DATA BASE DESIGN IN THE DISTRIBUTED ENVIRONMENT

C. Mohan

May 1979 TR-131

Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

DATA BASE DESIGN IN THE DISTRIBUTED ENVIRONMENT*

C. MOHAN

Software and Data Base Engineering Group
Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

Technical Report TR-131
16 May 1979

ABSTRACT

The potentials of distributed data base (DDB) management are currently being explored very vigorously. Many technical and organizational issues still need to be sufficiently understood. The 80's will see rapid progress in the exploitation of DDB's. This paper examines some of the constraints placed on a DDB design by the mechanisms proposed for handling DDB management problems. A DDB design framework consisting of many phases is presented and is illustrated in the context of the now popular DDB system SDD-1. Next the problems of DDB administration are analyzed. Some new administrative roles are proposed and the duties of the new data administrators are outlined.

CONTENTS

---

*This work was supported in part by the Air Force Office of Scientific Research under grant AFOSR77-3409.

1.0 INTRODUCTION

With the emergence of local and distributed networks, distributed data base (DDB) management has been gaining more and more importance and has become one of the key areas of research [Adiba 1978, Mohan 1979d]. Progress in this area has not been dramatic. A full understanding of the relevant issues is yet to be achieved. There are still many unresolved issues that require additional research. Because of the lack of experience in the usage of distributed data base systems (DDBS) many of the tradeoffs involved are unclear. Over the past several years research and, to some extent, development have primarily centered on the technical issues of DDB management. This paper addresses one of the central issues of DDB management: distributed data base design (DDBD) and administration. Many phases of the DDBD process and the responsibilities of DDB administrators have been identified.

Some coherent set of principles, guidelines, techniques and integrated tools need to be developed and established to aid the distributed data base administrator (DDBA) in the performance of his tasks (some initial work in specifying the support functions for the DDBA can be found in [Bunch 1975]). The DDBA will also be refered to as the Global Enterprise Administrator (GEA).

While many researchers have dealt with specific problems in the design of DDB, to my knowledge so far there has not been any attempt to develop a systematic methodology for carrying out such designs. At present, even a framework for carrying out DDBD does not exist. Sufficient attention has not been paid to the development of such concepts and methodologies. Some initial ideas have been presented in [Mohan 1979c]. Any proposed methodology should be suited to the design of data bases for general purpose DDBS, rather than for specialized DDBSs. Very little work has been done in the development of heterogeneous DDBSs. Hence almost no work has been reported on problems involved in the design of data bases for such systems. According to [Adiba 1978] even heterogeneity is a natural consequence of users' requirements.

Data base design for a particular DDBS should take into consideration not only the data requirements at the different sites, the performance requirements and (possibly) the security requirements, but also the characteristics of the DDBS (like the query processing strategy, and synchronization mechanisms employed by the system). The latter, to a great extent, determine the performance of the system for a given data base. Proper distribution could hopefully lead to reduced inter-site traffic and increased parallelism in query processing. Data base design should also take into consideration the reliability and the recovery mechanisms of the DDBS. A thorough understanding of these aspects of the system would lead to good and efficient

data base designs.

Work needs to be done for determining what statistics need to be collected and for devising mechanisms and methodologies for gathering and analyzing those statistics, in an operational system, for aiding the DDBA so that he would be able to reconfigure the distributed data base. These statistics could be used for load balancing and simulation also. While the designs described in [Bunch 1975, Chang 1977, Neuhold 1977, Pardo 1977, Small 1979] mention the existence of a statistics collection component enough analysis has not been done about its exact nature and operation.

I will now illustrate my earlier remark concerning the impact of the characteristics of the DDBS on the data base design process. The peculiar design features of some DDBS pose some additional data base design problems. In the case of IDER [Lien 1978] the concept of a cluster has been introduced. The system requires that each cluster be fully resident in only one site and that each query can deal with only one cluster at a time. The introduction of the cluster concept leads to a complication in the data base design process. The designer has to somehow decide on how to group the data into clusters and the users have to be aware (when they formulate queries) of what data items are present in each cluster so that they don't violate the above restriction. Although IDER provides network transparent access the above constraints more or less dictate that the users be aware of where what data is located (which is the equivalent of saying that the network is not transparent to the users). The data base design considerations resulting from the introduction of transaction classes in SDD-1 [CCA 1979, Rothnie 1979] have been analyzed in [Mohan 1979a].

