

Applications of Object Relational Database Management Systems at BCS

Barrodale Computing Services Ltd. (BCS)

<http://www.barrodale.com>

This article provides an overview of Object Relational Database Management Systems (ORDBMSs) and summarizes our company's experience in using ORDBMS technology to provide solutions to six applications involving the management of complex data.

Types of Databases

Modern databases can handle both simple and complex data, ranging from integers and text strings to satellite imagery, 4D weather grids, and video. Some of the types of data that might be stored in a meteorological/oceanographic database are indicated in the diagram below (from a Commander Naval Meteorology and Oceanography Command presentation). In this example, the data has a strong geospatial component, with each feature being localized in space and, optionally, in time.

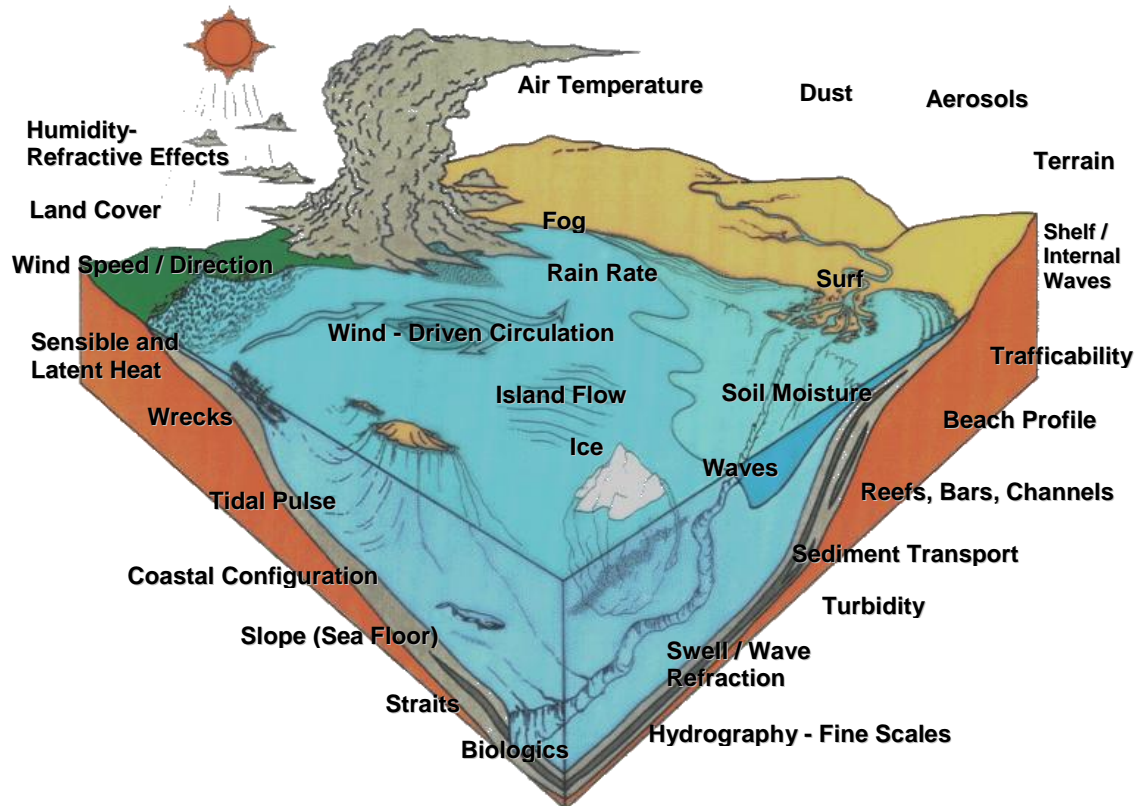


Figure 1. Data that could be stored in a meteorological/oceanographic database.

The following diagram (from p.2, *Object-Relational DBMSs – Tracking the Next Great Wave*, M. Stonebraker and P. Brown, Morgan Kaufmann, 1999) classifies databases by usage, although many real-life applications would fall into more than one quadrant. It indicates that when we want to make queries on a database containing complex data, there are certain advantages to using an ORDBMS. Queries are made using SQL, which is much simpler to learn and use than most programming languages.

