A GUIDE TO SCHEMA DESIGN METHODS

Stephen Neil

A supplementary Technical Report to

"A SURVEY OF A DATABASE DESIGN TECHNIQUE"

COSC 460 Honours project

TABLE OF CONTENTS Page

CHAPTER 1.0 Introduction.

In this document some of the major different schools of schema design methods will be outlined. This is important because the different schools, and different specific examples of these schools, will be referred to in the Project Report , (specifically in part 2). Furthermore, in researching this paper no suitable paper, or chapter in any book, was found that compared and contrasted these topics that would be suitable to be given as a reference.

Producing a technical report was therefore thought to be a worthwhile task. It also gives a solid grounding for the discussions in the report.

CHAPTER 2.0 Synthesis and Decomposition.

2.1 Introduction

At the very basic level of schema design there are two competing approaches: synthesis and decomposition. The synthesis approach has been widely developed by Bernstein. Whereas the decomposition approach owes much to the work of Codd and Fagin. Many of the other approaches that follow in this section will build on, or show similarities to these two techniques.

A knowledge of the concepts of normal forms, (1NF, 2NF, 3NF, BCNF, 4NF and 5NF, and their desirability), functional dependencies, multivalued dependencies, join dependencies, non-loss, and dependency-preserving, are assumed. A very good tutorial on all of these is given in [Date 1986] Chapter 17, and a guide to normal forms can be found in [Kent 1983b].

2.2 The Decomposition approach.

The decomposition approach proceeds as follows. At the start of the design process one is given as input: an initial set of relational schemas, (implicitly a Universal Relation), along with a set of functional dependencies (FDs), multivalued dependencies (MVDs), join dependencies, and inclusion dependencies. The initial relational schemas need only be in 1NF, and can be derived by taking all of the relevant empirical information and ‘flattening it out", (as described in [Date 1986]), to produce the atomic elements that are necessary for 1NF. This is made possible by the associated "Universal Relation assumption" that is described in the next section.

Then by taking these existing relations and applying a series of projections, using the information supplied in the constraints, (in the form of FDs, MVDs, and JDs), a new set of relational schemas are be derived. This process of taking projections of the relations resulting from the previous step can produce relations that are finally in 5NF, having proceeded from 1NF.

Date calls this process "Non-loss decomposition". Date gives a very comprehensive treatment of this material, and is the recommended reference. It is worth noting that the only mention that Date makes to the synthesis approach is in the reference section of the decomposition chapter, (Chapter 17).

Fagin [Fagin 1977a , Fagin 1977b , Fagin 1981] has also done a large amount of work in this area. He is responsible for the definitions of 4NF and 5NF, and the relevant decomposition techniques for both normal forms. He also gives a useful description of the difference between the decomposition and synthesis techniques in [Fagin 1977a], which is a guide to the 4NF step in the decomposition technique. This paper is somewhat dated, but the basic concepts are still applicable.

Codd [Codd 1970 , Codd 1971, Codd 1979], who initially formulated the relational model, is responsible for much of the work in developing 1NF, 2NF, 3NF, and BCNF.

2.2.1 The universal relation assumption.

The universal relation assumption can be formalised as:

"For a set S of relations under consideration, there exists in principle a singe universal relation U such that each relation in the set S is a projection of U". [Kent 1981]

This can be visualised as a combination of all of the necessary data that is of interest to the enterprise that has been rolled into one big relation. This relation obviously must be in 1NF. Theoretically it is on this relation that the constraints are applied to produce the final 5NF relations. In practice the designer will have some idea of what the relations should be, and this one relation need not be the starting point of the analysis. But from the theoretical point of view this universal relation assumption must be made.

The Universal relation is not just mathematical pedantry, it is very useful for interface design, query processing, and other areas. These are several outstanding issues and complex intricate problems that have been addressed in papers by [Kent 1981, Ullman 1982, Ullman 1983, Kent 1983a, Maier etal 1984]. This is still a topic of current research and argument, the exact details of which are beyond the scope of this paper.

2.3 The Synthesis approach

The synthesis approach proceeds as follows. The design process begins with only attribute, (potential column names), and dependency information, but not with an initial design. The input is simply a set of attributes, along with a set of functional dependencies. The output is a set of 3NF relational schemas that is some sense 'embody" the functional dependencies.

