, we can consider the coordinates of the index space as independent variables, and the values associated with points in the index space as dependent variables.

We form a *field space* by attaching a variable from a domain V , called the *variable space*, to each point in an independent space, called the *index space*. The index space contains the set of points with which a variable value can be associated. We construct a *field* by assigning a value in the variable space to each point in a subspace of the index space,

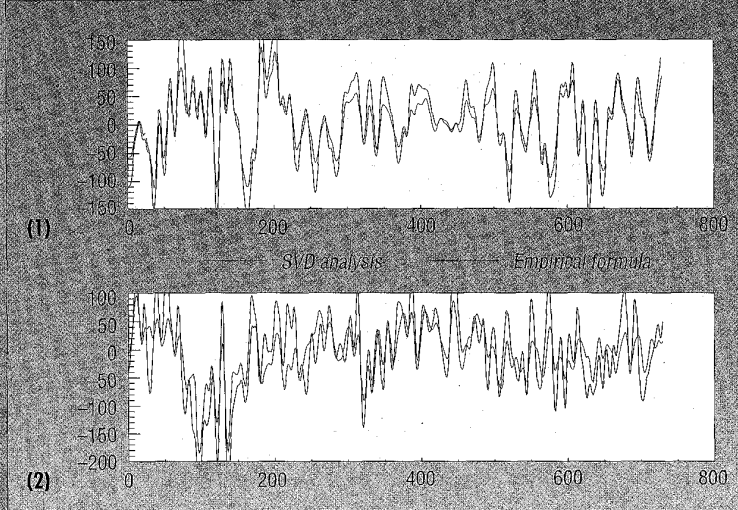


Figure 8. In each plot, a time series of blocking conditions calculated using a mathematical formula derived from empirical data is superimposed with a time series calculated from principal components extracted via singular-value decomposition (SVD) analysis. The horizontal axis represents one year (730 snapshots taken at 12-hour intervals). The superpositions illustrate that the SVD analysis was able to capture the underlying structure of the blocking signatures. (1) Alaska blocking; (2) European blocking.

and 500-mb levels). Next we locate waves of sufficient strength (amplitude) at two neighboring pressure levels in the upper troposphere (for example, 250-mb and 500-mb levels) by computing the first Fourier coefficient of data values, corresponding to the height above sea level required to reach a given constant pressure surface. By monitoring, at a given location, phase differences between neighboring pressure levels, we can locate periods of persistent upward propagation of wave energy.

Principal-components analysis

Another common method scientists use to better understand the variability of weather patterns is to look for events in which anomalous patterns (deviations from the mean value) have unusual amplitude or

deviation. The scientist usually identifies these events by selecting a parameter of the flow, computing its temporal history, smoothing the time sequence with a low-pass filter, computing the standard deviation, and looking for periods where the filtered value differs from the mean value by one or more standard deviations. With simple variables such as temperature at a given grid point or an aggregate value over a spatial-temporal window (for example, the average temperature over the North Atlantic in the winter months), this is a simple, albeit very I/O-intensive, computation. Several variations of this basic idea, however, can become very complex.

Blocking conditions represent a well-known example of a phenomenon where significance is measured in terms of temporal duration or frequency of occurrence. During a blocking event, the sea-level pressure exhibits highs and lows that persist for relatively long periods. For instance, during an Alaskan blocking condition, a persistent high-pressure area develops in the winter months in the vicinity of the Gulf of Alaska—an area bounded by (27° N, 164° W) and (65° N, 160° W)—and can relate to a sequence of storms into Southern California.

Geoscientists might search for such patterns, for example, by exploiting SVD ideas. SVD, also known as principal-components analysis, is a very popular data-mining technique with an extensive history in atmospheric modeling and analysis. It typically begins with the selection of a region containing several anomaly time series of an important variable—for example, (27° N, 164° W) and (65° N, 160° W). The time series at these various locations might, for example, combine with a different coefficient for each time series, to form a new time series representing a spatio-temporal pattern over the whole region. A suitable time series shows persistently strong behavior above a given threshold, which can be interpreted as a blocking condition.

SVD is a straightforward, automatic way to generate suitable time-series combinations. Starting with a small number of grid points that coincide with locations where blocking tends to occur, combinations generated automatically by the SVD methods indicate blocking behav-

called the *sampled index space*. We don't assign variable values to all points in the index space because the index space can be infinite and the finiteness of storage necessitates storing only values for a sampled set of points.

The basic operators supported in our initial prototype fall roughly into these classes:

- **Set-oriented operators.** We define selection, projection, Cartesian product, union, intersection, difference, and join operators similar to their counterparts in relational algebra. Although the logical schema for the result of these operators is well-defined, the resulting field often does not inherit the semantic properties of the inputs. For example, selecting cells

in a field on the basis of their variable values (cells in a regularly gridded sea-level pressure field that recorded a parameter value greater than 980 millibars) generally returns a field whose cell coverage is unstructured.

