SOME NOTES ON MULTI-LEVEL DATA BASE DESIGN*

C. Mohan

May 1979 TR-128

Department of Computer Science
The University of Texas at Austin
Austin, Texas 78712

---

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

SOME NOTES ON MULTI-LEVEL DATA BASE DESIGN

C. Mohan

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

17 May 1979
Technical Report TR-128

1.0 INTRODUCTION

[Yeh 1978a] described a multi-level data base design methodology as consisting of the following three major elements:

1. The separation of the design problem into "independent" levels of design.

2. The description mechanism and design criteria at each level.

3. The methods and computer based design aids for the generation, evaluation and selection of design alternatives at each level.

[Yeh 1978a] further stated that in their opinion a multi-level design methodology could be developed by utilizing many of the existing design levels, description mechanisms, and evaluation techniques. Those authors were concerned with both logical and physical design.

The aim of this note is to further analyze the above and to critically evaluate the applicability of some of the design approaches proposed recently.

It is felt that it would not be possible to come up with a generalized multi-level design methodology (which is independent of a particular data model or DBMS) since the identification of levels depends on the data model, and the constructs and facilities available in the DBMS, which would be used for storing the data base. The level of abstraction of the constructs of the different data models varies widely (see [Kleefstra 1978b, Schmid 1977, Senko 1978] for very detailed discussions of this point). Depending on the model the representation of entities, relationships and properties is going to vary (The levels identified in [Yeh 1978a] were clearly influenced by the choice of the relational model). Hence <u>generalized</u> description mechanisms and design criteria are not possible for all levels. These could exist for the top few levels - those levels which would not be influenced by the data model or DBMS chosen at the lower levels.

At the end of this paper I have outlined a refined multi-level design methodology for the design and implementation of relational and DBTG schemas.

2.0 CONCEPTUAL SCHEMA MODELS

Very frequently new data models are being proposed in the literature (See [Schmid 1977, Wong 1977] for surveys of some proposed data models. Some other data models that have been proposed are described in [Bradley 1978a 1978b 1978c, El-Masri 1979, Flory 1978a 1978b, Foucaut 1978, Hammer 1978, Kleefstra 1978a 1978b, Munz 1976a 1978, Shipman 1979]). Recently many researchers have started incorporating the time aspect in the data models, in many interesting ways [Breutmann 1979, Flory 1978b, Foucaut 1978, Laine 1979, Sundgren 1978]. Bradley has proposed a novel model which combines the aspects of a process or program library with those of conventional data models. This model has been shown to let the system answer complex queries involving the time dimension. An excellent discussion of the different components of the ANSI/SPARC architecture [Tsichritzis 1977] has been given in [William Olle 1978]. A major obstacle to the implemetation of the ANSI/SPARC proposal in its full extent has been analyzed in [Paolini 1978b]. The difficulty in defining and implementing the transforms between the different schemas has been noted to be the major hindrance.

Based on this observation, instead of looking at a view exclusively as an instance of a data model, a more general architecture has been proposed in which views are seen as translators of operations into lower level operations. A brief discussion of the implications of the ANSI/SPARC architecture in the distributed data base environment has been presented in [Mohan 1979a].

Although many authors claim that they have adopted the ANSI/SPARC architecture in the case of many models it is not clear whether they are being proposed as models to be used in external schemas and/or the conceptual schema. The proponents of such models should have a clear perception of the differing requirements of the models for the external and conceptual schemas. The shortcomings of the various currently popular data models (like relational and DBTG) for their use at the conceptual schema level have been amply illustrated in many of the papers in [Nijssen 1976a 1977a] and in [Munz 1976, Nijssen 1976b 1977b]. Researchers have realized the need for giving formal treatments to the semantics of data models and languages [Beeri 1978, Borkin 1978, Kalinichenko 1978, Kleefstra 1979, Kobayashi 1979, Konrad 1977, Laine 1979, Schmid 1975, Steel 1976, Subieta 1978].

At this point it would be useful to summarize (from the literature) the desirable properties of a conceptual schema model. They are:

1. It should permit faithful representation of real world relationships.

2. The concepts used for defining the schema should be simple enough to be understandable and usable by non-expert users.

3. It should be stable enough to withstand changes to the internal schema. It should be a data description basis for restructuring.

4. It should allow only non-redundant representation of facts. Presence of redundancies could be a factor of insecurity.

5. Changes in the real world system (Universe of Discourse) when reflected in the schema should cause minimal impact. That is, the conceptual schema concepts should have good stability properties. This criterion is called evolvability. The conceptual schema should be a long-term model of the enterprise.

6. It should be a basis for integrity and security declarations.

7. It should be a common denominator (or integrated view) between multiple local-user views and optimized storage descriptions. It should insulate the external schemas from changes to the internal schema and vice versa.

8. It should be formal so that no ambiguities could arise. Concepts should be so chosen that a formalized and precise definition of the semantics could be provided.

9. Orthogonality of the model should be high.

10. The constructs should be powerful enough to allow any data model and data manipulation language at the external level.

11. The mappings from the conceptual schema to the external schemas and internal schema should be as simple as possible (see [Mercz 1979, Paolini 1978a 1978b, Pelagatti 1977, Zaniolo 1979a 1979b] for discussions of some mapping problems). This criterion is called transformability.

12. The conceptual schema should serve as a centralized documentation of the data base contents.

3.0 DATA DICTIONARIES

The conceptual schema contents must be supplemented with a data dictionary. Excellent accounts of what should be in the data dictionary and how it could be utilized effectively have been given in [Clark 1978, Clough 1977, DDSWP 1977, Ehrensberger 1977]. The British Computer Society's Data Dictionary Systems Working Party (DDSWP) Report discusses a conceptual model, a subset or simple mapping of which would be the ANSI/SPARC conceptual schema. The former would contain some non-DBMS data information also. An implementation of a unique data dictionary system (for a DBMS called NDB), which uses the same architecture as the data base itself and which is physically integrated with the data base itself, has been described in [Sharman 1978]. The data in the dictionary controls the DBMS operations and provides user documentation, as well. An example of a simple data dictionary design can be found in [Clark 1976].

The 'database skeleton', introduced in [Chang 1978], has some of the data dictionary concepts embedded in it. A database skeleton will have some information which will not be found in a data base schema but at the same time it will not have certain information which would be found in a data base schema.

4.0 DESIGN AIDS

As for as the existing computer aided design tools are concerned they were developed, mostly independently, by different researchers. Hence they cover overlapping areas and leave many uncovered areas (particularly in the logical design phase). Most of the tools are restricted in their use for one or two data models only. Efforts are underway to integrate the existing tools and to develop new tools compatible with the existing ones (see [Chen 1977b, Fry 1978, Gerritsen 1978, Schkolnick 1978a 1978b, Scheuermann 1978, Teorey 1978, Wilens 1978, Yao 1978] for further discussion).

[Merten 1977] has listed the following as the likely characteristics of a design aid:

1. It can determine a set of designs, all of which satisfy the constraints and/or objectives.

2. It is an integral part of the other aids to the system development process. For example, the outputs of design aids for this activity correspond with the inputs of subsequent steps.

