

New Technology for Ocean Data Management

J.S. SARUPRIA AND R.M.S. BHARGAVA

National Institute of Oceanography, Dona Paula, Goa - 403 004, India

ABSTRACT

An overview is presented of the major computer systems currently available for use in oceanography measurements. The special features of different systems are delineated. More advanced systems in the offing are discussed.

INTRODUCTION

Considerable interest in multi-disciplinary oceanographic measurements and their management has been generated recently. This interest is driven in large part by the recognition that many interesting and important oceanographic problems require multi-disciplinary data sets and modelling approaches. Included among these problems are biomass determination, phytoplankton, primary productivity, secondary productivity, biogeochemical fluxes and cycles. Gas exchange across the air-sea interface, larvae fish recruitment, optical variability and bioluminescence etc. The two primary methods which will be available to ocean scientists for studying the oceanic ecosystem in near future are *in situ* sampling and remote sensing. However, both the systems require historical filtered databases for long-range prediction and solving of various problems of the ecosystem. Development of a multi-disciplinary measurement system has been motivated by the desire to understand complicated oceanic ecosystems.

Technological advances during the past decade have led to rapid expansion of the sampling domain of many of the important bio-optical and physical variables. In particular, new optical and acoustical sensors along with microprocessors have enabled biological and optical oceanographers to begin to study the phenomena on scales comparable to those studied by physical oceanographers. Integration of these sensors into CTD and current meter types of systems has made it possible to investigate bio-optical and physical interactions through co-located and concurrent observations [4]. Many of the future advances will most probably depend on continued progress in the development of fundamental instrumentation and *in situ* sampling systems utilizing this instrumentation. Additional satellite and derived oceanographic data will become available for regional and ocean basin scale

studies and *in situ* measurements should be designed to complement these data in terms of types of observations, particularly in depth and time.

Satellites are also being used to determine the surface features of the ocean like circulation, wind, wave, current directions, suspended matter, coastal pollution, etc. The data/information derived from satellites like SEASAT, NOAA, METEOSAT, IRS, GEOS, INSAT, etc. have been used to study the ocean features globally.

OCEANOGRAPHIC MEASURING SYSTEMS

A remarkable number of advances in multi-disciplinary *in situ* measurement systems have been made during the past decade. The types of multi-disciplinary measurements and their temporal and spatial scale ranges have been expanded significantly. Particularly important developments have occurred in optical and acoustical instrumentations. These have enabled many biological measurements on scales comparable to those of CTDs and current meters. In addition, new systems which collect essentially collocated and concurrent multi-disciplinary data have been developed. Further advances in multi-disciplinary measurement systems will depend on new sensor and system technology and the availability of economical bio-optical and acoustical sensors, fibre optics and telemetry. Study of global scale oceanic problems will require both satellite remote sensing and selective *in situ* observations. Moored, drifting and expendable bio-optical and physical measurement systems are quite attractive for the latter purpose because of their potential for expanding the ranges of temporal and spatial measurements and greatly increasing the number of observations. Hence, to handle these data/information, sophisticated data management system is required [3].

Data base management system (DBMS)

Techniques in the management of data have passed through very decisive stages. The first was the development and widespread use of file techniques. The major efforts during this phase were directed towards defining improved access methods, studying the workloads catered by these and producing languages by which a file system could be easily accessed.

File system served the needs of relatively small information systems well. However, as attempts were made to build larger and larger information systems; design methodologist found this inadequate and the file system started showing problems also. This led to the second stage in the development of techniques for the management of data. Gradually, database systems emerged and got stabilized after a fierce controversy in the 70's.

In fact, database systems even replaced file systems in some cases, particularly those where *ad hoc* queries rather than fixed pre-determined queries were encountered frequently. With the emergence of PCs, database management systems become quite common. In fact, the use of database management techniques has

become so widespread that these seem to be taken for granted in almost the same way as the operating system of a computer. This again is a far cry from earlier days when practicing professionals used to greet and mention database technology with derision and hostility. However, by the end of the 70's, data-base technologists had sorted out almost all serious problems in this subject area.

Databases are the major sources of information in a computer system. Whether that information is on oceanographic cruises/stations inventory or on major multi-disciplinary oceanographic data activities, it is stored in some kind of a database [7]. How that database should be organized and structured is a matter for consideration.

