Proceedings of the Third International Conference on Very Large Data Bases, Tokyo, JAPAN, October 6-8, 1977.

ARCHITECTURAL ISSUES IN DISTRIBUTED DATA BASE SYSTEMS

C. V. Ramamoorthy, G.S. Ho, T. Krishnarao and B. W. Wah

Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley, California 94720

ABSTRACT

The design of a distributed data base is a complex and difficult task requiring careful consideration of data organization, data distribution, user interface, updating/retrieval schemes, program/data placement, security policies and reliability issues. In this paper, we have discussed a design methodology which can be used to design a distributed data base. This design methodology is a systematic way to guide the designer in making decisions during the requirement process phase, the design process phase and the implementation phase. We conclude this paper by examining the architectural issues in the design of distributed data bases.

INTRODUCTION

The character, demand and economics of computer systems have been continuously changing and the need for new computer organizations and architectures seem unavoidable. Until recently major emphasis has been placed on maximizing the hardware utilization of uniprocessor systems or tightly coupled multiprocessor systems because of high hardware and associated software costs. However, with the recent developments in large scale integration logic and memory technology and with the exploding size and complexity of the application areas, the designers are encouraged to consider the distribution of processing, leading to distributed architectures.

Basically a Distributed Computer System (DCS) may be considered as an interconnection of digital systems called Processing Elements (PEs), each having certain processing capabilities and communicating with other. This definition encompasses a wide range of configurations from an uniprocessor system with different functional units to a multiplicity of general purpose computers (e.g. ARPANET). In general what people call "distributed systems" vary in character and scope from each other [RAM 76]. So far there is no accepted definition and basis for classifying these systems. In [JOS 76] Joseph gives a most comprehensive taxonomy of distributed systems.

There are several motivations for the development of distributed computer systems and among them the most important ones are i) modularity and structural implementation, ii) reliability and availability, iii) throughput and response time, iv) geographical distribution and resource sharing and v) application orientation. The modularity and structural implementation refers to the ease with which system modifications can be achieved. Modularity may refer both to functional and physical levels. Functional modularity provides a common basis for system design over a range of sizes. Physical modularity is the extent to which standard modules can be used with ease to achieve higher processing capabilities. Distributed processing approach with inherent capabilities of modularity reduce the developmental costs over a range of system sizes.

Another important motivation for distributed processing is the potential to provide more reliable systems with improved availability despite some system failures. Thus in a distributed processing system it is possible to provide a comprehensive error detection mechanism and isolate the faulty modules without impairing the entire system operation.

Improved throughput and response time can be a major consideration in certain applications. In a distributed system the throughput can be increased by exploiting parallelism. In a real time environment the DCS approach can meet the time constraints that might otherwise be impossible in a uniprocessor system.

Finally geographical distributions and resource sharing is another important motivation for many of today's distributed systems. In many of the applications such as airline reservations systems, processing nodes are dispersed geographically but working in cooperation with each other. In these applications the data base which is shared between various processing nodes may be centralized or distributed. In addition, several of the resources such as utility programs may also be shared by several users. The distribution of processing and/or data base provides improved response time and ability to handle large volumes of information which might otherwise be impossible in conventional systems.

Whatever may be the motivation, the word "distributed" refers to the distribution of either processing logic or functions or data or control or a combination of them. These factors together with the major motivation would dictate the architecture of the system. In the following sections we discuss the issues related to distributed data bases and the architectural considerations of distributed systems supporting DDBs.

DISTRIBUTED DATA BASES

A distributed data base (DDB), in the broadest sense, can be thought of as the data stored at different locations of a distributed system and can be considered to exist only when data elements at multiple locations are interrelated and/or there is a need to access data stored at some locations from another location. The rise of DDBs is the result of two trends in data processing -- growing need for large data base systems and recent advances in data communications. The prime concept of the generalized data base system is based on the definition of a data format to store the data and on a generalized data base management software for accessing the data. Due to the ever-increasing demand for on-line data processing, there is a need for decomposing very large data bases into physically/geographically dispersed units and for integrating existing data bases held in physically isolated nodes into a single, logically coherent data base that will then be available to each of the distributed nodes.

The main objective of any data base, whether distributed or not, is to provide a framework for integrating and sharing data across a community of users. The distributed data bases are motivated by the desire to decompose very large data bases into physically dispersed units and to integrate physically dispersed isolated systems for efficient management, and by the fast response requirements at the dispersed nodes.