<i>Query</i>	Relational Database Management System (RDBMS)	Object Relational Database Management System (ORDBMS)
<i>No Query</i>	File System	Object Oriented Database Management System
	<i>Simple Data</i>	<i>Complex Data</i>

Figure 2. Simplistic database classification matrix.

File Servers vs. Relational Databases for Data Storage

Generally, applications involving data storage comprise both data and metadata (i.e., descriptive data about the data actually collected and stored). There are several options for using file systems and databases to store data and metadata. One option adopted by many people is to avoid storing their data in databases altogether, using only files instead. A File Server containing both the data and the metadata is certainly simpler and often cheaper to use, but there are certain advantages in using a second option, which is to combine a traditional Relational Database Management System (RDBMS) containing the metadata with a File Server containing the actual data.

Using a ***File Server*** alone (the first option) to store the data and metadata has the following advantages (+) and disadvantages (-):

- + Greater simplicity.
- + Often lesser cost.
- + Fastest access to the whole data set (the entire file).
- Slower access to a small subset of the data.

Using an **RDBMS** to store the metadata and a file server to store the data (the second option) offers users several advantages over a file server alone:

- + Integrity checking of metadata - this can be performed by built-in RDBMS features (check constraints, triggers, etc.).
- + Efficient access to the metadata - e.g., indexes can be used.
- + Concurrent access to the metadata by multiple users.
- + Easier to locate data of interest - e.g., complicated queries on the metadata can be performed.
- Metadata separated from data - data inconsistencies are possible, since changes made to the data (stored in a file) may not be reflected simultaneously in the metadata (stored in the RDBMS).

This last point represents a substantial disadvantage of the RDBMS/File-Server strategy, in that the metadata is kept separate from the actual data. This can sometimes lead to data inconsistencies.

Object Relational Databases for Data and Metadata Storage

The above concern can be overcome through use of a third option, an **ORDBMS**, in which composite data types allow data to be bundled together with metadata, and queried using SQL. ORDBMSs have four features that set them apart from RDBMSs:

- User-defined abstract data types (ADTs) - ADTs allow *new data types* with structures suited to particular applications to be defined.
- User-defined routines (UDRs) - UDRs provide the means for writing *customized server functions* that have much of the power and functionality expressible in C.
- “SmartBLOBs” - these are disk-based objects that have the *functionality of random access files*. ADTs use them to store any data that does not fit into a table row.
- Extensible and flexible indexing - R-tree or Generalized Search Tree (GiST) indexing for multi-dimensional data enables *fast searching* of particular ADTs in a table. Traditional B-tree indexing is suitable for data that can be ordered in one dimension (scalar numbers or text strings), but for multidimensional data (such as polygons) an R-tree or GiST index is more suitable.

An example of the use of some of the above features is: *Sum the “area” (UDR) of all “lakes” (ADT) “contained in” (R-tree) “British Columbia” (instance of an ADT).*

Using an ORDBMS in which the metadata and data are integrated has the following advantages:

- + Improved concurrency - concurrent users can safely query the same data.
- + Composite data types - data is bundled with its metadata.
- + Improved integrity - ability to reject bad data before it is stored in an ORDBMS.
- + Database extensibility - easy addition of data types and operations.
- + Uniform treatment of data items - the SQL interface can perform complex queries based on *any* of these data items, e.g., metadata as well as data; hence there is less need for custom programming by users.
- + Custom data access methods - e.g., R-tree indexes.
- + Point-in-time recovery of data is possible.
- + Built-in complex SQL functions can be provided for data operations - e.g., aggregating, slicing, subsetting, reprojecting, etc.

BCS specializes in ORDBMS applications and we have taken advantage of some or all of these ORDBMS features in a variety of diverse fields. We have developed these applications using both Open Source (e.g., PostgreSQL) and proprietary ORDBMSs (e.g., IBM Informix, Oracle and SQL Server).