The changes that have been occurring recently with databases delve into the popular relational model and Structured Query Languages (SQL). In the past, most microcomputer databases have been of the stand-alone variety. Recently, however, with the popularity and proliferation of LANs (Local Area Networks), the problems of database incompatibilities among different microcomputers have become more pronounced.

General Format-3 (GF3)

This system was developed by the Intergovernmental Oceanographic Commission (IOC) Working Committee on International Oceanographic Data Exchange (IODE) as a generalized formatting system for the exchange and archival storage of data within the international oceanographic community.

GF3 is a system for formatting geo-scientific data series into sequential files and digital storage devices. It is not a fixed format in the conventional sense, but is a generalized system, which allows the user a number of ways to organize data, such that whichever way he chooses, the data fit into GF3 [5].

It is a highly versatile system capable of accommodating virtually any type of digital oceanographic data, including physical, chemical, biological, geological, geophysical and meteorological measurements. Being multi-disciplinary in nature, it is also applicable to other branches of environmental and geo-sciences outside the field of oceanography.

Data can be stored and retrieved in GF-3 with simple rather short programmes. The GF3 includes a Fortran-based software package (GF3-Proc) to facilitate the reading and writing of data in GF3. The package is designed as a suite of Fortran sub-routines and provides a complete software interface to GF3, enabling the user to handle GF3 in an automated fashion. It is also designed to be compatible between different computer systems so as to be readily available to scientists and data centres alike. The technical specification represents the standard that should be adhered to in preparing GF3 magnetic tapes for the purpose of data exchange. Thus, all the data files are arranged in sequence on the tape but are preceded at the beginning of the tape by a 'Test File' and a 'Tape Header File' and the tape ends

with 'Tape Terminator File'. These additional tape-related files are used primarily for administrative and labelling purposes.

The underlying structure of the GF3 system is built around logical records of a fixed length of 1920 bytes in character coded format. There are a number of different types of GF3 records, each with its own distinctive use and format. A proper understanding of the GF3 system requires knowledge of the usage and contents of these records. For example, there are header records to define the beginning of a GF3 file or series, plain language records for the textual, which may be used for data documentation purpose. However, the key of the GF3 system is the definition record used to define the content and format of the data which is stored records. The definition record makes automatic processing features of GF3 possible and can be interpreted by computer to find out what parameters are contained in the data cycles, where they are located in the records, what units they are in and in what format they are stored. The flexibility of GF3 results from the variety and number of possible usages and combinations of the GF3 record types. This makes it possible to include within the format scheme structures from very simple to those with several layers of hierarchy.

Structured query language (SQL)

SQL is a language for interacting with relational databases. SQL statements are embedded in programming languages, where they retrieve sets of rows and columns from the database, stepping a cursor through them one at a time and passing each to host language variables for further processing. The source code containing SQL is preprocessed to translate the embedded SQL statements into optimized database; these calls are specific to the host language [6]. Application Programming Interfaces (APIs) to SQL engines are also provided for programming languages. Now the SQL has been developed as an integral part of a 4th Generation Language (4GL). SQL is also incorporated in front-end tools other than programming languages. Thus, in forms-based systems, one can inlay SQL statements in forms or can hide SQL altogether with menu, prompt, or form-driven capabilities, such as query-by-example, query-by-form, or graphical interfaces. These approaches require good mapping between relational functions and these tools. The weakness of the SQL language itself causes some of this variation and the consequent implementation problems. It is the relational nature of SQL that has propelled it as the language of choice for connectivity. Its high level, set orientation, and support of physical data independence have made possible many of the current developments in the database arena.

Cooperative processing and database distribution among networked heterogeneous computers have become possible because workstations can now communicate in a standard language with different remote database servers. These things are also possible because database's functions, including integrity and security, are now centrally, relationally benefited by mixing and matching any

front-end tool with any SQL back end. The latter are simply becoming a commodity.

SQL is likely to be the dominant database language for servers for some time. The attraction of SQL is that it lets database applications issue multi-record requests to the server. The four primary SQL data-manipulation verbs - SELECT, INSERT, UPDATE, and DELETE - can all work on a group of records at a time. In addition, SQL is a data-definition language with which one can define databases. It offers commands for defining database tables and fields, as well as the security control on these items. SQL also lacks features that many critics consider important parts of the relational model. Despite this and other problems, SQL is clearly the database server language of today. All the major announced database servers support SQL. Both its technical virtues and the market forces behind it make SQL more popular.