There are three major components in a data base [BRA 76]. First, there is structured information or schema which describes the data structures and the validity criteria of the data. Second, there is the data itself. Finally, there are various programs or processes which control the operations of the data base. There is another structural component, the subschema, which describes the data base as user applications see it. The third component together with the structural component collectively form the Data Base Management System (DBMS) [FRY 76,BAC 75]. The DBMS allows data sharing among a community of users, while insuring the integrity of the data over time, and providing security against unauthorized access. It also provides the transparency of the data, by allowing the data to be stored in different formats in different parts of the system, thereby aiding in providing storage and retrieval efficiency. Finally, it provides an interface between the users and the system.

The data base can be classified according to how these components are being put together. In [ASC 74], Aschim proposes two classifications, the first is based on the number of DBMS in the network and the second is based on the centralization or decentralization of two of the data base components (the file directory and the data). Booth [BOO 76] classifies the DDBs into two structures, partitioned data bases and replicated data bases. A partitioned data base is one that has been decomposed into physically separate units distributed across multiple nodes of a computer network. The partitioning will normally be based on the distribution of access requirements. In a replicated data base, all or part of the data base is being replicated at multiple processing nodes. The amount of partitioning and replication will depend on the the architecture of the distributed system, the amount of traffic anticipated and other requirements such as reliability, security, etc.

ISSUES IN DISTRIBUTED DATA BASE SYSTEMS

There are several issues associated with the design of a data base. A good design involves judicious choice of various alternatives and trade-offs. In the following we will discuss some of the design issues and the associated architectural requirements.

Data Base Organization

A DBMS generally supports one of the data models: relational, hierarchical or network [DAT 75]. There are other models like the binary association model and the external set model which are not quite popular. The architecture of a distributed system is very much dependent on the type of organization since it affects the storage organization, access mechanisms and the communication requirements. The criteria for designing and selecting a model has not yet been established, nor is it likely to be established in the near future. The design of a distributed system supporting a distributed data base therefore encounters two decisions: which data model to utilize and how the data is to be structured for a chosen model [SIL 76].

Data Base Distribution

Data base distribution refers to the distribution of the data base components: schema, data and control programs. The architecture of the distributed system heavily depends on how these components are distributed among various nodes. However, their distribution is dependent on available communication facilities, secondary storage, nature of the application and the response time requirements [LOO 76].

User Interface

Another important issue in the design of a DCS supporting a DDB is the design of user interface for easy users' access to the data base. This interface basically consists of communication interface and a query language. This interface should help the users to find the exact location of the required data. The complexity of user interface depends on the required ease with which users access the data and there is a tradeoff between these two in the design of communication processors.

Updating/Retrieval

In a DDB where several users share the data, there are several problems associated with multiple accesses and updates. When several users try to access the common data, there would be interference between the users, and the communication protocols should be designed to minimeze this interference. Another problem related to consistency arises when data elements with multiple copies at different locations are to be updated. Simple locking mechanisms cause excessive delays and may cause throughput degradation in the distribbted system. Efficient updating schemes are needed and the architecture would be very much influenced by such schemes [ESW 76].

Program/Data Migration

In a distributed system the operation of the system can be improved by optimal placement of the data and programs so that the communication traffic can be reduced [MOR 77]. Such operation can be performed either by moving the procedures to the data or the data to the procedures, or a combination of these two methods. However an optimum placement of programs and data depends on the following: a) the extent to which common programs are used at all nodes, b) the volume of messages that would be transmitted in the system, and c) the data structures and usage. But frequent migration of data and program may introduce inconsistency and there is a tradeoff between the cost to maintain consistency and the cost to migrate a program or data to a node where it is frequently used.

Security and Privacy

Another important issue that influences the architecture of a DCS is security and privacy. Security refers to the protection of data against deliberate or accidental destruction, unauthorized access or modification of data. On the other hand, privacy refers to the right of an individual user to determine for himself what personal information to share with others as well as what infor- to receive from others. The threat to security and privacy increases as the size of data base increases. In addition, it is increasingly difficult to implement any measures in a DDB. Additional techniques such as data encryption would influence the transmission efficiency and the communication mechanisms.