- **Sequence-oriented operators.** Many geoscientific data-mining applications involve studying the change of time-varying parameters. For example, given a set of cyclone track fields represented as time series of polygonal cyclone extents, we might want to find all cyclone trajectories whose spatial extent shrinks for three consecutive days. So we introduce several sequence-oriented operators. Each operator generates fields by sequentially consuming cell records from its input

fields, performing the appropriate operation, modifying its own internal state in the process, and then outputting the resultant fields.

- **Grouping operators that collect related cell records in a field for further processing.** Data-analysis applications often involve the generation of aggregate information on collections of related data from a field. Several tuple operators collect subfields containing related cell records.
- **Space-conversion operators that derive new fields with different coordinate and variable spaces, as well as new coverages.** We often must reconcile the differences in the coordinate and variable spaces of heterogeneous fields before we can meaningfully compare and correlate

ior at roughly the correct times and locations. This encouraging result justifies the extension of the approach to include a search over the various possible physical locations of each time series to obtain the strongest principal-components signal. Figure B illustrates the use of SVD to detect two well-known blocking conditions—Alaska blocking and European blocking—originally discovered using empirical data. Figure B1 superposes the signature of the Alaska blocking condition at time t and year y computed using the empirically derived equation

$$AB(t, y) = \frac{1}{2} (\hat{Z}_{500}(27^\circ\text{N}, 164^\circ\text{W}, t, y) - \hat{Z}_{500}(63^\circ\text{N}, 160^\circ\text{W}, t, y))$$

where each \hat{Z}_{500} is a geopotential anomaly, with the principal components computed using SVD analysis. Figure B2 presents the results for the European blocking condition.

Hierarchical cluster analysis

In a more general context, the enormous dimensionality of the spatio-temporal features characteristic of geophysical data sets makes the finding of regularities and clusters in the data highly problematic: the well-known curse of dimensionality is at work here. One method of reducing the dimensionality of the large images while retaining the structure of important regularities is to apply cluster analysis to group images together according to shared spatial features. Hierarchical clustering is a classic mechanism for accomplishing this.

The algorithm seeks to cluster together many frames of a geophysical field of interest generated over time. It builds clusters recursively by starting with each of n frames as an individual cluster. First, the algorithm computes a Euclidean pointwise distance d_{pq} between every pair of images p and q . We define the sum of all such distances as the error of clustering:

$$\text{Error} = \sum_{\text{clusters } C} \left(\sum_{p, q \in C} d_{pq} \right)$$

At each step, the algorithm chooses two clusters to merge—namely, the cluster pair that minimally increases the error. The algorithm updates a simple mean image at each step, for each cluster, for use in future computations of the total error. We can use the resulting tree structure to reduce the data dimensionality by identifying *cluster images* containing most of the important information. Many of the

insights obtained correspond to similar observations that can be made on the basis of a somewhat more elaborate SVD analysis.

The hierarchical technique outlined here represents one broad class of approaches to clustering. The other class consists of grouping algorithms that assign membership of each sample point to one or more categories from the very beginning of these algorithms. The classic method in this context is K -means clustering.³ An algorithm assigns each sample point membership in only one class, on the basis of that point's closeness to each K center scattered throughout the sample space. As it proceeds, the algorithm iteratively adjusts these centers' positions to arrive at the best clustering.

A more recent development in the same spirit uses mixture models, where an algorithm assigns each sample point a weighted membership in all the possible K classes. Iterative methods then assign the various weights appropriately. We can interpret the iterations as the maximization of an overall function of the sample points and weights known as the objective function.⁴ It turns out that this objective function often has more than one minimum, which in days gone by was a large problem because of the paucity of computational power. However, the increasing availability of raw CPU power of modern workstations has substantially lessened (though not eliminated) this difficulty, letting scientists search many different minima to locate the best one. For this reason, computationally intensive techniques such as this have become common in the clustering domain. Although they do not guarantee optimal answers, good ones are achievable with ease.

References

1. R. Agrawal and R. Srikant, "Mining Sequential Patterns," *Proc. 11th Int'l Conf. Data Engineering*, IEEE Computer Society Press, Los Alamitos, Calif., 1995, pp. 3–14.
2. E. Mesrobian et al., "Extracting Spatio-Temporal Patterns from Geoscience Datasets," *Proc. IEEE Workshop Visualization and Machine Vision*, IEEE CS Press, 1994, pp. 92–103.
3. R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.
4. A. Gelman et al., *Bayesian Data Analysis*, Chapman & Hall, London, 1995.

the fields. Space-conversion operators let the application of arbitrary (possibly user-defined) functions change the representation of a field so that we can meaningfully compare and correlate differences between data fields from different sources.