3. Parametric analyses can be performed and take less time than the initial solution.

4. The underlying algorithms should be understandable and, if desired, visible to the designer.

[Merten 1977] has stated that so far it has been very difficult to compare methodologies and/or manual or computer aids which are often an integral part of the methodology. The reason attributed for this is that there has been no formal description of the design process which can be used to pinpoint the role of a specific methodology and/or aid. After having stated these that paper has presented a formal mathematical description of the data base structure design process. But, from a reading of the description it is not obvious as to how it could be used to pinpoint the role of a specific methodology and/or aid. The description is based on certain assumptions which are highly unlikely to hold good.

5.0 DESIGN METHODOLOGIES

Even though many models and design methodologies have been proposed recently by researchers (for e.g. see [Bubenko 1978b, Flory 1978a, Foster 1977, Laine 1979, Lindencrona-Ohlin 1978, Meng 1978, Methlie 1978, Scheuermann 1978, Shipman 1979, Sundgren 1978, Vincent 1976] and the papers in [NYU 1978]) one noticeable feature of these efforts is the fact that most of them do not consider the effect of evolutions in the real world (although they talk about changes in processing requirements). Since evolutions in the Universe of Discourse and/or changes in users' information requirements are frequent phenomena in realistic environments, any proposed methodology/model could be considered to be good only if it could gracefully accomodate (i.e. with minimal impact on existing application programs) such phenomena. It is highly unfortunate that many researchers have overlooked this point. There has been lot of discussion, about the importance of this point, at the IFIP TC-2 conferences (see [Nijssen 1976a 1977a]).

As pointed out in [Bubenko 1977] the most important requirement of a design methodology is that it should to a minimum extent rely on ingenuity and artistry of the designers. Flory and Kouloumdjian have pointed out that when a conceptual schema is being realized in a company a difficult problem arises: no data model is associated with a method allowing for its practical use. They have further stated that many authors do not distinguish between the notions of the model with those of the method. There experience has been that it is best for the model and the method to be created simultaneously if the model is to be used properly [Flory 1978a].

5.1 Requirements Specification

[Kahn 1978] has presented a logical data base design methdology (LDDM) consisting of six steps: Requirement Step, Entity Step, Relationship Step, Entity Structure Step, Refinement Step, and DBMS Accomodation Step. Kahn has stated that there are two orientations to the logical database design process (one based on Information requirements and the other based on Processing/usage requirements) and that a consequence of this difference is the need to have two parallel paths (Path I and Path P) in LDDM. Based on this premise she has listed as one of the activities of the Requirement Step the selection of a design path to determine whether Path I or Path P is most applicable. To me it is not at all clear as to how processing requirements could affect the initial modeling steps in logical design (since the entity model is independent of them). Only the Refinement Step and DBMS Accomodation Step are likely to be affected by processing requirements.

Recently many researchers have been studying the problems and approaches concerning requirements specification in data base design. [Solvberg 1977] has proposed a model for specification of phenomena, properties, and information structures. The specifications are expected to be enunciated during the information system design phase. Although the terminology is different the proposed model is found to have much resemblance to the ANSI/SPARC framework [Tsichritzis 1977]. The constructs proposed for the phenomena model are found to be very similar to the constructs present in many of the entity-based binary data models proposed for use in the conceptual schema (see [Nijssen 1976a 1977a] for many papers discussing such models). Of course Solvberg has provided some additional facilities for specifying quantitative information.

Solvberg's phenomena model and 'local application perspective' are found to resemble, respectively, the conceptual schema and the external schema of the ANSI/SPARC framework. But Solvberg states that the phenomena model (and the associated descriptor structure) represents those features about which there is a community agreement, rather than a union of all user views on every information systems detail. Due to this fact, one cannot say that the phenomena model is equivalent to the conceptual schema (as defined by ANSI/SPARC). It is not clear as to why this discrepancy is allowed and is desirable, and as to how the phenomena model is likely to be used by the data base designer and the users.

Since the utility of the phenomena model is found to be much smaller than the conceptual schema, Solvberg's model does not seem to have much value. Also, the constructs specified for the information structure are found to have the traditional file oriented flavour. I feel that the different user views should also be specified using an entity based model (see [Kahn 1978, Navathe 1978] for examples of such an approach). The extent to which the quantitative information, specified by the user, is going to be used in the subsequent design steps will be dependent on the DBMS that would be ultimately used (some systems would allow certain parameters to be specified by the user, while other systems might have them built-in). In this respect the 'view modeling' approach proposed in [Navathe 1978] is found to have some good features. The 'database skeleton', introduced in [Chang 1978], could also be used for conveying the type of information that could be conveyed by using Solvberg's model.

5.2 Top Down Or Bottom Up?

A majority of the researchers working on conceptual schema models are of the view that the external schemas should be derived from the conceptual schema (the top down approach), in contrast to the derivation of the community view from the different user views (the bottom up approach) as proposed in [Chen 1977a, El-Masri 1979, Kahn 1978, Navathe 1978, Smith 1978a, Yao 1978]. See [Curtice 1978, Falkenberg 1977a 1977b, Jefferson 1976, Palmer 1978a] for arguments supporting the top down approach. A panel discussion at the 1978 SIGMOD Conference on Management of Data also recommended this approach.

Some of the reasons given for not advocating the derivation of the conceptual schema from the collection of application or user views are:

1. The collection of particular application views, it seems, does not produce a coherent whole that can be a basis for evolution.

2. If one attempts to generate a model of reality by a consensual process, there is no guarantee that one will have a mathematically consistent perception.

3. The ANSI/SPARC Study Group's intention was that the conceptual schema should not simply be a union of the external schemas. It should be a model of the enterprise, a description of the business functions, not a description of the data processing.

The Inferential Abstract Modeling (IAM) Approach proposed in [Bubenko 1977] could also be criticised on this point. IAM also uses 'local user views' (expressed in terms of sets of anticipated output requirements, and anticipated and/or known input transactions) to derive an abstract information model. Another problem with IAM is that (as Bubenko himself has pointed out) it does not guarantee a non-redundant, consistent and semantically correct end product. This is in contrast to the already pointed out requirement that the schema should contain only non-redundant representation of facts.

[Yao 1978] has presented a classification of the phases of a logical data base design as Requirement Analysis, View Modeling, View Integration, View Restructuring, and Schema Analysis and Mapping. Based on this classification, existing design approaches and techniques have been analyzed to determine the phases in which they fit in. The View Modeling phase is intended to model individual user views, while View Integration is meant to integrate those views to get the community view. For the reasons given above it is felt that this is not the order in which the phases should be placed.

5.3 Richness Versus Evolvability

One criticism, made by [Senko 1977], of Schmid and Swenson's [Schmid 1975] distinction between associations (relationships) and characteristics (properties), could equally well apply to the 'view model' of [Navathe 1978] and the entity models of [Chen 1977a, Kahn 1978, Palmer 1978a, Pirotte 1977, Tozer 1976]. To counter Schmid and Swenson's arguments that the distinction might help in formulating consistency rules for update and insertion, Senko states that it would be better to treat the problem in the context of the consistency rules for update and insertion rather than indirectly in the data structure definition, where the distinction is point of view dependent and not fundamental. Similar things could be said about the view model's distinctions amongst the different types of associations (simple associations and identifier associations). The presence of those distinctions could lead to a complex data manipulation language.