Reliability

The determination of the needed hardware and data redundancy is another major issue in the design of a DDB. Multiple copies of data base realm offer fast recovery, checkpointing of realms, dumping and journal rollback and role-forward offers a slower but cheaper recovery. The effect of any recovery mechanism on the response time and the associated overhead must be weighted against the reliability requirements.

Standardization

One of the major inhibiting factors in the development of distributed systems supporting DDBs is the lack of standardization. Standardization can be established in the areas of programming languages, hardware (e.g. disk, tape), data formats, network protocols, user interface commands, etc. However, with an unknown and ever-changing technology at hand, standardization may cause costly refitting later and may even hinder acceptance of new ideas.

We have outlined some of the issues in the design of a distributed system supporting a DDB. However these issues are by no means complete and several other issues, both design and operational, have to be considered. All these issues together define the requirements for a distributed system. Based on these requirements one could proceed with the design following the proposed design methodology.

DESIGN METHODOLOGY OF DDBs

In this section, we will describe a systematic design methodology for the architecture of a distributed system supporting a DDB to reduce cost and to improve lead time, reliability and performance of the DCS [KUN 77,YEH 77]. The design methodology is able to:

i) provide an evolving system with controlled expansion and to preserve the integrity of the whole data base;
ii) represent effectively and efficiently the decision making constraints of a DCS by a specification language;
iii) provide a means for incorporating design alternatives and trade-offs at various design steps and design levels;
iv) provide design attributes and documentation for evolution, growth and modification so that changes can be made without reconsidering the whole design process.

The design methodology is aimed at designing the architecture of a broad spectrum of data base systems, e.g. relational and hierarchical data base systems. Its primary objective is to produce reliable, effective, modifiable systems with low costs and small lead time. The philosophy behind our approach is multi-functional multi-level modeling of a DCS. Basically, the methodology consists of four major phases, viz: i) requirement and specification phase, ii) design phase, iii) implementation and validation phase and iv) operation and maintenance phase (Figure 1). Each of these phases has its own unique characteristics with respect to engineering decisions, analysis and testability. In this paper, we will concentrate on the first two phases.

REQUIREMENT AND SPECIFICATION PHASE

The requirement and specification methodology basically starts with uniformly specified user requirements and elaborates on them in terms of the design specifications [THU 77]. This phase consists of four major steps, viz: i) requirements analysis, ii) requirements specification and iii) definition of attributes. Requirement analysis deals with the decomposition and translation of the user needs and objectives into system engineering terminology. This translates real world problems into functional structure of the system. For example, the overall requirements of a data base can be decomposed into data processing requirements, resource allocation and communication

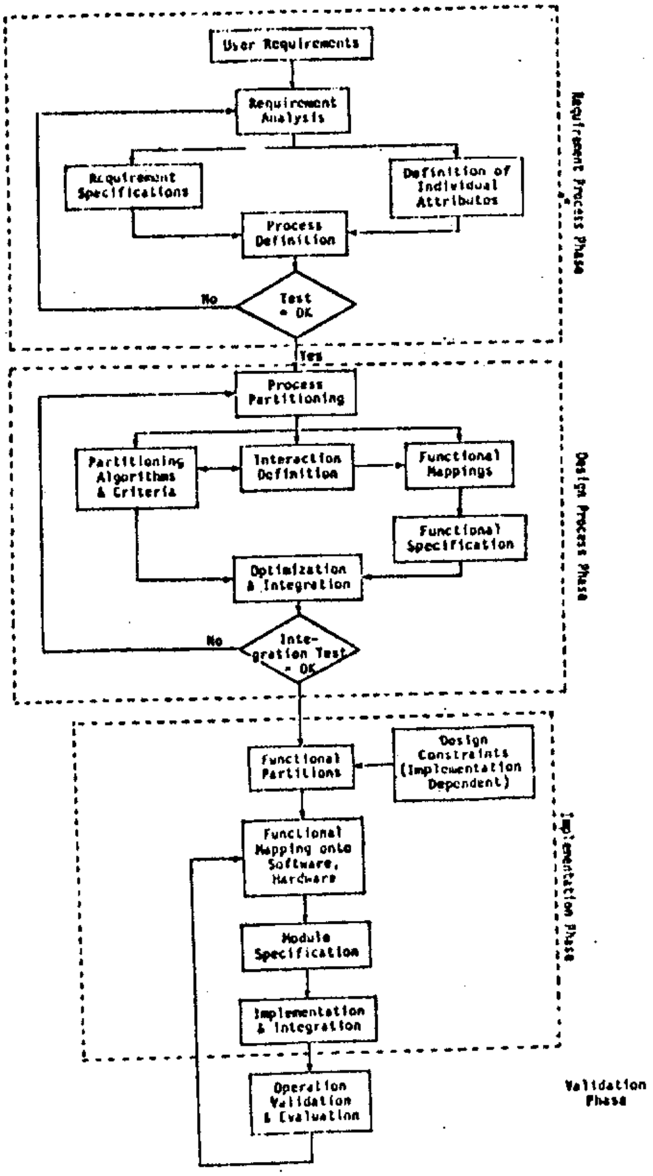
User Requirements
Requirement Analysis
Requirement Specifications
Definition of Individual Attributes
Process Definition
No
Test = OK
Requirement Process Phase
Yes
Process Partitioning
Partitioning Algorithms & Criteria
Interaction Definition
Functional Mappings
Functional Specification
Optimization & Integration
No
Integration Test = OK
Design Process Phase
Functional Partitions
Design Constraints (Implementation Dependent)
Functional Mapping onto Software, Hardware
Module Specification
Implementation & Integration
Implementation Phase
Operation Validation & Evaluation
Validation Phase