For example, we have developed a [Watershed Atlas](#) of British Columbia using an IBM Informix ORDBMS. The diagram below shows a portion of this Atlas, in which the rivers and lakes are in blue and the watershed boundaries (or heights-of-land) are in green.

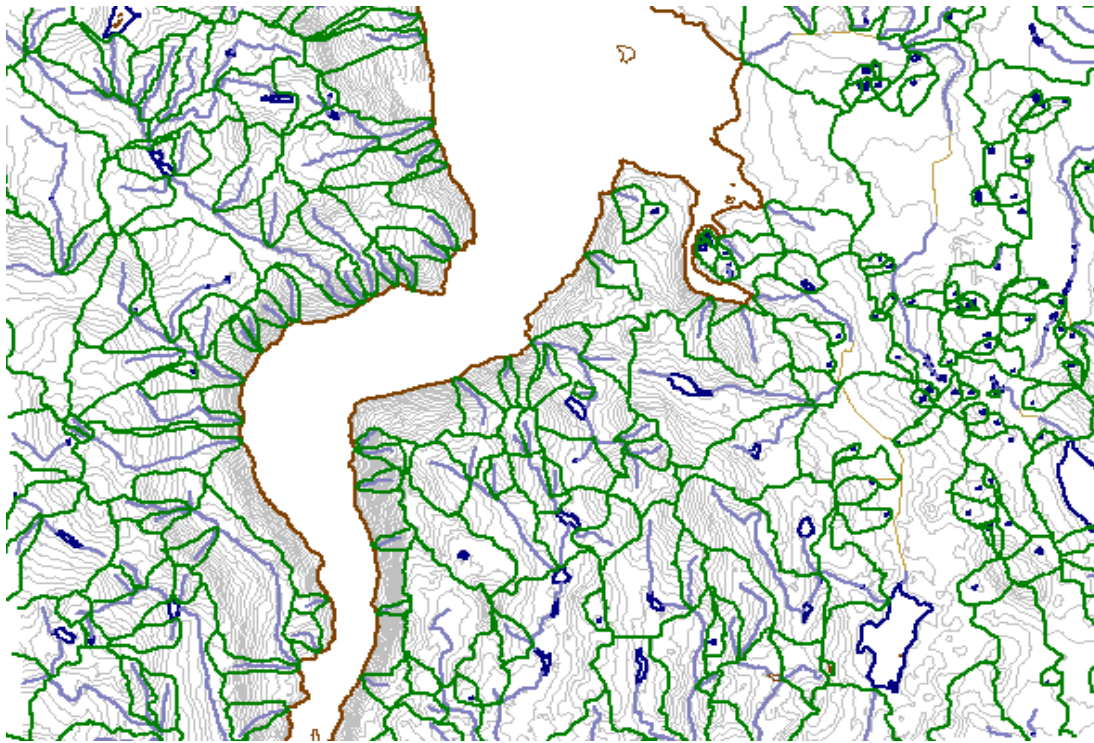


Figure 3. A small portion of the Watershed Atlas for British Columbia that was developed by BCS using an IBM Informix ORDBMS.

In this database, sophisticated queries can be made using just a few lines of SQL. For example, for a query like “*Find the watershed area that is upstream from where a given road crosses a given stream,*” an SQL command such as the following could be issued:

```
SELECT Area(Watershed(streamElement,
    (Intersection(streamElement, roadElement))))
FROM streamNetwork, roadNetwork
WHERE Overlap(Intersection(Box(streamElement),
    Box(roadElement)), userDefinedArea);
```

Note that:

- *userDefinedArea* is, say, a string provided by the user.
- There are five UDRs involved in the query, namely:
 - BOX - rectangle enclosing object;
 - INTERSECTION - common area;
 - OVERLAP - T or F;
 - WATERSHED - calculates watershed upstream from a point;
 - AREA - calculates area.