Query management, optimization, and execution. Conquest's major components are the query manager, the query optimizer, and the parallel query execution server. The query manager receives a query and sends it to the rule-based query optimizer, which optimizes, parallelizes, and transforms the algebraic expression into a parallel query execution plan. The plan then proceeds to the query execution server for evaluation (see Figure 2).

Meanwhile, the query manager connects to the visualization manager and prepares it for the data stream that the query execution server will send it in response to the query. The design of the query execution server is based on the Volcano extensible query execution engine.⁶ Conquest extends Volcano with a scientific data model that encompasses relational data, scientific data fields, and multi-dimensional array data; and an algebra that supports geoscientific queries. We have implemented generic algebraic operators in this data model, as well as application-specific operators, to support scientific studies.

We originally implemented Conquest on massively parallel supercomputing platforms (for example, the IBM SP1 and SP2 and the Intel Paragon) and on workstation

farms using the portable message-passing library Parallel Virtual Machine as the interprocess communication mechanism. Although the core Oasis services let users access heterogeneous distributed objects without regard to their underlying storage and representation, these services do not immediately support parallel processing of data retrieved from these objects. Therefore, we are reimplementing Conquest as an Oasis service to deliver automatic optimization, parallelization, and parallel execution for complex geoscientific queries within the distributed object-management framework presented by Oasis. The Oasis-based Conquest parallel query execution service takes advantage of the capabilities of the underlying distributed object-management system

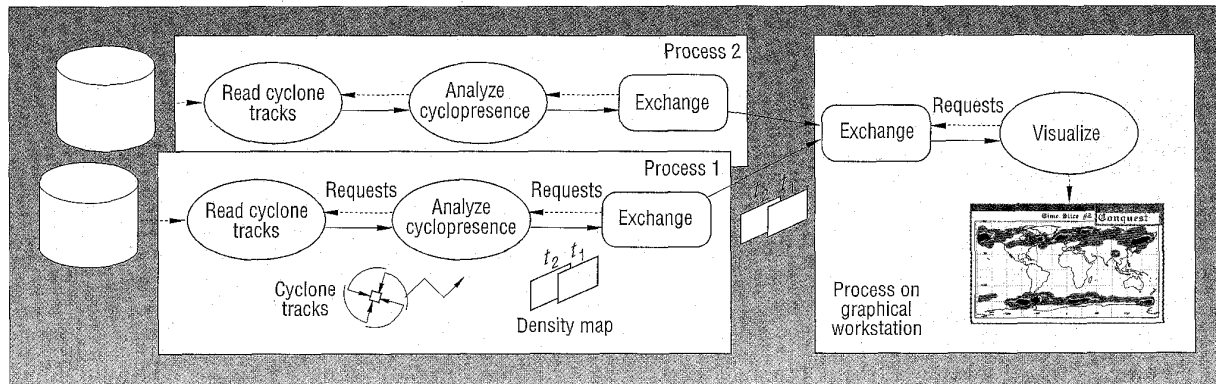


Figure 2. Parallel query execution in Conquest.