Figure 1. Design Methodology of Distributed Systems Supporting Distributed Data Base

requirements, reliability, availability requirements, etc. It also identifies individual attributes and performance parameters associated with the decomposed requirements. These requirements can be further decomposed into lower level functional requirements. For example, reliability/availability can be further decomposed into user requirements, recovery requirements, resource requirements and so on.

The second step, specification of the requirements, concerns with the logical aspects of the DCS and generates a set of specifications based on the analyzed requirements and the attributes. For successful execution of this phase, a formal way of specifying these requirements in terms of a specification language is used [CON 77]. The specification language enables the system designer to express the requirements in a natural way and to define the structure of the system.

The next step, the process design step, accepts the inputs from the requirements specification and attributes definition stages and identifies major functions to be performed. Based on this, the information flow in the system can be modelled and analyzed to identify major operations to be performed and their location of occurrence. Based on these identified operations, the features of the process can be defined to perform the required major operations.

DESIGN PHASE

The design phase starts with the defined processes which are the outputs of the requirement and specification methodology phase. The major steps involved in the design phase are process partitioning, functional specification and mapping, integration and optimization. Basically the design phase requires a provision to trace the system requirements through all the levels of design and a means of assessing tradeoffs at the functional level and comparing design attributes.

In the process partitioning phase the designer will decompose the system into a network of interacting tasks. The motivation for such partitioning is to increase modularity and testability and to decrease the interference. According to requirements of the system, the data files can be centralized in one node of the system or can be duplicated and distributed throughout the whole network. In the distributed case, data file integrity must be considered very carefully. The situation may be further complicated by the relaxation of the program/data file independence arising from the requirements [CAS 72,LEV 75]. This happens when both the relevant program and the relevant file must be accessed in order to process the given transaction and they are in different nodes. For example, a given transaction initiated at node i has to be processed by the relevant program at node j which in turn has to access data files at node k. As a result, different assignments of programs and data files to computer sites will yield different distributions of requests and therefore different traffic load and communication overhead to the whole system. In a heterogeneous computer network, such as the ARPANET, the distribution of program is expecially critical. The problems of compatability of programs and data file structures of different machines must be standardized so that the users view the system as a homogeneous data base.

ARCHITECTURAL CONSIDERATION IN THE DESIGN OF DDBs

The above described methodology is a generalized approach to design distributed architectures to support data base systems. However the specific architectural features required and the specific strategies adopted depend on the particular issues and their relative significance with respect to the overall design objectives. The distributed system supporting DDB can be divided into three major subsystems. The first one is the processing node subsystem which would enable the data to be stored and processed. The characteristics of the processing nodes are dictated by the nature and volume of the data and job load. This subsystem often consists of processors of diversified characteristics and is often evolved either from the overall design or from existing heterogeneous processing sites. The second major subsystem is related to the data model, data organization and storage. Currently almost all the existing data bases are based on three models: relational, hierarchical and network [DAT 75]. In all these models the data organization is built on three basic types of organization: sequential, random and list; and the choice of a specific organization is dictated by the cost, space, and response times requirements. These data base models, types of data organizations, the volume of information and the cost determine the storage and memory technology adopted.