[Navathe 1978] does not say anything about the DML or the implementation of the model. It is not known whether the model would be directly implemented or whether the view model schema would be translated into a schema in an existing implemented data model (like relational or DBTG). The latter seems to be the authors' approach, as discussed in [Yao 1978]. This creates problems because, as pointed out in [Yao 1978], one should be able to map a network like community view into any data model as dictated by a target system and this general problem has not been fully addressed yet. This need for complex mappings violates one of the desirable properties (namely, simple mappings) of a conceptual model. Even if the data structure mapping problem were to be solved there do not seem to be straightforward ways of mapping the entity and association insertion and deletion rules of the view model into the target system. Probably 'triggers' (in the spirit of System R) could be used to enforce them.

Real world changes which result in one type of association between given entities changing to a different type of association could necessitate drastic changes to programs operating on data stored using the view model. This means that the view model does not satisfy the requirement of graceful accomodation of the evolutions of the Universe of Discourse. As pointed out in [Senko 1977] the more the number of categories in a model the more complicated and harder it is to understand.

5.4 Data Abstraction

While the abstraction approach of [Smith 1977 1978a 1978b] is found to be conceptually appealing (when illustrated with tiny examples), it would be extremely difficult to apply those concepts to real world enterprise modeling. It demands a great deal of skill on the part of the data base designer for modeling large systems. Since the names of objects are expected to carry lot of semantics improper namings could lead to lack of understandability. Identifying the appropriate number of levels of abstraction will not be an easy task. The approach is likely to cause problems when one considers the evolutionary nature of the real world systems of interest. As pointed out earlier one of the requirements of a good approach is that small changes in the Universe of Discourse should cause minimal changes in the model obtained using the approach. The Smiths' approach is highly unlikely to meet this requirement. Introduction of new intermediate levels would warrant modification of existing programs.

Also one wonders whether an enterprise would really be interested in the abstractions that could be created for the elements in its Universe of Discourse. It is strongly felt that the need for or application of generalization would be very low. Such a need must be clearly demonstrated for a real world system before one could accept that the advantages of the approach outweigh its disadvantages. In [Smith 1978a] this model is claimed to be suitable for use at the conceptual schema level. I have already pointed out that the non-suitability of the relational model for use at the conceptual schema level has been very well illustrated in the literature. Since the Smiths' work is based on the relational model their claim is highly questionable.

While discussing the desirable properties of a design method Bubenko states that while there are areas where pure 'top-down' methods seem adequate it is his experience that those approaches require 'artistry' and an almost perfect perception of the whole problem and that inadequate high level abstractions may turn disastrous at lower levels and require extensive multi-level iterations, where every iteration is a potential source for errors and inconsistencies [Bubenko 1977].

Since Brodie's structured schema specification design approach [Brodie 1978a] and the approach of [Bonczek 1977] are based on the Smiths' work, the above criticisms apply to their methodologies also. One of the problems with these methodologies is that there are no guidelines or objective criteria for deciding how many levels of refinements to carry out. Even when a single designer is involved, this decision making would be difficult. If many people are involved in the design process then these methodologies require that they all come to an agreement about the levels of abstraction that need to be incorporated. Since the view model of [Navathe 1978] is an extension (to achieve a better modeling of the usage perspective and to incorporate relationships among data instances, especially those used for identification purposes) of the Smiths' work the above criticisms apply to that model also. The view model gives much importance to the explicit identification of instances of objects. It remains to be explained as to why this is a desirable feature. As Falkenberg points out, if one wants to understand what the elements of a Universe of Discourse are, then it is necessary to abstract from the various possibilities of signification or identification [Falkenberg 1976b].

Of late there has been a great tendency to use the abstract data type concepts in the data base area [Ehrig 1978, Hebalkar 1978, Lockemann 1979, Mylopoulos 1978, Palmer 1978b, Paolini 1978a 1978b, Pelagatti 1977 1978, Weber 1978b 1978c]. My pessimism about its uselfulness in the data base area has been, to some extent, supported by the comments in [Brodie 1978b]: "Abstract data types are not a panacea for data base problems. However they may be seen as a unifying concept which provides conceptual guidance for data base research".

5.5 <u>DBMS Specific Methodology</u>

Tozer has presented a method for the design of the CODASYL schema, subschema and storage structure from the information in the entity representation of the corporate data model [Tozer 1976]. Some qualitative criteria have been given for deciding on when to eliminate repeating groups (by introducing a set type), combine many record types into a single record type, split a record type into many record types and eliminate a set type. Results obtained from the application of this methodology to realistic situations have been summarized in [Palmer 1978a]. In this entity model also one notices the problems that arise due to the maintenance of the distinction between properties and relationships.

The 1973 CODASYL DDL specifications are now being replaced by the 1978 CODASYL DDL specifications. Those (old) DDL elements which dealt with storage and tuning aspects have been excluded from the (new) DDL and a new language, the Data Storage Description Language (DSDL), has been designed to specify such parameters [CODASYL 1978, Manola 1978, Stacey 1978, Tozer 1978]. To accomodate these changes Tozer's design methodology would need to be modified.

6.0 FRAMEWORK FOR DESIGN

6.1 Basic Premise

If one adopts the ANSI/SPARC architecture or the COEXISTENCE model [Nijssen 1977b] one could choose (from what has been termed as third generation data models in [Falkenberg 1977c]) an entity based model for the conceptual schema. Falkenberg's object-role model [Falkenberg 1976a] seems to be an ideal one. It allows one to represent the 'deep structure of data'. Elementary facts are represented by associations, where an association connects two or more objects and/or associations, with each association/object playing a specific role. A machine readable notation and a manipulation language (which incorporates both procedural and non-procedural elements) for the deep structure have been outlined in [Falkenberg 1975 1977a]. The need for basing the various data base management functions (like data definition, data retrieval, etc.) on one principle, namely signification, has been pointed out in [Falkenberg 1976b]. A language (called CSL - Conceptual Schema Language) based on Falkenberg's model and useful for describing both the static aspects and the dynamic behaviour of data has been presented in [Breutmann 1979].

Even if Falkenberg's model is not implemented on a computer it could be used as the starting point of the design process. Aggregation, etc. could be added to convert the object-role model schema into the equivalent schema in another model which has been implemented on a computer. Falkenberg calls as 'deconceptualization' the process of transforming a schema in the object-role model into a schema in some other model. Since the object-role model does not concern itself with semantically irrelevant aspects (like access paths, notational aspects, etc. - See [Falkenberg 1977c]) of the Universe of Discourse, even if some quantitative information (access frequency, query characteristics, etc.) change, the object-role model schema will remain stable. Even other evolutions (those that affect the semantics of the Universe of Discourse) cause only minimum changes to the schema.

Based on Falkenberg's model Nijssen is developing the Evolving NAtural Language Information Model (ENALIM) [Nijssen 1977b]. This model has been successfully used in a large scale real-world environment. It has been found to be highly useful [DeJong 1977]. An extended data dictionary has been needed to keep the large scale project under control. An implementation of ENALIM, called CYBER-EDMS, is now available through the CYBERNET of Control Data Europe.