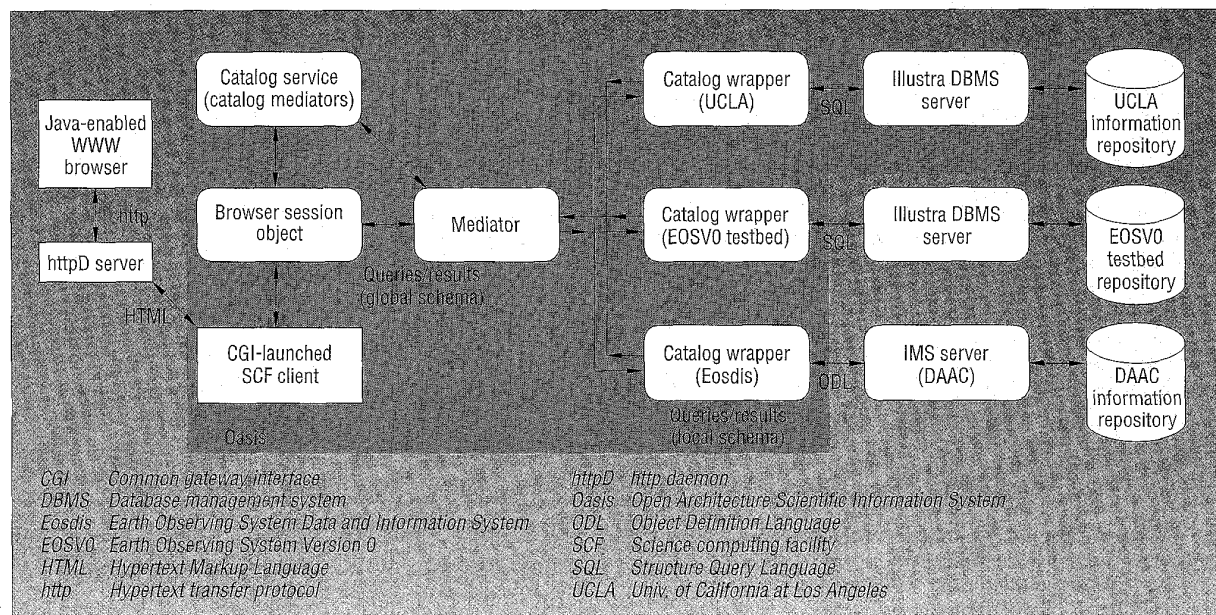


Figure 3. Oasis catalog service.

to simplify operator registration and the packaging of complex data during inter-process communication.

Catalog service. One of our project goals is to facilitate the location and retrieval of scientific information, be it data created and stored locally or data located at remote data repositories such as NASA's Earth Observing System (EOS) Distributed Active Archive Centers (DAACs). In this scenario, scientists pose queries via the Oasis catalog service infrastructure (see Figure 3).

The catalog service loosely models the Tsimmis system.⁷ Mediators realize global query schemas and provide query support via their query language (for example, ODL or SQL). Upon receipt of a query, a mediator decomposes the query and farms out sub-queries to its participating catalogs, which are

encapsulated via catalog wrappers. A catalog wrapper translates the query from the global query schema and language to the target catalog's local schema and query language. The catalog then processes the query and returns an iterator object (for example, a database cursor) to the wrapper. The catalog wrapper, in turn, passes an iterator object back to the mediator. However, this iterator returns tuples in the mediator's global schema. To do so, the catalog wrapper's iterator issues a request to the catalog iterator for the next tuple, translates the tuple from the catalog's schema into the mediator's global schema, and returns the resultant tuple. The major reason for this demand-driven evaluation of query results is to return query results to the user as necessary, thereby avoiding potentially unnecessary processing overhead (especially for large query-result sets).




User scenario

To demonstrate how Oasis can facilitate data mining, we outline a real-life, end-to-end scientific data-mining application involving the extraction and analysis of cyclone tracks. We developed the application within the framework of Oasis, taking advantage of Oasis's support of heterogeneous-data repository access and high-performance distributed computing.

Catalog browsing and data-object location. The catalog browser provides a graphical user interface for locating scientific data sets, derived products produced as the result of previous queries, and so on. The browser is written in JavaSoft's Java language and C++ CGI-bin programs (a JavaNEO version is under development) and executes via a

OASIS Catalog Browser

Query Form

Please fill in the form:
(It takes a while to start the applet)

Select one or more catalog(s):

UCLA AGCM AMIP Run 1

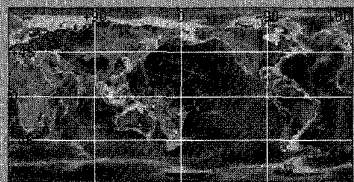
Time Envelope:

From: 1980/08/01 00:00:00
To: 1989/12/14 12:00:00

Year	Month	Day	HR	MM	SS
1951	01	01	00	00	00
1952	02	02	00	01	00
1953	03	03	00	02	00
1954	04	04	00	03	00
1955	05	05	00	04	00
1956	06	06	00	05	00
1957	07	07	00	06	00
1958	08	08	00	07	00
1959	09	09	00	08	00

Space Envelope: (latitude, longitude)

From: -90, -180
To: 90, 180



Click and drag the mouse to select the region

Select/Enter the appropriate values:

Parameter	Sensor
Geopotential Height	UCLA AGCM
Potential Temperature	
Sea Level Pressure	

Processing Level:

Browsable Image: Yes No Either





Figure 4. A query form produced to support Eosdis IMS schema.

Record #1:

Object reference = 1:2:color:47892:03baea8600008027:0

parameter	Potential Temperature
sensor	UCLA AGCM
browse	Yes
processing level	3
minRange	177.85
maxRange	310.994
unit	Kelvin
Time Envelope	1980/08/01 00:00:00 GMT to 1989/12/14 12:00:00 GMT
Space Envelope	(200, -86, -180), (930, 86, 175)
Dimensions	4 X 44 X 72 (Elevation, Latitude, Longitude)
Time Steps	6942 steps
Step Delta	0 d, 12 h, 0 m, 0 s
DataSet Size	346807296 bytes

Please click on the image for animation effect:
(for the first 20 time steps)

Save this dataset? Yes No

To save the marked datasets, click the "Save" button.