2.0 DESIGN PHASES

The distributed data base design (DDBD) process can be viewed as consisting of the following phases (See Figure 1):

I. Elaboration and Synthesis

II. Decomposition

III. Allocation

IV. Preliminary Analysis
   1. Reallocation (Return to Phase III)
   2. Repartitioning (Return to Phase II)

V. Implementation

VI. Usage

VII. Reconfiguration
1. Reallocation (Return to Phase III)
2. Repartitioning (Return to Phase II)
3. Evolution (Return to Phase I)

The above design process is applicable to the environment in which a DDBS and an underlying computer network already exist, and in which the individual data bases at the different sites do not preexist. The rest of this section discusses the various issues that need to be taken into consideration in the various phases of the design process. It has to be noted that the issues of the various phases are interrelated. This means that the decisions made in the various phases would affect each other. The DDB designer has to choose from the many ways in which a DDB could be organized.

The activities in Phase I (Elaboration and Synthesis) result in the definition of the GCS (Global Conceptual Schema). The data model chosen for the definition of this schema depends upon the DDBS that one is intending to use. Since one would expect a DDB to comprise of a very large number of data items (i.e. the size of the schema itself is likely to be very large), it would be prudent to start using a data dictionary in this phase. The usage of the data dictionary would greatly simplify the schema definition task. What things are included in the schema would be dependent upon the intended applications that would be using the DDB. A great deal of systems analysis work must be done to determine what should be included and what should not be included. Even though a GEA (Global Enterprise Administrator or Distributed DBA) may be defining the schema, he would have to consult with the LEAs (Local Enterprise Administrators) at the different sites where parts of the data base would exist. A brief discussion of the responsibilities of GEAs and LEAs can be found in Section 4.0. The GEA should analyze the enterprise's activities/functions and elaborate its information flows and usage. The latter should be used as the basis for the synthesis of the GCS.

The II Phase, Decomposition, is meant to partition/fragment the GCS into smaller pieces. This means that the partitions/fragments would be the units of data distribution in the network. A group of data items being present in a fragment is intended to reflect the knowledge/perception that accesses to those data items would be mostly in a clustered fashion. In most DDBS designs (like Distributed INGRES [Stonebraker 1977], POREL [Neuhold 1977], SDD-1, VDN [Munz 1978], etc.) that use the relational model, the

partitions are specified using simple predicates. Two of the criteria used in making the partitions would be the data accessing and updating characteristics of the different applications in the network. The organizational structure of the enterprise would also have an impact on the partitioning decisions. The size of the partitions would depend upon the granularity of the data references made by the applications. The greater the number of partitions (in general) the greater would be the time required to determine as to which partitions are involved in a given transaction. Whether the partitions are overlapping or non-overlapping would be dependent upon the DDBS to be used. The relative merits of overlapping and non-overlapping partitions have been discussed in [Schweppe 1977]. The information about the partitions should also be added to the data dictionary.

During Allocation, which is the III Phase, the fragments are allocated to the different sites of the network. The choice of sites for holding parts of the DDBS is expected to be the result of managerial decisions of the enterprise. Depending upon the DDBS that is to be used, the fragments may be allocated redundantly, i.e. a given fragment may be stored at more than one site. Some of the different ways of performing this allocation have been analyzed in [Ries 1978]. The allocation decisions would be influenced by the data accessing characteristics of the specific sites involved. The characteristics of the network (like the topology, the distance between sites, the type of switching done, etc.) would also have a significant impact on them. Also relevant would be the reliability constraints, availability requirements and response time requirements imposed on the data base, externally. As would be obvious, increasing the redundancy increases the time required for performing updates of those fragments. One of the objectives in carrying out the allocation should be to distribute the fragments in such a way that the chances of any sites becoming bottlenecks are minimal. Efforts should also be made to reduce unnecessary message/data traffic on the network. When this phase is completed, the definitions of the LCSs (Local Conceptual Schemas) of the different sites would have been performed. Once this is over, the directories, which give information about the locations of data, could be prepared. A brief outline of an algorithm for determining the sites involved in a transaction, when the data distribution is expressed in terms of predicates, has been presented in [Steyer 1977].