Finally the third major subsystem is the data communications system. The data communications system links terminal devices, transmission channels, and one or more computers to provide on-line data flow between several stations. The basic elements of such communication subsystems are the terminals through which the users communicate, the communication interfaces called modems which are used to connect the terminals to the communication links, and the transmission links that connect various computers in the system. There also exist communication software systems that handle communication between various users and computers in a reliable manner maintaining the integrity and security of the data being accessed. The characteristics of this communication system such as communication link capacities, cost, reliability, transparency of data transfer and interpretation mechanisms for the users, dictate the overall system architecture. From the operation point of view it is the communication subsystem that handles the issues related to routing of the queries and the associated responses, multiple requests and multiple updates, security and privacy of the users' access rights, data compression, and code conversion, etc.

The routing of queries and the associated responses depends on the availability of communication links and the associated communication protocols. The decisions associated with the routing are often decentralized in a distributed system. The status information regarding source and destination and any information related to routing is appended to the messages. In certain distributed data base systems this routing of queries and responses is handled by a separate message handling processors [FOS 76]. A message handling processor is associated with each of the nodes in the system and provides an interface between the users and the system (Figure 2).

The next important issue that is handled by the communication subsystem is the multiple requests and multiple updates. In order to maintain the consistency of the data, locking mechanisms are needed on the access paths while updating is being carried out. The specific approach adopted to solve the multiple updates problem is constrained by the communication requirements, response time requirements and whether or not data inconsistency is allowed for a shorter interval of time. The inconsistency is likely to be predominant in cases where there are multiple copies of data and the data base control is distributed among the several nodes. The approach to this problem is proposed by Mullery [MUL 75] where a locking mechanism is used with three possible states: available, prepared and locked. The conflicts are resolved by the protocol while the data is put in an intermediate state called prepared state.

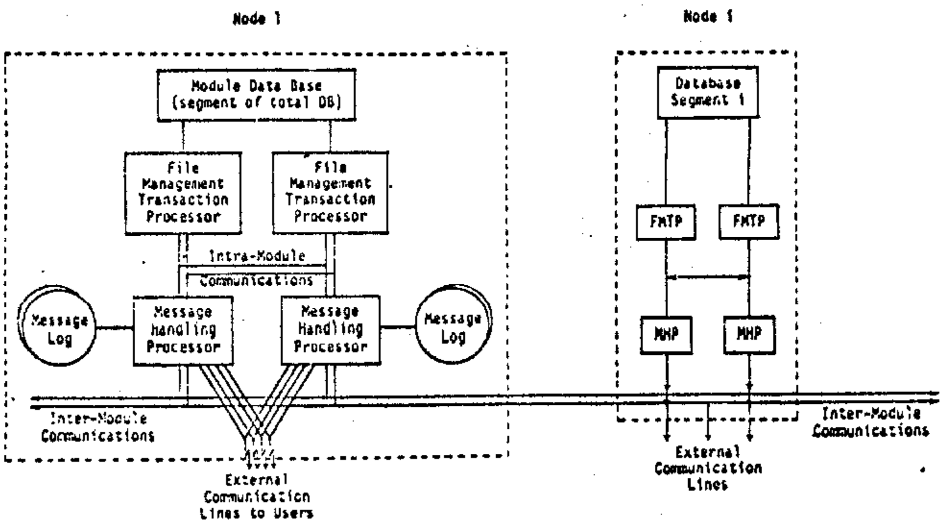
Node 1
Node i
Module Data Base (segment of total DB)
File Management Transaction Processor
File Management Transaction Processor
Intra-Module
Communications
Message Log
Message Handling Processor
Message Handling Processor
Message Log
Inter-Module Communications
External Communication Lines to Users
Database Segment i
FMTP
FMTP
MHP
MHP
Inter-Module Communications
External Communication Lines

Figure 2. Simplified Distributive Computing Facility Processing Module (Reference [FOS 76]).

Another important aspect of communication subsystem is the security and privacy of data being accessed. This is a very important consideration especially in a large distributed data base system where centralized access control is not possible. The problem of security of data may be accidental or deliberate. Accidental threat to privacy and security can be handled by providing redundancy and checking circuits at the storage level. However, the communication subsystem could provide a means to counteract the deliberate threat, by defining access control and processing restrictions, privacy transformations and by monitoring the procedures. With the recent development of powerful low cost microprocessors, the access control tasks can be implemented using microprocessors as front end processors in the communication subsystem.