Figure 5. Detailed information retrieved for a data set selected via a query.

Java-enabled browser. Users can invoke the browser in a stand-alone mode or can launch it from an application such as LinkWinds or Glint (see Figure 1). Upon invocation, the scientist chooses from a set of available keyword sets (global schemas). The catalog service provides this information to the browser. Once the global schema is chosen, the browser interacts with the mediator, which is responsible for evaluating queries against the schema. The mediator provides the browser with the schema information (keywords, valid values, and so on) necessary to build the appropriate query form.

Figure 4 presents a sample query form. In this example, the scientist can specify a time

range or a spatial extent (by drawing a bounding box over the map), select parameters and sensors of interest, and choose the catalogs to be queried (the default is "all catalogs").

Figure 5 shows the detailed information presented as the result of the sample query. The net result of a scientist's interaction with the browser is the location and selection of distributed objects (data sets). The object references of the selected objects can then go back to the launching application or to an application invoked through the browser. In this example, the located object corresponds to the potential temperature fields produced by the UCLA Atmosphere General Circulation Model (AGCM),⁸ which is a finite-difference model that includes sophisticated parameterizations of cumulus convection and planetary boundary-layer processes, as well as parameterizations of short- and long-wave radiative transfer. Grid cells of various resolutions represent the model's horizontal structure; we are using a grid size of 5° longitude and 4° latitude. A series of constant σ (pressure/surface pressure) layers represents the model's vertical component; the version in this study has nine layers in the vertical, with the top at 50 mb. The prognostic variables (horizontal velocities, potential temperature) and diagnostic variables (precipitation) of the AGCM are written as netCDF

