MODELS OF INFORMATION: THE FEASIBILITY OF MEASURING THE STABILITY OF DATA MODELS

Myron Murray Marche
London School of Economics and Political Science

Submitted in fulfilment of the requirements for the award of the degree of Doctor of Philosophy of the University of London

LONDON, ENGLAND
SUMMER, 1991

ABSTRACT

The theory of data modelling makes a variety of claims about schema stability. This research determined the current state of data modelling practice, and tested hypotheses related to measuring model stability. The research developed a method whereby the major elements of a data model can be consistently represented whatever process was originally used for modelling. This was achieved through a construction of a logical relational schema from the record design. The construction/reconstruction process attempted to identify the primary meaning primitives of a data model in order to track changes to them in different iterations of the application.

The stability data collection process was applied to a case study followed by a series of models to generate further data. The early evidence indicated that data model instability has it roots in errors in modelling, errors in the semantic analysis whether done consciously or intuitively, and in changes to the requirements brought on by changes to the "reality". This research suggested that some of the elements of a data model are significantly more important than others.

The research documented problems associated with the transformation of natural language into the constraints of data dictionaries. This exploration into the potential application of linguistic research into systems theory and practice identified a number of theoretically interesting problems, such as variable semantic determination. The discussion outlined some specific techniques an analyst can use to improve the process of semantic analysis. The work suggested that there should be greater concentration on the question of data model evolvability, and the appropriate preservation of meaning across model versions, and not necessarily on data model stability.

To Michael, Stephen, and Janet

ACKNOWLEDGEMENTS

While there is only one name on the title page of this work, there can be no question that it exists only due to the effort of many people. I must first acknowledge the original support of Professor Ronald Stamper, who encouraged me to begin studies at the London School of Economics, and who was willing to let his students think in very broadest of terms. While beginnings are very important, endings are just as important, and often substantially more difficult. Coming to the end of the research was in no small measure due to the efforts of Dr Jonathan Liebenau who provided effective supervision over most of the life of this research, and this he did with sensitivity, unfailing good humour, and good grace.

All systems professionals appreciate the importance of strong project management. I must thank the project manager for the data collection phase of this work, Mr. Ian Cameron, who in his typical fashion did not take no for an answer. The encouragement and confidence of many friends, clients, and business associates is also gratefully acknowledged. I have invariably learned more from them than they have from me.

My colleagues and friends at the London School of Economics have always made it a place for the best kind of open intellectual exploration. Thanks especially to Dr Jim Backhouse, and Professor Ian Angell.

Finally, the love and support of my wife Janet, as well as the patience of my children, Michael and Stephen were instrumental in reaching this goal. Graduate schools should really award research degrees to families, not to individuals.

TABLE OF CONTENTS

PART I THE THEORY AND PRACTICE OF DATA MODELLING

PART II - DEVELOPING AND APPLYING A TOOL FOR STABILITY MEASUREMENT

TABLE OF CONTENTS (continued)

PART III - TOWARDS A CRITICAL THEORY OF DATA MODELLING

TABLE OF CONTENTS (continued)

PREFACE

This preface has been written in the interest of preparing the reader for the discussion and debate which is to follow. This physical document shares one very important characteristic with databases. In each of these different circumstances, the author and the user are unable to negotiate the inevitable differences in understanding which might separate them. Users of modern technology such as photocopiers, often enter into a kind of conversation with the technology, especially when the technology does not respond in ways that the user might have anticipated. Of course, the technology is not able to respond very effectively, except for the occasional error message, perhaps announced with a beep. A document is not even able to do this much, so I have tried in this preface to provide some flavour of the sense of negotiation which I hope the reader will bring to the text.

The review of the literature on data modelling presented in this thesis concluded that the theoretical underpinnings of data modelling are quite weak, and that the research and data on data analysis as a specific subset of systems analysis do not support many of the assertions made by the theoreticians. While I stand by these conclusions, the reader should realise that I hold a rather more pragmatic view - that data modelling as an activity is an extremely valuable tool in the repertoire of the competent systems analyst. My primary concern is that claims about the technology of modelling have been made where there is no data to support the claim. In many cases, there are not even methods available to collect data which might test these varying assertions.

In the course of this document, the reader will come across many terms used in ways which might be at odds with the reader's usage. This is understandable for many reasons. First, differences in language use are extremely common in our daily lives, to the extent where we do not take any notice. We simply identify the misunderstanding, and make the necessary adjustment. The 'necessary adjustment' may or may not involve reconciling our use of the terms under negotiation. For example, the expression 'next Thursday' to me means the very next occurrence of Thursday in the weekly cycle, but for my wife 'next Thursday' means one week after the next occurrence of 'Thursday'. I am never going to successfully change her use of the expression, and my use is likewise set; therefore we have learned to expect a misunderstanding whenever this expression or any of its variants is used. In this case, the meaning gets negotiated not by agreeing to a common use of the expression, but by shifting the specification of the timetable to a date, such as June 13th, instead of depending on a day specification such as Thursday. The reader is encouraged to listen carefully over the course of a single day to detect similar incidents of meaning negotiation. Alternatively, the reader might conduct a simple survey, posing the question in a neutral way, to determine what 'next Thursday' means to associates, colleagues and family, as a way of confirming my claim that meaning often requires negotiation.

I expect that there might be many differences of linguistic opinion between the text as expressed and the understandings which the reader might bring to the text. In order to help the reader begin to fill in both sides of the meaning negotiation, this preface seeks to present my position on a number of potentially contentious terms.

For the sake of clarity, I make a distinction between data modelling as an activity vis. data models as products. A systems analyst has little choice about whether to model user data: his only option is whether to do it entirely on the basis of his experience and intuition, or whether he might apply one or more methods from the analysts' tool box. Data modelling is an extremely important process which helps the analyst and the user community to negotiate a common understanding about the circumstances of the problem or opportunity, and perhaps to agree to a new approach for the future. In part, the result of this process is usually a data model expressed in one way or another, be it hierarchy, network, relation, or a COBOL data division. How much this data model contributes to the way the end user reconstructs the meaning intended by the system is a matter for much further research and thought.

This last sentence points to a very important position which I assume over the course of the entire document. Databases, including both the data structures and the contents, contain no meaning. All meaning associated with information systems is found in the

the people who designed it, who use it to record and distribute data and information, and who retrieve that data and information. The database and its contents are the primary formal evidence of this intentionality on the part of the database designers and the database maintenance people. What might be meant by the database must be inferred by the end user on the basis of the direct cues and clues provided by the data structures and their contents, as well as a whole host of tacit semantic relations and knowledge about the application which are not directly reflected in the database. We might refer to this necessary process of interpretation as the hermeneutic of databases, and the orientation of the research programme was to attempt to recognise overtly the hermeneutic foundation.

There are a number of other presuppositions and assumed positions implicit in the following research. First among them is causality, the notion that our choices are actually instrumental in effecting change. Should the reader hold the position that causality is fundamentally a fiction, and that what happens is primarily stochastic, that individual will find much to dispute in this text.

Second is the question of whether counting is a form of measuring. On the one hand, we find that one of the definitions of the *Oxford English Dictionary* specifically excludes counting from the idea of measuring, the implication being that measurement must be done using a continuous function, while counting uses a discrete one. This question is critical to the research, since the objective was to develop a method to measure data model stability, and the proposed method involves a process of comparison, classification, counting and calculation. I offer a famous physics riddle in defense of counting as measuring.

The headmaster of a school was asked to adjudicate a difference of opinion between a student and a teacher over whether he should pass the final exam, given some of the controversial answers the student had provided. The principal, teacher and student agreed that there would be one additional question posed, and that the student had to answer it correctly whilst demonstrating some principle of physics. The question was to determine the height of a very tall building using a mercury barometer. The answer the headmaster anticipated was that the student would take the air pressure at the bottom of the building as well as at the top of the building, and by using the difference in pressures, calculate the height. Instead, the student stated that of the five other answers which he created for the problem, he proposed to drop the barometer from the top of the building and time how long it would take to smash at the bottom, and calculate the height using the gravitational acceleration factor. This answer clearly involved physics principles.

One of the other answers was to take the barometer to the bottom of the building, mark the top edge of the barometer, move the barometer up to the mark and make a new mark at the top, repeating the procedure until he got to the top of the building, whereupon he would be able to count the number of times he move it. The height of the building would be expressed in units of barometers. In my view, this process of counting is every bit as much a measure as calculating the height from the length of time it took the barometer to reach the ground from the roof, and might actually be more precise given the error which could be made in measuring the time to fall. Thus is counting a subset of measuring. (The other three methods of determining the height of the building are left as an exercise for the reader.)

The third potential area of tension between the text and the reader might be in the use of the term 'objectivity'. The text makes frequent use of this word and its derivatives, especially in the context of the philosophical question of whether data models are the dispassionate documentation of an 'objective reality'. The text does point out some of the problems with the notion of an 'objective reality', and I offer the opinion that most organisations do not **behave** as if there is an objective reality. However, the stability measurement protocol does not depend on the user taking a position one way or another.

The text uses also the word 'objectivity' and its derivatives in the sense of 'without bias', for example in the discussion about the qualities of effective measurement. The reader might be tempted to offer the view that if reality is entirely subjective, how is it then reasonable to raise the notion of objectivity in the application of the stability measurement protocol. I suggest that all measurement requires the application of some form of judgment, and that the judgment which is applied is done so in the context of two implicit questions: 1) how much wobble or uncertainty the measurement process admits, and: 2) is that degree of uncertainty consequential? When measuring a piece of wrapping paper for a gift, we implicitly understand that there are two basic mistakes to be made - too much or too little - and only one of these mistakes is likely to be important. The precision necessary beyond this general rule is not very high. Judgments must be made in a number of different areas of the measurement protocol. The idea is to make these judgments based on the available evidence, without favouring one outcome over another, as far as this is humanly possible.

My personal opinion about data modelling is that it has been mathematised in ways which contribute primarily to the technical design. Notwithstanding claims to the contrary, there is little scientific evidence to support the assertion that these various modelling forms have contributed significantly to the front end, that is to say the problem exploration and resolution. Modelling theoreticians admit that policy formulation is the foundation to successful data modelling, but it is very difficult to demonstrate how one formalism is superior to another in articulating the problem, developing alternative solutions, implementing the solutions effectively and providing continuing quality systems to the user community. Having declared this personal opinion, based on my review of the literature and on my professional experience, this opinion is of no consequence to the development and application of the measurement protocol documented in this thesis. Whatever view one might have on the art and science of data modelling, if one is going to make claims about data model stability, there must be a method to measure these claims. In this thesis, I document such a measure.

The meaning of all communication is negotiated among the parties to the communication. I invite the reader to pursue this text in the spirit of negotiation. I challenge each reader to exploit the tensions created by the text as a way to clarify their own interpretations while speculating on the intent of mine.

PART I
THE THEORY AND PRACTICE OF DATA MODELLING

CHAPTER 1
INTRODUCTION

Electronic data processing has been firmly established as a vital element in the life of the modern large organisation. Many large organisations think of information systems as a critical success factor without which it would be impossible to operate the business. Airlines, credit organisations, the military, and government all depend heavily on the use of Information Technology (IT). The penetration of IT has reached further and further into the economic infrastructure of the community as a whole to the point where even relatively small enterprises have implemented this tool.

In the early days of computing, applications were very narrow in their conception, limited to such traditional "number crunching" activities as accounting and operations research. However the development of multi-processing and on-line real-time processing technologies resulted in applications that could be developed in the context of a much larger group of simultaneous users, usually from the same functional area of the organisation.

Over the last 20 years, the price performance of computer hardware has changed dramatically. The economics of hardware has had a profound influence on the development of software, and the focus of the systems development effort. Twenty years ago, the relative expense of acquiring the hardware and operating it in special physical environments meant that the development of commercial software had a much greater focus on machine resource utilisation. Software education of the early 1970's placed a premium on executable code which ran quickly and used the smallest processing region possible. It was only when the relative expense of maintaining such code, much of which was quite idiosyncratic to the developer, outweighed the cost of the machine resources required to operate it, that the systems profession began to think about design decisions which influence the maintainability of the software code over its efficient use of the hardware.

The development and use of data processing systems in the 1970's continued to be comparatively restricted in functional scope. Many organisations developed their applications within the constraints of a relatively narrow organisational framework, e.g. accounting applications for the finance department, and marketing applications for the sales department with no shared information about the same customer. Even today, there are many organisations which have application profiles that clearly demonstrate the organisational boundaries which were in place at the time of system development. However, system developers eventually began to change their orientation to data and information.

The changes in data and information management might be viewed from the perspective of the transformation from calculation based thinking to knowledge engineering. The programming of the 1950's began with its primary focus on calculation, but gradually moved toward more generic symbol manipulation, that is towards "data" processing. With the focus shifting from number crunching to data processing came the challenge of changing the data definitions which users demanded in their applications. For example, when a traditional COBOL data element in a given file had to be changed for any given reason, each of the programs which used that element had to be changed as well. It became clear to academics and practitioners alike that the separation of the "data" management from "processing" management might improve the efficiency of the systems maintenance function.

Another source of technology change influenced the orientation of systems professionals to data and information. New operating system utilities and technologies, including changes in direct access storage medium, made multi-processing and re-entrant code possible. This in turn made possible multi-user on-line data processing. Once the technological possibility of relatively convenient shared data use was introduced, the potential for data and information to be viewed in the context of the organisation as a whole emerged, resulting in new ideas about the nature of this resource.

The opportunity to share data, and the parallel development of database software products which enabled this objective, resulted in the development of numerous formalising techniques to help the systems practitioner come to terms with corporate shared data needs as well as the specific data needs identified in the analysis of a given narrow application requirement. In keeping with the historical reason for the separation of data analysis from processing requirements (i.e. improved efficiency in systems maintenance, and improved opportunities for data sharing), the developers of these formalising techniques made claims about improving the stability of the data models generated by the techniques. For example, Navathe and Kerschberg (1986, 23) claimed that the enterprise model should contain relatively static or invariant data. This claim suggests that a data model should consciously exclude data which might be unstable. If this is so, then research should demonstrate that data models are highly stable. However, such claims of improved stability have never been formally demonstrated, for the simple reason that there has never been a method to measure such stability.

Therefore, the first objective of this research was to determine if it would be feasible to develop such a stability measurement tool. If such a development were possible, the tool should then be applied and evaluated. Finally, it would be valuable to apply it to a variety of application models which have changed over time, in order to develop a preliminary idea of what the sources of data model instability might be.

This thesis is structured into three major sections, each with three chapters. The first section describes the foundations and practice of data modelling. Chapter 2 reviews the intellectual and theoretical foundations and Chapter 3 reports on data modelling as it is practised. The second section of the thesis describes the model stability measurement tool, with Chapter 4 describing the development of the tool, Chapter 5 reporting the results of applying it, and Chapter 6 providing a critique of the tool and the method of its use. The third major section of the thesis develops a critical theory of data modelling. Chapter 7 discusses stability, instability, and evolvability of data models, Chapter 8 reviews the implications of the research findings, and Chapter 9 provides a conclusion and thoughts for further research.

CHAPTER 2
THE INTELLECTUAL FOUNDATIONS OF DATA MODELLING

Data modelling techniques are like the tools of any trade in that they have a fundamental impact on the process and content of the trade itself. Indeed, sometimes our tools define the "trade" itself. Naturally, we try to develop our tools to make our work more effective. The information systems profession has many different tools, which cover many different aspects of the process, including data analysis. The question we often fail to ask is how these tools subsequently shape and constrain the work we do, and how they contribute to the limitations we bring to our work.

Decision tables, flow charts, precedence networks, data models, formal specification, data flow diagrams, and structured methodologies are just some of the tools which have been used with varying degrees of success. Few, if any, of the tools have had a bigger impact on the development of automated information systems than data modelling in its various forms. Some form of data modelling technique is used across all of the phases of the systems development life cycle, from initial problem formulation to analysis, design, implementation, system use by the organisation, and to systems maintenance. The objective of this chapter in the thesis is to explore the intellectual foundations of data modelling to give us a starting place for considering what stability might mean and how we should go about measuring it.

In order to assess the data modelling techniques which are now so overtly used by the modern systems analyst, this chapter begins by examining the roots of data modelling theory. The second part of this chapter presents a description of the philosophical foundations of data modelling. Finally, we will review what the information systems community reported as the actual practice.

A. THE ROOTS OF DATA MODELLING THEORY

An overview of the roots of data modelling provides a context within which to develop our expectations of data modelling as a technique for systems analysis. Modelling tools of varying forms have been an important part of the systems tool-kit since the first computer systems were developed. The basic reason to use some kind of modelling tool as part of the problem solving attack is to organise and manage complexity (De, Sen, Gudes, 1982, 1). In their work, many professionals use one form of modelling or another in varying degrees of abstraction. Aeronautical engineers use model airfoil sections in the wind tunnel. Architects routinely build physical scale models. Architects are also working with computer assisted design products to develop models of the proposed end product which are lifelike to the extent that they are called "virtual reality" since the user will be able to "experience" the model in a real time way by "walking" through it and inspecting the result through special goggles. These virtual realities, as well as less sophisticated models, can also act as an important vehicles for clarifying and communicating an understanding of the requirement, in a way which encourages consensus about the nature of the problems and their possible solutions.

In the systems practice of the late 1960's and early 1970's, flow charts, decision tables and decision trees were popular. Critical path method dependency diagrams, and PERT charts were also often used. One of the reported problems with all of these tools was that they were very labour intensive, and therefore, the outputs of these techniques were often not maintained. More important, there was no commonly accepted theory or discipline in applying them. For example, different individuals would generate different flow charts for the same circumstance. Consequently, maintaining idiosyncratic flow charts overwhelmed many systems maintenance groups.

In addition to the idiosyncratic model diagrams, maintenance groups had also to maintain idiosyncratic programming code. This undisciplined code and its consequences gave rise to the "structured" techniques with the widespread promotion of "structured programming" (Dahl, Dijkstra, Hoare, 1972), top-down program design using decomposition techniques and stepwise refinement. This initiative was successful enough that the "structured" idea was then applied to technical design and finally to systems analysis generally, as well as data analysis too (Gane and Sarson, 1979). The structured disciplines sought to consider carefully the questions of data and processing hierarchies, communication, and control (Yourdon, 1989). The authors who wrote on the benefits of these structured approaches had high expectations of these methods. For example, Ross and Schoman (1977, 7) wrote that structured requirements analysis would "capture on paper, at the appropriate time, all relevant knowledge about the system problem in a complete, concise, comprehensive form."

Alongside these developments, the systems maintenance professionals of the 1970's observed that changes to the structure of data often had a ripple effect throughout an entire system. This observation in combination with new direct access technologies resulted in the development of database management technology. With specialised software to manage the data definition, data access, and data security, it was then possible to separate the management of data structures from the processing cycles which created, modified, and destroyed the data content. The technology which enables the technical separation of data from processing has had a major influence on systems methodologies. User communities are less often concerned with the "processing" part of the problem and far more interested in the "data" question, since they assume that they will be able to generate the information they need, and process it at the necessary level of summarisation, as long as they have the raw data available at the most detailed level possible. This thinking was encouraged by systems professionals who were likewise searching for "atomic" data elements in their analyses (Nijssen, 1977b, 39).

Systems development has thereby become data driven, and indeed the literature frequently refers to data driven development as a highly desirable approach. The processing side of the problem is considered partly as a way of clarifying what data structures will be necessary, and as a way of validating the proposed data model. There are some authors who were in favour of maintaining the connection between data and processing, for example the process model holder proposed to be managed parallel to the data model holder (Zahran, 1981), but most approaches separate the two.

Data analysis or data modelling has thus become a separate discipline with its own analytical tools. Data can be analysed or modelled with a wide variety of techniques. Martin and McClure (1985) reviewed no fewer than 17 different diagramming techniques used in systems analysis generally. Of these, they classify 4 types of diagramming techniques which are specific to data compared to processing activities: entity-relationship diagrams, data structure diagrams, Warnier-Orr diagrams, and Jackson diagrams. However, most modelling theorists insist that the processes which the data participate in are an important part of validating the data design. Data flow diagramming, decomposition diagrams, Jackson diagrams, data navigation diagrams fall into this class.

The data analysis theory is not clear about the relationship of "data" to "processing", and how separate these notions can be. As noted above, there is no doubt that the process-related considerations have a significant impact on the data structures. However, the data management tools currently available separate the data structures, definitions, content, access methods, from the processing elements of the application systems, by design.

Whatever the relationship between data analysis and process analysis, Wood-Harper and Fitzgerald (1982) classify both of these approaches under the science paradigm. We should consider the question of whether the intellectual foundations of data modelling can properly be described as science. If we are confident that data modelling is scientific, then we might have higher expectations about the process and products of the activity. The question of data modelling as a science may also be somewhat problematic given Popper's (1972, 33) observation science often makes mistakes, and pseudo-science may stumble on the truth. However, the general expectation is that the more scientific something is the greater the degree of control we expect to have in that domain.

Exploring this question of "science" and "art" as it relates to data modelling must consider the question of how each of these terms might be understood. We might begin with the ideas of "science" and "engineering". The information systems development community has published a significant number of articles based on the notion of information engineering. The idea of engineering is generally understood as applying science in order to plan, design, construct and manage towards a clearly conceived objective. The key to engineering is the existence of scientific principles which have been demonstrated to the satisfaction of the scientific community of the day. When we are confident that a specific conceptual framework has been tested rigorously enough in order to qualify as science, we can have a relatively high degree of confidence that if the principles are applied, the results will be highly predictable. Causality and subsequent predictability are the foundations to effectiveness in science and engineering. This predictability is typically achieved by testing assertions and assumptions empirically. The results of the such testing are carefully measured and interpreted.

The standards of empirical testing, careful measurement, well established causality, and high predictability are absent in data modelling methods research. For example, Flavin (1981, 99) claimed that information modelling and semantic analysis, as he described them, will pose questions and identify policy at "precisely the right level of analysis" because of its use of semantic modelling concepts, and a particular analytical language. In his work, Flavin has offered no way to measure and interpret such a claim. Indeed, it is difficult to imagine how one might go about demonstrating this statement in a way which would be accepted as scientific for the following reasons.

Let us begin with his ideas of semantic analysis. At the outset, we should note that the semantic modelling concepts which he refers to are primarily those related to entity relation modelling, and not to semantic modelling concepts which might be found in the linguistic literature. The information modelling processes referred to in Flavin (1981, 54) are functional analysis, scenario analysis, transaction analysis, abstraction analysis, and anchor point analysis. The first three of these analyses are used to explore the application area to search out entity types of interest. Abstraction analysis is used for determining the level of detail at which the objects of interest in the application are to be declared. Anchor point analysis is used to refine the first-cut model determined through the above analyses, to establish "well-defined relationships."