In the IV Phase (Preliminary Analysis) a mathematical/simulation/hybrid model of the existing allocation is developed and is analyzed to measure design effectiveness. This analysis would be at a gross level, since the implementation of the LCSs at the individual sites would not be considered. This analysis would necessitate a characterization of the workload that would be applied to the DDBS. One of us is working on a simulator which might be

useful for this type of analysis. Based on such an analysis it should be possible to predict the likely (approximate) future behaviour, in terms of certain performance measures. When these predictions are compared with desirable ranges of values for these measures, some changes may be deemed to be necessary. These may result in reallocation (i.e. a return to Phase III) or repartitioning (i.e. a return to Phase II). This analysis may also be used to determine 'optimal' values for certain parameters which would be set in the subsequent phases.

In the next Phase (Implementation) the implementation of the LCSs is done. This involves deciding as to how and where exactly each site's data base would be stored. Decisions pertaining to access paths, storage structures, etc. must be made. The activities of this V Phase are similar to what goes on during a centralized data base design (See [Mohan 1979b, Yeh 1978]) and are expected to be performed by the LEA at each site. During the begining of this phase, a translation of the LCS from one data model to another may have to be made, depending on the DDBS to be used (for e.g. if the DDBS is a heterogeneous federation).

During the Usage Phase, the DDB is loaded and the application programs, and terminal users start using it. As the usage of the DDB goes on, performance and usage statistics would be collected. This assumes the existence of system activity monitoring facilities in the DDBS. The statistics that are collected would help the GEA in tuning the DDB, so that the retrievals and updates are more efficient.

Such tuning activities take place in the VII Phase - Reconfiguration. Three types of changes may be found to be necessary. Some or all of them might be induced by the dynamism of the environment in which the DDB is being used. One would be Reallocation: that is, changing the locations of the existing fragments and/or increasing/decreasing the amount of redundancy. This means that Phase III and the subsequent phases would be repeated. This would necessitate changes to the directory entries.

The next type of change is Repartitioning: that is, redefining the fragments. This may result in increased/decreased/same number of fragments compared to the existing situation. Repartitioning may be found to be necessary if it is noticed that access to more than one fragment occurs in a clustered fashion very often. In such cases it may be worthwhile to combine the relevant fragments into a single fragment. If the accesses to a fragment are such that most of the time only non-overlapping parts of the fragment are needed, then it may be of benefit to split that fragment into two or more fragments. Introduction of new units/divisions or abolition of existing units/divisions in the enterprise could also lead to repartitioning. Another

reason might be that the initial partitioning was done based on guessed usage patterns. Once the usage of the DDB was observed, one would be in a better position to do a good job in partitioning the data base. Repartitioning would lead to the repetition of Phase II and the subsequent phases. One of the factors that make reallocation and/or repartitioning necessary are changes in the users' performance requirements.

The third type of change would result from the evolution of the enterprise's applications and the growth of the enterprise - i.e. the functonal requirements of the enterprise change. Accomodating them would warrant modifications to the GCS itself. Some data items may no longer be needed and some new data items may have to be included. Thus evolution leads to the repetition of Phase I and the subsequent phases.

3.0 A DESIGN CONTEXT

To be more specific I would like to outline the various activities that would be performed, in the different phases, in the context of a particular DDBS. The system that I have chosen to consider is SDD-1 [CCA 1979, Mohan 1979a, Rothnie 1979]. This relational system is being designed and implemented (on the ARPANET) by the Computer Corporation of America. A complete prototype system, for the DEC-10 and DEC-20 computers, is expected to be released in 1979. Compared to other DDBS designs, SDD-1 has some novel concepts and features embedded in its design. The major assumption which forms the basis of SDD-1's design is that the types of transactions that account for most of the data base activity are predictable.

Phase I: Three main activities would take place in this phase. The first one would be the definition of the DDB schema in the form of relations. The second activity would be the definition of the transaction classes (TCs). The predictions about the expected data base activity are embodied in the TC definitions. The TCs are like transaction schemas: an instance of a TC would be a transaction. A TC is defined by specifying the logical data items that would be read (the read set) and those that would be written (the write set) by a transaction of that class. The read and write sets are expected to be defined using simple predicates. The third activity would be drawing and analyzing the conflict graph (CG) to determine the synchronization requirements of the different TCs.

Phase II: This phase involves the definition of the logical fragments which would be the units of data distribution. These definitions again are expected to be expressed using simple predicates. In SDD-1 a fragment is obtained by first

applying a restriction on the tuples of a relation and then projecting on the tuple identifier and another column.