Choosing an ORDBMS: IBM Informix vs. PostgreSQL

We have worked primarily with IBM Informix, PostgreSQL, SQL Server and Oracle ORDBMSs. In our experience, choosing one over another may be application dependent, or it may be determined by customer preference. For example, the following points provide a brief comparison of the PostgreSQL and IBM Informix ORDBMSs, with an indication in parentheses of which product “wins” in each category.

- PostgreSQL is free while IBM Informix has an up-front cost (*P*).
- PostgreSQL is Open Source while IBM Informix is Proprietary. So, adapting OR extensions in PostgreSQL is generally simpler, although more maintenance is required since versions change more often than with IBM Informix (*P*).
- For data types not requiring large amounts of space (vector linework) development in PostgreSQL is simpler than in IBM Informix, but the opposite is true for data types requiring large amounts of space (say, 4D gridded data sets).
- PostgreSQL is single-threaded while IBM Informix is multi-threaded. This can make Informix more efficient in situations with many concurrent users (*I*).
- The BLOB implementation in IBM Informix is more efficient than that used in PostgreSQL (*I*).

BCS Solutions to Applications Involving ORDBMSs

The following sections summarize our experience in designing and/or using ORDBMSs in six diverse application areas.

Ocean Observatories

Overview

In recent years, there has been increasing interest in establishing ocean observatories to obtain information about the three quarters of our planet that is covered by ocean. For example, the VENUS (University of Victoria) and NEPTUNE (Canada and USA) projects will accumulate data from a variety of subsurface and bottom-mounted ocean instruments and will provide detailed information about the coastal regions of the Pacific Northwest. In addition, the Gulf of Mexico Seafloor Observatory (University of Mississippi) will generate data about the geochemistry, microbiology and stability of gas hydrate outcroppings in the Mississippi Canyon region of the Gulf of Mexico. The data from these observatories, which consists largely of time series, will be stored in Data Management and Archiving Systems (DMASs) for access by scientists worldwide, and will support research in fields such as:

- Oceanography: physics, chemistry, tides, currents, fluxes.
- Seismology: plate tectonics, slope stability, early earthquake/tsunami warning.
- Marine biology: plankton migration, behavior monitoring, whale detection.
- Climate modeling: global warming, model testing and verification.

These ocean observatories will not only enhance our knowledge base about fundamental ocean and crustal processes, they may also provide more economic benefits, such as early warning of offshore earthquakes and the improved prediction of tsunamis. The measurements will also be expected to be of importance in the monitoring and modeling of climate change.

Data

Oceanographic data can be obtained from a wide variety of sensors, and consist of many different data types, including surface, subsurface, bottom, and satellite sensors. The data characteristically have both spatial and temporal components and most data sets are near-real-time sequences of 1D or 2D data. In certain cases, the data acquisition rates and volumes to be stored can be very large. Some typical examples of oceanographic observatory data include:

- CTD (conductivity, temperature, depth).
- ADCP (acoustic Doppler current profiler).
- ZAP (zooplankton acoustic profiler).
- Chemical sensors (oxygen, methane, nitrate).
- Turbidity, optical backscatter.
- Acoustic and seismic time series from hydrophones and geophones.

- Still and video camera images.
- Satellite imagery.

BCS Contributions and Role of ORDBMS

BCS was involved in the VENUS and NEPTUNE projects, having prepared a comprehensive 238-page report on their proposed sensors and instruments. This report also contained descriptions of the data handling practices of organizations currently collecting oceanographic or similar data (see [VENUS/NEPTUNE DMAS Examination](#) (October 2004)).

In addition, we have recently developed special purpose techniques for handling data sequences in an ORDBMS. This has led to the development of [DBXten¹](#), which provides a fast, flexible and compact way of storing and indexing data sequences. DBXten speeds up insertion of data into indexed tables more than 50 times and decreases index sizes by more than a factor of 1000, while generating these indexes more than 1000 times faster. This technology is eminently suitable for ocean observatory database applications and is already available for PostgreSQL, IBM Informix, SQL Server, and Oracle.