Decision support system (DSS)

DSS has become a major area of study within information systems. DSS are interactive computer-based systems that utilize data and models for aiding an organizational decision-maker(s) in the semi-structured problems. DSSs are man-machine systems where man and machine communicate through an interactive interface [2]. Their focus is on semi-structured decision problems faced in an organizational context. Their role is to support a decision-maker to make decisions for him. They provide facilities which usually include data management capabilities for developing computer-based models for analyzing decision problems. The characteristics attributed to DSSs are:

- DSSs are information systems whose focus is decision making. In this sense they differ from EDP and MIS, which focus on data and information respectively.
- They rely on the man-machine working together for solving problems.
- They are support tools under the control of managers. They do not attempt to automate decision-making, predefine objectives or impose solutions.
- Their functional capabilities are provided more towards supporting decision-making activities rather than solving decision problems.
- They provide an environment where decision-making can be applied effectively to various kinds of management science methods.

DSS architecture is an active area of current research. The data management research is addressing the data handling requirements of DSS.

Though there are frequent overlaps with computer science DBMS research, there are also problems that are unique to the DSS context. For example, dealing with dynamically changing data, dynamic restructuring of data, time-series data, poorly structured data, unplanned and unanticipated data are unique in the DSS context. There is a quest for techniques that can unify data, models and knowledge

bases into homogeneous conceptual entities, so that they can be dealt with in a uniform manner [8]. Also, the model-base management is a unique research area of DSS. It is concerned with techniques for developing and configuring models and bringing a level of order to models that is similar to what database research has done to database management [1]. Dialogue management is another critical area of research in DSS, because DSSs are man-machine systems and system effectiveness depends critically on the success of man-machine interactions.

Other problems which are prominent in the general area of technology are: developing generalized architectures that can flexibly combine the various elements of a DSS during run-time, developing technologies, such as expert systems, uncertainty management techniques, non-monotonic reasoning advanced graphics and work stations, etc.

Distributed database management system (DDMS)

Distributed databases exist not only in theory, but in practice also. Two of the most talked about systems are *Ingres/star* and *Oracle*. *Ingres/star* is the distributed version of *Ingres*, one of the first relational database implementations. *Oracle* also offers distributed processing, but it does not provide as high a level of support as does *Ingres/star*. The distributed version of *Oracle* has no global intelligence overseeing the distributed database; rather it acts as a loose confederation of local databases linked by communications drivers. The distributed *ingres* architecture consists of three components : network nodes and their associated local databases, communications links between nodes, and distributed databases that can incorporate tables from multiple local databases. These distributed databases are the purview of *Ingres/star*.

One can link tables from local databases into a distributed database with the register as link or create link commands. These also create tables in the distributed databases. *Ingres/star* stores the table in a local database and places a link to it in the distributed database. *Ingres/star* designates the network node where one creates a distributed database to be the coordinator node for that database. This node then tracks the location of all the tables in the distributed database and manages the global data- dictionary table. *Ingres/star* has a very sophisticated query optimizer that can analyse network conditions with a high degree of precision. A distributed *Ingres/star* database can include local databases using any of these systems. The distributed database still looks like a single *ingres* database to its users.

Knowledge based expert system (KBES)

For the developments of a KBES, a knowledge of AI languages, such as LISP, PROLOG, etc. is required, which is a very time- consuming task. It is the expert system tool that incorporates a graphical interface and a text editor to integrate from the development and run-time stages. A specialized post-processor can be

used to convert OFF-LINE system to ON-LINE one. Intelligence is built into the post-processor to minimize the operator inputs. The expert system thus generated can be verified on a graphic screen.

The knowledge base is created by a knowledge acquisition programme. It consists of two interfaces, viz. construction interface and refinement interface. The construction interface is designed to provide a flexible framework for the developer to make changes and incrementally build up the knowledge base. Knowledge is generated by the interaction of the system with the developer (expert), who answers specific questions asked by the system or adds information which is relevant at a particular instance. Construction interface helps the developer to create a knowledge base and is initially empty to which knowledge can be added progressively.