6.2 A Refined Multi-Level Approach

I have illustrated in Figures 1 and 2 a multi-level design approach to the design of a relational schema and a DBTG schema. I have assumed that the Universe of Discourse is analyzed (through systems analysis techniques, etc.) to determine the information that needs to be represented in the data base (to reflect the enterprise's structure, objectives and functions). As a result of this analysis one gets to the 'atomic facts or semantic level', where only semantically relevant information is documented using the object-role model (see [Falkenberg 1977b 1977c] for examples of such schemas). Recognizing the fact that in the real world DBTG and relational systems are more widely available as implemented ones (and the fact that an implementation of the object-role model or of the ANSI/SPARC architecture is not widely available) I have assumed that the object-role schema would be used as the basis of an implementation of a relational or DBTG schema on an existing DBMS.

The ordering of the levels in the figure shows the order in which the different parameters (about the physical and logical aspects of the data base), that would be fed to a DBMS, are determined. The initial levels concern the logical aspects of the data base, while the subsequent levels are concerned with the physical aspects. The movement from one level to the next level is a step in the design process. While the steps are found to be distinct, the interdependency amongst the decisions made in the different steps needs to be recognized.

In the Figure I have indicated beside the arrows the factors which determine the movement from an upper level (one at the tail of the arrow) to a lower level (one at the head of the arrow), where each box represents a level. Besides each box I have indicated the parameters that would have been determined as a result of having moved from the previous level to that level. These parameters have been based on System R [Astrahan 1976, Chamberlain 1976], as a representative of implementations of the relational model, and the 1978 CODASYL specifications [CODASYL 1978, Manola 1978, Stacey 1978, Tozer 1978] for the DBTG approach. Although Figures 1 and 2 contain almost the same kind of levels there are certain differences.

6.2.1 The Relational Case -

The entity level consists of definitions of a set of relations each of which represents an entity type (one crude way of identifying this type of relations may be to say that their keys would consist of a single column name). The movement to the relationship level would result in the definition of a set of relations which represent relationships amongst entities (one crude way of identifying these relations is to say that their keys would consist of more than one column names). The process of obtaining these two sets of relations from the atomic facts could be considered to be one of synthesis [Chang 1978, Fagin 1977]. I have split up this preliminary design process into two phases to emphasize the fact that if one were to consider an enterprise data base with thousands of entities then the resulting complexity could be conveniently handled by first identifying all the entities of interest and then identifying the relationships amongst them (rather than doing both at the same time).

The set of relations obtained in the above two levels are now modified (by vertical and/or horizontal decomposition and/or composition - see [Chang 1977, Cheng 1977]) based on estimated or observed query characteristics, storage constraints and performance requirements to arrive at the refined or semi-optimized level. The relations existing at this level could be used to define application specific or individual user VIEWs and the security constraints for individual users (thereby reaching the applications level). With the increased emphasis on placing more and more of the application programs' functions in the data base itself (as a set of relations containing partially predefined queries) certain activities which were previously considered as being part of application program design now become part of data base design. One could be defining such tailored user interfaces while moving to the applications level, if one were to be using the type of system whose architecture is presented in [Spath 1977].

While the view definitions are being made one could in parallel move to the access path level by defining IMAGEs (i.e. indexes), LINKs and clustering conditions. The determination of where the relations and auxiliary information (indexes, etc.) would be stored (i.e. the contents of SEGMENTs, which are logical address spaces) would lead to the locative structure level (In System R SEGMENTs cannot be defined by a user of the SEQUEL2 interface to the system). The mapping of the SEGMENTs to physical storage leads to the physical storage level.

In the case of data base design for other relational systems the same set of levels may not be necessary. For example, if one were to perform data base design for the intelligent controller RAP [Schuster 1978] then the access path level will not exist (since RAP does not maintain indexes). At the locative structure level one would fragment the set of relations' tuples so that each fragment would be storable in a RAP cell memory. At the physical structures level the fragments would be assigned specific memory cells.

6.2.2 The DBTG Case -

On reaching the entity level record types which represent entities would have been defined. The movement to the relationship level would result in the definition of set types and some new record types to represent relationships. Now based on the expected usage of the data base some of the record and set types would be modified or eliminated (see [Tozer 1976] for guidelines). The resulting collection of record and set types could be used to define the subschemas, thus moving to the applications level. Upto this point the DDL would have been used. In parallel with the subschema definition activity the DSDL (Data Storage Description Language) could be used to define mappings from the logical record types to the actual stored record types (that is, the logical records would be materialized through the stored records). Complex mappings in which a schema record type is materialized by more than one storage record type (with the contents of the stored records being disjoint or overlapping) could be specified. The mappings could also be defined to be different for different occurrences of the same schema record type [Stacey 1978, Tozer 1978]. The process of defining the mappings brings one to the stored records level.

The movement to the access path level results in the definition of indexes and set representations. The movement to the locative structures level results in the determination of the distribution of the stored records and the auxiliary information on storage areas, which are logical address spaces. The definition of the mapping of these storage areas onto physical media results in the movement to the physical structures level.

7.0 CONCLUSION

An example design for the CODASYL environment has been worked out in [Yeh 1978b]. I am currently extending the methodology for the distributed data base environment [Mohan 1979a 1979b 1979c].

References to many other papers dealing with topics discussed here could be found in [Mohan 1978].

8.0 REFERENCES

1. Astrahan, M.M., et. al. [1976] System R: Relational Approach to Data Base Management, ACM TODS, June.

2. Baroody, A. [1978] The Evaluation of Abstract Data Types as an Implementation Tool for Database Management Systems, Ph.D. Thesis, Tech. Rep. 330, Univ. of Wisconsin - Madison, August.

3. Beeri, C., et. al. [1978] A Sophisticate's Introduction to Database Normalization Theory, Proc. IV Int. Conf. on VLDB, September.

4. Biller, H. [1979] On the Notion of Irreducible Relations, Proc. IFIP TC-2 Working Conf., Venice (Italy), June.

5. Biskup, J., Dayal, U., Bernstein, P. [1979] Synthesizing Independent Database Schemas, Proc. ACM/SIGMOD Int. Conf. on Management of Data, May-June.

6. Blanken, H. [1979] Automatic Generation of Storage Structures for a DBTG-Database System, Memo. Nr. 247, Twente Univ. of Technology, The Netherlands, March.

7. Bonczek, R., et. al. [1977] Conceptual and Operational Constructs for Knowledge Representation, Krannert Institute Paper, Purdue Univ., July.

8. Borkin, S. [1978] Data Model Equivalence, Proc. IV Int. Conf. on VLDB, September.

9. Bradley, S. [1978a] Operations Data Bases, Proc. IV Int. Conf. on VLDB, September.

10. Bradley, S. [1978b] An Extended Owner-Coupled Set Data Model and Predicate Calculus for Database Management, ACM TODS, December.

11. Bradley, S. [1978c] An Algebra for an Owner-Coupled Set Data Model, Tech. Rep., Technical University of Denmark, Lyngby.

12. Breutmann, B., Falkenberg, E., Mauer, R. [1979] CSL: A Language for Defining Conceptual Schemas, Proc. IFIP TC-2 Working Conf., Venice (Italy), June.

13. Brodie, M.L. [1978a] Specification and Verification of Data Base Semantic Integrity, Ph.D.

Thesis, Tech. Rep. CSRG-91, Univ. of Toronto, April.

14. Brodie, M.L., Schmidt, J. [1978b] What is the Use of Abstract Data Types in Data Bases? Proc. IV Int. Conf. on VLDB, September.