Elections Geographic Database

Overview

Managing elections has a very strong geographic component, since voters are generally registered to vote according to their addresses. Elections British Columbia uses a large geospatial database application developed by BCS to determine the electoral district corresponding to an address – with allowances for misspellings and aliases. This geospatial database, termed “[INDEA](#)” (for INtegrated Digital Electoral Atlas) contains the geographic information needed for managing elections in British Columbia, Canada.

The information and functionality contained in the INDEA ORDBMS include:

- Geocoding, i.e., determining the electoral district (ED) and voter area (VA) on the basis of a voter’s address (see [Know Your Electoral District](#)).
- Fuzzy matching for addresses to accommodate variant spellings and aliases.
- The storage and maintenance of numerous geographic layers, along with methods to ensure their consistent attribution.
- The presence of numerous distinct versions of the same layer (ED/VA boundary sets).

In addition to supporting the necessary tasks of registering voters and advising them where they can vote, the INDEA database application also provides a source of data for redistribution (redistricting). This process involves redrawing the boundaries for the voter areas as the spatial distribution of the voters changes. Previous to the development of INDEA, redistribution was done manually and was a very laborious process.

¹ Update: In December 2011 BCS was awarded US Patent 8077059 for DBXten.

Data

The underlying geographic data in INDEA was obtained from many sources and have different levels of accuracy, ranging from differential GPS to (in one case) a faxed hand-drawn sketch on a coffee shop placemat! A real challenge is maintaining consistency between the various spatial components of the database through conflation and other processes.

INDEA contains more than 50 separate layers of point, line and polygon data, and implements a large number of special data types and business rules. The data includes geographic and enumeration data from federal, provincial (i.e., state), and local governments and private sector data collection agencies. INDEA is also regularly updated as a result of new subdivisions, construction, secondary suites, infilling, etc.

Some of the types of data stored in INDEA include:

- Roads segments with address ranges.
- Hydrography.
- Railways, transmission lines, and parks.
- Buildings, including schools, colleges, universities, hospitals, etc.
- Municipal and Indian Reserves.
- Provincial and Federal EDs and VAs.

BCS Contributions and Role of ORDBMS

BCS developed INDEA using an IBM Informix ORDBMS, which provided a platform that allowed the tight integration of geographic and attribute data. It also provided the ability to implement user-defined types and user-defined routines that operate efficiently inside the database server to perform computations, the results of which are downloaded to the client applications.

In addition to INDEA itself, we have developed a GIS-like software system for management of the INDEA spatial and attribute data, and have also developed semi-automated redistribution techniques based on the information in INDEA database.

Watershed Atlas Database

Overview

It's a big job to map an area the size of British Columbia, which covers an area larger than the states of Washington, Oregon and California combined, or France and Germany combined. It's also a challenge to store the information seamlessly in a database, and to use it to automatically generate watershed boundaries and an "intelligent" directed single-line stream network (i.e., one with upstream-downstream information) through all the lakes and rivers in BC. That was the scope of the TRIM (Terrain Resource Information Management) Watershed Atlas project, in which both the original data and the derived data products were stored in an IBM Informix ORDBMS.

The derived watershed boundaries in the [TRIM Watershed Atlas](#) (TWA) have both geographic and economic value, as they are used to define forest tenures. These are the areas within which timber companies are licensed to harvest trees, and even small changes in their boundaries can have large financial and logistic implications for these companies. Our other derived product, the “intelligent” single-line stream network, can be used to compute upstream areas for a point on a stream, which is essential information for environmental and road engineering applications.

The TWA project involved first developing mechanisms to store all the hydrographic data present in the TRIM project in British Columbia. This included point data (e.g., a non-gridded digital elevation model or DEM), line data (e.g., streams) and areal data (e.g., lakes). We achieved this objective by combining and compressing the geographic data for the features into “chips” (we define a chip as a compact representation of a continuous block of tuples in a relational table) and storing these in BLOBs in the ORDBMS. This approach also allowed efficient construction and use of spatial indexes.