Flavin's claims about information modelling achieved through the application of the techniques described above are quite complex ones. The idea of "direct analysis" is to be contrasted with what he calls "indirect analysis" achieved through "organisation-wide data collection or of mathematical dependency analysis". Any evidence which suggests that this direct analysis produces a result superior to indirect analysis would be strictly anecdotal, and are not provided in his book.

The claim that these techniques achieve the "precisely right level of analysis" begs numerous questions. The book does not specify what is meant by the right level of analysis. The author does not tell us how to determine how many levels of analysis are there in the analysis of a given system. Even if we had a good idea of the levels of analysis which might be applied to a given problem set, the author does not provide any criterion which might be used to judge whether one level is preferable to another. Presumably there are consequences to the quality of the analysis if the incorrect level were chosen. Given the importance which the author places on achieving the right level of analysis, I assume this would affect the quality, size, or stability of the data model. Each of these important assertions are untested.

Things are no clearer when we look at some of the most frequently referred to elements of the literature, the contributions of E. F. Codd. For example, Codd (1979) published an article called "Extending the database relational model to capture more meaning." It is possible to be generous in the interpretation of the article and suggest that the idea of a data model "capturing meaning" is intended primarily as a metaphor, especially with his proviso that "the task of capturing the meaning of data is a never-ending one". However, when he said "in addition, a meaning-oriented data model stored in a computer should enable it to respond to queries and other transactions in a more intelligent manner", it is clear that he literally meant "capturing the meaning". Codd (1979) went on to say that the model itself will act as the mediator among the varying external views which the application programs and end users have as contrasted to the multiple internally stored representations.

What claims are implied by these statements? Firstly, it is not clear what is meant by the idea of a "meaning-oriented data model". The best interpretation which can be made is that there are variations between data models as to their degree of meaning orientation. It is difficult to imagine how he might propose some way to measure or to confirm the degree of "meaning-orientation" which a data model "contains". Secondly, a similar question might be raised about the effectiveness of models as intelligent mediators of external views. Developing a variety of models and then testing them for the greater or lesser intelligence and effectiveness as a mediator would be a significant achievement in applied systems analysis.

How can we judge whether such theories and proposals are "scientific"? Popper (1972, 33-37) posed exactly this question of theories in general. He summed up his discussion about the "problem of demarcation" by saying: *"the criterion of the scientific status of a theory is its falsifiability, or refutability, or testability"* [italics in the original]. He commented on what counts as confirming evidence:

> Confirming evidence should not count *except when it is the result of a genuine test of the theory*; and this means that it can be presented as a serious but unsuccessful attempt to falsify the theory. (I now speak in such cases of 'corroborating evidence'.) [italics in the original].

Thus data modelling theory is scientific, in the sense that there appear to be real ways of testing some of the assertions made in the literature. However, given the state of the testing, I conclude that data modelling is not scientific in the sense that the claims of modelling theories and principles have been tested severely and have been found to support effective predictability. What this conclusion basically says is that data modelling has an unrealised opportunity to become much more scientific.

The roots of data modelling theory are in the science paradigm. Modelling in general is a common scientific practice, whether these models are physical, mathematical, or schematic. Data modelling literature also seeks to place itself within the scientific rubric, as is evidenced by the many schools of "computing science". However, a careful reading of the data modelling literature demonstrates that it has emerged as a result of technological advances in hardware and software performance, not from a foundation in a scientific theory. Whatever the historical origins of data modelling, the tools and techniques which are in use have a variety of philosophical assumptions which must affect our understanding of the potential and limitations of the techniques.

B. THE PHILOSOPHICAL FOUNDATIONS OF DATA MODELLING

While the average working systems analyst probably does not think much about the philosophical elements of the data modelling process, these elements are extremely important in shaping our understanding and application of the tools. This is especially true given the number of variations in approach to data modelling specifically and systems analysis generally. The following section of the thesis discusses:

- the philosophical subjectivists vs objectivists;
- the semantic and linguistic issues; and,
- the use of a data model as a communication vehicle.

1. Is Reality Subjective or is it Objective?

One of the best reviews of the philosophical foundations of data modelling was presented by Klein and Hirschheim (1987). They classified the various methodologies under the two general headings of "objectivist" versus "subjectivist". The objectivist school of thought takes the position that there exists an objective reality populated by empirical entities. This orientation to a "real world" which is an objective reality to be discovered and documented by the data modeller is extremely common in the literature of data modelling. We can contrast this view to those who suggest that reality is primarily a social one, to be created, not simply to be discovered (Berger and Luckman, 1971).

There are many references in the literature to the "real world". For example Edgar (1987, 3) in his dissertation on the design of a unified data model, criticised existing data model constructs, which in his judgment do not provide a complete set of constructs to ideally model all aspects of the real world. His objective was to identify constructs to represent the real world as accurately, precisely, and unambiguously as possible. The main objective behind this precise representation of the "real world" was to reduce the incorrect retrieval and manipulation of data. Zahran (1981, 193) wrote that data dictionaries must be able to describe all relevant elements of the "real world".

Gordon C. Everest in his textbook on database management (1986, 143-44) talked about seeking to capture all "essential aspects of the real world" being modelled. However, he did allow that someone has to make a judgment about which information is necessary and which is spurious or premature to the modelling process itself. Therefore, for Everest there are two problems to address in data modelling; 1) identifying the fundamental aspects of the real world, and 2) making a value judgment about their relevance.

Even those parts of the literature which seek to criticise the most common forms of data modelling methods, the relational models, make important assumptions about the existence of an objective real world. For example, Davenport (1978, 83) indicated that the relational model achieves a high degree of data independence but loses some of the important semantic content about the real world. This observation suggests that there is some kind of trade-off between the cost of losing semantic content against the benefits of data independence. The idea of real world entities is further complicated by the idea that some of things in the world are entities which eventually result in the creation of record types, while all of the other things in the application world are not entities (Kent 1985, 176).

Sundgren in his *Theory of Data Bases* (1975, 5) was quite straightforward about stating his position. He said:

> We postulate the existence of an objective reality outside the mind of every beholder. As we have thereby taken a particular position vis-a-vis one of the eternal philosophical issues, some argument on the subject should be provided. The assumption of an objective reality is equivalent to the assumption that it is always possible to determine objectively at least in principle, whether a particular statement about reality is true or false. It seems reasonable to maintain that statements, which users cannot agree on as being very viable, have no place in the data bases of the kind to be discussed here.

In addition to the important issue of whether there exists an objective reality or not, Sundgren has made other statements which deserve some attention. The first is that databases are about making statements which are true or false. This assertion is presented because much of the subsequent database theory is based on predicate calculus. However, it is not difficult to think of a circumstance where the truth value of a statement might be true in one context and false in another. The second serious consideration in the above quote is that non-viable statements have no place in the database. I am unclear as to what the word viable might mean in this context. Finally, there is certainly some question about whether databases contain statements that all potential users can agree on.

On the other side of the argument, there is a group of theoreticians who are either not prepared to make a philosophical claim about the subjective versus objective nature of reality or who sense that it is more precise to describe the world as being primarily a function of the agent who perceives it. For example, in the linguistic community, Aitchison (1985, 71) noted that semanticists who use the term "real world" would probably agree that "the external world as perceived by humans" is really what they mean. The implication to this statement is that perceptions vary, and therefore, we can have more than one "real world". If this is true, we would then have the potential for multiple data models, based on the varying perceptions of the user community.

The data administration answer to this particular problem is that a given data model must be constructed to reconcile the various user or application data models into one comprehensive model, such that each user can have his "view" accurately and adequately represented. The proverb of the blind men asked to describe an elephant fairly represents the understanding which is prevalent in the data modelling community. Each blind man was in contact with a different part of the elephant's anatomy. The man who had the tail described it as a rope. The man who could feel the leg described it as a tree. The man who had the trunk described it as a snake. The man who had the ear described it as a fan. The traditional lesson of this parable is that one must integrate all of these different views in order to describe the objective reality completely and accurately. Data modelling professionals propose to be the ones who will facilitate the integration of the varying application and business perceptions into the one "true" model. But what happens when one of the blind men actually has a snake, when the second is touching the trunk of a real tree, and the third has a rope? The danger is that the data analyst will insist the blind men just have different parts of an elephant.

Chua (1986) documented the same concerns about these objectivist assumptions where they are built into the world view of accountants. He noted that mainstream accounting thought begins with the claim that there exists an objective reality which has a determinate nature. He suggested that this appearance of a neutral technical rationality in accounting information is particularly useful in legitimising activities. This argument can be extended to forms of information well beyond accounting applications. As Chua noted, this fiction of neutral rationality may narrow the public debate to issues circumscribed by the systems in place. We talk about the data in our accounting systems and other applications, perhaps to the exclusion of other topics, because the existence of the systems themselves constrains our understanding of what might be real and what part of the reality is important. The fiction of rationality implied by the automation of information is part of the magical belief about computing found in some user communities.

Marche (1978) documented evidence of a magical belief in the credibility of computer generated information when he reported seeing clerical staff re-typing computer generated output which had been manually corrected due to input and processing errors. The report was being re-typed back onto computer paper to preserve its credibility. The credibility has been implied by the apparent objectivity of systems technology, data modelling and information engineering.

Instead of the objectivist approach, it is possible to take the approach that the world is comprised primarily of a socially constructed reality and that this reality is achieved through the consensus developed in a given community. In this case the data modelling process is primarily one of developing, extending, and negotiating the existing consensus in the community and documenting what that might be.

The sharing of any resource in a large and probably diverse organisation may have improved control and efficiency as an objective, but attempting to develop a consensus about the data resource is also likely to produce conflict. In *Database Management: Objectives, System Functions, & Administration* (Everest, 1986, 577), the chain of events was set out as:

SHARING
produces CONFLICT
necessitating MEDIATION
resulting in COMPROMISE for one,
but yielding GLOBAL OPTIMA for all [emphasis in the original]

Where sharing is proposed across a wide number of traditional corporate boundaries, the resulting conflict is likely to reach high levels in the organisation. This factor has also encouraged centralised control over, if not centralised creation of, the data models which are the necessary precursor to the use of a Data Base Management System (DBMS).

R.A. Frost (1986, 113) in the proceedings of the Fifth British National Conference on Data Bases, identified two other important philosophical assumptions in the data modelling approach. The first assumption is the "closed world assumption" (CWA) which states that if a fact is not explicitly recorded in the database, then it is assumed to be false. Given the socially constructed nature of data models, the CWA is unlikely to be consistent with the interpretation that the average user will make of a given database.

The second philosophical assumption Frost documented was the "domain closure assumption" (DCA), which asserts that the universe of discourse is comprised of named individuals only. In the context of data modelling, two things are implied: first, that the named entity types are the only entity types which are of consequence; and second, that individuated entities represented by each data record are the only instances of this entity which exist.

Frost had numerous concerns about these philosophical assumption, most of them directed at the varying possible interpretations which developers might make of them. According to him, one's interpretation might have a significant impact on the design and implementation of semantic integrity constraints, as a minimum. He complained that these assumptions are typical of the foundations of data modelling in their vagueness, and likely that they have an impact on the model.

Frost's entire discussion has at its basis first order predicate logic. Predicate logic is the strength and foundation of the relational techniques developed by Codd (1970, 1982), Chen (1976), Date (1986) and numerous other writers. We must consider the contribution and the limitations of predicate logic as a theory for data modelling and systems analysis. The idea implicit in the relational model is that once we have established the relations and relationships in a way that the user community will agree with, we will be able to depend on the implied truth functions of the agreed data structures. We can then exploit the mathematical and logical purity of these model-theoretic structures to apply the deductive rules enabled by the structures. Lockemann et al. (1979, 61) underscored the importance of the formal properties by declaring that they should have priority over descriptive concepts.

There are problems with depending on this theoretical foundation. First, according to Allwood, Andersson, and Osten (1977, 16) deductive logic has been explored much more thoroughly than inductive logic. However, inductive logic is far more common in business analysis than deductive logic. Second, as Allwood, Andersson, Osten (1977, 29) noted, the translation of natural language into "an unambiguous system of formal representation" has proved to be difficult and in the end, somewhat arbitrary. Specifically, applying the set-theoretical ideas as the system of formal representation for solving the business systems problems creates difficulties. Dowty (1979, 375) offered the following thought: "to get to the point right away, let me confess that I believe that the model-theoretic intension of a word has in principle nothing whatsoever to do with what goes on in a person's head when he uses that word." What goes on in a person's head when he uses the terms which will form the basis of a given data model is exactly the concern of the data analyst. It would be unreasonable to suggest that we can ignore what goes on the users' heads when they use data structures and their contents.

Lastly, the notion of determining the "truth value", which is the model theoretic objective of deductive analysis, might be problematic in the social context where information systems are used. "True or false" may not be nearly as important to business systems as useful or not useful. The strength and contribution of predicate logic underpinning the relational model is the Aristotelian notion of true or false, in all contexts. Predicate logic depends on the idea that the truth value of a given statement is preserved over all contexts of the statement.

The question of "truth" may be less important than the very different question of whether it is "useful" to declare something as true. Let us look at what may seem to be a trivial example. A recent American court case has adjudicated the truth of the statement that the children's action figure, GI Joe, is a doll, as opposed to being a toy soldier. Hasbro, the distributor of GI Joe, asserted that this toy is a soldier, while the American government asserted that it is a doll. The ontology of GI Joe determines whether Hasbro must pay the substantial duty imposed on dolls which is not paid on imported toy soldiers. It is hard to imagine that there could be some philosophical formulation which would adjudicate the question. The debate seems to me to be primarily linguistic, not ontological. Quite apart from the linguistic and philosophical question of GI Joe's ontology, the American government has the power to pass a law which deems GI Joe to be one or the other. The act of deeming is basically one of saying that it is useful for a given purpose to agree that a circumstance is the case, notwithstanding what most people would believe about the circumstance. While the American public might classify GI Joe as a toy soldier, for the purpose of customs and excise taxation, he is a doll.

There are many examples of this phenomena of "deeming" a fact, notwithstanding the facts. In most Western nations, the date of an individual's birth is an extremely important element of biographical data. One's birth date determines one's age, and one's age defines whether one is subject to a wide variety of legislation. The right to drive, to go to school, to have access to government assistance programmes, to get married without parental permission, to consume alcoholic beverages, to vote, to serve in the army, and to collect old age pension are all a function of age.

In some remote areas of Canada 60 years ago the record of births was not nearly as disciplined as it usually is today. This situation poses something of a problem for administrators of public registries. They have each developed policies for evaluating the often contradictory evidence relating to the circumstance of a given birth. In the end, each registrar will make a judgment on the basis of the evidence and "deem" a birth date for the individual. The actual birth date of the individual is not what is under debate; it is the evidence for the event that is key. So when this data is entered into the appropriate database, we cannot actually say that this is the true birth date of the individual; we can only say that we have deemed it to be so, based on the best available evidence.

I would be much happier with the statement "it is 'useful' for us to say such and such is the case", because this formulation preserves two important principles of effective modelling: 1) agency or responsibility for the statement; and 2) the teleology or purpose, since the word 'useful' implies a purpose or a goal implicit in the assertion. Depending on the truth value of a database assertion is a simply a problem because of the assumption that a database assertion is always true or false, whatever the context.

Whether there is an objective reality or not, most organisations do not operate as if there is one and consequently the social consensus is more important than trying to achieve an objective data model. In other words, the existence of an objective reality is not relevant to the systems development process if the various sub-communities in an organisation do not have the same understanding of reality. If the organisation operates from a framework of overlapping socially constructed realities which are at some disagreement, the prospects are likely dim for convincing each sub-community to modify their world view to the that of the "objective reality" as articulated by the systems community. The analysis of empirical data models undertaken by this research yielded valuable anecdotal data on the question of world views and how these views create inconsistent data models.

In any event, the idea of an objective, immutable reality does not really account for such practical abstractions as contract, accounts receivable, employee, or asset. From a practical perspective, systems analysis generally and data modelling specifically may be more of a process of generating consensus, than it is a process of describing an objective reality.

In summary, the development of data modelling techniques was originally driven by the practical need to introduce more control into the systems development and maintenance process. While data modelling literature places itself in the science paradigm, there is no consistent set of philosophical assumptions and scientific principles to underpin the basic concepts. Finally, the most solid elements of the theory, those based on predicate logic, are used after the problems has already been constrained by language formulated by arbitrary methods. In short, the philosophical foundations of data modelling are found wanting.

2. Semantics

Data modelling is in part about interpreting what data and information might be meaningful to a user community. Where we find an examination of the idea of meaning, we must look to the linguistic issues which affect meaning. Many of the researchers and theorists in information technologies have identified the issue of meaning as being a particularly difficult one to address. For example, Stamper (1985) discussed the relationship between signs and what they signify as being primarily a question about meaning. A DBMS is just such a tool for organising and reporting signs and their signification or meaning. In the classic work by Ogden and Richards (1923), they examined the question of the meaning of meaning and identified well over a dozen different ways of understanding this basic idea.

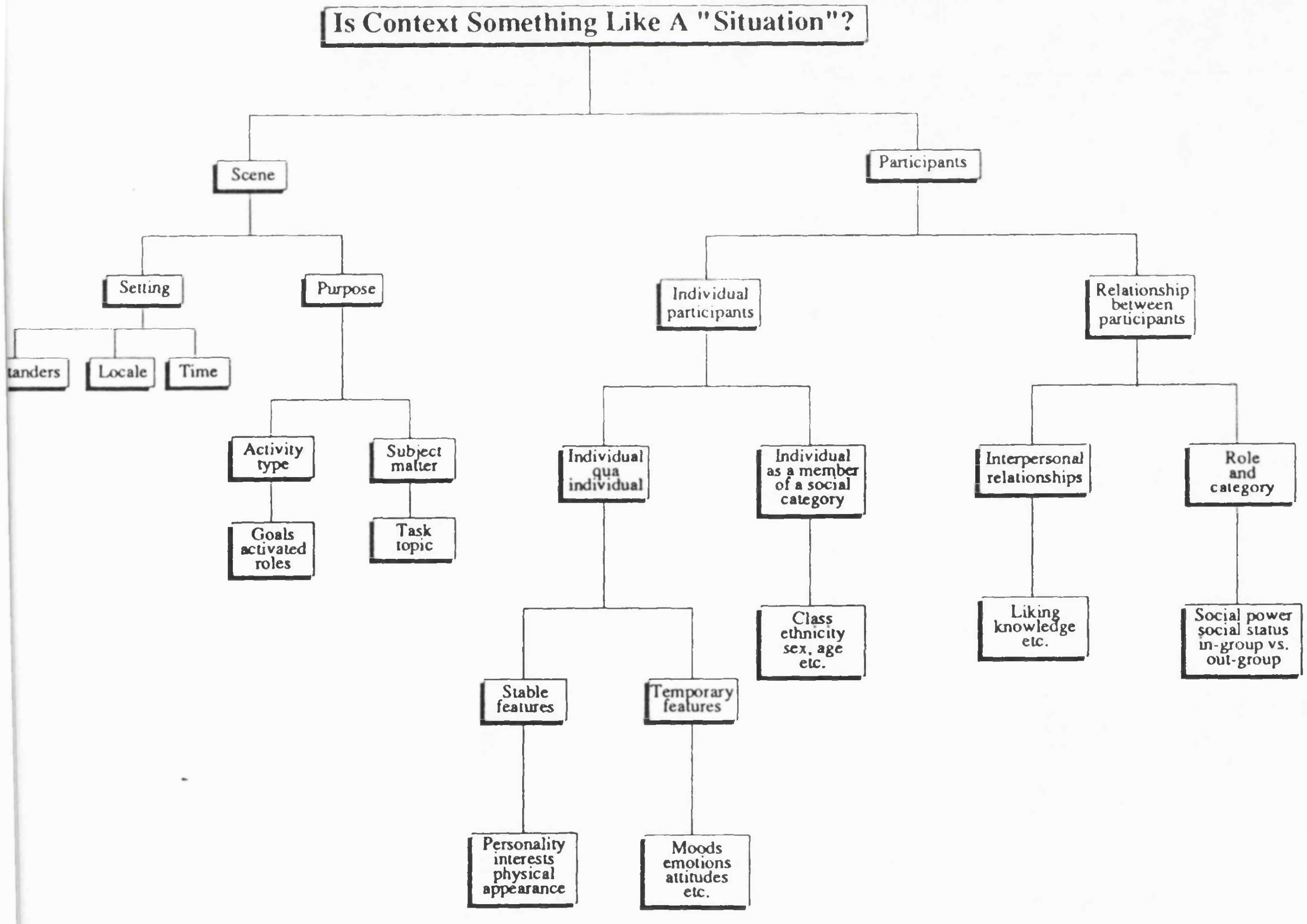
Even the question of the stability of meaning over time is one that is not easy to deal with. For example Stamper (1984, 9) said:

> Only within a definite group of people working towards a common end can there be a reasonable assurance that meanings do not shift. The assumption of constant meaning can be made to work by tightly restricting the exchange of data. The alternative, to live with shifting meanings, is at present unavailable because we do not possess the analytical tools to deal with the problem.

The problem with this formulation is that information systems may have no alternative other than to deal with shifting meanings. There is little evidence in the literature that systems research has determined whether meanings shift in an organisation and how these shifts in meanings might contribute to difficulties in the process of systems analysis. There might be a significant cost to an organisation if it made a policy decision for "tightly restricting the exchange of data". This notion would run directly counter to one of the fundamental rationales for having centralised data management, that of data sharing.

The purpose of a particular data element has an important impact on the interpretation of that data element by an information user. Database management systems have no way of overtly recording purpose. Therefore the meaning and use of a data element is often not clear to the person who subsequently uses it. This may pose a problem for communicating meaning over the distances of organisation, time, or place in the enterprise. For example, in one large Canadian business of over 10,000 people, 6,000 people regularly access the automated personnel directory. Among other things, the directory reports the names and addresses of employees. What is not generally understood by users of the directory is that the address does not record where the individuals work, but where their payroll cheques should be sent for distribution. Anyone who uses this data to locate an individual in his place of work will be disappointed and probably confused by the directory in approximately 10% of the cases.