A mildly different definition of FDs is called for, since the decomposition definition requires a FD to be true in the context of a given relation. Whereas in synthesis approach, FDs are supposed to exist independent of the relations that embody them. That is, FDs, and not relational schemas, are the primitive objects. This approach was largely developed by Bernstein [Bernstein 1976], who uses Armstrong's mathematical axiomation of FDs [Armstrong 1974], to present a technique for deriving 3NF schemas from the functional dependencies.

2.4 Comparison between synthesis and decomposition.

There are problems with both the synthesis and decomposition approaches. In decomposing a set of relations, firstly the designer must be able to find an initial design which is then improved by decomposition, (theoretically this initial design should be the Universal Relation). Fagin [Fagin 1977] conjectures "that it would be nice if instead, the designer could begin with just dependence information, rather than with an initial design". This is the basis of synthesis, which is a more ambitious approach.

Synthesis's problems are that the FDs must suffer under the uniqueness assumption. That is if there are two FD's on the same set of attributes then they are assumed to be the same FD. This is obviously an unrealistic assumption. Also it is only very recently, [Unger 1987], that a synthesis technique for MVDs that will produce 4NF - and then only for a specific class of 4NF - relations has been formulated. This is still an area of continuing research as it is far too much of a disadvantage for these techniques to achieve only 4NF in special cases.

As a footnote to this section it is noted in [Kent 1981] that under the universal relation assumption, the decomposition approach becomes virtually indistinguishable from the synthesis approach.

CHAPTER 3 Record and Fact-based.

3.1 Introduction.

There are two different approaches which differ in the basic unit of data that they use.

3.2 Record-based approaches.

A record-based system means that the system is based on a fixed sequence of field values, conforming to a static description usually contained in catalogues. The description usually consists mainly of a name, length, and data type for each field. Each such description defines one record type, (or in relational terms, a "relation" or "table"). This clearly includes the traditional hierarchical, relational, and (CODASYL) network models and is the most common and well understood model.

Record structures do not provide the semantically self-describing base needed for conceptual schemas. While some information cannot be represented in records, other information can be represented in so many ways as too become ambiguous. Record-based models that do provide additional file structures around the records, (for example: sequencing, hierarchies, and CODASYL networks), overcome some of the semantic limitations. None of them overcome all the limitations. Indeed if they did then there would be no purpose for research into new techniques such as DLS, and this project would not exist.

The above problems also applies to the Entity-Relationship model of Chen, [Chen 1976], which is really driven by record structures, (relations), as the primitive concept, rather than entities and relationships.

In information modelling purposes, one has to account for such concepts as entities and entity types, relationships and attributes, and naming. It is natural to associate a record with an entity. Both are elementary units of creation and destruction, as well as data transmission, and records are classified into types just as entities are. However a record can not be characterised as containing "all the facts" about an entity. This is most graphically illustrated by the characteristic that relationships between entities are usually not contained in the records for the entities that partake in the relationship. This is usually achieved by creating records that represents the relationship, or by placing the necessary information in only one of the participating entities.

Often there is no way of distinguishing between a multikeyed relationship record or an entity that needs the multiple keys to qualify it. In the absence of any additional semantic structures it is difficult to guess at what the underlying semantic structure is by looking at the record structure alone. [Kent 1979]

This is a very negative introduction to record-based systems. Their principles are the best known, and that is why many of the points are glossed over. Many of the negative aspects are not fully argued, but complete arguments are beyond the scope of this paper. It is done so that the idea that the record-based structure, if not the most common, does have limitations and is not the only option.

It is suggested in [Kent 1979] that binary relations, (or elementary facts), models are generally more successful in coping with some of the problems mentioned here. This lead to fact-based systems.

3.3 Fact-based approach

There is a comparatively new technique that is in competition to models that use the record as the basic unit of construction. This is the fact-based technique. Date, [Date 1983], gives a good discussion on many of the issues that are relevant to several different implementations of this approach. The title which this approach is discussed is the "Binary relational model". A specific implementation of this approach will be concentrated upon because of its relevance to DLS. This is Kent's "fact-based" approach.

Kent has done much of the work in this area, and the previously cited [Kent 79] is a paper that tries too point out the drawbacks of the record-based systems in some form of justification for using the new technique. Also in this paper, a brief description of the concept behind this new technique is given:

"Binary relation, (or elementary fact), models are generally more successful in coping with the modelling problems, though none have done so completely. Such models are more directly based on semantic concepts, eg entities and the network of relationships among them, rather than record like structures."