15. Bubenko, J.A. [1977] IAM: Inferential Abstract Modeling - An Approach to Design of Information Models for Large Shared Data Bases, IBM Tech. Rep. RC6343, January.

16. Bubenko, J.A. [1978a] Some Fundamentals and Problems of Information Analysis, Tech. Rep., Chalmers Univ. of Technology, Sweden, February.

17. Bubenko, J.A. [1978b] Database Design in the Perspective of Information System Development, Proc. Symp. on Database Design and Software Engineering, Chalmers Univ. of Technology, Sweden, September.

18. Bubenko, J.A. [1979] On the Role of 'Understanding Models' in Conceptual Schema Design, Proc. V Int. Conf. on VLDB.

19. Chamberlain, D.D., et. al. [1976] SEQUEL2: A Unified Approach to Data Definition, Manipulation and Control, IBM J. of R & D, November.

20. Chang, S.K., Cheng, W.H. [1977] Structured Database Decomposition in the Medical Information System Environment, Proc. COMPSAC'77, November.

21. Chang, S.K., Cheng, W.H. [1978] Database Skeleton and Its Application to Logical Database Synthesis, IEEE Trans. on Software Eng., January.

22. Chen, P.P.S. [1977a] The Entity-Relationship Model: A Basis for the Enterprise View of Data, Proc. NCC.

23. Chen, P.P.S., Yao, S.B. [1977b] Design and Performance Tools for Data Base Systems, Proc. III Int. Conf. on VLDB, October.

24. Cheng, W.H. [1977] Optimization Techniques in Designing Relational Database Systems, Ph.D. Thesis, Univ. of Illinois at Chicago Circle, October.

25. Cherniavsky, V., Schneider, H.J. [1978] On Database Systems Viewed as Programming Languages, information Systems 3, 3.

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

27. Clark, J.D. [1978] A Utility for the Generation of a Preliminary Data/Dictionary Directory, Proc. ACM Annual Conf., December.

28. Clemons, E. [1977] An External Schema Facility to Support Data Base Update, Tech. Rep. 77-08-04, Univ. of Pennsylvania.

29. Clemons, E., Gerritsen, R. [1978] Self-Managing Data Base Systems, Working Paper 78-03-05, Univ. of Pennsylvania, March.

30. Clemons, E. [1979] Rational Database Standards: An Examination of the 1978 CODASYL DDLC Report, Information Systems 4, 3.

31. Clough, H. [1977] Control of Data: Definition and Description, In Software Engineering Techniques, INFOTECH State of the Art Report, Infotech International.

32. CODASYL [1978] Report of the CODASYL Data Description Language Committee, Information Systems 3, 4.

33. Codd, E. F. [1979] Extending the Data Base Relational Model to Capture More Meaning, ACM TODS, December.

34. Cook, M.M. [1978] Data Base Facility for COBOL 80, Proc. NCC, June.

35. Curtice, R.M., Jones, P.E. [1978] Key Steps in the Logical Design of Data Bases, In [NYU 1978].

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

37. De, P., et. al. [1978] Toward an Optimal Design of a Network Database from Relational Descriptions, Operations Research 26, 5, September-October.

38. DeJong, W. [1977] Experience with a New Database Approach, Proc. IAG Conf. on "Managers, Data Bases and Information Systems", November.

39. Deutsch, D., et. al. [1979] Architectural Issues for Data Base Standards, Proc. COMPCON'79 Spring, March.

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

41. Ehrig, H., et. al. [1978] Algebraic Specification Schemes for Database Systems, Proc. IV Int. Conf. on VLDB, September.

42. El-Masri, R., Wiederhold, G. [1979] Data Model Integration Using the Structural Model, Proc. ACM/SIGMOD Int. Conf. on Management of Data, May-June.

43. Fagin, R. [1977] The Decomposition Versus the Synthetic Approach to Relational Database Design, IBM Tech. Rep. RJ1976, March.

44. Falkenberg, E. [1975] Structuring and Representation of Information at the Interface Between Data Base User and Data Base Management System, Ph.D. Thesis, University of Stuttgart, Germany, June.

45. Falkenberg, E. [1976a] Concepts for Modelling Information, In [Nijssen 1976].

46. Falkenberg, E. [1976b] Significations: The Key to Unify Data Base Management, Information Systems 2, p19-28.

47. Falkenberg, E. [1977a] Concepts for the Coexistence Approach to Data Base Management, Proc. International Computing Symposium, April.

48. Falkenberg, E. [1977b] Future Perspective in Information Systems Technology, Proc. IAG Conf. on Managers, Data Bases and Information Systems, Amsterdam, November.

49. Falkenberg, E. [1977c] Data Models: The Next Five Years, Proc. INFOTECH Conf. on Data Bases: The Next Five Years, London, December.

50. Flory, A., Kouloumdjian, J. [1978a] A Model and a Method for Logical Data Base Design, Proc. IV Int. Conf. on VLDB, September.

51. Flory, A., Kouloumdjian, J. [1978b] A Model for the Description of the Information System Dynamics, Proc. II Conf. of the European Cooperation in Informatics, October.

52. Foster, C., Gamble, O. [1977] Information Systems Architecture - An Analysis and Design Methodology Founded on the Data Base Approach, Working Paper, Cincom Systems, Inc.

53. Foucaut, O., Rolland, C. [1978] Concepts for Design of an Information System Conceptual Schema and its Utilization in the REMORA Project, Proc. IV Int. Conf. on VLDB, September.

54. Fry, J., et. al. [1978] Database Design Tools and Methods - A Survey and Tutorial, Tech. Rep. 78 DB 14, Univ. of Michigan - Ann Arbor, October.

55. Gerritsen, R., et. al. [1977] Automated Data Base Programming, Tech. Rep. 77-07-05, Univ. of Pennsylvania, September.

56. Gerritsen, R. [1978] Tools for the Automation of Database Design, In [NYU 1978].

57. Goldschmidt, D., Reiter, A. [1978] An Implementation of FORAL Using IDMS, Proc. Int. Conf. on Data Bases: Improving Usability and Responsiveness, August.

58. Griffiths Selinger, P., et. al. [1979] Access Path Selection in a Relational Database Management System, Proc. ACM/SIGMOD Int. Conf. on Management of Data, May-June.

59. Hammer, M., McLeod, D. [1978] The Semantic Data Model: A Modelling Mechanism for Data Base Applications, Proc. ACM/SIGMOD Int. Conf. on Management of Data, May-June.

60. Hebalkar, P. [1978] Application Specification for Distributed Data Base Systems, Proc. IV Int. Conf. on VLDB, September.

61. Hitchcock, P., Pace, F. [1977] An Approach to Conceptual Data Analysis, Tech. Rep. UKSC-0090, IBM Peterlee Scientific Center, England, December.

62. Hutton, D. [1978] DDL '78: A Critical Assessment of the CODASYL Data Description Language 1978, In [INFOTECH 1978].

63. INFOTECH [1978] Data Base Technology, INFOTECH State of the Art Report, Infotech International, England.

64. Jefferson, D.K. [1976] The Role of the External Schema, The ANSI/SPARC Model, D. Jardine (Ed.), North Holland.

65. Kahn, B.K. [1978] A Structured Logical Database Design Methodology, In [NYU 1978].

66. Kalinichenko, L. [1978] Data Models Transformation Method Based on Axiomatic Data Model Extension, Proc. IV Int. Conf. on VLDB, September.