Phase III: Two main activities would take place in this phase. During the first activity the data base designer decides on how to distribute the logical fragments amongst the sites of the network. Once this is done the directories can be generated (The SDD-1 directories are expected to contain relation and fragment definitions, fragment locations and usage statistics). Since SDD-1 treats directories also in the same way as data base data, this phase concerns decisions regarding how the directories also would be distributed amongst the sites of the network. During the second activity the number of TMs (Transaction Modules) that would be present is determined (The execution of all transactions belonging to a particular TC would be controlled by a particular TM and each TM could be present at no more than one site). Decisions are also made concerning which TC/TCs would be handled by which TM and at which site which TM/TMs would be located.

Phase IV: The design resulting from the activities of the three previous phases can be subjected to certain analyses, to predict the performance of the design. If simulation is chosen as the analysis technique, then that would require the existence of a simulator to model the operation of SDD-1's concurrency control mechanism. Currently I am planning to build such a simulator. A variety of performance measures can be used to analyze the effectiveness of the design. Mathematical analysis of the design is highly likely to be very difficult.

Phase V: This phase involves storing the fragments using the data modules (DMs) at the different sites and filling up the TMs' tables, with the synchronization requirements (what protocols to run against which TCs). Certain parameters concerning time stamps, data reduction and access characteristics of transactions (for use by the query processing algorithm), and frequency of issue of NULLWRITE and PROBE messages would also be set. Since SDD-1 is a homogeneous DDBS (i.e. the same data model is used at all sites) no translation of the LCSs from the relational model to some other model need to be performed.

Phase VI: During the usage of the DDB all kinds of statistics could be collected: inter-TM traffic (traffic generated due to the receipt of a transaction by a TM which is not meant to handle that transaction's TC), number of transactions executed by each TM, the TM-DM traffic, the access/update frequencies at the different DMs, the transaction arrival patterns, the PROBE and NULLWRITE message traffic, etc. As of now no desings exist for mechanisms for monitoring the performance and collecting such statistics in SDD-1. The design of these mechanisms should be a fairly easy task.

Phase VII: The statistics collected in the previous phase are analyzed and the results are used in performing reconfiguration. As should be evident from my classification of the SDD-1 data base design activities into the above phases, the redistribution of the fragments does not necessitate the redrawing of the conflict graph (CG). Only a reclassification of the existing TCs or the introduction of new TCs would result in the redrawing of the CG.

4.0 DISTRIBUTED DATA BASE ADMINISTRATION

A proper administrative apparatus is critical to the success of DDB management. The importance of this organizational setup cannot be neglected. Technical advances in the DDB area need to be complemented with the development of approaches to DDB administration. In this section I have outlined some of my ideas about the latter.

4.1 Roles And Responsibilities

The role of the data base administrator (DBA) has been more or less well defined in the literature. [De Blasis 1978] states that DBA's are individuals and/or teams who perform the functions of planning, designing, documenting, operating, coordinating and controlling the data base of an organization at both the policy and operational levels. Practical experience has been gained in employing DBAs in organizations (The results of a survey on DBA functions can be found in [De Blasis 1978]). The control of the definition and description of data in the DDB context requires the creation of some new roles. Some initial ideas can be found in [Booth 1977, Joyce 1977]. In this section I have identified two such roles: the Global Enterprise Administrator (GEA) and the Local Enterprise Administrator (LEA). I have sketched the scope of the responsibilities of these two types of administrators. It needs to be pointed out that each role need not be played by only one person but could be shared by many people.

The possible responsibilities of a global enterprise administrator (GEA) or network DBA are:

1. Interact with local enterprise administrators (LEA) and coordinate their activities. Analyze data requirements of different groups.

2. Define logical structure of DDB using a global schema definition language.

3. Specify integrity and security constraints.

4. Define logical structure of data base in each site, given an existing communication network. This results in schema partitioning and/or replication. If communication network does not exist define the most appropriate network topology and other needed network characteristics.

5. Formulate ways in which data would be searched for, acquired and validated.

6. If the DDBS has parameters (as characterized in [Mohan 1979d]) that could be specified at DDB installation time, then specify those.

7. In the case of SDD-1, the GEA has to decide about transaction classes, materializations, and then draw the conflict graphs and determine protocols to be used by different classes.

8. Decide on actions to be taken in case of site failures, network partitioning, etc.

9. Monitor the DDB usage and performance, and modify data distribution accordingly.

10. Determine (i) level of security to be enforced at different sites, (ii) restrictions on acceses to data at different sites by each of the sites in the network and (iii) whether minor inconsistencies could be tolerated.