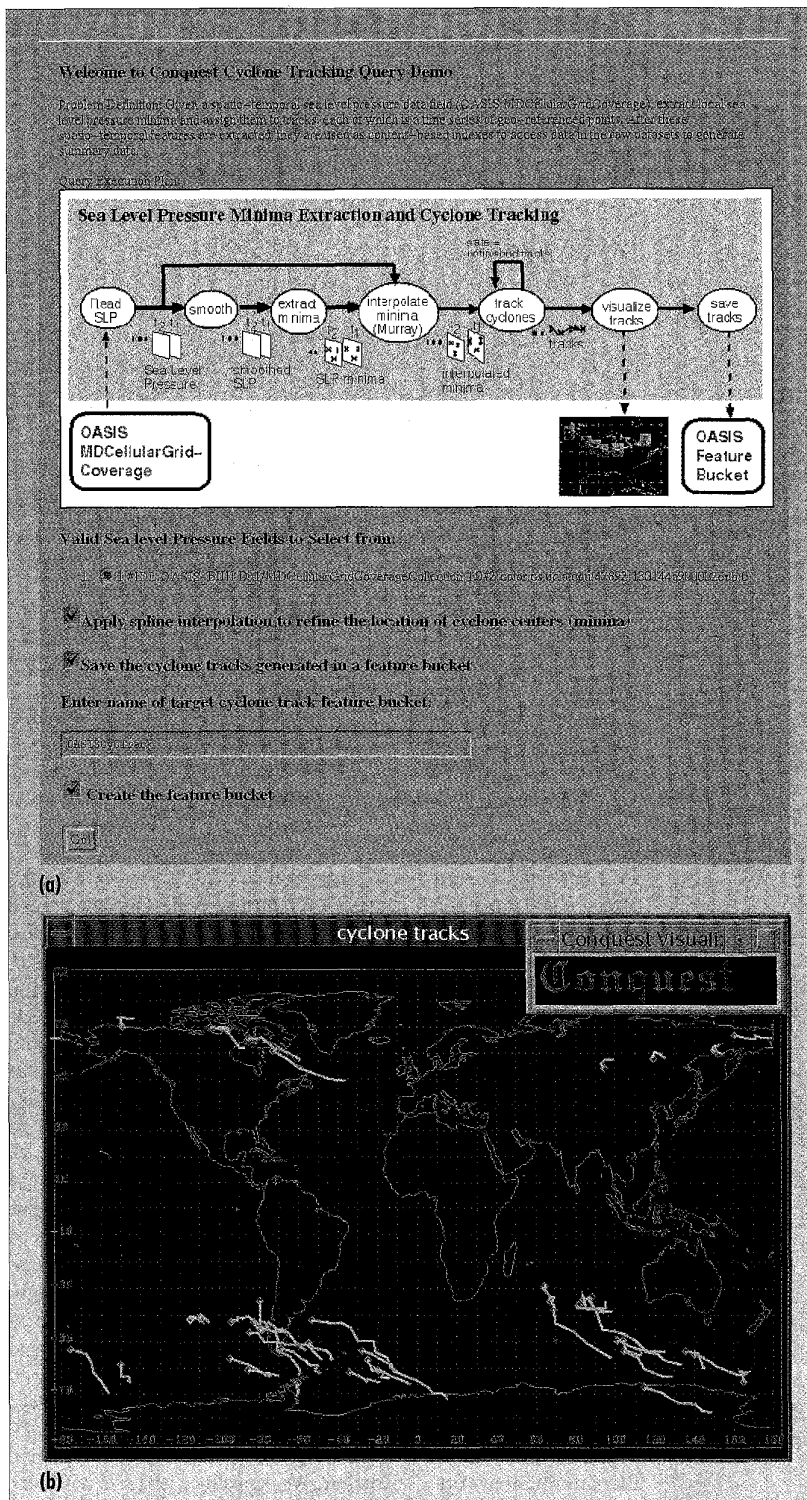


Figure 6. (a) Cyclone-tracking query panel; (b) visualization of partial results.

files at 12-hour (simulation time) intervals; however, we can modify this frequency, depending on the database's storage capacity. At the lowest spatial resolution ($4^{\circ} \times 5^{\circ}$, 9 levels) with a 12-hour output interval, the

AGCM produces approximately 5 Gbytes of data per simulated year.

Applying data-mining and analysis algorithms. After users have selected Oasis sea-

level pressure objects (AGCM data products) during their catalog-browsing session, they can bring up a WWW query panel (see Figure 6) for a precompiled Conquest query that extracts cyclone tracks from sea-level pressure data. (Conquest semantically checks the objects to make sure that they contain sea-level pressure data.) The cyclone-tracking algorithm (described in the sidebar) consists of two steps that can each be implemented as a Conquest operator with a well-defined interface: (1) extraction and refinement of minima in a sea-level pressure field, and (2) assignment of minima to tracks.

After extracting cyclone tracks from large, sea-level pressure data sets and ingesting them into an Oasis feature bucket object (a container for GIS data such as polygons or line strings), scientists can further study the extracted features to better understand these natural phenomena. Figure 7 shows the query panel for invoking the precompiled query for generating and visualizing summary data from the cyclone track collection created in the previous cyclone-tracking query. The panel's foreground shows a snapshot of the animation of the spatio-temporal migration pattern of monthly cyclopresence concentration for the AGCM data set from which the cyclone tracks are extracted. Note that most cyclones are formed and migrate within a few zonally elongated extra-tropical regions (storm tracks) in the Northern Atlantic and Pacific and around the Antarctic continent.

OASIS OFFERS SCIENTISTS AND application developers a view of the world as a collection of distributed location- and platform-independent objects. The system achieves this view for application developers by using well-known object interfaces based on OGIS and object implementations based on CORBA. Oasis offers scientists the view of the world as location-independent objects through high-level services and applications such as the catalog browser, which provides query facilities to locate data sets, and visualization and analysis tools that directly handle Oasis data objects. In addition, the extensible Conquest query-processing service provides high-performance query processing for complex geoscientific data-mining applications through parallelism. This service can handle complex scientific queries involving computationally expensive

calculations on large geoscientific data sets stored in different formats and managed by many different autonomous storage subsystems. UCLA and the JPL have used Conquest during the past two years for exploratory data analysis and data mining of spatio-temporal phenomena produced by the AGCMs at UCLA and European Centre Medium Weather Forecasting (Ecmwf) and by satellite-based sensor data such as the Ecmwf Global Basic Surface and the National Center for Atmospheric Research (NCAR) Upper Air Advanced Analyses. Examples of spatio-temporal features mined from the data include frequency, intensity, and tracks of cyclonic storms; blocking features; warm pools in oceans; and propagation events of wave energy in the upper atmosphere. Once extracted, these phenomena serve as high-level indexes for content-based access to raw measurements stored in large data sets (gigabyte-terabyte range).

We have demonstrated the utility of Oasis and our data-mining facility, Conquest, by executing queries to extract cyclone tracks and blocking events in several observational and AGCM simulation data sets. We have also described applications of unsupervised learning techniques and global change study (the study of changes in the global atmosphere and world ocean system, including chemical tracers). The ability to extract, analyze, and visualize these features from such large data sets lets scientists easily compare model simulations and observational analysis to gain a better understanding of the physics behind such features.