From this beginning the model has been developing to an almost workable system, but there does not exist a technique that is as widely documented, or as practical, as the more well known record-based systems, (for example the Entity-Relationship model).

The approach works on these principles:

- Shift the focus from entities to facts about entities i.e. relationships.
- Use a simplified form of the E-R model.
- Defer the problems of identification and representation until later in the process than the record-based model.

The first step of the theoretical model is to first identify facts that are to be maintained, in terms of Entities and Relationships. This is deriving all of the FDs, MVDs, and JDs that are in the model. This has to be done in the E-R model as well and is an area of continuing research. There are NP complete problems in the identifying of the JDs in a schema.

The next step of this model comprises:

1 - Collect all single-valued facts about the same things into one record.
2 - Provide a separate record for each many-to-many relationship, including single-valued facts about the relationship. The term many-to-many is commonly used and is described in Date [Date 1986],and Howe [Howe 1983].

It is important to make it clear at this stage that it is not stipulated that all of the dependency information must be gathered at once, before the modelling step. There is latitude for spreading this step over the entire modelling process. This avoids the problem of collecting all of the dependencies at once, as is often the case for decomposition systems.

A partially working methodology for this process has been proposed in [Kent 1984]. This briefly comprises of the following steps:

1 - Generate a "pseudo-record" for each fact.
2 - Identify "pseudo-keys" for each pseudo-record.
3 - Merge pseudo-records having compatible keys.

The first two steps are fairly well defined tasks, it is the third step where development of this model is still being made.

This approach is fundamentally different from the record-based approach in that the basic unit of data is different. A data record usually contains a number of facts: facts about things, (attributes), and facts that connect things, (relationships). Facts about an employee might include his salary, department, full name, and birthdate. Connecting facts might connect an employee to a project, or to his/her spouse.

To make the distinction between what is a fact about the thing, (i.e. the employee) or facts that connect two things together (i.e. the employee and project) is a problem which is encountered frequently. Why is an employee's department a fact of that employee, and not a fact connecting two things? The answer to this question is "we don't know the answer". Fact-based methods avoid the question by making no distinction between the two, and considers all facts as connections between things. It is proposed in the model that these assumptions, though surprising at first, simplifies the modelling process.

3.4 Comparison between fact-based and record-based.

Fact-based approaches are fundamentally simpler than record-based ones, but sometimes appear more elaborate because they deal with more complex and realistic situations. They provide a simple framework that higher level concepts like n-ary relationships , multivalued attributes, attributes of relationships, and attributes of attributes can be built on, as opposed to building these concepts into the framework, (as is done in the like of the E-R model and the EER model). [Kent 1984]

The algorithm in [Kent 1984] is not fully developed and as this model is still in the research stage the practical implementation is still quite general. In fact one of the stages in the algorithm is "consider alternatives to the relations produced", which in essence is to consider the alternatives designs that would result if other choices in the design process were taken. It is my conclusion that DLS is a more concrete implementation of the fact-based methodology, but it initially does not appear to be so because much of the fact-based ideas are somewhat camouflaged.

The focus has been moved from storing what facts are about, to what facts are to be maintained. The starting point is no longer one record per entity but one record per fact.

CHAPTER 4 Bottom-up and Top-down.

4.1 Introduction.

Another family of conflicting design philosophies are the Top-down and Bottom-up approaches. Comparisons to the previous pairs of design techniques can be drawn, but it will also be formulated that they both are actually combinations of the previous techniques. The top-down approach has a semi-formalised method presented, but it has a less structured background. These two approaches will sound familiar, because the terms top-down and bottom-up have been widely used in the topic of Software Engineering.

The traditional top-down approach has a "divide and conquer" style. The designer starts with the initial objectives and breaks it up into smaller intercommunicating parts. From there each part, or module, is subdivided keeping the necessary communication with the other modules in mind. This process continues until modules that are of manageable size are produced.

Alternatively the bottom-up approach works more on the library routine concept. A large application will be dealt up by using calls to library routines that will do a specified task, and implementing enough linking processes that binds all of the library calls together. This process will not produce the hierarchical structure that top-down methods do, and therefore is considered by many to be an inferior approach.