The whole concept of "context", which is fundamental to the idea of shared purpose, is one which has consumed a great deal of time and attention by both computer scientists and linguists. Sperber and Wilson (1987) in their book *Relevance: Communication and Cognition* invested a significant amount of their analysis on the question of context. The preservation of context which contributes to relevance is not formally identified or discussed in any coherent way by data modelling literature, even though most discussions about data modelling want data models to be relevant.



Is Context Something Like A "Situation"?
Scene
Participants
Setting
Purpose
Individual participants
Relationship between participants
tanders
Locale
Time
Activity type
Subject matter
Goals activated roles
Task topic
Individual qua individual
Individual as a member of a social category
Interpersonal relationships
Role and category
Class ethnicity sex, age etc.
Liking knowledge etc.
Social power social status in-group vs. out-group
Stable features
Temporary features
Personality interests physical appearance
Moods emotions attitudes etc.

Figure 1.

The complexity of the idea of context is demonstrated in Figure 1 above from Brown and Fraser (1979), indicating a number of the characteristics which contribute to context. This diagram comes from the literature on communication, and not that of data modelling. Very few of the contextual elements as identified in this diagram are preserved in the final data dictionary, or data structures.

As Cook and Stamper (1979) indicated "the language division provides the link to the user, who, generally speaking, is expected to know the meanings words given in the context." From a data analysis perspective, the foundation for describing the organisation from a data processing point of view are the entities, attributes, and relationships found in the data model. We will have to consider seriously how well current database systems help to preserve context given the limits which might be inherent in these notions.

Another common notion in data modelling is that of the data element as a receptacle or conduit of meaning. This idea of a meaning receptacle is often carried into language through our understanding of the nature of words. Most people think that words carry meaning. Stewart Chase in *The Tyranny of Words* (1937, 42) called it "soul box" theory of meaning. He talked about the word as the magical receptacle which carries the essences of meanings. This notion of "carrying" is fundamentally a metaphor which highlights the importance of the word in communication, and hides the significance of the agent who utters the word and the receiver who chooses how to create meaning from it given the context. This point is thoroughly discussed in Reddy (1979) where he summarises his discussion with the view:

> We have the mistaken, conduit-metaphor influenced view that the more signals we can create, and the more signals we can preserve, the more ideas we "transfer" and "store". Storing with the complementary understanding of the need for interpretation constitutes the major weakness of the meaning-in-words hypothesis.

Everest (1986) underscored this general understanding about this meaning-in-words hypothesis. He said "data are 'facts' represented by values -- numbers, character strings, or symbols which carry meaning in a certain context." It is not clear why the word "fact" is in quote marks. Perhaps Everest was aware of the philosophical problems associated with the idea. Or he may have heard of McCorkle's (1977) definition of "fact": a fact is any assertion that people have stopped arguing about.

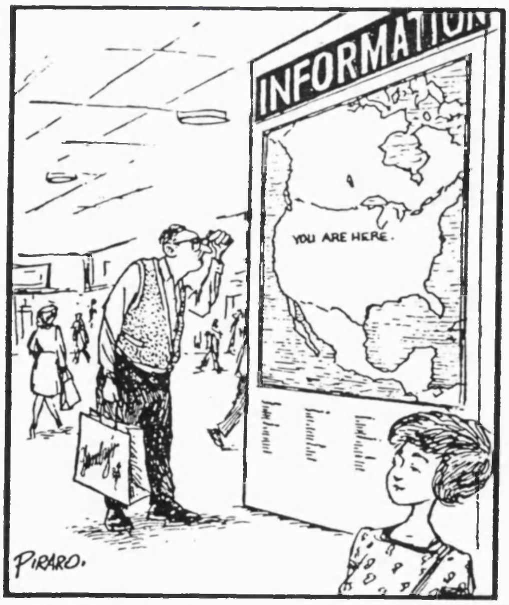
Chase (1937) also noted that one of the major causes of communication failure is the idea that a word has meaning independent of its use and that an independent, one proper meaning, should actually control its use. Data modellers follow the same path as they work through the process of data analysis towards developing a prescriptive lexicon. The objective of the data model may partly be to constrain the language permitted when using automated information systems, so as to limit the interpretations and potential misinterpretations which can be made from a given schema and its data.

Of the people who are critical of the denotational underpinnings in data modelling, William Kent (1978, 185-186) is probably among the best known. In his article "The realities of data reconsidered", he said that the main things needed to "capture the meaning" of data are: 1) the real organisation of the data as implemented, 2) the conceptual schema, and 3) links between data structures and implementations. The data structures, conceptual schemas, and their inter-relationships provide the meanings, i.e., "the interpretation of the data in terms of the relationships among the entities in the enterprise." Once again we are confronted with the concept that meaning is captured and once more the significance of the person who has to interpret the conceptual schema, the data dictionary, and the data itself is underplayed.

One of the fundamental characteristics of data modelling involves a process of data abstraction. Abstraction is primarily a semantic classification process which attempts to hide un-necessary details. Probably the best known proponents who assessed this particular process are Smith and Smith (1977a, 405 and 1977b). They identify two major data abstraction processes: 1) aggregation, and; 2) generalisation. They offered the following notion of "abstraction" as a place to begin: "by abstraction we mean the suppression of all details about some object (or activity) except those relevant to the understanding of some phenomenon of interest."

There is a marked resemblance between the "suppression of detail" in the abstraction process and the "highlighting and hiding" function of metaphors. Perhaps the key in understanding both metaphors and abstractions is not to confuse them with "reality". The above definition of abstraction raises the question of what details of an entity are relevant. The results of the research reported later in this thesis demonstrate that data models sometimes fail to collect important details of these objects, such as the inclusion criterion, since the current modelling methods have no way to represent this meta-data. The second major assumption implicit in the Smith and Smith discussion is that the "details of interest" are relatively stable. The review of the literature demonstrated that there is currently no reported evidence to support this assumption.

Necessarily implicit in the idea of abstraction is the notion of the abstraction hierarchy. There are two problems which relate to the abstraction hierarchy. Firstly, the level of abstraction designed into the data structures contributes to the meaning and appropriateness for the end user. There is such a thing as an inadequately low level of abstraction, in the case where the Director of Finance is presented with the detailed account balances and the transactions underpinning them, when he needs the accounts receivable balance for the general ledger. The other end of the abstraction hierarchy might be equally inappropriate as the following cartoon demonstrates.



(Used with the kind permission of the copyright holder, Chronicle Features, San Francisco)

One has to work hard to imagine a circumstance for this man where the information implicit in the abstraction "continent" would be useful. The level within the abstraction hierarchy is, within certain limits of natural language, a design choice on the part of the systems analyst. Note that this design choice is made early in the process, at the data planning and analysis stage. The second problem with the abstraction process is that there is little in the literature which provides guidance on the question of how to design an appropriate abstraction hierarchy which will maximise the probability of correct end-user interpretation of the data.

Flavin (1981) makes reference to the Smith and Smith (1977a, 1977b) concepts of generalisation and aggregation but adds two additional forms of abstraction. He includes functional differentiation and characterisation. These language abstraction processes which are such an important part of the problem exploration in systems analysis are an important precursor to data structure development. Fundamentally the abstraction process is one of language classification. However, most of the systems literature does not demonstrate any understanding of the documented problems of linguistic classification systems and its relationship to the language in use. For example, the classification of vehicles for those who use English would include certainly truck, car, bus, and probably aeroplane. However when we get into the more difficult examples, the classification system begins to pose a problem. For example would we include a roller skate or a hydrofoil or an all-terrain vehicle as examples of vehicles? As Aitchison (1987) documented, the semantic classification processes, which are necessary in creating the abstraction hierarchy, are not universal between linguistic groups. She indicated that most French people classify a ski as a vehicle. This demonstrates that language in use varies among linguistic communities with regard to classification approaches.

The question is whether such classification systems are also inconsistent between subsets of the same language group. For example in an organisation there might be different understandings of a particular entity type such as "asset". The personnel department might think that staff resources are clearly an important element of production and therefore constitute an asset to an organisation. Most organisations make various forms of investment in this asset, by way of career counselling, performance evaluation, training and education. Yet under generally acceptable accounting practices, the value of this asset is never recognised on the balance sheet.

It is clear that the choice of abstractions and generalisations which form the foundation of the conceptual model are fundamental to what can be done with the model later in its logical and physical design. It is also clear that the appropriateness of the abstractions could have a significant impact on data model stability. The choice of entities and attributes which the systems analyst makes constrains this process of choosing the problem space is akin to the process which C.S. Peirce (1958) identified as abductive thinking. Peirce indicated that while inductive and deductive reasoning are important to solving a problem, the process of selecting which problem is to be solved is also a fundamental part of the process. He termed this process of choosing the question to be asked as "abductive" thinking. A data modeller probably uses a similar process when he explores the requirements of a particular user community, prior to naming the entities.

Choosing and naming the entities, the attributes, and the abstractions for the data model is partly a design decision on the part of the modeller. It is not just a simple question of using natural language to describe the "real" world of the application described by the user and mediated by the analyst. Leech (1981, 26) pointed out that language itself imposes structure on the world by emphasising some distinctions and ignoring others. He noted that the foundation to this process is "supplied by cultural norms, rather than external reality." Perhaps it is reasonable to think of data analysis as a form of cultural anthropology.

There is one modelling process which takes as its fundamental starting point the idea that the norms of the community must be explicitly articulated in the analysis process. Backhouse (1991) presented case studies using the NORMA techniques for semantic analysis. This particular approach to the specification of user requirements makes two important assumptions (Stamper, 1986) which are at odds with most of the data modelling literature. First, there is no knowledge without an agent responsible for the creation of the knowledge. Second, the agent constructs his reality through action.

One other important notion which underpins the NORMA conceptual framework is that there exist relatively invariant behaviours for a social group, which constitute the functional evidence of the norms for that group. These norms are an important element in articulating the functional information requirements of the group. Notwithstanding the major differences between NORMA as a method of requirements explication, and the more traditional data planning and analysis methodologies, Backhouse (1991) made one claim which is common to both camps. That is, if different analysts use the technique on the same problem in the same context, the results of the efforts will be quite consistent. This claim has yet to be tested.

Veryard (1984, 7) raises the issue of semantic relativism as opposed to semantic absolutism in the creation of data models. By absolutist, he means those who believe that there is "only one correct or ideal way of modelling anything". We find a similar notion implied by Martin (1985) where he talks about designing systems from "provably correct constructs". The literature documents others who support various forms of semantic absolutism. For example, Gorman (1984, 5):

> ...There must be a formally defined set of rules or policies for data definition, capture, and maintenance. The data must also reside within its natural contexts; it must not be pulled from these contexts as it is with traditional systems. Finally, each piece of data must say the same thing to all who utilize it.

The idea of data "residing within its natural contexts" raises once more the question of how well each of the model development techniques help to preserve context. This idea should be compared with those thinkers who would like to develop context-free data management environments.

Contrasted to the semantic absolutists are the semantic relativists who Veryard (1984, 7) describes as those who believe "that most things in the real world can be modelled in many different ways, using any of the basic constructs". Hammer and McLeod

(1981, 354) are firmly in the relativist school of thought. They propose to use a semantic database model description language of their own development because, in their opinion, "contemporary, record-oriented database models do not adequately support relativism". In their view, a relativist view of data modelling is mandatory. Finally, Smith (1985, 85) falls into the relativist camp when he wrote that there is seldom one best design for a particular solution.

Flavin (1981, 25) certainly understood the problems associated with nailing down the components of the model. He said:

> In order to complete our information model, it is necessary to define each component of the model (objectives, relationships, data elements, and operations) unambiguously and consistently in the context of their meaning in the real world system. This is not an easy task: producing meaningful definitions of model components that are agreeable to a diverse user community can be a challenging and exhausting task.

The idea of entity types or object types having a precise identity for designers and users begs a number of questions. Precision in such a definition can only be in degrees of relative precision in the same way that a measurement is never completely precise but only precise to an acceptable degree (see Stamper, 1973). On the one hand, as Zadeh (1982, 27) said, "in some instances we elect to be imprecise because we do not need a higher degree of precision." Increased precision often means increased cost in generating the level of precision.

On the other hand, technology may contribute to producing an unnecessary degree of precision. For example, if you asked someone for the time of day thirty years ago, you would have heard an answer like "just after twelve thirty", or perhaps "twenty-five to one" if more precision seemed warranted. Today, if you ask the time of a person with a digital watch, the answer is much more likely to be "twelve thirty-four". Not only does this answer probably give you a level of precision you do not need, you would now have to make an intermediate calculation if your real concern was how much time was there to get to a one o'clock meeting.

The "required degree of precision" is part of the social consensus of relevant meaning in a given context. From an epistemological point of view, philosophers have understood this problem of consensus in meaning for some time. Douglas C. Arner (1959, 108) said "we always want to be correct but not at the price of endless checking and hedging. Considerations of reliability are always of the first importance, of course." Reliability is a key element of data models, and in this context, the reliability we are looking for is related to the interpretation which the end user of a system will make of the data model and its content compared to the intended interpretation. Data modellers intuitively attempt to achieve a similar kind of a consensus and reliability. Their objective is to construct abstractions and generalisations which will have the kind of reliability not requiring constant reconfirmation and checking within the organisation across the users of a particular data model.

However, there is a possibility that higher levels of abstraction introduce higher levels of ambiguity. One of the problems about the questions of aggregation, generalisation, and functional dependency as a process for data aggregation, is that such abstraction poses some potential confusion for the user community when the database is subsequently implemented. As Chase (1937, 5) said:

> When this tendency to identify expands from dogs to higher abstractions such as "liberty", "justice", "the eternal", and imputes living, breathing entity to them, almost nobody knows what anybody else means. If we are conscious of abstracting, well and good, we can handle these high terms as an expert tamer handles a lion.

The key in this statement is the original emphasis of the consciousness of abstraction. Chase went on to make the point that the true meaning of a term is to be found in observing what a man does with it, not what he says about it. This idea of meaning is completely consistent with Wittgenstein's notion of meaning in use.

However, data modelling seeks to separate what we say about data (i.e. the data model) as contrasted to what we do with the data (i.e. the process model). The difficulty is that the user who has access to a database may see the structure of the data by way of the entity classifications and the attributes underneath it, but he may have no idea how other users of the database actually apply the data in their business processes. Hiding the use of data abstractions helps to obscure the context in which a particular data construct is created.

The process of data abstraction, the naming of names in part, is always a risky business. Whorf (1956, 137) said "we always assume that the linguistic analysis made by our group reflects reality better than it does." The difficulties of data analysis and data abstraction have also been pointed out by Davenport (1978, 97). He wrote that entity analysis can be a major project involving a team of people over several months for just part of an organisation. He wrote that this entity analysis will be a continuous activity to reflect the changes to the organisation and its "reality".

Davenport's thinking implied that the models will require continuing maintenance. This seems to run contrary to the expectation that most users have of such a data analysis project. In many organisations, the separation of data from processing through data modelling by way of a database management system is done primarily because data structures are supposed to be more stable than the processing which depends on the data (see for example, Feldman and Fitzgerald, 1985, 81). Avison and Fitzgerald (1988, 180) noted that this stability of data types over processes is a major philosophical belief of information engineering, although there has been the occasional dissenter on the prospects for stability in shared database environments (Carden, 1986, 364).

As we have discussed in the section on the philosophical foundation of data modelling, data analysts talk extensively about "capturing the meaning" in their data models without explicitly providing a theory of meaning to support their assertions. If we cannot find much in the way of a discussion and debate about meaning in the literature on data modelling, we can find much to think about from the perspective of linguists. Leech (1981) has written a commonly used textbook called *Semantics - The Study of Meaning* which deals with meaning extensively. He identifies seven types of meaning:

- conceptual;
- connotative;
- social;
- affective;
- reflected;
- co-locative; and,
- thematic.

Of these seven types, conceptual, connotative, and thematic meaning are of most interest from the perspective of testing the meaning-preservation characteristics of data models.

The conceptual meaning is described as the "denotative" or "cognitive" meaning. Cognitive meaning is probably the kind of meaning that is implicit in most theoretical descriptions of modelling techniques. Connotative meaning is the communication value of an given expression which goes beyond the purely cognitive meaning. For example, the use of the term "physician" would normally mean, cognitively speaking, a duly qualified, registered practitioner of modern scientific medicine who seeks to identify and treat physical, mental and emotional disease. However, the connotative meaning of doctor in our society might also make implicit references to the typical social class, earning power, social influence, education and personal competence which are associated with this profession. By definition, any term or expression which is interpreted beyond the strict denotative sense has a connotative meaning. The shape and strength of the connotations tied to a term or expression is also influenced by the context. If data modelling theory focuses exclusively on the denotative meaning of entities and attributes, the connotative meaning could be stripped out and left behind with the context, especially if the end user depended solely on the data model and the data content to reconstruct the meaning intended.

Leech (1981, 18) used the collective idea of "associative meaning" as a summary for all of the meaning elements outside of cognitive, connotative, and thematic meaning. He claimed that the collection of associative meanings contains so many imponderables that the area can only be approached using approximate statistical techniques. There is no data modelling technique which exists today which uses approximate statistical techniques to determine the associative meaning elements of the data structures which are being proposed.

The last category of meaning which Leech addresses is that of the thematic meaning which is the result of the way the message is organised and delivered. This includes the particular technology which is used to deliver the message, the sequencing and focus of the message, and the emphasis. I suspect that there may be some thematic considerations to data modelling, but none of the modelling methods or techniques makes specific allowance for this element of meaning.

More generally speaking, we might ask whether meaning is abstracted from a database or whether it is projected onto it. Martin Gardiner addressed the mathematical equivalent of this question in the August, 1968 issue of Scientific American where he discussed the meaningful patterns in random numbers. It seems clear that random numbers cannot have intended meaningful patterns, by definition. However, when a person is given a set of numbers from a random number table, the individual is quite capable of describing the pattern "inherent" in each of the numbers. Is the pattern or meaning inherent in the number or in the person looking at it? The lesson from this article is that the role of the interpreter in creating meaning is absolutely vital. If a person can project pattern onto completely random numbers, then users who are confronted with a schema and its data will be clearly capable of creating patterns and making interpretations of the data which may or may not be intended.

Data models which are specifically designed to deal with semantic specification issues continues to pose a challenge to the information systems community. Agosti (1984, 7) claimed that semantic data modelling was not yet supported by a formalism with an adequate balance between completeness and simplicity.

To summarise the relationship between data modelling and semantics, the data modelling literature talks about the idea of the stability of meaning over time, and the problems related to this question but does not have an operational definition of "meaning" in this context. The theory of meaning in data modelling depends on mistaken notions of conduit and receptacle metaphors, and has little to contribute to the many other forms of meaning which have been identified by the linguists. As expressed in traditional data modelling techniques, meaning in the literature makes no allowance for variations amongst differing linguistic groups. The modelling literature underestimates the role of the user in interpreting and making meaning out of a model and its data. In short, for the most part data modelling techniques have no comprehensive and coherent theoretical foundation in respect to semantics.

3. Schemas and Databases as Forms of Communication

Among the many claims of data modelling theory is the notion that data models are a vehicle for organisational communication. One of the specific objectives of the relational theory as presented by Codd (1982, 110) was to provide a common understanding of the data so that users and programmers could communicate effectively about the data. Does data modelling using a particular approach or method improve the communication effectiveness of a given application database? How does such communication effectiveness affect the stability of a given data model, or does stability improve the communication effectiveness? Intuitively, if relational claims are true with respect to improved communication, one would expect the stability of these data models to be higher than data models developed using other techniques.

Separating the idea of meaning (as described in the previous section of this thesis) from the idea of communication is somewhat arbitrary, but a brief examination of communication theory will provide further insights in another area of theoretical weakness where data modellers make claims. The process of communication is founded on language, whether the expression of that language is verbal, written, automated information systems output, or by gesture. The way communication works is of vital interest to the information technologist, since it is at the foundation of the development and operation of all information systems.

There are many theories and models of human communication which have their basis in the idea that meaning is coded **into** the message and subsequently decoded on the receiving end. One of the communication metaphors which persists to this day was developed by two telecommunication engineers. Shannon and Weaver (1949) documented the simple straightforward model presented in Figure 2. Versions of this model have become imbedded in the thinking of disciplines a varied as politics, social psychology, and accounting (e.g. Arnold and Hope, 1983).

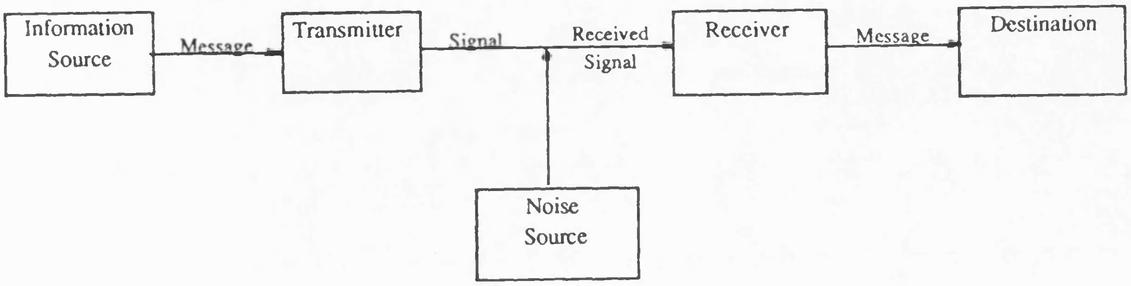


Figure 2.

According to the understanding which many writers have of this model, the information source creates a message with the intent to communicate it. Based on the message, the information source chooses a transmitter to code the message into a signal. The signal is then transmitted through a channel. On the receiving side, the signal is received, decoded by the receiver, and finally the reconstructed message is passed along to its destination. As the model implies, the primary explanation of why the received message might be different from the sent message is the introduction of noise into the communication channel by an external source.

The intent of this model was to provide a general diagram of communication which would help a telecommunication specialist focus on problems specific to his field, such as the questions of channel capacities, code design, encoding mechanisms, signal to noise ratios, and the optimum level of information redundancy. In the original article, the primary intent was to develop a theory of signal transmission, and not a comprehensive theory of human communication. For example, the communication model makes no distinction between the communication of a vital message compared to a trivial one. However, many other disciplines and authors have used this model extensively to the point where it is understood by many people as a law rather than a metaphor. Note that a database schema likewise cannot distinguish between something which is trivial versus something that is vital.

There are a number of serious issues which arise from the things said and unsaid in this early model. At the outset, there is a strong coding and decoding bias to the models, which has a stimulus response kind of thinking behind it. The following story from Condon, Jr. (1974) nicely demonstrates this code approach to communication.