11. Determine the extent of local control over data that should be permitted.

12. Maintain utilities for loading, restructuring and reorganization of data bases.

13. Loading, maintaining and modification of data dictionary facilities.

14. Educate users about the usage of the system.

The GEA should not only be well versed with data base concepts, but also with computer network concepts. He must be much more sophisticated than a DBA in a centralized environment.

The possible responsibilities of a local enterprise administrator (LEA) are:

1. Interact with the GEA.

2. Interact with other LEAs.

3. Interact with application administrators in his site.

4. Design and implement the local schemas. If necessary, perform translation from global schema's data model to local site's data model.

5. Develop guidelines, procedures, standards and data base description for proper usage of the data base.

4.2 Data Dictionary

In order for the data base administration function to fulfil its specified role in the control, coordination and administration of data there is a great need for a data dictionary.

This need has been recognized in the centralized data base environment and considerable work has been done [Butler 1977, Clark 1976, DDSWP 1977, Ehrensberger 1977, Sharman 1978]. The usage requirements for a data dictionary and the scope of its operations, in the DDB environment, are yet to be studied. Some initial ideas have been presented in [Joyce 1977].

5.0 REFERENCES

1. Adiba, M., et. al. [1978] Issues in Distributed Data Base Management Systems: A Technical Overview, Proc. IV Int. Conf. on VLDB, September.

2. Booth, G. M. [1977] Distributed Data Bases, In Distributed Processing, INFOTECH State of the Art Report, Infotech International.

3. Bunch, S., et. al. [1975] Research in Network Data Management and Resource Sharing: Preliminary Experimental System Design Report, CAC Doc. No. 170, University of Illinois, August.

4. Butler, D. [1977] Distributed Processing: Implementation Experience, In Distributed Processing, Infotech State of the Art Report, Infotech International, England.

5. CCA [1979] A Distributed Database Management System for Command and Control Applications: Semi-Annual Technical Report 4, Tech. Rep. CCA-79-12, Computer Corporation of America, January.

6. Chang, S.K., McCormick, B.H. [1977] Intelligent Coupling of the User to Distributed Database Systems, Tech. Rep. KSL-3, Univ. of Illinois at Chicago Circle.

7. Clark, I. A. [1976] Relational Data Dictionary Implementation, In Data Base Systems, H. Hasselmeier, W. G. Spruth (Eds.), Lecture Notes in Computer Science - Volume 39, Springer Verlag.

8. DDSWP [1977] The British Computer Society Data Dictionary Systems Working Party Report, ACM SIGMOD Record, December.

9. De Blasis, J-P., Johnson, T. H. [1978] Review of Database Administrators Functions from a Survey, Proc. ACM/SIGMOD Int. Conf. on Mgmt. of Data, May-June.

10. Ehrensberger, M. [1977] Data Dictionary - More on the Impossible Dream, Proc. NCC.

11. Joyce, J. [1977] Software Requirements for Distributed Data Base Management, In Distributed Processing, INFOTECH State of the Art Report, Infotech International.

12. Lien, Y. E., Ying, J. H. [1978] Design of a Distributed Entity-Relationship Database System, Proc. COMPSAC'78, November.

13. Mohan, C. [1979a] An Analysis of the Design of SDD-1: A System for Distributed Data Bases, In Distributed Data Bases, INFOTECH State of the Art Report, Infotech International (Also Tech. Rep. SDBEG-11, Software and Data Base Engineering Group, Univ. of Texas at Austin, April).

14. Mohan, C. [1979b] Some Notes on Multi-Level Data Base Design, Technical Report TR-128, Software and Data Base Engineering Group, Univ. of Texas at Austin, May.

15. Mohan, C., Yeh, R. T. [1979c] Distributed Data Base Systems - A Framework for Data Base Design, In Distributed Data Bases, INFOTECH State of the Art Report, Infotech International (Also Tech. Rep. SDBEG-10, Software and Data Base Engineering Group, University of Texas at Austin, April).

16. Mohan, C. [1979d] Distributed Data Base Management: Some Thoughts and Analyses, Technical Report TR-129, Software and Data Base Engineering Group, Univ. of Texas at Austin, May.

17. Munz, R. [1978] Gross Architecture of the Distributed Database System VDN, VDN Report 15/78, Technical University of Berlin, W. Germany.

18. Neuhold, E., Biller, H. [1977] POREL: A Distributed Data Base on an Inhomogeneous Computer Network, Proc. III Int. Conf. on VLDB, October.