The corresponding top-down and bottom-up approaches for schema design are based on similar concepts. With the top-down approach using a more hierarchical structure to develop a schema, and bottom-up staying on a flatter plane.

4.2 Bottom-up

The Bottom-up technique may be considered to start with the lowest level of semantic construct, and to mechanically work its way up to fully normalised tables. The technique does not support the notion of entities or relationships, only the notions of attributes.

Bottom up can be summarised as the following sequence of operations:

1 - Select the attributes of interest to the enterprise, and
2 - Combine the attributes into fully-normalised tables.

The first step is to identify all of the FDs, (also called determinants in this case), and MVDs. This is primarily considered to be a one step operation, i.e. all of the determinants must be identified at this stage, as noted in section 3.3 this is not a trivial task. The next step is to create 1NF tables that represent these determinants. The second step comprises of purely decomposing these into fully-normalised tables by the process of non-loss decomposition.

If there are many attributes to consider, it can be difficult to sort out all the functional dependencies, particularly when composite determinants exist. A real world application can have hundreds or even thousands of attributes. It therefore is generally considered undesirable to have the "determinant identifying stage" as one step. It is worth noting that this basic framework is the one that is often used in teaching courses on the subject of data analysis and design. "I cannot see the merit of this approach, which seems to have become prevalent for historical reasons rather that for its intrinsic worth, and I have a sneaking suspicion that it is sustained chiefly by its convenience as a source of examination questions." [Howe 1983]

4.3 Top-down.

The Top-down approach does incorporate the ideas of entities and relationships. This approach relegates the step of assigning attributes to their relevant entities and relationships too last of all. The idea is to build up the framework of the model by building a skeleton of the entities and relationships, and then to decorate the skeleton with the attributes. The building of this skeleton can be accomplished by a subset of the well known Entity-Relationship model and represented by a stripped down version of E-R . It is supposed that this skeleton will be fully normalised and represent all of the determinants in the model.

The process can be summarised as:

1 - Select the entities, (such as customers, parts and suppliers), and the relationships between them, (such as customer_order_parts, and supplier_supplies_parts);
2 - Add attributes to these entities and relationships in such a way that a set of fully-normalised tables is obtained.

It is generally accepted that there is no absolute way of classifying data into the three groups of entities, relationships, and attributes. [Howe 1983] though recognising this problem offers no way of ratifying it, apart from suggesting that one classification may be in better accord with the enterprise's own view of its data than another.

4.4 Comparison between bottom-up and top-down.

It would appear that the Top-down technique has a record-based start but it uses a fact-based type of analysis to derive finished tables. The starting point of identifying the entities and relationships is the classic start to the traditional E-R model. It is the exclusion of the attributes that is the main difference.

It would appear that the Bottom-up technique has a fact-based start but it then proceeds to produce the finished tables with the more traditional record-based model of non-loss decomposition.

CHAPTER 5 Semantic Modelling.

Part of the motivation behind implementing a database is to represent information. A database's schema is a representation of the structure of the information that the enterprise is dealing with. Unfortunately when using a relational database the schema that results is of a low level of abstraction, (but a higher level than its predecessors). Everything must be represented by tables, even objects that have complex interactions with other objects, and objects with complex descriptions and properties.

The information of the interactions and descriptions that represent objects is Semantic Information. We wish to retain as much of this semantic information as possible when designing a database. The overall activity of attempting to represent the meaning of data will be termed Semantic Modelling. The term Semantic data model is sometimes used to refer to one or other of the "extended" models, for example [Hull et al. 1987], [Abriel 1974]. Where an "extended" model is a schema design model that tries to represent more semantic information than the base data model that it uses.

[Date 1986] suggests that this is a presumptuous term, because "capturing the meaning of the data is a never-ending task, and we can expect to see continuous developments in this area as our understanding continues to evolve". I am inclined to agree with this point of view, and much of the research in this paper is centred on this topic.

CHAPTER 6 Conclusions.

The different schools of schema design methods presented so far, position themselves at different levels of semantic representation. These schema design methods are more the basic ideas and philosophies behind different techniques of producing a database schema. They are the building blocks or frameworks of the more complex design methods that are in common usage. An example is the extremely well known E-R model [Chen 1976]. This uses a record-based, decomposition approach to schema design. The E-R model is an example of semantic modelling.