> An American and a Frenchman had an opportunity to sit beside one another on a flight from New York to Paris. With the noon time meal, the Frenchman raised his glass and said, "Bon appetit." The American raised in glass in reply and said "Ginzberg."
>
> The same exchange took place at dinner, the Frenchman saluting with "Bon appetit" and the American responding "Ginzberg." However, a passing airline steward overheard the exchange and took the American aside to explain that "bon appetit" meant "have a good meal."
>
> As it turned out the Frenchman and the American were staying at the same hotel in Paris, and found themselves sharing a breakfast table. The American held up his glass and said "Bon appetit." The Frenchman raised his glass and replied "Ginzberg."

The anecdote is completely consistent with the code oriented, denotational stimulus-response characteristic implied by the Shannon and Weaver's model. It also clearly demonstrates the problems inherent in the code and decode construction, which is a common element of database design. As we have noted in the discussion on semantics, data modelling theory has its major focus on denotational meaning that shares the code and decode process of interpretation as its primary mode.

The second major limitation of the Shannon and Weaver model is that it carries with it a "meaning in the message" metaphor. That is, the signal has the meaning "packed into" it, and the task of the listener is to "unpack" the meaning. Expressed another way, the meaning gets coded into, carried by, and decoded out of the message. The negative effects of this implicit "words/sentences as containers" has been discussed by Lakoff and Johnson (1980). It is possible to develop communication models where the meaning is not carried by the message, but is instead initially created by the sender and then recreated by the receiver, based on the evidence of communication presented by the signal. The evidence of this thesis' data modelling research is that the code and decode is an inadequate explanation of data hermeneutics.

More recent thinking on the question of communication is described by the approach of Sperber and Wilson (1985), which they call ostensive - inferential communication. In their book, *Relevance - Communication and Cognition*, Sperber and Wilson begin by posing two questions:

- What is communication?
- How is communication achieved?

The authors strongly disagree with the code model as the only approach, although they do allow that there are strong code elements in some communication. Instead, they propose an alternative approach which is based on inference. How is coding and decoding different from inferential thinking? The code approach implies a mapping correspondence between a phonetic, semantic, and syntactic representation which closely associates a particular thought with a particular sound. Inferencing is a broader process which starts from a set of premises and ends in a set of conclusions which logically follow from or which are warranted by, the premises. Note that the idea of "logically follow from" (i.e. deductive) is a much more precise process from a classical logic perspective than the idea of "warranted by" (i.e. inductive).

One of the most important elements in the inferential model of communication is the concept of "context". Context for Sperber and Wilson includes the factors which were discussed in Figure 1 above (Brown and Fraser, 1979), as well as previous utterances, expectations on the part of both hearer and listener, anecdotal memory, belief about the speaker's state, and cultural assumptions. Thus, in their view, communication and cognition naturally use an heuristic approach rather than the algorithmic approach implied by code based models. This heuristic approach uses context, rules of relevance, and the intent to communicate, as they exist in both the sender and receiver. This is fundamentally different from the assumptions built into data modelling theory.

The heuristic inferencing approach may use more than one method for making meaning out of a communication. Some of the techniques suggested by Sperber and Wilson are: hypothesis formation and confirmation; pattern recognition; jumping to conclusions; subjective analogies; and reasoning without knowledge. Data modelling constructs have no way to represent any of these important elements of communication.

In the opinion of Sperber and Wilson, semantic representations must be inferentially enriched before they can be used for meaning making. This applies even to cases where one might expect that a code and decode interpretation of a communication would be possible. The term "ostensive" is used to describe communication acts in which the sender makes manifest the fact that he intends to communicate something. This distinction is necessary for their development of the concept of relevance and its impact on communication and cognition.

The context of the situation may seriously influence what is filtered into or out of the cognitive system. In whatever way this is achieved, the message must subsequently be used in inferring what the original meaning was. The receiver of the communication takes the message, his understanding of the context, his understanding of the intent of the communicator, and his conceptual / expressive repertoire, and infers what the original cognitive, connotative, affective, and thematic meaning was. In short, the meaning is re-created by the receiver, through a process of inferential enrichment. The point of this discussion is that human communication, whether face-to-face, by telephone, by written document, or by database storage and retrieval, is not simply a question of code - pack - transmit - unpack - decode. The context of the message is vital for its interpretation.

As it turns out, misinterpretations in human communication are extremely common. However, in natural language these misunderstandings usually become apparent quite quickly, and the meaning intention is negotiated naturally, frequently without the participants recognising that such meaning negotiation is taking place. My purpose in this discussion is to demonstrate the inadequacy of the code and decode model of communication. I suggest that these arguments about the inadequacies of the code and decode model of communication apply equally to the use of database schemas and their contents as communication vehicles. The problem with database schemas is that they inadequately preserve the context of the database. Most importantly, it is impossible to negotiate meaning with a database in the event of inevitable misunderstanding. A schema and DBMS cannot recognise in any meaningful way that you have misinterpreted or misunderstood the data.

Perhaps the expectations which the community has for the idea of "capturing meaning" in a database for subsequent communication is unrealistic. Raskin (1984, 63) offered the opinion that the usefulness of isolated meaning is highly dubious. He suggested that if the context is not given explicitly, the communication receiver is unlikely to comprehend the message at all or at least fully. Thus from a communication theory perspective, data modelling has a relatively naive understanding of how communication is achieved, and how communication is used linguistically. The primary weaknesses in modelling theories is a bias of the code and decode basis of communication and an understatement of the role of the user in interpreting models and their data.

C. DATA MODELLING - THEORY INTO PRACTICE

While we have looked at the foundations of data modelling theory, we must also consider the theory of data modelling as it is supposed to be applied. As we shall see, data modelling theorists make claims about the stability of data models which are based on recommended or proposed methods of creating a data model. Thus the objective of the following section of this chapter is to review the existing literature on how data modelling techniques are supposed to be applied. The data modelling literature in this area is also controversial along these dimensions.

- Are there solid modelling principles which are generally accepted?
- Is it practical to separate technical design from data analysis and planning?
- What level in an organisation structure is the right one to start for "top-down design"?
- How is the articulation of business policy done to support the development of data requirements?

In reviewing the literature on the practical application of data modelling, I am struck by the lack of a strong theoretical framework which might guide the data analyst in creating the best model possible. On the one hand, Date (1986, 487) made the observation that the relational approach introduces a modest theoretical foundation into database management, "a field sadly lacking in solid principles". We have seen that the modest theoretical basis of relational methods is the application of set theory and predicate logic primarily at the level of logical record design. Many other theoreticians have extended the relational theory by providing complex formulations designed to increase the logical rigour of data models and the subsequent data processing. Unfortunately most of these formalisms do not exist for the first phases of any data analysis - that of exploring the problem and specifying the user requirements.

On the other hand, Veryard (1984, 1) took the position that there are well known principles which have been well tested:

> ...data analysis is a branch of systems analysis and therefore shares its principles. Of particular relevance are the separation of analysis from design, the clear statement of objectives, assumptions and priorities, the systematic top-down and iterative approaches to analysis, and the unambiguous documentation of results.
>
> The data analyst often has to make arbitrary decisions on a criterion of elegance, or according to what the users are likely to understand and agree with.

There are a couple of points in this statement which bear some consideration. First the separation of analysis from design is a rather difficult objective to achieve since most analysts are acutely aware of the limitations of the DBMS which they will subsequently use. Therefore, the capabilities and limitations of the eventual technical design tool cannot help but influence the analysis to some degree. Date (1986, 474) said that the DBMS's marketed at that time severely constrained what the designer could do at the conceptual level.

The reference to a top-down and iterative approach is probably the influence of the structured techniques reviewed earlier in this thesis. Intuitively speaking, this idea of starting at the global and proceeding to the specific appears to be a reasonable approach. The relationship of the level at which the data analysis is undertaken to the subsequent data model and its stability would be an interesting research question. However, there is no advice or instruction in the literature about where in the organisational hierarchy the top-down design should begin. Other authors have also discussed the question of top-down analysis. Avison (1985, 60) noted the practical problem of achieving a complete top-down analysis. He suggested the completion of "local" entity analysis, that is, analysis done in the context of a particular functional area, such as marketing. This runs contrary to writers such as Martin (1983), who devoted a whole chapter in his book on the process of enterprise analysis, which is supposed to precede data analysis. He used a diagram (p. 109) which clearly demonstrated top-down analysis followed by bottom-up design. The global approach implied in this method posed a problem for Jones (1986, 67) who noted that "it is difficult to see where such a global view of the organisation might come from". In contrast, Flavin (1981, 2) supported the Martin view where he said that information modelling "is a 'top-down' procedure rather than a 'bottom-up' one".

Veryard's comment about the iterative nature of solving the problem would seem reasonable to all analysts who have had to return to the user community again and again during their problem analysis. The final point which Veryard made in the above quoted passage involves arbitrary decisions on a criterion of elegance, user understandability, and user agreement. This statement seems to be contrary to prospects for developing provable formalisms which will be usable for the whole, or even most, of the modelling cycle.

Wherever one begins the modelling process, whether at the enterprise level or at the application level, many authors make specific reference to the need for business policy analysis to form the justification or rationale for the model which will be subsequently developed. The business analysis which necessarily parallels the data analysis was seen by Avison (1985, 54) as "a two-way exercise -- an opportunity to inform, help and convince, as much an opportunity to find out about organisation." This is clearly not just a matter of dispassionately documenting an immutable objective reality.

As far as Flavin (1981, 11) was concerned, the mathematical techniques which might be used later in the analysis process cannot substitute for business policy analysis. He made the forceful statement:

> Business policy is the bedrock against which logical databases are validated. A logical database design procedure *must incorporate* a means to validate the logical database design against stated, and verified, business policy. [emphasis in the original].

On one hand, as Martin (1981, 34) noted, the literature on business planning and business policy is vast, and that such policy planning is an unavoidable part of the data planning process. On the other hand, almost no reference is made at all to the question of policy and its influence on data models in Everest (1986) in over 700 pages of text on database administration. The role of policy articulation in data planning may have an impact on model stability.

If we are going to consider the question of measuring data model stability over time, we will first have to come to some operational definition of what a data model actually is. The literature often talks about three different levels of a given data model: the physical data model implementation, the logical data model (LDM), and the conceptual data model (CDM).

While there is little debate about what physical database model means, there is less agreement about what an LDM is, and even less agreement about what a CDM is. Some consideration of these concepts is necessary. Physical database design is the process of choosing the optimum degree of data redundancy, the best data compression, data set generation management, and storage allocation of the available physical resources. Technical systems staff usually undertake this design activity relatively late in the system development life cycle, when most of the functional systems issues have been resolved.

The CDM attempts to recognise the essential information elements of the enterprise which should or could be managed by information technology. The CDM seeks to document the entities, the characteristics of the entities, and the various relationships among the entities, without concern for technological considerations. Graphically, this is sometimes represented as in Figure 3. (See also Biller and Neuhold, 1977, 4-5; Biller and Neuhold, 1978, 12; Deen, 1985, 58).

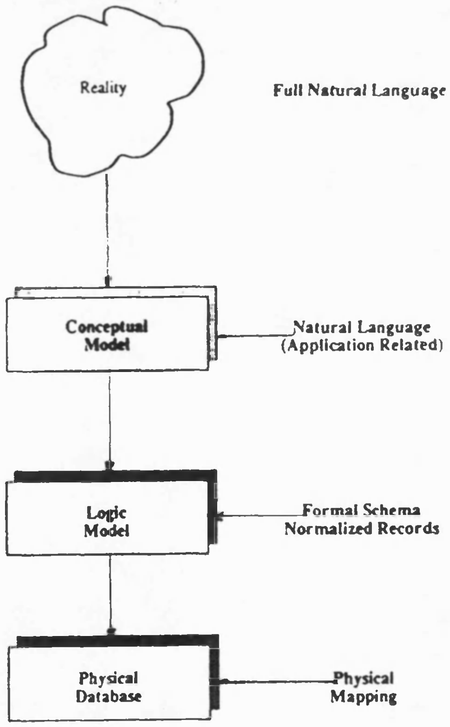


Figure 3.

However, the way that such models are to be constructed is not at all consistent throughout the literature. We would probably all agree that the user community would be heavily involved in helping to create the CDM. But from a research perspective, what would have to be in place before we could confidently say that a CDM had been created? Would a list of major system entities, and a Bachman (1969) diagram be sufficient? Would they be necessary?

According to Ruchti (1976, 122), a conceptual schema writer must meet two requirements. First is the test of empirical truth where the reader of the description of a conceptual schema should arrive at the same empirical interpretation as was intended by the author of the schema. Second is "semantic truth" where the conceptual schema conforms to semantic rules defining data structures to the DBMS. The first point, empirical truth, does not help us discriminate between conceptual and logical data models. Presumably, the test of empirical truth should apply to describe both levels of modelling. The second point, semantic truth, is a problem in that conceptual data modelling is supposed to be technology independent. The test of semantic truth is also not going to help us discriminate between the conceptual and logical data model. Note that Ruchti uses the word "semantic" in a completely different way than a linguist would.

Antoni Olivé (1985, 397) implied that the conceptual model is a necessary pre-cursor to the logical data model. He took the position that the CDM constituted the specification from which the remaining models were generated. He stated that it was fundamentally impossible to verify formally the correctness of the conceptual framework. If the impossibility of formal verification is correct, this suggests to me that we should lower our expectations about the stability of data models. Olivé's discussion about data modelling also does not provide any help in answering the question of how to discriminate between a conceptual data model and a logical one. Some authors have high expectations of such a specification, such as Mason (1985, 81) who wrote that the external presentation of the model must be in a form which is "capable of complete and unambiguous interpretation by the user" [emphasis in the original].

A. Solvberg, and C.H. Kung (1985, 205) were a little more helpful in their description of the conceptual data model when they stated that a "conceptual model consists of both structural and behavioural aspects of the application" [emphasis in the original]. According to these authors, a CDM would be much more extensive than many data model representations since the processing aspects of the system would also have to be represented somehow.

Everest (1986) did not use the term "conceptual data model" but discusses instead the process of "designing the 'natural' data structures". His description and example of the schema which comes from this natural data structure included a preliminary analysis of the entities, attributes and relationships of the application area. The schema presented documented a number of other aspects of the model such as whether the relationships are optional, whether they are unary or n-ary, whether they are reflexive or other forms of relationship. The prospective key to the entity was also documented. This data structure was then analysed in the logical data modelling process in order to normalise the structures. Thus the difference between CDM's and LDM's for Everest is the normalisation of the data structures.

Avison (1985) took the approach where the CDM requires more than a conceptual schema as described by Everest. Avison said that the conceptual schema must also document the entities, attributes, relationships and events to the extent where the size of the various data elements are chosen, and the data dictionary names have been assigned. This distinction is primarily an administrative one, but adds the benefit of identifying some of the potential synonyms, or inconsistent language usage in the problem domain. Brachman (1977, 139) wrote more extensively about the need to make use of the relationships between all of the entities in a semantic network in order to make most intelligent use of the knowledge represented in the network.

Ross (1986, 10) talked about beginning the data analysis with entity types which are named "using a noun in the singular form that clearly identifies the object". Softech (1981, 22) wrote that data involves "information, objects, or anything that can be described with a noun phrase." This search for the noun is an understandable process given the usual definition of entity as some*thing* about which an organisation wants to store data. Many authors have noted that there is no agreed upon way to classify any observation according to entity, attribute, relationship (Kent, 1978; Curtice and Jones, 1982; Date, 1986). The most common example which demonstrates this classification problem is the notion of marriage, which can be legitimately represented as an entity (for example in the case of government departments responsible for vital events registry), as an attribute, or as a relationship. This classification question does not begin to take into account changing social mores which are beginning to suggest acceptance of homosexual marriages. The notion of marriage as modelled in most designs does not address the question of circumstances where polygamy or polyandry might be permissible, such as is occasionally the case among North American Mormons.

How important is diagramming to the data model? There is some difference of opinion about whether diagrams should be used or not. Modelling is partly about the management of complexity, and authors such as Ross (1986) certainly depended heavily on the use of diagramming conventions. By contrast, Veryard (1984) expressed serious reservations about the use of diagrams when it comes to explaining logical data dependencies:

> Attempts have been made to develop diagrammatic conventions to show logical dependencies between relationships, e.g. by means of arcs and dotted lines. Experience shows that such conventions do not work. They are difficult to remember, and can only be applied to the simplest cases. There is usually no substitute for describing the dependency in precise, logical sentences.

Martin and McClure (1985, 1) claimed that "good, clear diagrams play an essential part in designing complex systems..." Perhaps the middle ground to this debate is that effective diagramming is essential but not sufficient for data modelling specifically and systems analysis generally. There have been other criticisms of diagramming, such as Floyd (1986, 26) who wrote that developing a logical model out of data flow diagrams was judged to be "entirely unreasonable," primarily in cases where the system is a large one.

What is not clear from the literature is how a given diagram is presented to an end user once an application is implemented. If the diagrams of a data model are important during the analysis phase for communication of the model structure and intent, presumably they would be important to users during the operational phase of the system, especially for anyone who had not participated in the original design. Are these diagrams maintained and kept current? Do end users consult the various diagrams which may have made a significant contribution to the development of a data model when they are trying to interpret the data which is being presented through the vehicle of the data model?

Notwithstanding the differences in the literature, generally authors agreed that the conceptual data model seeks to document the entities, the characteristics of the entities, and the various relationships among the entities, without concern for the technology. Logical database design or logical data modelling is the process of mapping or fitting the natural data structures which come out of the conceptual model into structures which can take advantage of the data structures and processing in the available DBMS. Theoretically, this is the first stage in which the technology is given any consideration.

Everest (1986, 143-44) underscored this concern about the relationship between the logical data model and the database definition language when he said "the logical database definition language should make it easier to formulate, comprehend, and change database models of the real world." As the above Figure 3 indicates, each of these steps has a narrower constraint to the languages available within the particular modelling step. When we deal with business processing problems, we use the full natural language in whichever native tongue we care to pick to deal with the situation, whether we see it as an objectively or subjectively created reality. Within the bounds of a particular conceptual model however, we continue to use natural language but we use a subset of natural language, which is application specific.

For the most part the terms and vocabulary of a conceptual model are constrained by the application related understanding of both the data modeller and the user community to whom he relates. In the logical data modelling step, this vocabulary and language is further constrained through a process of creating a formal data schema, usually involving normalised records. These records may be normalised to varying degrees, but the process of normalisation further constrains the way in which the vocabulary may be used. Finally in the physical database process, the formality of the model is increased to include a physical mapping which further constrains the access processing and management of the subsequent data within the logical structures documented in the logical data model.

This general process is one of describing or creating an appropriate "reality", and through an increasingly formal process, preparing a physical design. The objective of the process is to determine how the understanding of the users in a given application area can be represented to increase the likelihood of a subsequent user of the data understanding what was intended. Mijares and Peebles (1976, 27) noted that "in order to describe a semantic view the user of the relational model must carry the semantics of the relations in his head." It is not just the semantics of the relations that the user carries around in his head, but also the semantics of the entities, attributes and relationships, too. Senko (1975, 6) confirmed this opinion when he wrote "...in existing systems technology, the burden for associating meaning with representation is placed almost entirely on the user."

It is important at this point that we distinguish between the concepts of *relation* as in "association of attributes within a given entity" and the notion of *relationship* which involves a stated or unstated association between different entities. The Mijares and Peebles quote referred to the specific tuples which have been designed in the structure of the relational design. Their particular point about the semantics or the meanings of the connection of the relation attributes being part of the tacit understanding of the environment is frequently understated in data modelling literature. This of course poses a specific problem where databases are shared across large communities within a user company, where the distant users of a particular database may not have the same clarity of understanding about the semantics of the relations in a given tuple.

Looking back on the question of how the theory gets turned into practice, we see that there many serious practical concerns about how to put into effect the recommendations of the literature. Many of the "principles" of design are under debate by the various modelling theoreticians. There is serious doubt about whether it is practical to separate the technical considerations of the application DBMS from questions of data structure design. There is no simple and direct answer to the question of what level to begin the data analysis process. While virtually all writers agreed on the need for effective articulation of business policy, there is little in the way of practical advice about how data modelling ought to assist the process. Finally, there is an apparent contradiction between the two competing objectives for a data model - that of stability versus that of evolvability. Whether an author's view of reality was objective or subjective, he agreed that realities change over time, and that data models must evolve to reflect those changes.

In summary to Chapter 2, we can say that data modelling theory in both its foundations and its application is missing major elements. It is not based on any firm and coherent theory of meaning. Its philosophical foundations are highly controversial. Data models cannot be effective tools of communication on the basis of the models alone because they do not satisfy the fundamental principles of communication theory.

The specific directions of the literature for building a data model are contradictory, unclear, unspecific, and incomplete. One of the major areas of disagreement among authors is where certain levels of detail in the analysis belong: in the conceptual data model or in the logical data model. There is also no generally agreed upon documentation standard at any of the levels of analysis which are described. If a model is to be developed from the top down, where is the top? What diagramming conventions or rules are necessary at what level of detail? How do we find an entity and classify it? If business policy is the foundation to a data model, how does this policy get articulated? Is it always consistent from one place in the organisation to another, and should it necessarily be consistent? The data modelling literature has no answers for these questions.

We can conclude from this review of the literature that data modelling is probably not consistently practised, and that styles and approaches will likely vary in the community. The actual practice of modelling in the community might give us some insight into what we might expect along the stability dimension. In order to get this preliminary view, a survey of the data modelling practices of organisations in Canada was conducted early in 1988. The next chapter of the thesis will review the structure of this survey on data modelling practices, and the results which it generated.

CHAPTER 3
A SURVEY OF DATA MODELLING PRACTICES IN CANADA

The review of the literature on data modelling presented above yielded contradictory results at the level of the fundamental theoretical foundations. The literature is likewise inconclusive about how the tools and techniques of data modelling should be applied in practice. Since the literature presented such a confusing picture of data modelling theory, it was decided to determine what the actual practice of modelling in industry was to provide data to help in the preparation of a method to measure stability.

There were a number of reasons for conducting such a survey. First, I wanted to determine what the actual modelling practices in the information systems community were, in order to identify any obstacles to designing the stability measurement tool. Second, comparing the practice to the theory should provide some data to develop a sense of the likely prospects for data model stability. Third, I wanted to identify any existing examples of stability monitoring processes being applied in a commercial setting. Finally, the survey might identify potential volunteers for the next phase of the research.

The data modelling literature identified a number of factors which various authors claim are significant in the modelling process.