19. Pardo, R., et. al. [1977] Distributed Services in Computer Networks: Designing the Distributed Loop Data Base System (DLDBS), Proc. Computer Networks Symposium, December.

20. Ries, D., Epstein, R. [1978] Evaluation of Distribution Criteria for Distributed Data Base Systems, Memo No. UCB/ERL M78/22, Univ. of California - Berkeley, May.

21. Rothnie, J., et. al. [1979] SDD-1: A System for Distributed Databases, Tech. Rep. CCA-02-79, Computer Corporation of America, January.

22. Schweppe, H. [1977] On Different Classes of Predicates for Distributing Data, VDN Report 4/77, Technical University of Berlin, W. Germany.

23. Sharman, G., Winterbottom, N. [1978] The Data Dictionary Facilities of NDB, Proc. IV Int. Conf. on VLDB, September.

24. Small, D., Chu, W. [1979] A Distributed Data Base Architecture for Data Processing in a Dynamic Environment, Proc. COMPCON'79 Spring, March.

25. Steyer, F. [1977] Processing Queries in Distributed Database Systems, VDN Report 3/77, Technical University of Berlin, W. Germany.

26. Stonebraker, M., Neuhold, E. [1977] A Distributed Data Base Version of INGRES, Proc. II Berkeley Workshop on Distributed Data Management and Computer Networks, May.

27. Yeh, R., Chang, P., Mohan, C. [1978] A Multi-Level Approach to Data Base Design, Proc. COMPSAC'78, November.

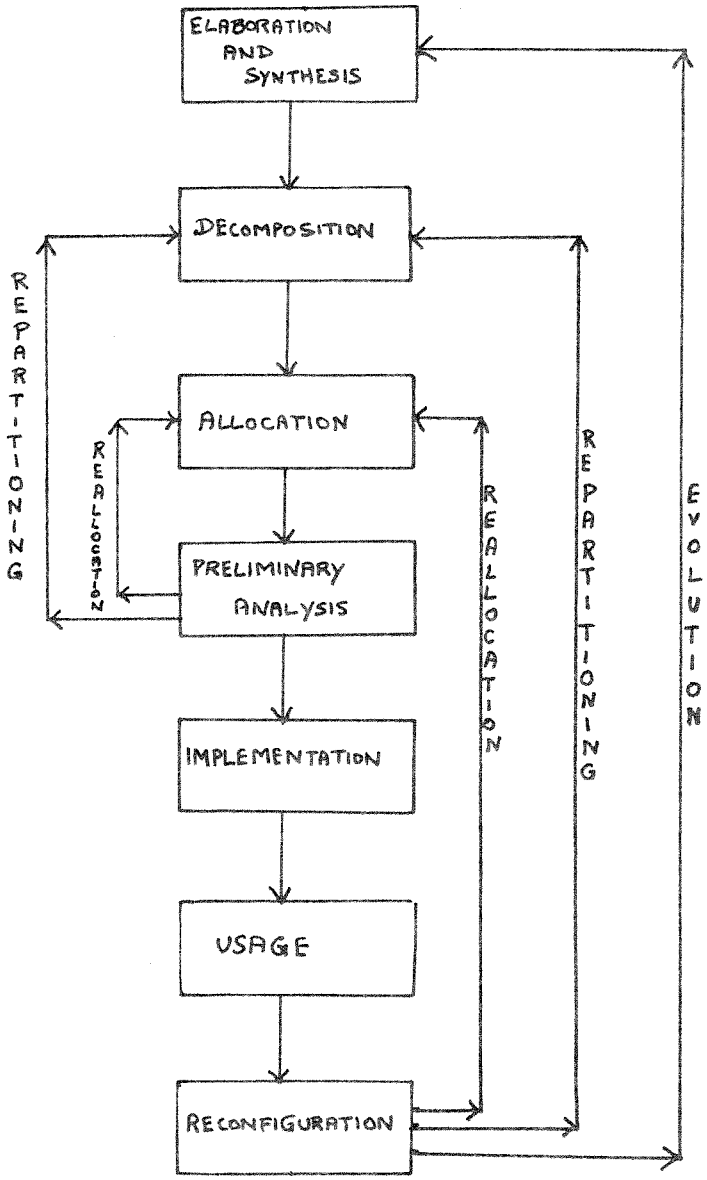

Fig.1. DISTRIBUTED DATA BASE DESIGN PHASES