Another part of the project involved development of algorithms for automated computation of watershed boundaries from the hydrography data and the DEM. The resulting boundaries were also stored as chips in the BLOBs of the database. In addition, we developed methods for computing a single-line stream network representing the connectivity and flow direction for lakes and double-sided rivers, and stored these as a directed network in the ORDBMS. The resulting network provided the framework for upstream-downstream computations such as the upstream area query (noted earlier in this article).

Some areas of application of the TWA ORDBMS system are:

- Legal boundaries (forest tenures).
- Stream gradients (fish habitat).
- Upstream area computation (road bridge/culvert design).
- Downstream tracking (spills of toxic or hazardous materials).
- Hypsometric curves (elevation vs. area).

Data

The data used in the TWA originated from British Columbia’s TRIM mapping project. The TRIM project involved the construction of a 1:20,000 digital basemap based on aerial photography, and produced more than 500 million 3D data points overall. It includes hydrographic features (coastlines, lakes, double- and single-sided rivers, ponds, wetlands, etc.) as well as certain breaklines (ridges). It also includes a non-gridded DEM, with points sampled every 50 – 75 m, depending on local relief. The absolute accuracy of the data is 10 m for horizontal linework and 5 m for elevations.

All this data, along with the derived watershed boundaries and the single-line stream network, was stored in the TWA ORDBMS using the chips/BLOBs technique. This resulted in a much more compact data set than could have been achieved without the

compression afforded by chips, and one which could be efficiently indexed using R-trees at the chips level.

BCS Contributions and Role of ORDBMS

As well as developing algorithms for delineating watersheds and generating single-line stream networks, we developed novel methods for seamless and efficient storage of geographic data using the chips/BLOBs approach. We also implemented a suite of geospatial operations, *TerrainWorks*, which includes algorithms for spatial predicates such as “contains” and “overlaps”, as per the OGC Simple Feature Specification. The *TerrainWorks* database extension runs in an IBM Informix ORDBMS.

Commercial Shipping Information System

Overview

The global maritime shipping industry is vast, with thousands of cargo and container ships in transit at any given moment, all needing current meteorological and other data while also representing a potential source of such data. In many cases, the information is vital to navigation and also to safety. Some areas in which maritime-oriented data is used include:

- Traffic management and route planning.
- Scheduling and course setting based on weather, tides and currents.
- Determining the most fuel-efficient route between two places, given the predicted weather and sea state.
- Ship, port and homeland security.
- Surveillance, coastal monitoring, pollution detection and control.
- Safety/security (congestion information; pirate avoidance).

Much essential meteorological information used in maritime applications originates in the form of large 4D gridded data sets, and a useful tool in the effective provision of the required weather information to clients is BCS’s [Grid DataBlade](#). This software allows data products from these large 4D grids to be generated very efficiently and then transmitted to users.

Data

Some of the types of data required by the commercial shipping industry as well as their ports of call include:

- Meteorological conditions and predictions.
- Satellite images.
- Tide and current predictions.
- Output from ocean and weather models.
- Vessel locations and reports.

BCS Contributions and Role of ORDBMS

BCS developed its Grid DataBlade in order to provide effective storage of 4D grids (such as those generated by weather models) and fast extraction of gridded data products. This software system, running in a PostgreSQL ORDBMS, is currently in use by BMT Asia Pacific as a key component of their weather information dissemination system.

We are currently also working with BMT to identify additional applications of DBXten, our ORDBMS technology for efficient storage and indexing of data sequences.

Weather Modeling and Prediction

Overview

Weather forecasts are of universal interest and importance, and are vital to many areas of human activity, particularly in the areas of transportation, military operations, and agriculture. These forecasts are produced by inputting large volumes of data from global and local networks of sensors into a weather modeling program which operates by solving mathematical equations on a 4D time-space grid. The results of these simulations can be directly provided to clients as grids or grid subsets or they can be processed to yield forecasts and maps. Some applications in which modeling of such weather-related variables as wind, temperature, precipitation, and cloud cover plays a large role include:

- Planning of military, agricultural, marine, aviation, and ground transportation operations.
- Storm/hurricane track prediction, severe weather warnings.
- Stop/go or management decisions based on weather.