CONCLUSIONS

In this paper, we have looked at some of the issues of a distributed data base. In particular, we have discussed some of the architectural requirements in its design. These requirements, when coupled with the requirements of the distributed system, will give the complete description of the requirements of the system. The issues we have discussed are by no means complete. New issues will arise during the design and operational phase of the system. The design methodology discussed in section IV alleviate some of the problems that arise when new issues come up by providing a systematic way for expansion and evolution.

REFERENCES

[ASC 74] Aschim, Frode, "Data-Base Networks--An Overview," Management Information, Vol. 3, No. 1, 1974.

[BAC 75] Bachman, C., "Trends in Data-Base Management," Proc. of AFIPS National Computer Conf., 1975, Vol. 44, AFIPS Press, Montvale, NJ, 1975, pp. 569-576.

[BOO 76] Booth, G. M., "Distributed Databases--Their Structure and Use," Infotech State of Art Report on Distributed Systems, 1976.

[BRA 76] Bray, Olin H., "Distributed Database Design Considerations," Trends and Application, Computer Networks, 1976.

[CAS 72] Casey, R. G., "Allocation of Copies of a File in an Information Network," Proc. AFIPS 1972, Vol. 40, AFIPS Press, Montvale, NJ, 1972.

[CON 77] Conn, A. P., "Specification of Reliable Large Scale Software Systems," Ph.D. Dissertation, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1977 (in progress).

[DAT 75] Date, C. J., An Introduction to Database Systems, Addison-Wesley Publishers, 1975.

[ESW 76] Eswaran, K. P., Gray, J. N., Lorie, R. A. and Traiger, I. L., "The Notions of Consistency and Predicate Locks in a Data Base System," CACM, Vol. 19, No. 11, Nov. 1976, pp. 624-633.

[FOS 76] Foster, J. D., "The Development of a Concept for Distributive Processing," COMPCON, Feb. 1976.

[FRY 76] Fry, J. P., and Sibley, E. H., "Evolution of Data-Base Management Systems," Computing Surveys, Vol. 8, No. 1, Mar. 1976, pp. 7-42.

[JOS 76] Joseph, E. C., "Distributed Processing Architectures--Past, Present and Future Trends," Infotech State of Art Report on Distributed Systems, 1976.

[KUN 77] Kunii, T. L., and Kunii, H. S., "Database Design," Proc. of the Tenth Hawaii Intl. Conf. on System Sciences, Western Periodicals Company, 1977.

[LEV 75] Levin, K. D., and Morgan, H. L., "Optimizing Distributed Data Bases--A Framework for Research," Proc. AFIPS 1975, Vol. 44, APIPS Press, Montvale, NJ, 1975.

[LOO 76] Loomis, M. S., and Popek, G. J., "A Model for Data Base Distribution," Symp. on Trends and Applications 1976, Computer Networks, IEEE 1976, pp. 162-169.

[MOR 77] Morgan, H. L., and Levin, K. D., "Optimal Program and Data Locations in Computer Networks," CACM, Vol. 20, No. 5, May 1977, pp. 315-322.

[MUL 76] Mullery, A. P., "The Distributed Control of Multiple Copies of DATA," IBM Research Report RC-5278, Yorktown Heights, 1975.

[RAM 76] Ramamoorthy, C. V., and Krishnarao, T., "The Design Issues in Distributed Computer Systems," Infotech State of Art Report on Distributed Systems, 1976.

[SIL 76] Siler, K. F., "A Stochastic Evaluation Model for Database Organizations in Data Retrieval Systems," CACM, Vol. 19, No. 2, Feb. 1976, pp. 84-95.

[THU 77] Thurber, K. J., "Techniques for Requirements-oriented Design," Proc. AFIPS 1977, Vol. 46, APIPS Press, Montvale, NJ, 1977.

[YEH 77] Yeh, R. T., and Baker, J., "A Hierarchical Design Methodology for Data Base Systems," TR-70, University of Texas at Austin, 1977.

ACKNOWLEDGEMENT

Research supported by National Science Foundation Grant MCS72-03734-A02.