- At what organisational level was the modelling was undertaken?
- To what degree does formal business planning support formal data planning?
- What role does the data administration function play in systems planning?

- What data modelling tools and techniques are actually in use?
- To what extent are automated modelling tools used?
- How formal are the documentation standards applied to data elements?

This survey was also interested in questions such as whether differences in data definition were a significant issue in the modelling process and whether commercial sites monitored the stability of their data models over time.

The survey of data modelling practice was designed and tested in Great Britain in 1987. The British Computer Society database special interest group provided a list of 22 professionals who had an interest in data modelling. The list was comprised of 12 data administrators and 10 academics or consultants. The test questionnaire was sent to each of them. Of the data administrators, 6 out of 12 (50%) replied, and of the academics / consultants, 3 out of 10 (30%) replied. The results of the survey were somewhat problematic since counting complications inevitably arose. For example, one consultant responded by answering the question on behalf of three of his clients, while another consultant answered on behalf of his firm. Some returns were not usable because of incomplete responses or misunderstandings about the question intent. The incomplete responses and misunderstandings were used to modify the questionnaire.

The results from the British test survey were interesting from a number of perspectives. Data modelling in its various forms and techniques is commonplace in systems analysis. Of the British respondents, 100 % reported that it was used at the application level, 83% at the divisional level, and 60% at the organisational level. While data modelling was reported 100% of the time, formal business planning occurs only 57%. Stability monitoring was reported in 12.5% of the cases, and all of the respondents reported difficulties in developing and rationalising data definitions across organisational boundaries.

The average time to complete this survey was reported to be 12 minutes. This test survey community and other systems professionals provided suggestions about modifying the survey instruments. Because of the short amount of time necessary to complete the survey, further questions were added in the area of specific products for data modelling management as well as the DBMS's in use. The survey also took the opportunity to solicit other related data of interest such as the size of the data processing organisation as a percent of all personnel resources, and the size of the data administration function within the systems group. A copy of the final survey is attached as Appendix I. In order to encourage maximum response from the community, the final survey was designed to take less than 20 minutes to complete.

The survey was sent as a data gathering tool to provide an indication about the data modelling practices and what impact these practices might have on the narrower question of data model stability. A covering letter explaining the purpose of the survey was prepared and translated into French for those organisations where it was likely the French would be the preferred language of choice. The survey was sent to approximately 700 organisations, comprising the 500 largest companies in Canada, and major government agencies selected from the Canadian Almanac. The mailings were sent to a named systems professional in each of these organisations where such individual could be uniquely identified. Included in each survey envelope was a postage pre-paid return envelope for the convenience of the respondent.

Over 30 of the surveys were returned with addressee unknown. Of the remaining, 73 surveys were returned and 69 of these were usable. There are a number of reasons for the relatively low return rate. First, there was no second mailing. Second, there may be low interest in the topic area, or the user may have little exposure to and/or interest in the methods of data modelling. Finally, there may have been a problem with survey fatigue on the part of the survey population. One return letter documented exactly this situation. This subset of the surveyed population is somewhat larger than the survey group which comprised the respondents for Kahn's (1983) report on data administration in the United States.

The number of employees in the organisations of the respondent group varied from 75 to 75,000 with an average of 6,860. The number of full time EDP staff varied from 1 to 2,000, so a wide range of organisations was represented in the data. EDP staff averaged 2.4% of the total employee resources in the organisations. In one case the EDP professionals represented over 90% of the staff since the organisation was a software development house. The average number of database analysts was 4, or 2.7% of the data processing staff. The median percent of EDP staff which were database administrators or equivalent was 2.9%.

Of the companies which provided details of the database products in use, over 35% reported that they used more than one product. The typical case was a company which preferred a particular product for their own development purposes, but had to buy a separate one in order to take advantage of off-the-shelf packaged software particularly well-suited to their business. While almost 2/3 of the respondents report using DBMS's for packaged software and software currently under development, only 38% use such products exclusively. On average the user population had over 7 years of experience with DBMS's.

Data modelling is done 77% of the time for individual applications. It is applied at the department or divisional level (e.g. to integrate data views) 48% of the time, and is used at the fully integrated level across the whole organisation in 23% of instances. This is significantly lower modelling rate than in the questionnaire test group in Britain, probably because the test group in England had a stronger interest and commitment to the techniques given that they were members of the data administration special interest group of the British Computer Society (see Table 1). The Canadian response group would likely represent a much broader cross section of the community. One of the most important commonalities between these two groups is that the frequency of model integration drops in the organisation as the level of abstraction and integration rises.

Table 1.

ORGANISATIONAL LEVEL WHERE MODELLING IS APPLIED

LEVEL AT WHICH MODELLING WAS FOCUSED	BCS SPECIALIST GROUP	CANADIAN MODELLERS
Individual Application	100%	77%
Department or Division	83%	48%
Fully Integrated	60%	23%

While some modelling theories (e.g. Martin, 1983; Flavin, 1981) emphasise that strategic business planning is a necessary preliminary to data modelling, only 57% of Canadian organisations report having conducted such work. This was exactly the same response as the British test group. Completing a strategic business plan has been a relatively recent phenomenon with 1986 reported as the first year that this was conducted on average, notwithstanding the reports that DBMS's have been in use since 1979-80 on average. In other words, the typical EDP organisation used DBMS's capable of integrating organisational data for 6-7 years without the benefit of a strategic business plan. Enterprise modelling as part of the data modelling process was reported 34% of the time.

The survey asked which groups in the organisation were involved in formal periodic planning for information technology. According the survey results users were involved 50% of the time, EDP/MIS management were involved 77%, and data administrators were involved 43%. This implies that users are not involved in technology planning 50% of the time.

Table 2 below reports the incidence of the various modelling techniques reported to be in use. The percentages do not add to 100% since multiple techniques were reported in use at many sites.

Table 2.

INCIDENCE OF THE APPLICATION OF DATA MODELLING TECHNIQUES

MODELLING TOOL	PERCENT REPORTING
Data flow diagram	66.2%
Entity relationship	61.8%
Entity-attribute-relationship	50.0%
Data analysis	38.2%
Data navigation	22.1%
Action diagrams	20.6%
Decision tree	16.2%
Other techniques	7.5%
Warnier-Orr	5.9%
HIPO	2.9%
Jackson Design	2.9%
HOS	0.0%
Nassi-Schneiderman	0.0%

The most common combination of techniques reported were data flow diagrams with some form of entity relation modelling. Of the techniques classified as "other", Bachman (1969) diagrams were mentioned most frequently. Chen (1976) claimed that the one tool which unifies the various popular data modelling approaches is the entity relation discipline. Whether this assertion can be supported or not, one form or another of the entity-relation concept is used by most data modellers today. This is probably not surprising since Kerschberg, Klug, and Tsichritzis (1973, 61) reported that of the 15 data models on which they concentrated their discussion, 12 of them can be classified as being entity-relation oriented.

For an activity that is generally understood to be labour-intensive and complex, I was surprised that less than 40% of survey participants reported using an automated modelling aid. Of those who use automated aids, over 20% use more than one product, the most common of which were IEW and EXCELERATOR. A number of organisations reported that they planned to develop their own product. I infer from this result that it must be difficult for most data modellers to change their data models easily during the data analysis phase. This might suggest that the final quality of many data models could be improved, and that they would therefore be subject to a higher degree of pressure for change over time. It also raises the question about the maintenance of models over time. Less than 20% of participants reported monitoring the stability of their data models. Almost 62% reported having difficulty resolving data definitions. This was especially the case where models were expected to span major organisational boundaries.

The literature clearly indicates that data modelling is a labour intensive activity. Since data modelling can be only part of the function of data administration, these organisations cannot deliver a significant amount of effort through the limited centralised resources that they report. Given the relatively low level of investment in specialised data administration resources in the data processing organisation at less than 3% of the data processing manpower available on average, I conclude that where data modelling techniques are applied in an organisation, they must be applied by the application analyst himself, or through the use of external consulting resources.

In those cases where a centrally controlled application of a given methodology cannot be delivered, at least the data administrator can set standards for how a methodology is to be applied. They can also set documentation standards for data modelling output. However, the survey results indicated that fewer than half of the reporting groups had documentation standards that were enforced, either formally or informally.

The survey participants reported that the average length of experience with DBMS's was over 7 years, with more than one product in use. It is clear from the reports on the products in use that there is no single answer to the question of which database is the best, even within a particular organisation. Multiple database management products must pose something of a special challenge for those organisations which are attempting data integration across group or application boundaries.

Notwithstanding the literature which recommends that data modelling should be a top-down process beginning with the enterprise model, most of the modelling is done at the application level and not across the organisation as a whole. Data modelling could not have been supported in many cases by strategic business planning, because 43% of the surveyed organisations do not have this form of planning as part of their on-going business discipline. Even where the groups undertook formal information technology planning, the data administration function was involved only 41% of the time.

The practice of data modelling may vary by organization size. The results from the survey and the analysis of the data was classified according to two groups, "smaller" organisations which were below the median size of 2 100 employees, and "larger" organisations which were above that median. Using the arbitrary threshold of the median as a way to discriminate larger from smaller, the average number of employees in the larger organisations was 12 771 compared to 948 for the smaller. The smaller organisations allocated proportionately more resources to data processing staff generally (3.6% compared to 2.5%) and to data administration staff as a percent of the EDP staff particularly (6.8% compared to 1.9%).

Larger organisations tended to have more experience with DBMS's (8.0 years) than smaller organisations (6.4 years). Larger organisations reported a much higher incidence of business planning, at 67.6% compared to only 42.9% of the smaller organisations. In terms of information technology planning, the picture was somewhat mixed, as Table 3 below shows.

Table 3.

COMPARISON OF PARTICIPATION IN INFORMATION TECHNOLOGY PLANNING BY SIZE OF ORGANISATION

FORMAL INFORMATION TECHNOLOGY PLANNING	SMALLER ORGS	LARGER ORGS	ALL GROUPS
User management participation	48.5%	54.5%	51.5%
EDP/MIS participation	70.6%	84.8%	77.6%
Data administration participation	35.3%	47.1%	41.2%

Larger organisations tended to report a higher incidence of formal information technology planning than smaller organisations, and this finding parallels the reported incidence of overall business planning which is lower among the smaller organisations. According to the literature, this business planning is part of the foundation upon which the subsequent data modelling techniques are based.

Table 4.

INCIDENCE OF THE APPLICATION OF DATA MODELLING TECHNIQUES - ACCORDING TO SIZE OF ORGANIZATION

MODELLING TOOL	SMALLER ORGS.	LARGER ORGS.
Data flow diagram	60.0%	72.7%
Entity relationship	54.3%	69.7%
Entity attribute relation	42.9%	57.6%
Data analysis	34.3%	42.4%
Data navigation	20.0%	24.2%
Action diagrams	20.0%	21.2%
Decision tree	20.0%	12.1%
Other techniques	5.7%	9.4%
Warnier-Orr	5.7%	6.1%
HIPO	2.9%	3.0%
Jackson Design	2.9%	3.0%
HOS	0.0%	0.0%
Nassi-Schneiderman	0.0%	0.0%

Table 4 above compares the reported incidence of use for each of the data modelling techniques reported in Table 2, according to the size of the organisation.

Of the respondents, 18% reported that they monitored the stability of their data models. The brief description provided by the respondents indicated a wide variety of ways that this monitoring was undertaken, from a general reconciliation of the data models during the annual system planning cycle, to detailed comparisons of individual application models against the corporate data model. Tracking changes at the level of the physical implementation was reported as a stability tracking mechanism. Problems in resolving data definitions were quite common, at 64% overall. There was a major difference between the smaller firms and the larger ones on this dimension. Smaller firms had a frequency of problems in reconciling definitions in 50% of the questionnaires. In the larger firms, this figure rose to 78%. The respondents' narrative descriptions of the difficulties underscored the percentage results. Larger firms reported serious problems more frequently than smaller firms.

Some of those respondents who reported serious differences in resolving data definitions among or between applications provided a brief description of the their problems in this area. Some of the typical problems described were:

> "...getting agreement from user that two similar elements mean the same thing e.g. customer and client."
>
> "...multi data names used for the same data items. Inconsistent definition of content and meaning. Very unproductive to track data usage, meaning, etc."
>
> "Difficulties in integrating new applications / modules in our existing environment; we have had to build 'bridges' and 'interfaces' and 'temporary' entities and data bases to cater for this situation."
>
> "The conflict occurs between users of different business areas and is always related to different interpretations of the same information."

In summary, data processing professionals reported practices which are at serious variance from the documented theories of data modelling. Practitioners had to contend with the constraints of varying DBMS constructs, varying data analysis techniques, often without automated help for the complex task. Their task of integrating data models was complicated by the fact that the implementation DBMS has an influence on the design of the data model, and in the larger organisations, the data administrators often must use more than one DBMS product.

Frequently the business planning which should underpin the systems development has not been done, and in many cases, even technology planning is not routinely done. The organisations where they work present them with problems in resolving data definitions, especially where models cross group boundaries. Fewer than one in five organisations monitor the stability of their data models, and those that do, monitor it relatively superficially through changes in the physical implementation or by comparison to previous overall business models. On the basis of this evidence alone, we should have limited expectations about the stability of data models over time.

The survey results confirmed the use of entity relational modelling supported by data flow diagrams is the most common modelling approach. The respondents also confirmed that there were no data model stability measuring techniques in use. Any stability monitoring that is done tended to be done at the physical implementation level. Most respondents expressed an interest in a method to review model stability. We therefore concluded that developing a careful tool to examine the question of data model stability at the logical level was worth pursuing.

The answers to the survey questionnaire provided some insight into the widely varying modelling techniques in use by the commercial data processing community, and also suggested a technique for resolving the problem through the use of the relational practices. The subsequent measurement tool used the results of this survey as a foundation for the specific proposed process.

PART II
DEVELOPING AND APPLYING A TOOL FOR STABILITY MEASUREMENT

CHAPTER 4

DEVELOPING THE STABILITY MEASUREMENT TOOL

The overall objective of this research was to develop and apply a measurement tool to track the stability of data models. In order to achieve this objective, we must be as clear as possible about three basic notions:

- What do we mean by "measurement"?
- What is "stability" in the context of data modelling?
- What is a data model, and what elements of it are most important, especially from the perspective of stability measurement?

Once we have addressed each of these questions, we can consider the question of how to collect data to measure data model stability.

A. WHAT DO WE MEAN BY MEASUREMENT?

Measurement is an essential element of information systems development. Mason and Swanson (1982, 29) lamented that management information systems commonly use measurements which were "crudely done, reflecting a lack of sophistication...". Churchman and Ratoosh (1959, 84) proposed that the purpose of measurement is to develop a method for generating a class of numerical data that will be useful in a wide variety of problems and situations.

The idea of measurement is so common in our individual, collective, and institutional lives, it is mostly taken for granted. There is very little in the systems literature that has much to say about it. In the literature on social research techniques, however, there has been some thinking about what constitutes effective measurement. Ellis (1966, 41) was quite specific about a set of conditions that must be satisfied in creating a scale of measurement:

> (a) we have a rule for making numerical assignments;
> (b) this rule is *determinative* in the sense that, provided sufficient care is exercised the same numerals ... would always be assigned to the same things under the same conditions; [italics in the original]

These conditions are similar to the operational definition quoted by Churchman and Ratoosh (1959): "one cannot know what 'length' means until one knows the operations that were performed in order to obtain the figure given as the length of an object." Therefore, the objective of Part II of this thesis must be in part to document the approach used and the operations performed in the application of the stability measurement tool.

The literature generally agrees on a number of criteria for good measurement. Osgood, Suci, and Tannenbaum (1957, 11) summarised the usual criteria for measuring instruments, as follows:

- Objectivity - the results of a given measure are independent of the person who applies the tool.
- Reliability - the tool yields the same results, within acceptable variance, when applied under the same circumstance.
- Validity - the results should be consistent with any other acceptable "independent index of meaning" .
- Sensitivity - the results as measured are graduated differentially to the same degree to which the measured object varies.
- Utility - the results can be obtained in a relatively cost-beneficial way.

Osgood, Suci, and Tannenbaum recognised that this list is not complete, but suggested that it is normally a sufficient set of criteria. Ghiselli, Campbell, and Zadeck went further than this in presenting a more extensive discussion on the single issue of validity. Their definition of content validity, for example, made no reference to any other independent index of meaning. Instead the authors described content validity as the answer to the question "to what extent do the operations measure what they are supposed to measure?" Ellis (1966) noted that measurement is primarily relative and not absolute. This notion applies to the stability measurement which is developed in this thesis, where models can be declared to be stable or unstable only in comparison with other models.

B. THE CHARACTERISTIC OF STABILITY IN A DATA MODEL

Information technologists do feel the pressure of having to make rapid adaptations in both application and technology related arenas. Therefore, it is not surprising when the data modelling literature refers to "stability" as an important objective and benefit of the modelling techniques and methodologies. This section of the thesis will:

- review what the literature has to say about stability;
- briefly review the potential consequences of model instability, and;
- provide an overview for the specific stability measurement objectives of the research.

1. The Literature is Not Specific in its References to Stability

The data modelling literature is not specific in its use of the term "stability". Many authors simply state that data models are by nature "stable". For example, Veryard (1984, 2) wrote that the data structure of users' requirements tend to be much more stable than their functional requirements. According to him, the stability of data structures would mean that a system design based on "proper analysis" of the data structure would be less likely to need major modification when the requirements change, than one based on "current operations and functions".

One of the problems with these assertions is that most data modelling methodologies require that the data analyst create the data models in conjunction with current process models. In other words, we come to an understanding of entities by understanding the affordance of the entity, that is, what the notion permits us to do. If the data model must be validated against the process model or from a "functional viewpoint" to use Shave's (1981, 42) expression, one can only assume that the data structures would have to be re-validated whenever the processing changes.

There is another example of the claim for stability in Bravoco and Yadav (1985, 71):

> ...it [the information model] represents a stable information structure, a stable set of rules and definitions upon which a viable database design can be constructed, and based upon which, rationality and consistency are injected into the arena of integrated systems definition.

In this article, there is no further discussion about what might be meant by "stable", and how such rationality and consistency are "injected". The results of the data modelling survey described in Chapter 3 of this thesis make it clear that integrating systems definitions creates conflict in developing data models. The notion of consistency of an integrated systems definition is also interesting. It might be possible that different parts of a large organisation might have different and inconsistent data definitions, where such inconsistencies are functionally valuable and perhaps even necessary.

Martin (1985, 159-60) was insistent and even repetitious on the topic of data stability. He claimed that the types of data used in an enterprise do not change much, that the basic entities in an enterprise remain the same unless the nature of the enterprise changes radically. He went so far as to say that a well constructed data model created 20 years ago would still be valid today except for minor changes. He contrasted the databases that are specifically designed to be stable with traditional data processing files. Martin claimed that data models can be specifically designed to be stable.

He then presented an extensive discussion on data planning. However, the survey reported in Chapter 3 on data modelling as it is actually practised shows that most data modellers fall far short of the standards he set out. If Martin is correct in his thesis, we should expect to find relatively higher degrees of instability in data models which do not use the techniques and methods he described.

When Flavin (1981, 9) mentioned stability, he talked about it in the context of bringing together the different views of varying applications into one data model:

> How does one unify all the data describing a bank customer, but still provide modularity in the design that guarantees the stability of logical data structures? This very real problem for data base designers is caused by a lack of precision and agreement upon what is meant by the term "business entity." This problem of customer representation can be solved with a set of rules and definitions that remain consistent and used universally applicable across all such problem domains.

The implication here is that the stability in the data structures is partly a function of the modularity of the design. Modularity appears to be achieved in this context through the classification of "business entities". How effectively Flavin achieved this through the methodology he proposed for the process is a matter of substantial debate not directly related to the specific question of measuring stability.

Deen (1985, 85) proposed a narrow definition for stability which revolved around the need for "recompilation in the event of changes in the other views". This definition of stability was applied to both application programs and external schemas. Prakash (1984, 41-42) was a little more specific about what might be meant by stability:

> ...the representation of a conceptual schema should be stable. By stability we mean two things. First that when, eventually, the conceptual schema is represented in the computer, it should be immune to changes in the physical organization of the data in the computer.

> The second implication of stability is that the representation of the conceptual schema should be immune to changes in the view that a user has. This means that, in the example considered above, if the scientist who is interested in reptiles at some time *t* finds that, at a later instant he has narrowed down his field and is no more interested in the lizards then he should be able to change his view without causing any changes in the representation of the conceptual schema itself.

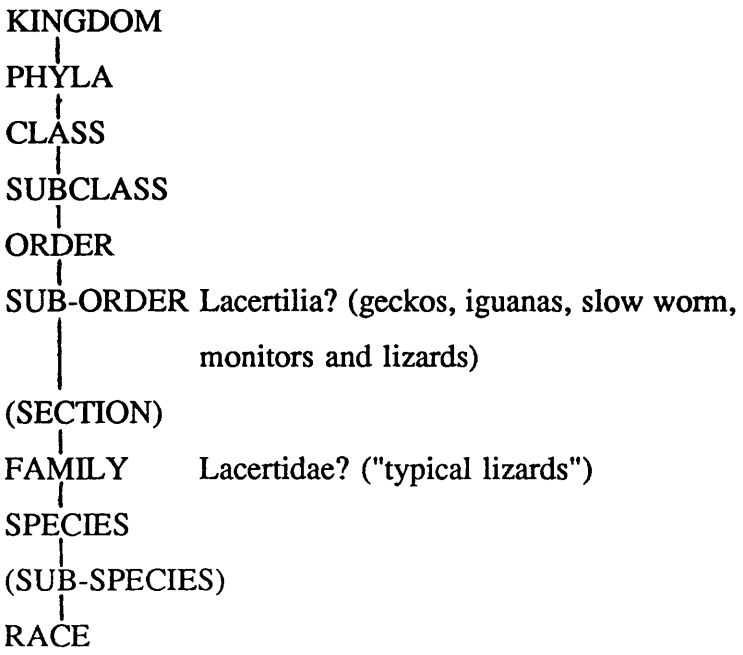
The first paragraph of this excerpt is generally accepted. In the review of the literature, everyone agreed with the idea that there is practical and theoretical benefit in the complete independence between the physical representation of data and the conceptual schema. However, the second paragraph is rather more provocative both in the principle and in the example which is provided.