Acknowledgments

We sincerely acknowledge support from NASA HPCC Grants NAG 5-2224 and NAG 5-2225, from NASA Aisrp Grant NAGW-3915, and from NASA Eosdis Grant HAGW-4292. Computer time was provided by the DOE AMIP project, UCLA Office of Academic Computing, NASA HPCC-sponsored Intel Paragon at the JPL, and NASA Ames. We thank Bob Haskins, Tientien Li, and Charles Thompson of JPL for Glint and for validating Oasis. We thank Andy Berkin and Michael Orton of JPL for integrating LinkWinds into the Oasis framework. Finally, we thank Larry Fillion of BBN for integrating OpenMap into Oasis.

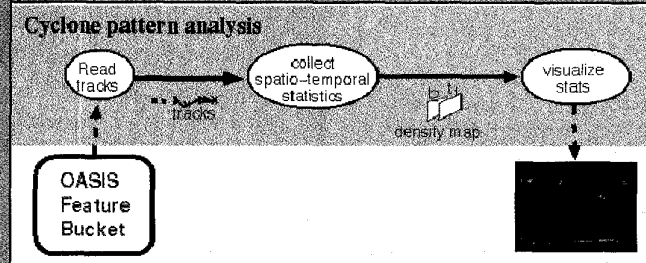
References

1. E. Mesrobian et al., "OASIS: An Open Architecture Scientific Information System," *Proc. Sixth Int'l Workshop Research Issues*

Conquest Cyclone Statistics Generation Query Demo

Problem Definition: Given a collection of cyclone tracks (OASIS Feature Buckets), spatio-temporal patterns of cyclone behaviors can be extracted and visualized to allow scientists to gain a better understanding of this natural phenomenon and the dataset from which the tracks are generated.

Query Execution Plan:



Enter name of source cyclone track feature bucket:

OASIS_CycTrack

Choose a statistics type

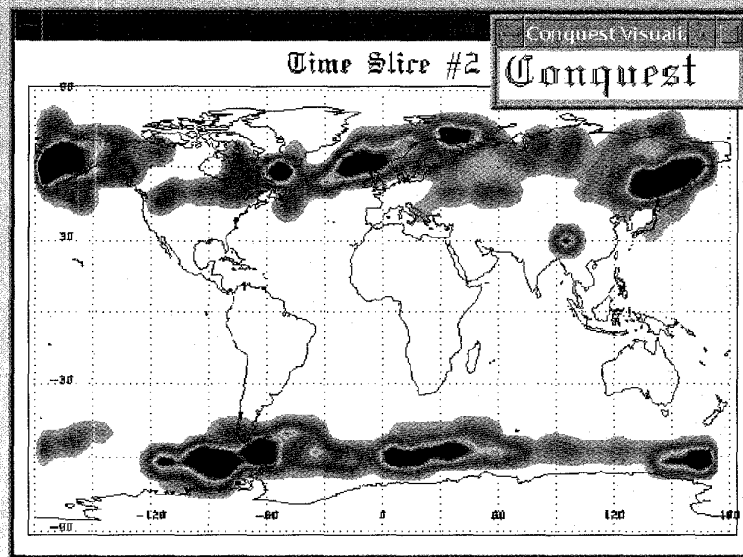
1. < Cyclopresence count
2. > Cyclogenesis count

Choose a duration for which statistics are collected

1. < On a monthly basis
2. > On a seasonal basis

Go!

(a)



(b)

Figure 7. (a) Cyclopresence query panel; (b) result visualization.

in Data Engineering, IEEE Computer Society Press, Los Alamitos, Calif., 1996, pp. 107-116.

2. A.S. Jacobson, A.L. Berkin, and M.N. Orton, "LinkWinds, Interactive Scientific Data

Analysis and Visualization," *Comm. ACM*, Vol. 37, No. 4, Apr. 1994, pp. 42-52.

3. E.C. Shek, E. Mesrobian, and R.R. Muntz, "On Heterogeneous Distributed Geoscientific Query Processing," *Proc. Sixth Int'l Workshop*

Research Issues in Data Engineering, IEEE CS Press, 1996, pp. 98-106.

4. R.M. Soley, ed., *Object Management Architecture Guide*, 2nd ed., Object Management Group, Framingham, Mass., 1992.
5. K. Beuhler and L. McKee, eds., *The OpenGIS Guide: Introduction to Interoperable Geoprocessing*, Open GIS Consortium Inc., Wayland, Mass., 1996.
6. G. Graefe, "Volcano: An Extensible and Parallel Query Evaluation System," *IEEE Trans. Knowledge and Data Engineering*, Vol. 6, No. 1, Feb. 1994, pp. 120-135.
7. S. Chawathe et al., "The TSIMMIS Project: Integration of Heterogeneous Information Sources," *Proc. IPSJ*, Information Processing Soc. of Japan, Tokyo, 1994, pp. 7-18.
8. C.R. Mechoso, S.W. Lyons, and J.A. Spahr, "The Impact of Sea Surface Temperature Anomalies on the Rainfall over Northeast Brazil," *J. Climate*, Vol. 3, No. 8, Aug. 1990, pp. 812-826.