The refinement interface facilitates integration of new rules with previously acquired knowledge and for performing high level analysis of the knowledge base. While creating the knowledge base, rules and data are drawn from different sources with the involvement of several experts. Though their mutual consistency is ensured by the construction interface, it is possible that the set of rules is not complete. This requires analysis and the refinement interface allows the developer to integrate the knowledge by adding or modifying the set of rules and data.

The knowledge base is represented by a binary tree structure where each node (question) divides into two branches (yes or no); while top-down traversing of the tree, the leaf of the tree contains a branching or a statement. A statement can also be followed by a branching or a question. The system also provides for indicating uncertainty and evidently supports in a situation where the developer is not very certain of a rule. This is usually necessary for heuristic rules which the domain expert draws from experiences. To represent uncertainty, a probabilistic weight factor called certainty factor (CF) is used. The conventional threshold level for cumulative CF is assumed to be 0.2. If it is found that the cumulative CF is less than the threshold value, a warning will be displayed on the screen.

The knowledge-based expert system generated by the tool gives its user accurate advice or correct solutions to his problems. This is done by a mechanism known as Knowledge Base Debugger. Debugger process checks that the knowledge base is correct and complete. Two dynamic data files are generated during the creation of the knowledge tree, and graphic representation of the knowledge base: textual representation of the knowledge tree, and graphic representation of the knowledge tree. A post-processor is a programme used to change the interface mechanism of the expert system. There are three interfaces with the knowledge base: keyboard, database or serial port.

The consultation programme offers advice to the end-user on a problem of a domain. The consultation programme draws out information about the problem by asking specific questions to the user and making decisions from the replies given. The decision is made on the basis of knowledge of facts in the knowledge.

base. There is a facility which enables the user to obtain the chain of reasoning which underlines the conclusion or advice obtained from the system.

Object-oriented databases

Object-oriented databases are designed to meet the data-handling needs of complex design applications, such as in oceanographic application. While the data modelling facilities of these systems resemble object-oriented programming languages, the database systems embedded persistence, concurrency, control, recovery, consistency management and a query language. In addition, object-oriented databases might design the data model somewhat differently from their language counterparts to effectively support database features. The history of database management is filled with proposals for competing data models. Each model has its own set of strengths and weaknesses. The object-oriented approach can unify some of these dissimilar approaches by providing a model that is based on abstraction, and that allows type designers to use whatever technique that best suits their applications as an implementation of basic functionality [9]. The object-oriented data model holds the promise of providing advanced data handling for today's increasingly complex applications in oceanographic environment.

ACKNOWLEDGEMENT

The authors are grateful to Dr. B.N. Desai, Director, for his help and encouragement. They are thankful to their colleagues for the willing help rendered by them.

REFERENCES

1. Applegale, .M., B Kansyuski and J.F. Mumamaker. 1988. Model Management System. Design for Decision Support, Decision Support System, 2, 1986.
2. Ariav, G. and S Weber. 1988. The contemporary in DSS88 transactions, TIME 1-3.
3. Desai, B N., R.M.S. Bhargava, J.S. Sarupria and G.V. Reddy. 1989. Present status of oceanographic studies and characteristics of the EEZ of India, Proc. International Ocean Technology Congress, Honolulu, 22-26 January.
4. Diakey, T.D. 1988. Recent advances and future directions in multi-disciplinary *in situ* oceanographic measurement system In: B.J.Rothschild (ed.), Toward a theory on biological, physical interactions in the world ocean: 555-598.
5. IOC. 1987. Manuals and guides No. 17, Vol. 2, A general formatting system for geo-referenced data, IOC/UNESCO, Paris.
6. Mark L. Van. Name and Bill Catchings. 1990. SQL: A database language SEQUEL to dbase, *Byte*. 14 (11): 175-182.
7. Sarupria, J.S. and P.D. Kunte. 1988. Integrated Inventory Information System (IIIS), Comput Soc. India Communications Jan.: 25-30.
8. Raghavan, S.A. and R. Chand. 1988. A perspective on Decision Support System, *J. Comput. Soc. India*, 18: 7-36.
9. Thomas Dave. 1989. What's in an object? *Byte*. 14 (3): 231-240.