67. Kleefstra, W. [1977] Database Description with the Single Name Category Data Model, Memo No. 180, Twente Univ. of Technology, Dept. of Applied Mathematics, The Netherlands.

68. Kleefstra, W. [1978a] Database Definition with the OC Model, Internal Report, Twente Univ. of Technology, The Netherlands.

69. Kleefstra, W. [1978b] Data Base Description With a Single Category Data Model, Proc. IV Int. Conf. on VLDB, September.

70. Kleefstra, W. [1979] Formula Structuring: A Framework for Data Models, Memo. Nr. 240, Twente Univ. of Technology, The Netherlands, January.

71. Klug, A. [1978] Theory of Database Mappings, Ph.D. Thesis, Tech. Rep. CSRG-98, Univ. of Toronto, Canada, December.

72. Kobayashi, I. [1979] The Generalized Relational Database Model, Res. Rep. 4, Int. Institute for Advanced Study of Social Information Science, Japan, March.

73. Konrad, E. [1977] Logical Semantics of Data Bases, Proc. Workshop on Logic and Data Bases, November (Also Tech. Rep. CIS 11/77, Technical Univ. of Berlin, W. Germany).

74. Laine, H., et. al. [1979] Grammatical Data Base Model, Information Systems 4, 4.

75. Levin, M. [1978] An Introduction to DIAM Levels of Abstraction in Accessing Information, Proc. ACM Annual Conf., December.

76. Lindencrona-Ohlin, E. [1978] Methods for Conceptual Level Data Base Design, Proc. Symp. on Data Base Design and Software Engineering, Chalmers Univ. of Tech., Sweden, September.

77. Lockemann, P., et. al. [1979] Data Abstractions for Database Systems, ACM TODS, March.

78. Manola, F. [1978] A Review of the 1978 CODASYL Database Specifications, Proc. IV Int. Conf. on VLDB, September.

79. Meng, D., Vincent, D. [1978] How to Design a Data Base that Mirrors Your Organization's Needs, Tech. Support E70-400-410, Datapro Research Corp., January.

80. Mercz, L.I. [1979] Issues in Building a Relational Interface on a CODASYL DBMS, Proc. IFIP TC-2 Working Conf., Venice (Italy), June.

81. Merten, A., Oberlander, L. [1977] A View of Database Structure Design, Proc. COMPSAC'77, November.

82. Methlie, L. [1978] Schema Design Using a Data Structure Matrix, Information Systems 3, 2.

83. Meyer, B., Schneider, H. [1975] Predicate Logic and Data Base Technology, Proc. Advanced Course on Data Base Languages and Natural Language Processing, August (Also Tech. Rep. CIS 4/75, Technical Univ. of Berlin, W. Germany).

84. Mohan, C. [1978] An Overview of Recent Data Base Research, ACM SIGBDP's DATABASE 10, 2, Fall (Technical Report SDBEG-5, University of Texas at Austin, April).

85. Mohan, C., Yeh, R. [1979a] Distributed Data Base Systems - A Framework for Data Base Design, To Appear in Distributed Data Bases, INFOTECH State of the Art Report, Infotech International (Also Tech. Rep. SDBEG-10, Software and Data Base Engineering Group, Univ. of Texas at Austin, April).

86. Mohan, C. [1979b] Distributed Data Base Management - Some Thoughts and Analyses, Technical Report 129, Software and Data Base Engineering Group, Dept. of Computer Sciences, Univ. of Texas at Austin, May.

87. Mohan, C. [1979c] Data Base Design in the Distributed Environment, Working Paper WP-7902, Software and Data Base Engineering Group, Univ. of Texas at Austin, May.

88. Molina, F.W. [1979] A Practical Data Base Design Method, ACM SIGBDP's DATABASE, Fall.

89. Munz, R. [1976a] The WEB-Model as a Possible Conceptual Schema, Proc. Advanced Course on Data Base Languages and Natural Language Processing, Freudenstadt (Also Tech. Rep. CIS 9/76, Univ. of Berlin, W. Germany).

90. Munz, R. [1976b] Extensibility of Database Systems, Tech. Rep. CIS 10/76, Univ. of Berlin,

W. Germany.

91. Munz, R. [1978] The WELL System: A Multi-User Database System Based on Binary Relationships and Graph Pattern-Matching, Information Systems 3, 2.

92. Mylopoulos, J., et. al. [1978] A Language Facility for Designing Interactive Applications, Proc. ACM/SIGMOD Int. Conf. on Management of Data, May-June.

93. Navathe, S., Schkolnick, M. [1978] View Representation in Logical Database Design, Proc. SIGMOD Int. Conf. on Mgmt. of Data, June.

94. Nijssen, G.M. (Ed.) [1976a] Modelling in Data Base Management Systems, Proc. IFIP TC-2 Working Conference, North Holland, Amsterdam.

95. Nijssen, G.M. [1976b] A Short Evaluation of the ANSI and CODASYL Database Approach, Proc. Annual Meeting of the Italian Computing Association, October.

96. Nijssen, G.M. (Ed.) [1977a] Architecture and Models in Data Base Management Systems, Proc. IFIP TC-2 Working Conference, North Holland, Amsterdam.

97. Nijssen, G.M. [1977b] On the Gross Architecture for the Next Generation Database Management Systems, Proc. IFIP'77, August.

98. Nijssen, G.M. [1978] The Next Five Years in Data Base Technology, In [INFOTECH 1978].

99. NYU [1978] Proc. Symposium on Data Base Design, Graduate School of Business Administration, New York University, May.

100. Palmer, I.R. [1978a] Practicalities in Applying a Formal Methodology to Data Analysis, In [NYU 1978].

101. Palmer, I.R. [1978b] Record Subtype Facilities in Database Systems, Proc. IV Int. Conf. on VLDB, September.

102. Paolini, P., Pelagatti, G. [1978a] Specification of Views and Mappings in Multilevel Databases, Tech. Rep. 78-2, Polytechnic of Milan, Italy.

103. Paolini, P. [1978b] An Alternative Structure for Data Base Management Systems, Proc. IV Int. Conf. on VLDB, September.

104. Pelagatti, G., et. al. [1977] Mappings in Database Systems, Proc. IFIP'77.

105. Pelagatti, G., et. al. [1978] Mapping External Views to a Common Data Model, Information Systems 3, 2.

106. Pirotte, A. [1977] The Entity - Association Model: An Information - Oriented Data Base Model, Proc. International Computing Symposium 1977, North Holland, Amsterdam.

107. Poschick, S. [1978] A Portable Relational Interface for the Distributed DBMS POREL, Working Paper, University of Stuttgart, Germany.

108. Scheuermann, P. [1978] On the Design and Evaluation of Data Bases, Computer, February.

109. Schkolnick, M. [1978a] Physical Data Base Design Techniques, In [NYU 1978].

110. Schkolnick, M. [1978b] A Survey of Physical Database Design Methodology and Techniques, Proc. IV Int. Conf. on VLDB, September.