Due to the very nature of the relational model, there are limitations on how much semantic information can be retained in designing a schema. Different design techniques produce widely differing schemas [Hull 1987]. The major reason is that different design methods use different techniques in representing pieces of semantic information, if it is capable of representing that information at all. The process of semantic modelling may be paraphrased as:

> We wish to represent as much semantic information, as naturally and easily understandable as possible, while keeping the amount of information that must be dealt with at once to a manageable size.

This introduces the idea of documenting parts of a databases schema where the relational model may accurately reflect the information. But represents it in such a way that the original semantic structure could not be recreated from the relation implementation alone.

Part 2 of the associated paper will concentrate on researching ways of maximising the objectives of semantic modelling, with respect to the DLS diagramming technique.

REFERENCES

[Date 1986] Date C.J ;
An introduction to Database Systems ;
Vol 1 Fourth Edition.
Addison-Wesley Pub.

[Kent 1983b] W. Kent.
'A simple guide to five normal forms in relational Database theory';
Communications of the ACM, Vol 26 (2) , Feb 1983, pp 120 - 125.

[Fagin 1977a] R. Fagin
'The decomposition versus the synthetic approach to relational database design';
Proc 3rd international conference on very large DB, 1977 , pp 441 - 446.

[Fagin 1977b] R. Fagin ;
'Multivalued Dependencies and a new normal form for relational Database';
ACM Transactions of Database systems, Vol 2 (3), Sept 1977
pp 262 - 278.

[Fagin 1981] R. Fagin
'A normal form for Relational Databases that is based on domains and keys';
ACM Trans. Database Systems, Vol 6 (3), Sept 1981, pp 387 - 415.

[Codd 1970] E.F Codd
'A relational model of data for large shared data banks';
CACM, 13 (6), June 1970, 377-387.

[Codd 1979] E.F. Codd
'Extending the database relational model to capture more meaning';
ACM TODS 4 (4), Dec 1979.

[Codd 1971] E.F. Codd
'Further normalization of the Database relational model';
Courant Comp Sci Symposium 6, D/B System,
Prentise Hall Pub, New York, May 1971 , pp 65 - 98.

[Kent 1981] W. Kent.
'Consequence of Assuming a universal relation';
ACM Transactions of Database systems, Vol 6 (4), Dec 1981
pp 99 - 121.

[Kent 1979] W. Kent.
'Limitation of record based systems';
ACM TODS 4 (1), March 1979

[Ullman 1982] J.D. Ullman
' The U.R. strikes back';
Proc 1st ACM SIGACT - SIGMOD Symposium on principles of Database systems, March 1982.

[Ullman 1983] J.D. Ullman
'On Kent's ' 'Consequence of Assuming a universal relation'';
ACM TODS Vol 8 (4) , Dec 1983.

[Kent 1983a] W. Kent
'The Universal Relation revisited';
ACM TODS, Vol 8 (4) , Dec 1983.

[Maier etal. 1984] D. Maier and J.D. Ulman and M.Y. Varchi
'On the foundations of the Universal Relation Model';
ACM TODS, Vol 9 (2), June 1984

[Bernstein 1976] P.A. Bernstein
'Synthesizing 3rd normal form relation from functional dependencies.';
ACM tran on DB system, 1, 277 - 298, Dec 1976.

[Armstrong 1974] W.W. Armstrong
'Dependence structures of data base relationships';
Info Processing 74, North-Holland Pub. Co, Amsterdam, 1974, pp 580 - 583.

[Unger 1987] E. Unger and C. Angaye
' A synthesis algorithm for a class of 4NF';
ACM SIGSMALL/PC NOTES, Vol 13 (4), Nov 1987.

[Chen 1976] Peter Pin-Shan Chen ;
'The Entity-Relationship ;model- toward a unified view of data';
ACM Transactions of Database systems, Vol 1 (1), March 1976
pp 9 - 36.

[Date 1983]
An Introduction to Database Systems;
Addison-Wesley (Pub), 1983

[Howe 1983] D.R. Howe
'Data analysis for DataBase design';
Edward Arnold (publishers) 1983

[Hull et al. 1987] R. Hull and R. King
'Semantic Database Modelling: survey, applications, and research issues';
ACM COMP SURV, vol 19 (3), 1987.

[Abrial 1974] B. Abrial
'DataBase Management';
North-Holland, Amsterdam (Pub), pp 1 - 59.