I believe that the reason Prakash takes this position on changes in the user view is that most of the modelling literature presents the notion of user view strictly from the perspective of the various attributes and relationships which are set out in the data model. In other words, the system accommodates the need for differing user views by selecting or suppressing different entities and attributes for different users. The notion of "user view" assumes complete consistency in such matters as the inclusion criteria and the level of abstraction and generalisation which applies to the entity. Biller and Neuhold (1978, 12) also assumed that all users had conceptually consistent views of the shared database. De, Sen and Gudes (1982, 2) wrote that such consistent view entities could be called "clean" ones, in that they have distinct meanings "perfectly understood by the conceptual designer." There was no discussion of "dirty" objects, and how these might be treated.

In the specific example provided by Prakash, the user would only be able to change his "view" of the data model to exclude lizards if the original data model included an attribute which could distinguish between lizard and non-lizard. As it turns out, the biological category "reptile" is a "class" in the following abstraction hierarchy.

Figure 4.

BIOLOGICAL CLASSIFICATION HIERARCHY



KINGDOM

PHYLA

CLASS

SUBCLASS

ORDER

SUB-ORDER Lacertilia? (geckos, iguanas, slow worm, monitors and lizards)

(SECTION)

FAMILY Lacertidae? ("typical lizards")

SPECIES

(SUB-SPECIES)

RACE

According to the *Larousse Encyclopedia of Animal Life* (1967), the category "lizard" might mean might mean the suborder Lacertilia, which includes geckos, iguanas, chameleons, slow-worms and monitors or alternatively might mean "typical lizards" of the family Lacertidae. I appreciate that Prakash did not intend to present a biologically precise example, but this problem of the relationship between the abstraction hierarchy of the model and its stability is easy to find. The question remains: what processes do we undertake to optimise the probability that the stability Prakash wrote about will be present in the final data model?

Feldman and Miller (1986, 354) touched on one of the fundamental elements of data models and their stability - that of meaning. They claimed that "the basic meaning of the major entity type remains static throughout the model; it can just be interpreted in different ways in different contexts." This statement presents certain problems. I find it difficult to understand how the meaning of something can remain static when users interpret it in different ways and in different contexts. Major parts of the literature on semantics, as well as selected areas of philosophy, show how context and interpretation change meaning, instead of keeping it static.

Other authors have had concerns about the stability of meaning within a given data dictionary. Symons and Tijsma (1982, 411) wrote that definitions for the same concept changes over time, and between authors, to the extent that the data dictionaries themselves contribute to the obscurity of meaning.

As a final example of authors who make claims about the stability of data structures in contrast to the processes using the structures, Cook and Stamper (1979, 12) noted:

> More stable than the rules are the structures of entities used to depict the world. The person who fixes these has the power to determine what part of the world is seen and how it is structured by the formal organisation in question.

This particular reference introduced a new factor into the data modelling process, that of power and influence. If the statement is true, it suggests the question: how might the power elements of a data modelling context affect the stability of the model?

2. The Consequences of Instability

Changes to the data model, whether at the conceptual, logical or physical level often require the significant application of technical resources. Typically, a major change in a database means the new logical and physical structures need to be designed and tested. Then the data from the old structures are unloaded, and mapped onto the new data structures. Sometimes this process requires some kind of interim processing in order to transform data values correctly, as would be the case if data structures were being converted from imperial measure to metric.

The first consequence of such changes to a data model and its physical implementation is the simple cost of the resources (machine, time, and human) which the process consumes. In the case of large mainframe databases, the process can take many hours, a lot of processing cycles, and data storage resources. One database re-organisation in my consulting experience took so long and consumed so much of the shared system resources, that the data centre would only undertake the task over the weekend, since running this process during the regular work week resulted in a measurable degradation in system performance for many other users. Such major data re-organisations often mean that the applications which use the database cannot function at the same time as the database is being re-organised. Clearly this would not be acceptable to many critical business applications, in banking systems, airline reservations and other real time applications.

The second major consequence involved in rebuilding a database due to changes in the schema involves the integrity of the data. Take for example, the major re-organisation of an accounts receivable system. Generally accepted accounting practices require some kind of process which would demonstrate that the closing balance of each account in the old version of the database is exactly the same as the opening balances in the new system. If for some reason the conversion from the old database to the new one is unsuccessful, the systems professionals must have procedures and resources in place to recover and re-install the old database.

Thirdly, notwithstanding claims about the independence of data from processing, systems analysts must also seriously consider how changes to a database might affect the application processing programs, as in the case where changed edit criteria assumptions may influence processing.

Finally, changes in the data model may result in limiting data comparisons from one iteration of the database to another. For example, assume that a personnel database kept track of whether company employees were married or not. This might involve defining an attribute on the employee record *marital status*, with three possible states: married, single, or divorced. This year the personnel department wants to redefine the classification structure to include "common-law marriage", "divorced", "separated", and "other" (for widow/widower, and possibly the category abandoned). In this particular scenario, no change to the data structure would be necessary. However, any attempt to compare the ratio of single to married employees from the current database to the previous one will generate a problem in data comparison. No data management software will assist the end user with this problem, which essentially entails the question of data interpretation.

There is no doubt that the issue of data model stability is a serious one from the perspective of both the efficient application of technical resources and the effective use of the data resources in a business. Keeping in mind both the stability question and the measurement considerations, we must next turn to the question of what a "data model" is before we can develop a method for measuring its stability.

C. THE DATA MODEL AND ITS IMPORTANT ELEMENTS

As noted in the general review of the data modelling literature, authors often talked about three different levels of a given data model: the physical data model implementation, the logical data model, and the conceptual data model.

1. Physical Database Design

Physical database design is the process of choosing the optimum degree of data redundancy, the best data compression, data ordering, data set generation management, and storage allocation for the physical resources which are available to the programmers. Technical systems staff usually undertake this design activity relatively late in the system development life cycle, when most of the application issues have been resolved.

2. The Conceptual Data Model

The conceptual data model attempts to recognise the essential information elements of the enterprise which should or could be managed by information technology. However, the process for constructing such a model is not at all consistent throughout the literature. It seems clear that the user community would be heavily involved in helping to create the conceptual model. But from a research perspective, what would have to be done before we could confidently say that a conceptual data model had been created, as distinct from a logical data model? Would a list of major system entities, and a Bachman diagram be sufficient? Would they be necessary?

According to Ruchti (1976, 122), a schema writer in a conceptual language should meet two requirements - empirical truth and semantic truth. By empirical truth, he said that the reader of a model should arrive at the same empirical interpretation as the author of the model intended. The empirical truth does not help us discriminate between conceptual and logical data models. Presumably, the test of empirical truth should describe both levels of modelling. By semantic truth, Ruchti specifically required the schema writer to conform to the data structures of the DBMS. By linking the idea of semantic truth to the data structures of a particular DBMS, Ruchti denied the idea that conceptual data modelling is supposed to be technology independent. The test of semantic truth is also not going to help us discriminate between the conceptual and logical data model.

Antoni Olivé (1985, 397) implied that the conceptual model is a necessary pre-cursor to the logical data model. He noted that it usually is not possible to verify the correctness of a conceptual model since the conceptual model itself is supposed to be the specification against which subsequent models are verified. He made a particular point about the impossibility of formally verifying conceptual model validity against the user's "real" requirements. If the author's statement about the validity of a conceptual model being impossible to verify formally is correct, this suggests that we should lower our expectations about the stability of data models. Olivé's discussion also does not provide any help in answering the question of how to discriminate between a conceptual data model and a logical one.

A. Solvberg, and C.H. Kung (1985) were a little more helpful in their description of the conceptual data model. According to them, a model is called a conceptual model when it consists of both structural and behavioural elements of the application. The primary role of the conceptual model is to serve as a common reference framework which is used by the systems analysts to communicate with future users of the system.

Everest (1986) did not use the term "conceptual data model" at all, but discussed instead the process of "designing the 'natural' data structures". His description and example of the schema which comes from a "natural" data structure included a preliminary analysis of the entities, attributes and relationships of the application area. The schema as he presented it would document a number of other aspects of the model such as whether the relationships are optional, whether they are unary or n-ary, whether they are reflexive etc. The prospective key to the entity is also documented. For Everest, the logical data modelling process consisted of analysis to normalise the data structures. Avison (1985) said that the conceptual schema must also document the entities, attributes, relationships and events to the extent where the size of the various data elements are chosen, and the data dictionary names have been assigned.

Notwithstanding the differences in the literature, generally authors agree that the conceptual data model seeks to document the entities, the characteristics of the entities, and the various relationships among the entities, without any technological considerations.

3. The Logical Data Model

Logical database design (logical data modelling) is the process of mapping or fitting the natural data structures which come out of the conceptual model into structures which can take advantage of the data structures and processing in the available DBMS. Theoretically, this is the first stage in which the available technology is given any consideration.

One of the major areas of disagreement among authors is where certain levels of detail in the analysis belong, whether in the conceptual data model or in the logical data model. There is also no one generally-agreed documentation standard at any of the levels of analysis which are described. For example, Avison (1985) took an approach where the conceptual data model requires more than a conceptual schema as described by Everest.

The specific objective of this research was to develop a measurement tool which would provide data to track the degree of change in a data model over time. The literature review above suggests some of the likely obstacles to this objective:

- There is no commonly agreed upon standard for preparing the conceptual data model.
- There is no common standard format and content to the conceptual data model.
- A logical data model is usually dependent on the DBMS in use.

Determining model stability necessarily implies comparing different versions of an application model. How are we going to be able to compare models which have been created using different modelling methodologies, involving possibly different DBMS's? The answer to this question is that there is one aspect to data analysis which is common to all systems, all applications, and all analysts. In the end, the designer must come up with a record design of some kind. Therefore, I have chosen to reconstruct the data model primarily on the basis of the evidence presented by the logical record design which is inferred from the physical design, if necessary. The reconstruction process creates a relational data model for each version of a given application and its data model so that models which have been created using entirely different techniques, whether by NORMA or by Jackson design, can be consistently compared. This approach also means that the model metrics from application to application can also be compared relatively conveniently.

The data model which is reconstructed in this way will document the entities and attributes of a model as a minimum. Careful consideration about the definition of "entity", "attribute", and "relationship" is necessary. One of the obvious challenges in this reconstruction process was to ensure that this did not introduce more normalisation than would likely have been intended by the original designer.

4. Elements to a Data Model Which Might Be Important to Stability Measurement

While the terms "entity", "attribute", and "relationship" seem to be relatively straightforward, there are some problems. No-one has been able to provide a series of rules for the modeller on when a given thing is an entity, an attribute, or a relationship. As far as the idea of relationship goes, it appears from most models that there are explicit relationships which the analyst identified and documented. It also appears that there may be implicit, un-named relationships which are part of the general user community understanding.

In determining what elements of a data model might be important to measuring stability, I have taken an end user hermeneutic perspective. In other words, the question which must be answered is: what aspects of systems data might affect the end user interpretation of the data?

When the system users come to interpret the data which is stored in a given system, what resources do they typically have at their disposal? First, the standard input documents and output reports which are used day to day are important hermeneutic clues. The standard inputs and outputs provide names for the entity types, attributes, and occasionally for the relationships. The inputs and outputs might use the same names as the data dictionary. The data element name as represented in the data dictionary provides another interpretation point for the end user, assuming that the end user has access to the data dictionary. The data model diagram when it is available and in use might also be helpful in understanding the system, as would the user and technical system documentation.

The question of naming, whether naming the entity type or the attribute, is not a trivial one, since the name itself is an important clue about its meaning for the person using the data. In satisfying the requirements of most formal modelling theories, we could acceptably refer to an entity type as "147", provided this was done consistently.

Such a name would not contribute to the meaning which must be made of the data structure and its data content by end users, or by database administrators for that matter. So the model stability measurement tool should track whether the entity names get changed during model evolution.

There is also the question of whether modelling results in the creation of entity types which are artifacts of the modelling process itself. For example, consider the case of an organisation which contracts out the maintenance of its vehicle fleet. It may want to arrange different contracts for different elements of the vehicle, such as mechanical repairs, tyres, and exhaust systems. The organisation might also want to keep track of the individual contracts, as well as the maintenance experience of each vehicle. Finally, they may want to keep track of which vehicle was serviced under what contract (i.e. a many to many relationship). Establishing an entity type to manage the data requirements of the contract-to-vehicle cross reference results in an entity type which is in some measure an artifact of relational design.

When we define a given entity type, we must have some way of telling when something is classified as this entity and when it is not. The classification rule is called the "inclusion criteria". For example, we may want to create an entity called "employee". Do we include:

- trainees?
- part-time employees?
- probationary employees?
- retirees?
- members of the Board?
- the chairman of the Board?
- applicants?
- contract staff?

The notion of inclusion criteria applies to the process of attribute classification as well.

I suspect that the inclusion criteria is rarely specified overtly but is found informally in different parts of a model implementation or is part of the "given" knowledge on the part of the application users. Some of the attributes may be part of the evidence for the inclusion criteria, and the edit rules in the database and in the application programs may be the remaining formal evidence. For the purpose of measuring model stability, do the inclusion criteria change over time?

There may also be informal inclusion criterion rules as well. For example, there may be an informal convenience rule of thumb which determines whether a particular item is included in the entity type inventory to be managed and tracked formally. Pencils might be understood as a low cost, low risk, readily available item to be excluded from inventory considerations.

There is also some question about the overall quality of the modelling investment. Given the significance which the technical literature gives to normalisation theory, it might be worth carefully examining how well this is done in the logical data model. When we look at how models change over time, it is possible that one of the subsequent causes of model modification may be related to modelling failure at the technical level of normalisation.

D. CREATING THE DATA COLLECTION SHEETS FOR MODEL CHANGES

In order to collect data which will contribute to considering these issues, I created four data collection forms attached as Appendix II.

- the attribute data sheet;
- the entity data sheet;
- the relationship data sheet, and;
- the model summary sheet.

The overall objective of this protocol was to collect data about a given data model and its revised versions which will help us to understand the changes to the data model. The primary source of data for this study is the logical record itself, but the analyst must be sensitive to other elements of a system which may contribute to the ways in which users interpret the system and its data.

1. Creating the Attribute Data Collection Sheet

I began with the design of the attribute data collection sheet to encourage relatively complete information on entities where possible. The attribute data collection sheet was documented first, since information from this tool is carried forward and summarised on the entity data collection form. I collected the attribute data for each entity in the model. The name of the attribute probably contributes significantly to the meaning which the application user can re-create in using the data. The values permitted in a particular data element are also important clues to the meaning of the model. A shift in the permissible range might be evidence of an important shift in meaning related to the attribute. When we extend the classification options, we change the data model in a subtle but important way. We would now have a special problem when it comes to comparing records of today with the previous database. This might constitute some kind of meaning instability in the data model, but not a structural one.

We should also pay attention to the edit rules which are usually performed prior to changes or insertions to the database. This applies only to those edit rules which are part of the application software or the DBMS rules. Edit rules are the way in which semantic integrity constraints are usually enforced. Thus, if possible we should ensure that our database will not store data on the age of men who have died in childbirth. We want to track changes in the edit criteria, since this is also an important part of the evidence which helps us understand how meaning can be constructed from the database. Changes in the edit criteria might be evidence of change to the model (Frost and Whittaker, 1983).

In addition to the stated integrity constraints, there may be rules which are implicit, but which might not be specified. For example, the date of birth for an employee record might have the usual constraints about 30 days in November, etc. but might not specify that an employee cannot be older than 65 years old, or younger than 14.

Some attributes are clearly more important than others. At the beginning of the design of the data collection forms, I thought that there might be a sub-set of attributes which might be called ontological. These are attributes which circumscribe the concept of the entity; that is, without which the entity would be meaningless. "Ontological" attributes are contrasted to "descriptive" ones. In the early field testing of the measuring instrument, prospective users could not come to terms with the notion of "ontological", and I substituted the idea of primary, secondary, and tertiary attributes. A primary attribute is an attribute which must be present before someone looking at the record can understand what the record represents. Primary attributes are contrasted to secondary attributes which are of interest, but are not fundamental.

The tertiary attributes are those data elements in the list of attributes which are used to control processing needs, such as for audit trail considerations. The existence of these tertiary attributes are interesting since one of the generally accepted objectives of data modelling is to un-couple data definitions (which are supposed to be more stable) from processing specifications (less stable). The presence of tertiary attributes which are process control related is an indication that the separation of data from processing has been unsuccessful to some degree.

The existence of a foreign key might give us some insight into the nature of the entity. A foreign key is an attribute which is used as key to a different entity. Tracking attributes which are used as indexes might be a way to confirm the judgment of the analyst about whether attributes are primary or secondary. For example, one would expect the incidence of indexing to be highest among primary attributes, and lowest among tertiary attributes. I also wanted to know if the attributes were derived or original. Is this attribute calculated or somehow derived from other attributes, or is it stored in its original form? How this factor changes from data model to data model might be a source of model instability.

2. Creating the Entity Data Collection Sheet

The entity seems to be the intuitive foundation for conceptual data model specification. Therefore, I created a data collection sheet for each of the entities, beginning with the name of the model and its original date. The question of the name of the entity might have some hermeneutic value to the end user of the data, by way of providing part of the data context. Changes to entity type names deserve to be tracked.

A key to understanding the ontology of the entity is to identify those attributes which the analyst defines as the primary way of instantiating any given member of an entity type set. So I tracked the number of attributes used which uniquely identify a individual entity (i.e. the entity type key). I also tracked the number of attributes which are foreign keys, that is, which are used to access data in other entity records, partly as a way of identifying potential relationships which were implicit and not explicit.

A number of pieces of information about the entity were collected primarily as descriptive elements, to provide some overview of what might be considered "typical", if such a thing appeared possible or likely. For example, the data collection tool was designed to record:

- The number of primary attributes for this entity, where a primary attribute is an attribute which must be present before someone looking at the record can understand what the record represents. Primary attributes are contrasted with secondary and tertiary attributes described below.

- The number of attributes that are secondary. A secondary attribute is one which is of interest to a community of users but which is not necessary to understand the fundamental meaning of the entity.
- The number of tertiary attributes. The tertiary category of attributes is used when unsure about the role a given attribute plays, or when the attribute obviously does not describe the entity in question, but supports other processing requirements.
- The number of indexed attributes. The number of indexed attributes is an indicator of the access paths which are of interest to the end user. An examination of the access paths might be a useful technique for reviewing the structure of the data to determine if the data structure accurately represents what is truly most meaningful or useful to the end user.
- Is the entity itself is primary or secondary to the application? Because normalisation rules force the creation of entities to avoid repeating groups, some of the entities may be more important than others. There also may be entities which are artifacts of the modelling process.

It is also important to describe any differences or disputes in definition or understanding across the organisation for each entity. The inclusion criteria (if known, or specified) are the rules which specify the conditions for membership in a given entity type category. These criteria are fundamental elements of the meaning context for any given entity, especially for entities which are primary to the application.

3. Creating the Relationship Data Sheet

Relationships are the way of associating entities and their attributes. The relationship data sheet collects limited data on the relationships in the model under study. The initial version of this data collection sheet provided for classifying relationships according to their symmetry, degree, optionality. However field testing of the data collection sheet demonstrated that it was quite difficult to classify many of the relationships these ways, and that the classification process did not look promising from the perspective of measuring data model stability. Determining whether the criteria for this relationship was explicitly specified, or whether it was generally understood by the user community without detailed explanation was a question of special interest. Most of the data collected on this form was straightforward.

4. Creating the Background and Model Summary Data Collection Sheet

The background and model summary data collection sheet was designed to collect a variety of data related to general questions such as when a model was developed and for what general purpose. Some of the more important data which were elicited through this form were:

- The organisational unit which commissioned the development of the model. The survey results in Chapter 3 suggested that most data models do not cross major organisational boundaries.
- The general application area - e.g. finance, payroll, personnel, inventory, accounting, scheduling, or some other.
- The date(s) of model origin and later revision(s) are important data in determining the rate of change over time.
- Why was the model developed? Prior to conversion to a new DBMS? As part of a new requirements definition? Or in support of an information resource planning project? For some other reason?
- Documentation standard and technique used. The objective of this question is to determine the documentation rigour which is used in the development of a logical data model. If automated modelling tools such as Excellerator, or IEW were used, or if a standard development methodology such as Jackson design methodology, the protocol has a place to indicate this.

For each of the original model and its major revision(s) I completed the following, where possible:

- Scope of modelling effort. Was this model created at the level of sub-application, application, division, or organisation as a whole.
- Estimate of modelling effort (in man days).
- Level and consistency of normalisation. Examining the normalisation level and consistency are ways to determine the technical quality of the development.
- Number of entities. This is a simple count of the entities in a given model. It was to be used to generate descriptive statistics on the extent and complexity of models.
- Total number of attributes for all entities. The number of attributes might also be used to create an index of model complexity.
- Number of formally established, or named relationships.
- A description of update processes to the original model. What was the motivation for the update? Were the original authors responsible for the modification? How much effort did the modification take?
- The degree and quality of management participation.
- The degree and quality of user participation.
- Entities, attributes and/or relationships in this model which have been the subject of marked differences in definition in the organisation. The objective was to describe the differences, and document how these differences were resolved.
- The degree of integration of one data model to other models from different parts of the organisation. Difficulties which this process had to overcome were identified. Specific examples of disagreements which required a negotiated settlement were specifically sought out.
- The areas in the models which have been particularly unstable.

A draft version of data collection forms and the instructions for their use was prepared. The package of forms and instructions was tested by two independent systems professionals. Based on their response, the protocol was modified to improve its understandability and clarity. At the outset, the above approach to documenting the changes to a given model over time seemed like a reasonable idea in theory. However, the application of the protocol in practice yielded a number of lessons about both the protocol, and the processes of data modelling.

CHAPTER 5 - A CASE STUDY OF THE TOOL IN USE

In order to provide a sense of the problems inherent in applying the data collection tools, and in analysing the data which has been collected, the following section of the thesis describes a specific case study in which the protocol was applied. This chapter of the thesis will:

- introduce the case study application;
- describe the first iteration of the application model;
- describe the major differences in the application after a significant software conversion,
- describe the differences introduced to the data model after a minor revision to the application, and;
- provide an analysis of the changes in the model over time in each revision.

A careful examination of the data collection tool in use is necessary in order to determine the limitations of the tool in serving the purpose of assessing data model stability, and to assess its effectiveness as a measurement device.

A. INTRODUCTION TO THE CASE STUDY

The first application of the data collection tool was in a relatively large real estate organisation which served the needs of a telecommunication company. The real estate group in this company was responsible for an annual capital building budget of approximately £20 millions, and an operating budget of £15 millions. The organisation had a building space inventory application call the Distribution of Office Space (DOFS) although the database tracked space for purposes other than office use.