111. Schmid, H.A., Swenson, J.R. [1975] On the Semantics of the Relational Model, Proc. SIGMOD Int. Conf. on Mgmt. of Data.

112. Schmid, H.A. [1977] An Analysis of Some Constructs for Conceptual Schema Models, In [Nijssen 1977].

113. Schuster, S.A., et. al. [1978] RAP.2 - An Associative Processor for Data Bases, Proc. V Ann. Symp. on Computer Architecture, April.

114. Senko, M. [1977] Conceptual Schemas, Abstract Data Structures, Enterprise Description, Proc. International Computing Symposium.

115. Senko, M. [1978] Classification: A Basic Tool for Data Bases and Information Systems, In [INFOTECH 1978].

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

117. Shipman, D. [1979] The Functional Data Model and the Data Language DAPLEX, Proc. ACM/SIGMOD Int. Conf. on Management of Data, May-June.

118. Smith, J., Smith, D. [1977] Data Base Abstractions: Aggregation and Generalization, ACM

TODS, June.

119. Smith, J., Smith, D. [1978a] Principles of Database Conceptual Design, In [NYU 1978].

120. Smith, J. [1978b] A Normal Form for Abstract Syntax, Proc. IV Int. Conf. on VLDB, September.

121. Solvberg, A. [1977] A Model for Specification of Phenomena, Properties, and Information Structures, IBM Tech. Rep. RJ2027, July.

122. Spath, C.R., Schneider, L.S. [1977] A Generalized End-User Facility Architecture for Relational Database Systems, Proc. III Int. Conf. on VLDB, October.

123. Stacey, G. [1978] The Concepts of the Data Storage Description Language, In [INFOTECH 1978].

124. Steel, T.B. [1976] Formalization of Conceptual Schemas, The ANSI/SPARC Model, D. Jardine (Ed.), North Holland.

125. Subieta, K. [1978] Linguistic Approach to Database Theory: DDL-s for Hierarchical Model, Information Systems 3, 3.

126. Sundgren, B. [1978] Database Design in Theory and in Practice - Towards an Integrated Methodology, Proc. IV Int. Conf. on VLDB, September.

127. Teorey, T., Fry, J. [1978] Logical Data Base Design: A Pragmatic Approach, In [INFOTECH 1978].

128. Toh, T., et. al. [1977] Multi-level Structures of the DBTG Data Model for an Achievement of the Physical Data Independence, Proc. III Int. Conf. on VLDB, October.

129. Tozer, E.E. [1976] Database System Analysis and Design, Proc. I Conf. of the European Cooperation in Informatics, K. Samelson (Ed.), Lecture Notes in Computer Science - Volume 44, Springer Verlag.

130. Tozer, E.E. [1978] The Data Storage Description Language (DSDL), In [INFOTECH 1978].

131. Tsichritzis, D., Klug, A. (Eds.) [1977] The ANSI/X3/SPARC DBMS Framework - Report of the Study Group on Database Management Systems, AFIPS Press (Also in Information Systems 3, 3, 1978).

132. Vincent, C. [1976] Data Base Design Methodology, The Bureau of Naval Personnel, U. S. Navy,

February.

133. Wasserman, A. [1978] A Software Engineering View of Data Base Management, Proc. IV Int. Conf. on VLDB, September.

134. Weber, H. [1978a] Design of Decentralized Coherent Application Systems by Autonomous Users, Tech. Rep., Hahn-Meitner-Institut, Berlin, W. Germany.

135. Weber, H. [1978b] A Software Engineering View of Data Base Systems, Proc. IV Int. Conf. on VLDB, September.

136. Weber, H. [1978c] A Software Engineering Approach to Data Base Systems, Proc. COMPSAC'78, November.

137. Wilens, M., et. al. [1978] Interactive Data Base Design Laboratory, Tech. Rep. 78 DE 13, Univ. of Michigan - Ann Arbor, April.

138. William Olle, T. [1978] Multi-Stage Data Definition in a Multi-Component DBMS Architecture, Proc. Int. Conf. on Data Bases: Improving Usability and Responsiveness, August.

139. Wong, H., Mylopoulos, J. [1977] Two Views of Data Semantics: A Survey of Data Models in Artificial Intelligence and Data Management, INFOR 15, 3, October.

140. Yao, S.B., et. al. [1978] An Integrated Approach to Logical Database Design, In [NYU 1978].

141. Yeh, R., et. al. [1978a] Data Base Design - An Approach and Some Issues, Tech. Rep. SDBEG-4, Univ. of Texas at Austin.

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

143. Zaniolo, C. [1979a] Design of Relational Views Over Network Schemas, ACM/SIGMOD Int. Conf. on Management of Data, May-June.

144. Zaniolo, C. [1979b] Multi-Model External Schemas for CODASYL Data Base Management Systems, Proc. IFIP TC-2 Working Conf., Venice (Italy), June.