Often, many individual needs may be satisfied and handled simultaneously by a central weather modeling and forecasting center, with distribution of selected data products to large numbers of individual clients. This can be handled effectively by storing the results in an ORDBMS and distributing selected portions of this data to users.

Data

The input data used by weather models typically originates from worldwide networks of meteorological sensors and imaging systems on fixed and mobile platforms. All these data can be stored in an ORDBMS before being input to the weather models.

The output of the models consists largely of 4D time-space grids of predicted values of physical parameters such as humidity, pressure, temperature, wind speed and direction. These grids can be stored in an ORDBMS and subsetted (e.g., sliced across space and time) to deliver targeted data products to clients.

BCS Contributions and Role of ORDBMS

As with the commercial shipping information system outlined above, a useful tool in the storing and handling of the 4D gridded data forecasts stored in an ORDBMS is BCS's

[Grid DataBlade](#). This software allows data products from these grids to be generated very efficiently and then distributed to users. Both the US Navy and National Oceanic and Atmospheric Administration (NOAA) are using the Grid DataBlade for this purpose.

Life Sciences

Overview

The Life Sciences arena involves large numbers of distributed databases, each containing rapidly changing data, and a vast number of scientific and medical users generating and accessing this information. The types of data in these databases are very disparate indeed, and they range from physicochemical properties of small molecules to genomic sequences, from medical images to full-fledged clinical trials, and from viruses and bacteria through to humans. The overall amount of data in these databases exceeds the exabyte level (i.e., millions of terabytes).

Life Sciences information is used in a wide variety of applications, ranging from agriculture to health care. There is currently a great deal of interest in drug discovery - pharmaceutical R&D based on molecular modeling of candidate drug compounds and receptors. Also, studies of gene expression are critical in understanding many life processes and abnormalities, and proteomics can provide a window into the functional activities of cells. Moreover, data mining has the potential to identify biomarkers for the presence of various disease states. The opportunities provided by Life Sciences information for betterment of human life and the health of the world in general are virtually unlimited.

Some areas in which ORDBMSs can advantageously be used in Life Sciences include:

- Bioinformatics.
- Genomics
- Proteomics.
- Diagnostics.
- Drug discovery.
- Biomarker detection.
- Drug interactions.
- Genetic engineering.
- Gene expression and therapy.
- Data mining for detecting patterns and providing insights.
- Environmental bioremediation.

Data

Sources of Life Sciences data include universities, research institutes, international agencies, and commercial enterprises. Some of the types of data used in Life Sciences applications include:

- Chemical and biochemical data: structures, isomers, spectra.
- Nucleic acid and protein sequences and molecular configurations.
- Spectra (mass, IR, UV, NMR, ESR, etc.).
- Medical images (ultrasound, X-ray, CT, PET, MRI, etc.).
- Biological activity and biomedical data.
- Clinical trials results.

BCS Contributions and Role of ORDBMS

In order to obtain rapid access to items of interest in large volumes of stored complex data, it is essential to be able to build indexes on that data. This is often not possible for certain ORDBMSs due to the lack of appropriate built-in indexing capabilities. BCS is currently developing a framework for users to create Generalized Search Tree (GiST) indexing schemes for their own complex data. The resulting indexing methods can be applied to complex multidimensional data and will allow users to obtain faster and more effective access to essential information.

Summary

Over the past ten years, Object Relational Database Management Systems (ORDBMSs) have gained acceptance as a powerful technology for handling complex database applications. BCS has specialized in using ORDBMS technology to provide solutions to problems arising in geospatial and other challenging fields. The purpose of the current article has been to summarize BCS's involvement in six such applications.

Contact Information

In order to discuss how ORDMBS technology could be applied to provide solutions to your applications, please contact:

Barrodale Computing Services Ltd. (BCS)

E-mail: BCSinfo@barrodale.com

Web: <http://www.barrodale.com>