For example the data included a large amount of space used for equipment, storage, warehousing, garages.

The data in the database was used for a number of purposes:

- tracking building space inventory;
- identifying vacant space for new tenants;
- allocating utility costs (such of heating fuel, power, water) of approximately £6.5 millions to tenants, and;
- generating revenue sharing data which accounted for income in the order of £7.5 millions.

While the application system was relatively simple and simplistic, the consequences of the data to the organisation were quite substantial. The initial system had been installed on a mainframe computer in 1984 using an unsophisticated file management utility called A Departmental Reporting System (ADRS). ADRS was written in APL as its base language for data manipulation.

B. THE DISTRIBUTION OF OFFICE SPACE SYSTEM

Researching the historical information on the Distribution of Office Space (DOFS) system was seriously impeded by incomplete systems and user documentation. Since the system had not been developed with a professional systems team using up-to-date tools and techniques, the rationale for the particular approaches which were used were unclear or not documented. To complicate matters further, the original author/designer of the system had retired and had subsequently died. The organisation had changed substantially, and there was extremely limited capability and understanding in the organisation that used and maintained the data in the DOFS system.

In order to understand how the model changed over time, we must have a good understanding of the entities, attributes, and relationships of the model. A copy of the completed attribute data collection sheet from the DOFS analysis is included in Appendix III. In order to demonstrate the problems inherent in the completion of the data collection protocol, the following discussion analyses each attribute. For clarity, the data elements are presented in italics exactly as they were used in the system. The upper and lower cases of the names have been preserved as well.

The research method began by collecting data at the attribute level for each of the entities in the application. The original system was designed to track the information on space areas, and on floors in buildings. However, the technical design and the documentation treated the data for these two things of interest in the same physical and logical record. Normally, one would expect to find different entities in different logical records. The documentation does give some indication of how it is possible to tell one record type from the other in the file, even though they are stored in the same physical file, record types interspersed. We must make a judgment from the record layout and the documentation about how many entities there are. A judgment was made by the researcher that the designers intended two separate concepts, one the floor and the other the space.

The first data element posed the first problem. It was called the *Assignment Number.* Unless the researcher or analyst had a reasonably complete understanding of the application and the file management software, he would not have any way of knowing what this data element was and how it was used. In this case, the name of the data element, its permitted values, and the application documentation were insufficient to understand the meaning of the data element. As it turned out, the assignment number was used by the software as the key to a particular record. In fact, it is the only unique data element for a given record. However, it is difficult to describe this as a true attribute of a given space, or a floor. The correct interpretation of this item in the context of reconstructing the conceptual data model is not to count it as a true attribute, but as a technological artifact introduced by the file management software i.e. tertiary type attributes.

The second data element was the *Account Code.* On the face of it, this data element might be interpreted in a number of ways. It might be used to allocate the operating cost of the building to the end user. Or it might be used to allocate revenues to the general ledger. Without knowing how other parts of the organisation use this data, it is impossible to interpret, since the documentation was incomplete on this point. While there was no formal edit of the data, the staff expected that any account code in this field must exist in the corporate chart of accounts. Classifying this attribute according to whether it is primary or secondary is also problematic in this case. The researcher had to make a judgment on the basis of the software and its documentation about whether this entity's primary purpose required an account code. In this case, the judgment was made that the attribute was secondary to this entity in this application.

The third data element was named *S.C.* This name was not meaningful unless the user referred to the documentation. The user community reported that S.C. stood for Settlement Code. Substantial reading was required, followed by discussions with staff outside the organisation responsible for the application, before I was able to come to any understanding of this data element. The settlement code was a separate code established by a Canadian national organisation in order to share revenues between companies who provided service to clients on a shared basis. Part of the complex revenue sharing algorithm required the space information be classified according to the agreed system of settlement codes.

The fourth data element was the *BLDG GRP CODE.* Given the subject of the database and the other data elements, one might have guessed that this name stands for Building Group Code. One would not likely guess what the significance of the item was without referring to the documentation and the chart of codes which were provided. The accounting department collected costs on groups of buildings where the buildings were small, and then allocated the costs to the users of the space in these building groups according to the space calculations.

The fifth data element was named *HOUSE SERV CODE.* Once again, the list of entities, relationships, attributes, etc. would not help the researcher in understanding what this data element is used for. In fact, most of the direct user community continues to have a limited understanding of the purpose of the field. Suffice it to say that the House Services Code is also related to the building cost allocation process.

The sixth data element poses yet a new problem. It was named in both the database and in the documentation as *OLR*. The documentation said that the OLR field was used to classify space according to whether it was "owned", "leased", or "rented". In the application, this attribute of a space is primary to the purpose of space planning, since the organisation had an important responsibility to monitor each of the three categories. There was one problem with the classification. Any discussion with the user community about it resulted in confusion about the differences between "leased" and "rented". The intent of the original analyst was to discriminate between space which was owned by the organisation and was leased/rented ***to*** an outside company, contrasted with space which was leased/rented ***from*** an outside group. The values permitted for these codes (i.e. O, L, R) actively interfered with efficient user interpretation, since few of the users could remember whether "rented" meant rented to, or rented from.

The next data element was *Lease Rental Code.* From the model itself and the fact that this data element was a numeric field, 6 characters long, one might have guessed that the lease rental code was an accounting related code. But to get a reasonable understanding of the field, one would have to consult the user system documentation in detail. This was also true of the eighth data element, *% of invoice.* The Lease Rental Code and the % of invoice data elements are used to allocate the costs of rental, that is space which is rented from others, to the end users of the space.

The thirteenth data element was called *City.* This data element would seem pretty straightforward except for the fact that it was only 4 characters long, and was redundant to the first four characters of a later data element, Common Language Location Indicator, (CLLI). The data element City was completely useless to the users of the application.

The fourteenth data element was interesting. The *Location Code (Accounting)* was a separate location coding system which was almost entirely redundant with the subsequent CLLI. This data element was no use whatsoever to the Real Estate organisation. The Location Code (Accounting) is not really a location code at all, but a unique asset identifier.

Each of the records, whether a space entity or a floor entity, contained the building name and building address. The system was not designed with a separate notion of building. Therefore, there was a significant amount of data redundancy and inconsistency in this data from record to record, where the records belonged to the same building. The physical sequence of the attributes related to location as found in the record design was a little puzzling. One would have expected to see the building name and address next to each other rather than separated by the floor indicator.

The reader is encouraged to review the other attributes which describe each of the floor and space records. A variety of other problems will become apparent. Note for example that the database stores data which is mathematically derived from other data elements (i.e. equiv. square meters, total assignable area, and Actual in Sq. Ft.), thereby trading off the cost of permanent data storage against the cost of occasional processing.

There are also two more data elements which have nothing to do with the ontology of the entities, *Effect Date*, and *New*. The first of these is the effective date of changes to the record, and the second is a field where the data entry operator can indicate whether this record has been added, changed, or is an available record area for a new space or floor record. The functionality of these data elements is entirely related to the audit trail requirements, and the need to identify changes in the database from month to month.

The 11 data elements from 31 to 41 were something of a mystery for everyone involved. The documentation said they were reserved, and did not provide any data names. There was no-one in the user environment who had the slightest idea what they might be for, and the actual database held nothing but zeros in these data areas for every record. I suspect that these data elements were used at one time for intermediate calculations during the month end processing. This question was never resolved.

The final curiosity was the last data element, the *Locator Code*. The name of this attribute might give the impression that this field had something to do with a location indicator. In fact it was an employee position code of the manager who occupied the space. The purpose of the position code was to be able to aggregate all of the space which had been assigned to a given manager. While one might expect that the account code might achieve this, the chart of accounts provided no convenient way for a manager to identify all of the account codes which were subordinate to his. Thus the position code was necessary.

In trying to reconstruct the conceptual data model from the existing physical and logical records, what have we learned? The conceptual data model which can be inferred from the record design is relatively straightforward, provided the documentation and the user expertise is consulted extensively. The record design and system documentation themselves are inadequate for reconstructing the intended meaning of the model. There is a demonstrated high degree of dependence on tacit user knowledge which gets passed along in the organisation, often by word of mouth.

This specific application, which was randomly chosen, was a good example of the problems which arise. It demonstrated some interesting data modelling problems:

- code scheme redundancy;
- mystery data elements;
- semantic confusion;
- complex and poorly documented interfaces to other automated systems in other administrative areas of the company;

- data elements which are not true attributes of the entity, but which exist to support routine file handling and processing, and;
- an artificial access key which also has nothing to do with the ontology or reality of the entity.

In order to examine the stability of this model, we must now compare it to the revised data model, the Space Recording and Control system.

C. THE SPACE RECORD AND CONTROL SYSTEM (1) - OCTOBER 1988

In mid-1988, a systems review by an independent professional indicated that the DOFS system described above was in poor shape for numerous reasons, ranging from the amount of mainframe resources it consumed during prime-time computing to the quality of the actual data. The real estate organisation accepted the recommendation to replace the system as soon as possible. The systems professionals assessed the options, and by successfully demonstrating a prototype, they convinced the organisation to switch from a mainframe product to a microcomputer using a relational database product.

Because of the urgency in coming to terms with the system weaknesses, the managers in the real estate department elected to design the new system to replace the existing functionality without extending the application scope, except for allowing opportunistic improvements such as could be made without interfering with a speedy replacement. The new system was called SPAce Recording and Control (SPARC). The same processes of data gathering and detailed analysis described above were applied to this subsequent model. The detailed data collection sheets are attached in Appendix IV. The comparison of these two models for stability assessment was then completed.

The most obvious change between the two logical models, given that the application design was intended to be extremely close to the previous one because of the pressure of time, is that there are now four entities instead of the previous two. The *building* entity emerges when the record design of DOFS is normalised. The simple step of normalisation saved significant data storage room, and eliminated many nuisance value inconsistencies and redundancies. The *floor coordinator* entity was part of the original documentation for DOFS but had never become part of the automated design of the previous system. In fact, *floor coordinator* was a record used in the space planning function served by DOFS, but it was managed entirely manually. This finding raises the questions of how, and whether, conceptual data models should include entities which are not part of the automated database.

The names of many of the attributes in the model have also changed, although the intent of most of these attributes remains essentially the same. The change of names improved the probability of a user understanding the data and its meaning without having to refer to the user documentation. "Properties District" instead of "Dis" and "Property Supervisor" instead of "F/M" are two good examples. In the building entity, two new attributes have been introduced: 1) the date of construction, and; 2) the date of acquisition.

It is also worth noting that the *Effective Date* attribute which was used in DOFS still existed in SPARC, because the new database management software also had no utility for routinely recording all changes to the database. In other words, *Effective Date* is an attribute of the building **record** not of the building **entity** which is described by the record.

The second entity is the *Floor*. A number of the attributes of this entity were changed as well, to improve the comprehensibility. For example, the OLR (owned/lease/rented) attribute was changed to OFT (owned, leased from, and leased to). It is hard to decide if this semiotic change should be counted as a change in the data model. The number of attributes of the floor dropped from over 40 to just six. Note that only two of these attributes are classified as primary, the building code (CLLI) and the floor number. The usable and rentable areas, like the effective date, were not created as true attributes of the floor, but to control certain aspects of processing in the space record. Even though users intuitively expect a notion such as "floor", the only real reason that the model needs this entity is for control purposes, to ensure that the database does not record space on floors which do not exist. In other words, the system was designed to ensure that a space record on the 13th floor could not be created if the building designer had been superstitious, and skipped the 13th floor when he named them. If the user could be sure that no error would ever be made on the related attributes in the *Space* entity, the floor entity would not be necessary since the attributes can all be derived from the space entity alone.

For the *Space* entity, the number of attributes dropped from over 40 to 23. More important, however, was the wholesale change in the definitions of certain fundamental attributes of the space entity, those related to the measure of space. Instead of the eight different attributes dealing with area measurement in the DOFS records, SPARC changed these to three measures only. All of these area measures, usable area, rentable area, and construction area, are defined in the American National Standards Institute Building Owners and Managers document.

There are two other primary attributes which have been added in this iteration of the data model. First, a space indicator, which is part of the key for the space entity, is a code which connects the database record to an area on a physical floor drawing which corresponds to that specific record. Second, the user community defined a number of categories of occupancy:

- V - Vacant;
- U - Under Construction or Renovation;
- O - Office;
- E - Equipment;
- W - Warehouse;
- G - Garage;
- H - House;
- C - Committed (i.e. the space has been committed for some future purpose);
- P - Parking.

This occupancy code was considered a fundamental attribute of the particular space belonging to the space inventory.

The entity *Floor Coordinator* appears to be a new one, but it is an exact duplicate of the manual version of this record. A careful reader of the floor coordinator entity might wonder why the location, city and address are necessary when this data is on the building entity. The address of the floor coordinator is a mailing address, as distinct from a street address. In a number of cases the mailing address for a person would be a postal box not located in the same building.

The effect of record normalisation was quite clear in that the number of attributes for the space and floor entities have both dropped significantly, while the number of foreign keys has risen. We can also see that the ratios of attributes to entities has shifted from 35:1 in the case of the original model to 12:1 in the normalised model. The ratio of relationships to entities has changed from the original model 0:2 to the new model ratio of 3:4 .

A number of other changes should be noted:

- Number of attributes added - 26
- Number of attributes dropped - 18
- Number of attributes with changes in name - 14
- Number of attributes with the same name - 9
- Of those with the same intent, the number of attributes having **exactly** the same range and domain - 12
- Of those with the same intent, the number of attributes which have the **approximately** the same range and domain - 3

There were other, more subtle changes, such as a shift in the order of the attributes under each of the entity types. Address-oriented data elements were all clustered, as are space measurement data. It is unclear from this research and from the literature whether such small aspects of a model design enhance its subsequent interpretation.

It is also useful to classify the reasons for the changes in the attributes of this application data model. The attribute changes which are included in the following table exclude changes to the attribute name, where that was the only change associated with that attribute.

Table 5.

REASONS FOR ATTRIBUTE CHANGES BETWEEN DOFS AND SPARC

REASON FOR CHANGE	Dropped	Added	Changed
Application design needs	5	5	0
	27.8%	19.2%	----
Technical design needs	2	3	1
	11.1%	11.5%	33.3%
Semantic reasons	0	1	2
	----	3.8%	66.6%
Useless attributes	11	0	0
	61.1%	----	----
Extending functionality	0	17	0
	----	65.4%	0.0%
TOTAL	**18**	**26**	**3**

A examination of the data elements which had their range and domains changed but which were still part of the model after conversion is presented in Table 6.

Table 6.

REVIEW OF RANGE AND DOMAIN CHANGES TO DATA ATTRIBUTES

Old Attribute Name	New Attribute Name	Range / Domain Change
City	Municipality	Changed from 4 characters to a text field representing the name of the municipality closest to the location of the building.
New	Delete/Add/Modify	This data element is used to control the audit trail requirements of the application. The number of codes has been expanded.
Ownership Code	Ownership Code	The first coding scheme for this attribute was confusing to all who used it. The alternative choice was made for improved semantic clarity.

Note that in two of the three cases in the above table, the names of the attributes have changed. In the case of "municipality", there is no doubt that most users of the system would find the data "Medicine Hat" more meaningful than "MDHT". The municipality data is not necessarily more meaningful than what was once held in the city data. The municipality is usually more helpful, but there are many remote locations for which knowing this data is not useful to the user. The value of the "municipality" as a data element is a function of how well the user understands the geography of the region.

By changing the coding structure of Delete/Add/Modify, this data element is more complete than previously. As noted above, the need for this data element is a function of database management software, which had no routine change reporting facility.

When the data from the attribute data collection sheet were carried forward onto the entity data collection sheet, a number of other aspects about the stability of this data model began to emerge. The model has been changed by creating two new entity types, one of which has been classified as primary to the application. If this entity, the *building*, was primary to the application, where was it in the first iteration of the data model? The building related data in the original model were hidden in the structure of the floor and space entities. Normalisation forces this entity out into the open.

The following table summarises the changes in the number of attributes:

Table 7.

TOTAL NUMBER OF ATTRIBUTES BY ENTITY AND MODEL VERSION

ENTITY NAME	DOFS VERSION	SPARC (1) VERSION
Building	0	12
Floor	27	6
Space	43	22
Coordinator	0	8
TOTAL	70	48

When we look at the detailed changes in the ranges and domains, a clearer picture of the degree of change emerges. Of the 70 model attributes in the original model, only 17 have survived the redevelopment process. More important, of the 16 attributes classified as "primary" in DOFS, only four remained in SPARC.

The semantic clarity of the model appeared to have been improved. There were still a few hidden semantic traps in this data model, however. For example, the occupancy code permits "V" to indicate "vacant" with an associated usable area, yet the space record has a data element called "surplus area". It is not clear at all from the logical record design what the distinction between surplus space and vacant space might be.

There is also a problem related to the question of the "inclusion criteria" at both the entity and the attribute levels. One the one hand, the inclusion criteria for the attribute is often suggested by the coding scheme of the attribute, in conjunction with the name of the attribute and its edit criteria. On the other hand, sometimes the classification schemes of the attributes may be a problem. What is the difference between a space which is classified as "storage" versus one which is described as "warehouse"? How does one classify office space used for storage purposes? There were also special problems for the definition of inclusion criteria for the entities. The model as set out by the logical data model, or even the conceptual data model, does not describe what is to be included as a "building" and what is to be excluded.

For a development that was supposed to preserve the same application functionality, this model has undergone major changes in almost every area except the names of the existing primary entity types. Connections and logical relations have become clearer through the foreign keys. The existence of these foreign keys is likely of more consequence to the technical system staff than it is to the casual user, who probably does not know what a foreign key is, or why it is important.

D. THE SPACE RECORDING AND CONTROL SYSTEM (2) - AUGUST, 1989

Less than a year later, the senior executive of the parent organisation embarked on an ambitious program to pass occupancy costs back to the users of the space on a cost recovery basis, in order to sensitise managers to the "true costs" of doing business. Up to this point, most of the cost of building construction and operation had been borne by the real estate department. With the existence of SPARC, with significant improvements in the quality of the data, and with the improved flexibility of the micro-computer based DBMS, the organisation decided to extend the application to include charging building occupancy costs back to the end user. This major change provided an unusual opportunity in this research to track the changes to the data model in yet another iteration.

The first major question which faces the person using the data collection protocol is one of classification. SPARC(2) has been significantly extended to include the elements of data necessary to support transfer pricing of space resources. In order to achieve this, the data model has added new physical record types:

- *Art Rate* - the rate charged to supply art from the corporate art program.
- *Run Date* - the day, month and year of the last month end run date.
- *Account* - a data set down-loaded from the corporate accounts for editing purposes.
- *Lease Information* - data for managing leases, such as the terms and expiry / renewal dates.
- *Adjustments* - financial transactions which permit adjustments to previous period calculations.

How should we classify these new record types? It seems clear that the *Art Rate* cannot be classified as a new entity even if it has a logical record all to itself. In my opinion, this data element was more properly classified as an attribute which is connected to the space record of those spaces which participate in the art program. The *Run Date* is also clearly not an entity, but an attribute which would be classified as neither primary nor secondary, but tertiary - that is, an attribute which contributes to the processing control of the cycle.

The *Account* record is also not an entity as most modellers would understand the notion, but is a lookup table used to validate an entity or subset of entities found on the space record, and possibly on the *Real Estate Coordinator* record. It is also clear that the last two logical records, *Lease Info* and *Adjustments*, would correctly be classified as new entities, although they should be described as secondary to the model and not primary.

The following discussion points out the more salient changes which have occurred in the data model as is evidenced by the reconstructed conceptual model as documented on the data sheets, comparing SPARC(1) to SPARC(2). The comparison of these two models showed that all of the entities from SPARC(1) are still present in SPARC(2). However, the number of primary attributes on the space record doubled, from 8 to 16. The fact that the number of primary attributes has changed so significantly is a sign that the focus of the application has begun to change. Originally, the design intent was limited to the space inventory function, but now the inventory function must share the limelight with the transfer pricing function.

We can also see that the ratios of attributes to entities has shifted from 12:1 in the case of SPARC(1) to 16:1 in the normalised model. The ratio of relationships to entities has changed from the original model 3:4 to the new model ratio of 4:6.

A number of other changes should be noted:

·	Number of attributes added	-	18
·	Number of attributes dropped	-	0
·	Number of attributes with changes in name	-	0
·	Number of attributes with the same name	-	50
·	Of those with the same intent, the number of attributes which have **exactly** the same range and domain	-	46
·	Of those with the same intent, the number of attributes which have the **approximately** the same range and domain	-	49

Consistent with the prior stability analysis, the reasons for the changes in the attributes of this application data model were classified.

Table 8.

REASONS FOR ATTRIBUTES CHANGES BETWEEN SPARC(1) AND SPARC(2)

REASON FOR CHANGE	Dropped	Added	Changed
Application design requirements	0	0	0
		----	----
Technical design requirements	0	2	1
		11.0%	25.0%
Semantics	0	1	3
		5.4%	75.0%
Extending functionality	0	16	0
		84.3%	----
TOTAL	**0**	**19**	**4**

A careful examination of the attributes which had their range and domains changed but which were still part of the model after conversion is presented in Table 9.

Table 9.

REVIEW OF RANGE AND DOMAIN CHANGES TO DATA ATTRIBUTES

(Previously Existing Entities Only)

OLD ATTRIBUTE NAME	NEW ATTRIBUTE NAME	RANGE/DOMAIN CHANGE AND IMPACT
Building Use	Building Use	The classification codes were extended to allow new classifications of buildings which had not been anticipated at the original design of this code.
Ownership Category	Ownership Category	The classification choice for this code was extended slightly.
Occupancy Code	Occupancy Code	The occupancy code as structured in the original SPARC design was semantically incorrect. The new design reflects a clarification of classifications.

In this case the attribute names have not changed, but the meaning has changed to varying degrees with the change in the domains and ranges. When the data from the attribute data collection sheet are summarised forward onto the entity data collection sheet, a number of other elements about the stability of the SPARC(1) data model begin to emerge. Two more new entity types were created, although I have classified them both as secondary. The following table summarises the changes in the number of attributes:

Table 10.