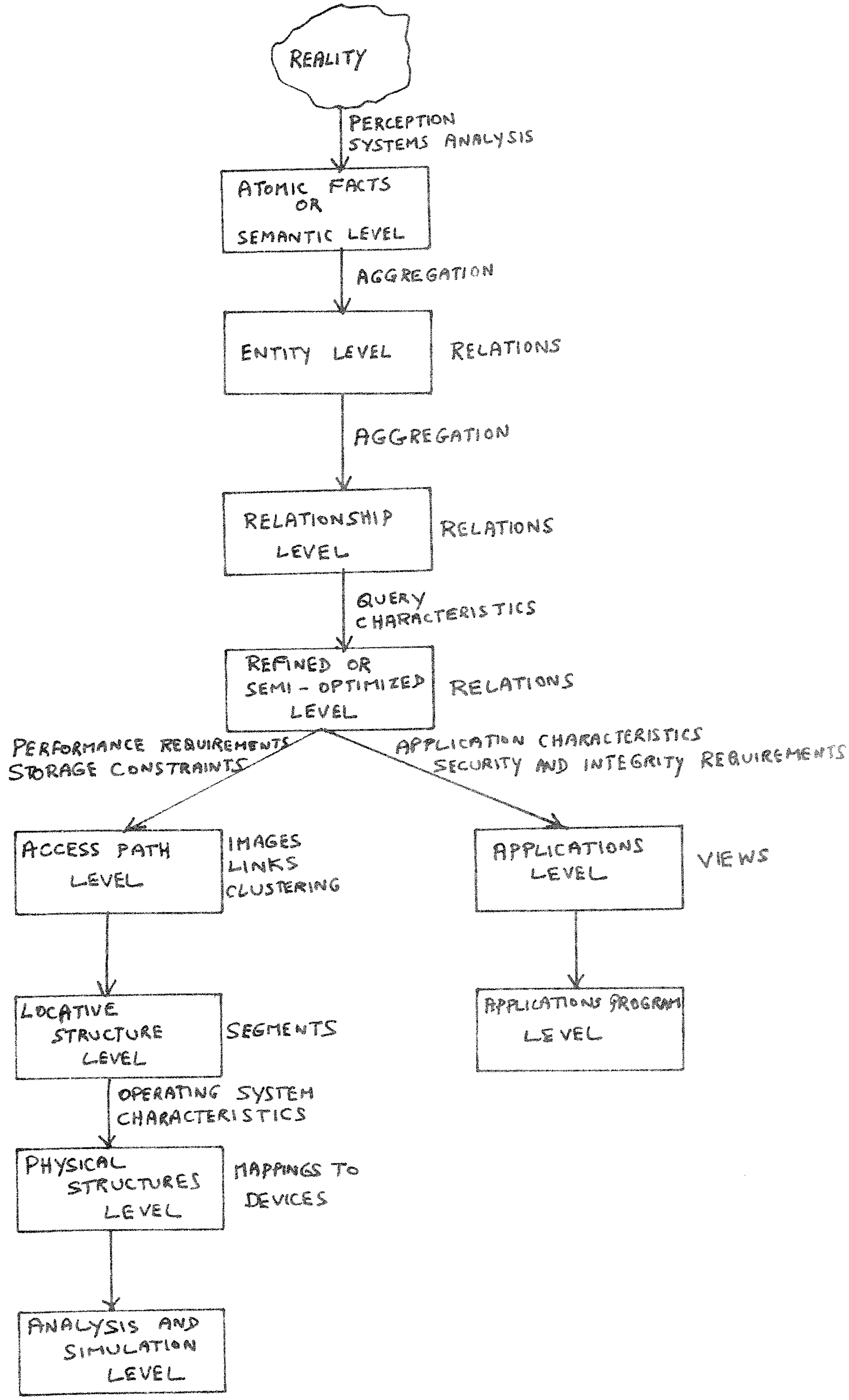
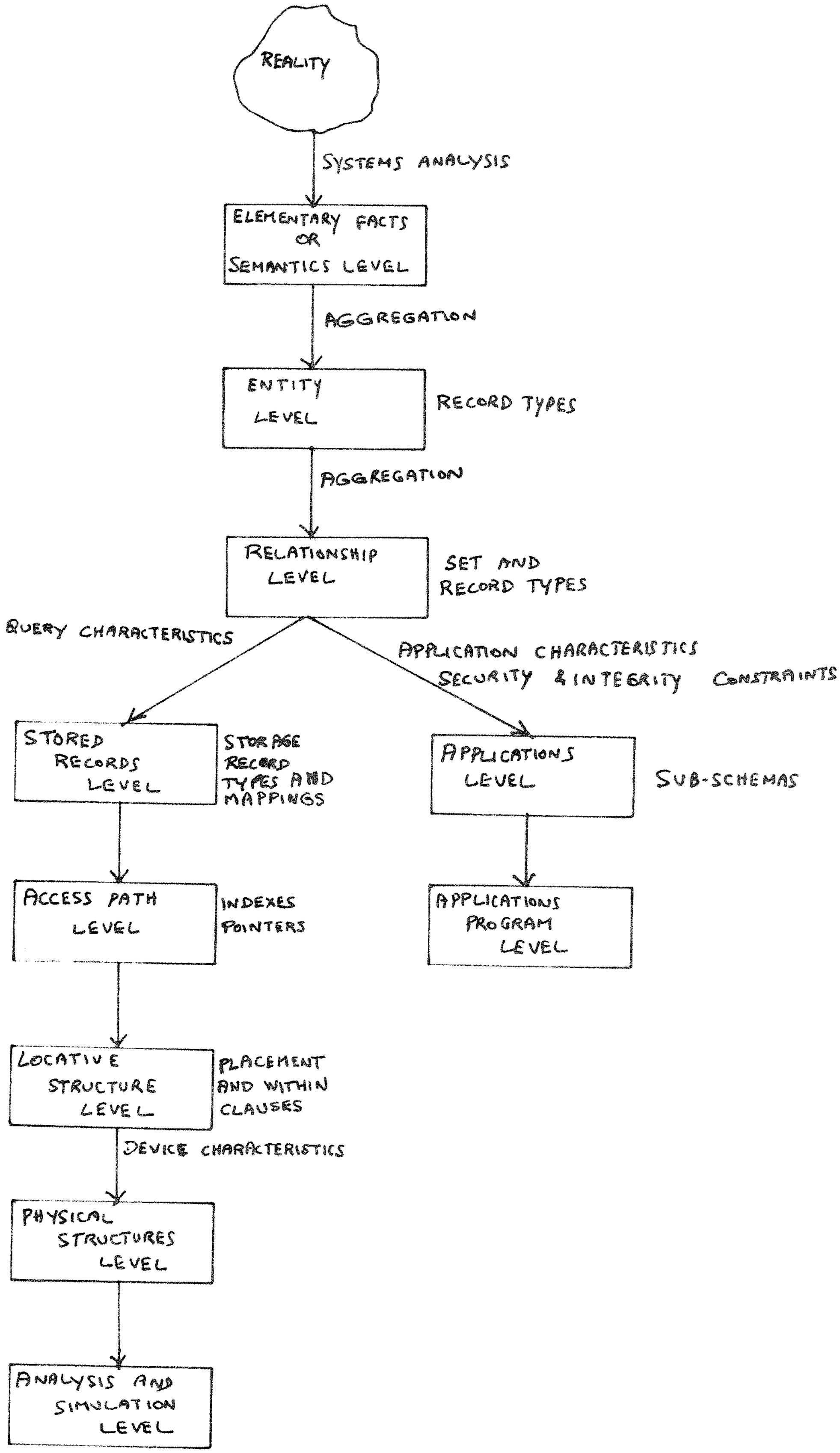

REALITY
PERCEPTION
SYSTEMS ANALYSIS
ATOMIC FACTS
OR
SEMANTIC LEVEL
AGGREGATION
ENTITY LEVEL
RELATIONS
AGGREGATION
RELATIONSHIP
LEVEL
RELATIONS
QUERY
CHARACTERISTICS
REFINED OR
SEMI - OPTIMIZED
LEVEL
RELATIONS
PERFORMANCE REQUIREMENTS
STORAGE CONSTRAINTS
APPLICATION CHARACTERISTICS
SECURITY AND INTEGRITY REQUIREMENTS
ACCESS PATH
LEVEL
IMAGES
LINKS
CLUSTERING
APPLICATIONS
LEVEL
VIEWS
LOCATIVE
STRUCTURE
LEVEL
SEGMENTS
APPLICATIONS PROGRAM
LEVEL
OPERATING SYSTEM
CHARACTERISTICS
PHYSICAL
STRUCTURES
LEVEL
MAPPINGS TO
DEVICES
ANALYSIS AND
SIMULATION
LEVEL

Figure. 1



REALITY
SYSTEMS ANALYSIS
ELEMENTARY FACTS OR SEMANTICS LEVEL
AGGREGATION
ENTITY LEVEL
RECORD TYPES
AGGREGATION
RELATIONSHIP LEVEL
SET AND RECORD TYPES
QUERY CHARACTERISTICS
APPLICATION CHARACTERISTICS
SECURITY & INTEGRITY CONSTRAINTS
STORED RECORDS LEVEL
STORAGE RECORD TYPES AND MAPPINGS
APPLICATIONS LEVEL
SUB-SCHEMAS
ACCESS PATH LEVEL
INDEXES POINTERS
APPLICATIONS PROGRAM LEVEL
LOCATIVE STRUCTURE LEVEL
PLACEMENT AND WITHIN CLAUSES
DEVICE CHARACTERISTICS
PHYSICAL STRUCTURES LEVEL
ANALYSIS AND SIMULATION LEVEL

Figure. 2