Richard Muntz, who chairs UCLA's Computer Science Department, is the principal investigator of three NASA-sponsored efforts to develop novel, distributed scientific information systems. Muntz has focused on distributed query processing, data mining, and the design of Conquest's data model. **Edmond Mesrobian**, a coprincipal investigator on these efforts, has led the design and development of Oasis and has been involved in developing data-mining algorithms. **Eddie Shek**, a doctoral candidate at UCLA, is responsible for the Conquest environment.

Siliva Nittel, a visiting researcher at UCLA's Computer Science Department, has developed support for fine-grained GIS data, and now is focusing on GeoPOM, a persistent object manager for geographic objects. **Mark La Rouche** and **Marc Krieger** are staff members of the Data Mining Laboratory, where they focus on developing Oasis. La Rouche has worked on the Java-based catalog browser, catalog service, spatial/temporal reference systems, and configuration-management issues. Krieger has concentrated on developing Oasis catalogs, and on catalog interoperability with NASA centers.

Carlos Mechoso, of the Atmospheric Science Department at UCLA and a coprincipal investigator, has provided invaluable guidance in developing algorithms to detect geophysical phenomena and in ensuring that Oasis offers an environment conducive to scientific investigations. **John Farrara**, an associate researcher at UCLA's Atmospheric Sciences Department, and **Hisashi Nakamura**, a professor in the Department of Earth and Planetary Physics at the University of Tokyo, have concentrated on data-mining algorithms for detecting cyclonic activity, cyclopresence, blocking, and wave propagation. **Paul Stolorz**, a member of the JPL's Artificial Intelligence Group, has concentrated on the application of statistical and neural network data-mining algorithms for phenomena detection and on executing algorithms using massively parallel computers (including the IBM SP1 and SP2, Cray T3D, and Intel Paragon).



Available January '97!

Stiquito™ The Design and Implementation of Nitinol-Propelled Walking Robots

by James M. Conrad and Jonathan Mills

The Stiquito robot is a small, inexpensive, six-legged robot that is intended for use as a research and educational tool. This book, describes how to assemble and build Stiquito, provides information on the design and control of legged robots, illustrates its research uses, and includes the robot kit.

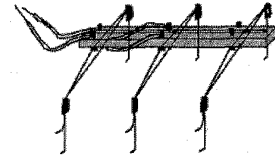
The experiments in the text lead the reader on a tour of the current state of robotics research. The hobbyist with some digital electronics background will also find this book challenging.

The book describes the birth of Stiquito, the building process, its modifications, and its increased load capacity. It examines designs for simple controllers to enhance the functionality of the robot while giving the robot intelligence and SCORPIO hardware designs for performing independent, intelligent operations. The book concludes with a discussion of the future for nitinol-propelled walking robots.

Contents: Preface • Stiquito Introduction and History • Walking Robots • Control of Walking Robots • Using Stiquito for Research • The Future of Stiquito • Bibliography • Appendixes

250 pages. 7 x 10" Hardcover. January 1997. ISBN 0-8186-7408-3.

Catalog # BP07408 — \$28.00 Members / \$35.00 List



Digital Design and Modeling with VHDL and Synthesis

by K.C. Chang

Available December '96!

Digital Design and Modeling with VHDL and Synthesis

by K.C. Chang, Boeing

Combines VHDL and synthesis in a step-by-step sequence that addresses common mistakes and the hard-to-understand concepts in a way that eases learning. VHDL is introduced with closely related practical design examples, simulation waveforms, and schematics so you can better understand their correspondence and relationship.

This book is the result of the author's practical experience in both design and teaching. Many of the design techniques and design considerations, illustrated throughout the chapters, are examples of real designs. Included with the book are numerous examples of real, complete working code for practical applications with simulation waveforms.

Contents: VHDL Basics and Modeling Concepts • Sequential and Concurrent Statements • Subprograms and Packages • Design Unit, Library, and Configuration • Writing VHDL for Synthesis • Finite State Machines • More on Behavioral Modeling • A Design Case and Test Bench • ALU Design • A Design Project • VHDL '93 • Appendixes

280 pages. 7" x 10" Hardcover. 1997. ISBN 0-8186-7716-3.

Catalog # BP07716 — \$45.00 Members / \$55.00 List

50 YEARS OF SERVICE

IEEE
**COMPUTER
SOCIETY**
1946-1996

Phone Orders:

+1-800-CS-BOOKS

CS Online Catalog:

www.computer.org/cspress