TOTAL NUMBER OF ATTRIBUTES BY ENTITY AND MODEL VERSION

ENTITY NAME	SPARC Version 1	SPARC Version 2
Building	12	17
Floor	6	7
Space	22	37
Coordinator	8	8
Lease Info	0	18
Adjustments	0	8
TOTAL	48	95

When we look at the detailed changes in the range and domain, a clearer picture of the degree of change emerges. All of the 48 model attributes in SPARC(1) have been carried forward to the revised model. However, there are still a few interesting observations to be made on this data. Recall in the first version of SPARC, the user community defined a number of categories of occupancy:

- V - Vacant;
- U - Under Construction or Renovation;
- O - Office;
- E - Equipment;
- W - Warehouse;
- G - Garage;
- H - House;
- C - Committed (for some future use);
- P - Parking.

Further analysis of this coding approach indicated a major semantic problem. It was possible that a given space could be described correctly by more than one occupancy category. For example, it is possible to have a house (H) which was vacant (V), or equipment (E) space which had been committed (C). Therefore, this data element was broken into two separate ideas - 1) a space use code; and, 2) an occupancy code. This is an example of inadequate semantic analysis in the original model.

While there have been a number of relatively minor changes to the part of the data model which served the previous inventory application, the vast majority of changes are a direct result of the extension of the application scope. The primary indicator of model stability is the number of primary attributes which are exactly the same in subsequent versions of the data model. Thus SPARC(1) is judged to be relatively stable compared to the SPARC(2) model. DOFS is judged to be relatively unstable compared to SPARC(1).

E. ANALYSIS OF THE CASE STUDY RESULTS

With the use of the data collection protocol and the analysis of the results, we can now consider the strengths and weaknesses of the protocol. The first conclusion that I reached was that any stability analysis must go beyond the superficial assessment in order to provide an index of stability. It would be easy to conclude on the basis of the normalisation which occurred on the first model re-iteration that the model was stable. The change in the total number of attributes in the DOFS model might have been dismissed as being primarily driven by the elimination of 11 mystery attributes, and by normalisation. On the basis of their names alone, we might say that the entities have not really changed over the entire life cycle of this application, that is 1984 through 1989, except where required to extend the functionality. However, the change in the primary entity, *space*, in the first re-iteration is profound. The single most important piece of evidence of instability which the data collection protocol and the above analysis provides is in the number of attributes which survived the process of re-modelling unchanged, especially the attributes in the original model which were classified as "primary".

Unfortunately, the data collection protocol does not draw to our attention other important areas of change in the model. The significance of some of the data elements may be found in the processing consequences of that element. The balance of the application focus has begun to shift to transfer pricing from simple space inventory, and the data element which is primary in determining the cost which each space is charged is the "space category". Prior to the introduction of transfer pricing, "space category" was simply a space planning data element. Thus the space category attribute has shifted from a secondary to primary one.

The data collection protocol did indicate that the meaning of the Real Estate Coordinator is beginning to shift. The evidence for this shift is found in the discussion of the inclusion criteria for this entity. The normal use of this record has changed quite substantially from the original data model. Reviewing the attributes and coding structures will not indicate this but the change in the inclusion criteria does. Note that nothing in the model structure and presentation would cue a user to this important change in the meaning of the model.

The indices of attributes (primary, secondary, and tertiary) to entity ratios do not seem to be an effective way of identifying the degree of change, but may be useful for establishing the rate of model growth and model complexity. The protocol does help in identifying a number of areas where semantic issues are important. The questions of which attributes are indexed, and which are derived, seem more closely related to system performance issues than to the meaning which a user can interpret from a given database and its data content.

There are a number of important disadvantages to the protocol as it stands. First, some shifts in the meaning of data elements, whether attributes or entities, are not identified by the protocol. A detailed understanding of the processing impact of the data element, and the use of the data element in practice are important ways of understanding the meaning of the element. Secondly, accurate and complete data collection requires the researcher to have a good appreciation of the application functionality, and the organisational norms as they may relate to the various terms and usages implied by the data model.

Finally, the use of the data collection protocol is very labour intensive. For the relatively modest model described in the case study, I invested on the order of 14-18 hours researching the fundamentals of the application, 12-14 hours documenting it, and a further 30-34 hours of analysis. While the analysis time for subsequent applications of the protocol should be easier, I happened to be familiar with the dynamics of the case model. The total investment of effort is something in the order of 56-66 hours of effort for a model with 4-6 entities, and fewer than 100 attributes. In order to subject a large complex model to this level of detailed analysis, weeks of effort would be required.

This chapter of the thesis has documented the tool which was designed to collect data for the analysis of data model stability. The application of this tool to a case study was presented in detail to document some of the problems when the tool was used. The case study also clearly demonstrated some of the semantic and hermeneutic problems present in a data model management. This chapter has demonstrated some of the analytical approaches of data model stability which will be applied to a group of data models included in the following chapter. Finally, we reviewed some of the problems associated with the measurement process.

CHAPTER 6 - THE RESULTS OF THE SUMMARISED CASES

Through contact with the professional systems community, I identified a number of models which were suitable candidates for further stability measurement research. The system professionals involved were co-operative and able to provide a minimum amount of documentation. The 7 applications which are the basis of the following summarised data were highly varied in their areas of administrative interest. They varied from project tracking, to real estate inventory, to accounting and sales management, to government administration of the skill trades and apprenticeship programs. A number of important findings emerged out of the analysis of how these models changed over time. The reasons given for the move from one data model to another varied from application to application. To paraphrase Tolstoy, stable models are stable for the same reasons, but each unstable model is unique in its own way.

In the first model, a combination of technical problems, and a major shift in business focus away from understanding the business as accounting for equipment space, to managing real estate assets, resulted in a change to the system. In another case, it was the failure of the technical system and the application support, and the need for closer control over the application functions. In a third case, the model was changed because of the application integration between two separate, but related systems. In the fourth case, moving the application from one processing platform to another created the opportunity to re-examine the application data structures. In another example, correcting semantic issues created by an initial data model, and extending the functionality was the basis for changing models.

The following section of the thesis reports the descriptive parameters of the models generally, followed by an assessment of the model changes over time.

A. A QUANTITATIVE ASSESSMENT OF THE MODELS AND THEIR CHANGES

As noted in the discussion of the literature on data modelling, there is little published on the typical size and complexity of practical data models. The data collection protocol developed above for measuring stability contributes some data in this direction. Table 11 below provides a summary of the size of the various original data models. The number of entities is a rough approximation of the complexity of the model, although this assessment of complexity ignores the number of relationships, named and unnamed, in a given model.

Upon inspecting the results in Table 11, one will quickly note the high variation in the number of entities across the sample. The number of entities varied from 2 to 26, and there are significant variations in the number of attributes per entity within each category of attributes. The frequency of tertiary attributes, that is to say attributes which are not fundamental to the entity which is being described, but which are used for control of processing, among other things, is particularly surprising.

If this group of models is any indication, models will vary dramatically in their size and complexity. Of course, the size and complexity of a model is not necessarily an indication of the consequence and importance of the model to the organisation, although one would not expect a large and complex model which was unimportant to an organisation.

Table 11.

SUMMARY OF THE ANALYSIS OF FIRST ITERATION MODELS

APPLICATION	ENTITIES	ATTRIBUTES				
		Primary	Secondary	Tertiary	Keys	Total
Personnel Skills	9	39	48	6	12	93
Sales and Payments	11	96	51	22	18	169
Apprenticeship	9	88	25	5	17	118
Project Tracking	9	61	88	14	20	163
Property Inventory	2	16	32	22	2	70
Lease Invoicing	4	21	24	3	7	48
Faculty Staff	23	95	69	7	33	171
TOTALS	**67**	**416**	**337**	**79**	**109**	**832**
Avg. # Attr. per Entity		6.2	5.0	1.2	1.6	12.4
Maximum # Entities	23	96	88	22	33	
Minimum # Entities	2	16	24	3	2	
Avg. # of Attr. or Ent.	9.6	59.4	48.1	11.3	15.6	118.9

The presence of the tertiary attributes says something about the purity of data modelling as practised, as distinct from process modelling. The complete separation of processing from data structures appears difficult to achieve. Table 12 below presents the summary of the same data for the second version of the models under analysis. Tables 11 and 12 provide the base data about the models which were examined, and give an approximation of the size and complexity of the models in this study.

Table 12.

SUMMARY OF THE ANALYSIS OF SECOND ITERATION DATA MODELS

APPLICATION	ENTITIES	ATTRIBUTES				
		Primary	Secondary	Other	Keys	Total
Personnel Skills	12	64	40	5	19	109
Sales and Payments	6	61	40	6	9	107
Apprenticeship	24	145	83	4	45	232
Project Tracking	8	113	163	23	10	299
Property Inventory	4	21	24	3	7	48
Lease Invoicing	6	40	43	12	14	95
Faculty Staff	14	51	32	0	25	83
TOTALS	74	495	425	53	129	973
Avg. # Attr. per Entity		6.7	5.7	0.7	1.7	13.1
Maximum # Entities	24	145	163	23	45	
Minimum # Entities	4	21	24	0	7	
Avg. # Att. or Ent.	10.6	70.7	60.7	7.6	18.4	139.0

Table 13 presents a comparison of the summary descriptive statistics comparing the old models to the new. As is noted in Table 13, the number of entities per model increased slightly. The ratio of primary attributes to entities rose slightly from 5.1 to 5.8 as did the ratio of secondary attributes to entities, from 4.1 to 5.0. The fact that the ratio of tertiary attributes to entities dropped from 1.2 to 0.6 is noteworthy. The increase in the average number of primary and secondary attributes is completely expected given the usual interest in increasing the amount of data which is stored. However, the reduction in the number of tertiary attributes is possibly an indicator that the database management software has improved in its sophistication by reducing the overhead which the programmer must take into account for managing requirements such as audit trails. To give a specific example, for those database management packages which time stamp or date stamp changes in records during the update process, the database would then no longer have to maintain this as data element on the entity record.

Table 13.

COMPARISON OF MODEL VERSION RATIOS

MODEL RATIOS	OLD	NEW
Avg. Number Entities	11.7	12.1
Avg. Number Prime Attributes	59.4	70.7
Ratio of Prime to Entities	5.1	5.8
Avg. Number Secondary Attributes	48.1	60.7
Ratio of Secondary to Entities	4.1	5.0
Avg. Number Other Attributes	11.3	7.6
Ratio of Other to Entities	1.0	0.6
Avg. Total Number Attributes	11.7	12.1

Table 14 below presents a summary of the changes which occurred in each of the attributes from the original version of a given model compared to its revision. With the identification of each attribute and entity which had changed over the life of the model, these changes were then classified according to whether the attribute was dropped, whether it had changed its structural coding characteristics, or whether it had changed for some other reason. The percentages in Table 14 are calculated on the basis of the number of attributes present in each of the categories of the first model results.

Table 14.

SUMMARY OF HOW MODEL ATTRIBUTES CHANGE OVER TIME

KIND OF CHANGE	PRIMARY		SECONDARY		OTHER	
	#	%	#	%	#	%
Added Functionality	28	7%	31	9%	1	1%
Dropped	134	32%	136	40%	55	70%
Moved	68	16%	45	13%	0	---
Expanded Coding	14	3%	5	1%	0	---
Contracted Code	7	2%	8	2%	0	---
Structural	9	2%	5	1%	2	3%
Extended Functions	76	18%	68	20%	8	10%
No Change	172	41%	128	38%	22	28%
Semantic	12	3%	10	3%	0	---
Original Attributes	416		337		79	
% Unchanged	41%		38%		28%	

Table 14 documents the way in which this method measures the stability of a given data model. It is done by closely comparing the number of unchanged data elements by attribute category against the original number of attributes by category in the data model as a whole. For example, in the seven models which were analysed, 41% of the primary attributes in the first iteration of the model are present in an unchanged way in the second interaction. An average of 38% of the secondary attributes continued exactly the same, and only 28% of the tertiary attributes remained after model revision. This last finding tends to confirm those authors who stated that the data model is more stable than the process model. That is to say, process control attributes (tertiary attributes) are less stable than primary and secondary attributes.

As demonstrated in Table 14, there are a number of reasons why a data model will change over time. In the first case is the need to add functionality. "Added" functionality is distinguished from "extended" functionality in that the added functionality is clearly an application capacity or capability which was not intended in the initial data model. By contrast, "extended" functionality is where the richness of the original application intent is enhanced. For example, in the case of an accounting application, extending the accounts receivable function by adding other cues for credit risk would be classified as an extended functionality, whereas integrating a general ledger capability would be adding functionality.

The classification "dropped" is quite straightforward. In these cases, the attributes were present in the initial model and were absent in the revision. Those data elements which are classified as having been "moved" are data elements which were present on one entity and were present with the same intent in a different entity, typically an entity created by changed relational design. The number of attributes which move from one version to another is often a function of the change in degree of normalisation.

It also became clear from analysis of these data models that some attributes experienced expanded coding where a given attribute type was split into two in the revised model. For example, the space occupancy in the case study became the occupancy code and the space use code. "Contracted code" is where the code design has been changed such that two or more attributes in one model become one coded attribute in the second model.

Some data elements change in the range and domain of their coding. Where the structure is significantly changed, this was counted as a change to the data model. Where a marital code used to be "single", "married" and "other", and this code was subsequently extended substantially to include other forms of cohabitation, this data element had changed sufficiently so that it could not be classified as being completely stable. Finally, there are a number of examples in the analysis whereby data elements were modified or changed primarily for semantic clarification. A classic example of this was in a software sales application, where there was a clarification of client addresses. The revised model included a mailing address, a shipping address, and a billing address.

Table 15 presents a calculation based on the analysis of the relative degrees of data model stability for the seven sample data models under discussion. As shown in Table 11, there are some potential surprises awaiting researchers who collect further data in this area. Intuitively one would expect that the degree of stability of a data model would change over time; that is, that the number of changes to a data model would increase as the management of a given application changes over time. One might expect that change of management personnel and user requirements typically varies over time, and that therefore instability is a function of the time between the two model versions.

As is demonstrated in Table 15, the application models compared over the longest period of time had an above average degree of stability (the apprenticeship management system). On the other hand, the system which was in place for six months, the faculty staff system, had the lowest degree of data model stability. This result was not a function of the degree of normalisation, since the original faculty staff model was fully normalised. In the case of the apprenticeship system, the original data model was not normalised whereas its replacement system was.

Table 15.

SUMMARY OF STABILITY INDICATORS BY MODEL

PERCENT OF STABLE ATTRIBUTES COMPARED TO THE ORIGINALS					
APPLICATION	PRIMRY	SECNDRY	TERTRY	MONTH	REVISION REASON
Personnel Skills	46%	50%	100%	24	Enhancements
Sales and Payments	43%	43%	32%	14	Maintenance
Apprenticeship	43%	28%	40%	80	Application Integration
Project Tracking	46%	20%	0%	55	Redevelopment
Property Inventory	31%	31%	18%	33	Redevelopment
Lease Invoicing	90%	96%	100%	9	Enhancements
Faculty Staff	24%	35%	0%	6	Redevelopment
Average Stability	41%	31%	28%	31.6	

From a quantitative perspective, the measurement protocol generates some data which is useful in tracking instability in data model management. The measurements generated through this analysis also provide data which might be useful in assessing data model complexity. However, there is also qualitative data which deserve discussion.

B. QUALITATIVE COMMENTS

One of the keys to interpreting the meanings implied by systems design choices is the presence of the individuals who were involved in the application design and use. In the case of many of the older data models, interpreting the data structures was extremely difficult, since the original system designers, both technical and application specialists, were no longer with the organisation, and in some cases were no longer alive. One of the important lessons of this research is confirmation that most of the semantic relations involved in a data model are a matter of the heuristic hermeneutic of the user community. The data model, whether represented by record design, data flow diagrams, ER diagrams, decision tables, or combinations thereof, is subject to significant interpretation by those who use the data. In other words, the meaning is not in the model and its data, but meaning is made of the model and its data. Inferring data from the documentation in the absence of those who were originally responsible for making meaning of the original models is a major limiting factor in the results.

The analysis presented above is missing a significant amount of data which might have been expected from the data collection questionnaire in Appendix II. However, much of this data could not be collected consistently. First, the data collection on relationships between entities was not considered useful by the data modellers who participated in this research. Important relationships are invariably reflected in one of two situations: 1) where the relationship is demonstrated through the existence of a foreign key in the entity records; and, 2) where the user community wants to collect data on the relationship, and thereby turns the relationship into an entity. In the latter case, data about the relationship is reflected in the attributes of the entity.

There were a number of questions on the Model Summary sheet which were not useful. The names of the model authors posed something of a problem. In the vast majority of modelling cases, no one person can be said to be the sole architect of the model, since model construction usually depends heavily on the contribution of the user community, and possibly a team of systems developers. The question on the estimate of modelling effort was also too imprecisely phrased to be answered sensibly. Users asked a number of questions to clarify the intent. For example, where does background investigation into a system requirement end and data modelling begin? The question as phrased does not indicate whether the effort of the users should be included in this estimate, or whether the days invested in refinements to the model in subsequent phases of the development should be included. Finally, none of the users had any reasonable tracking measures in place which might give some indication of the actual effort spent on modelling, especially for older systems.

The questions related to the quality and extent of user and management involvement were naive and simplistic ones. In retrospect, and after discussion with data analysis practitioners, it is clear that user and management commitment and involvement are difficult things to measure at the outset, and assessments of this typically vary from person to person in the user community. Defining and determining the quality of participation is a question which deserves a separate research initiative. Finally, it appears from this research that normalisation is not a major element in addressing the issue of data model stability, and therefore measuring the degree and consistency of normalisation is not useful.

C. JUDGING AND ASSESSING THE PERFORMANCE OF THE MEASUREMENT PROCESS

The following section of this thesis will assess the quality and effectiveness of the tool developed and applied earlier in this research by summarising the process as a whole, and by applying these measurement effectiveness criteria to the protocol.

1. Summarising the Stability Measurement Process

There are three basic phases that are integral to the measurement protocol used to assess stability in the above models. The first phase is that of reconstructing the conceptual model from the logical and/or physical record design as determined from the DBMS information and the user and system documentation. This is done by identifying the entities that are part of the model, and classifying them according to whether they are fundamental entities, whether they are artifacts of the modelling process, or whether they are a function of the DBMS which has been used. For example, in the case study the SPARC(2) model documented a number of logical records that are really attributes, such as the *Art Rate*, or which are used to control the processing cycle, in the case of the *Run Date*.

The second step of the reconstruction phase is the documentation of the attributes that form each entity type relation. The major attributes of the attributes which are recorded are:

- attribute name;
- attribute range/domain;
- attribute category (i.e. primary, secondary, or other); and,
- attribute edit rules (whether explicit or implicit).

The last step in this phase is to summarise the data on the attribute data collection sheets, to the entity data sheets and then forward to the model data collection sheets. The entity inclusion criteria are also documented.

The second major phase of the process is to identify and classify the changes in attributes, entities and relationships. Most of the substantive changes occur at the attribute level, while the names and inclusion criteria of the primary entities remain stable for the most part. Once the changes have been identified, they are classified according to the cause of the changes.

In the third phase of the research protocol, the changes in the model as a whole are analysed. In the models which were studied, most of the serious changes occurred at the attribute level. The first step in the analysis should be to confirm the status of the entities. Have any of the entities that were considered primary to the application in the first version of the model been dropped? Have the inclusion criterion changed significantly, and if so, for what reason? The percentage of new entities is likely a crude measure of the change to the application focus, unless these primary entities have emerged from the normalisation process.

The most direct measure of data model stability is the percentage of the primary attributes which are present in subsequent models without significant change. In the case of the transformation of DOFS to SPARC, the user community claimed that the functionality of the application was supposed to be same except for opportunistic improvements. But the number of primary attributes that were dropped, or which were significantly changed is much higher than in the next iteration of the model where the application focus of the system was shifted significantly. There are two direct measures of data model stability that can be calculated:

- the percentage of primary entities which are present in revisions of the data model where the inclusion criteria has not changed significantly; and,
- the percentage of the primary attributes which are present without substantial changes in the range/domain, or in the entity type relation.

Taking this direct analysis of the data collected in the research as the two major indices of data model stability, the criteria of measurement effectiveness should now be applied in evaluating the protocol itself.

2. Applying the Criteria of Measurement Effectiveness

How well does the measurement protocol of this research satisfy the first criterion, that of measurement objectivity? One of the characteristics of the protocol as developed is that the stability index is a direct function of the change in primary entities and attributes. The researcher must make the classification of the entities and attributes according to his judgment about the primacy of these elements to the data model. The definition of what is "primary" entity or "primary" attribute is not scientific, and judgments about this classification will likely vary somewhat from one researcher to another. The quality of these judgments is also prejudiced by poor documentation, and users who were unfamiliar with the design intent of original models and their replacements. The objectivity of the stability measurement is improved by careful research.

When completing the data collection process, I found cases of attributes where the classification decision was inconsistent from one assessment of the model to another. This typically occurred where there was some doubt whether a given attribute was really "primary" to the entity, or not. I suspect that if different individuals completed the data collection sheets for the same model, results would differ depending on the degree of familiarity the researcher had with the application, and of the generally understood meanings implicitly shared by the users in a given business environment. The protocol satisfies the need for a well defined, relatively easy-to-understand procedure. How much the results of applying the procedure would vary according to who applied it is a question for further research.

From the perspective of reliability, I judge the protocol to be relatively reliable, assuming that the researcher applies a reasonable level of care in the classification process which is applied to the attributes. The validity criterion as defined by Osgood, Suci, and Tannenbaum (1957) (i.e. co-variance with other independent "index of meaning") cannot be applied since no other independent measure of stability exists. Intuitively speaking, the protocol calculation of stability seems to be satisfactory.

Judging the sensitivity of the protocol is difficult. The protocol is sensitive to most changes but not all of them. The protocol as it stands does miss one important likely area of model instability, and that is in the attribute inclusion criteria. The names of the entity types tend to persist over time, even where there is evidence that the inclusion criteria has begun to shift significantly. This happens in the language generally. The notion of a "hansom cab" in London has gradually changed over the years from a horse-drawn carriage to a gasoline powered vehicle. It is not enough to say that the affordance of the hansom cab and the taxi are the same, in that they are both vehicles for hire and therefore, the inclusion criteria of cab vs. taxi are irrelevant. Understanding the inclusion criteria is part of the process of clarifying the semantics of the notion. The protocol does not track the changes to the inclusion criteria of attributes carefully, and the primary measure of instability does not take this factor into account. Therefore, I conclude that the protocol is not as sensitive to changes in model meaning as it could be.

The applicability of the protocol is quite good. It is specifically designed to apply to all forms of modelling which result in a traditional database file. It might not be as easy to apply in the case of some mathematical modelling methods, using FORTRAN for example.

The utility of the protocol is variable in that some aspects of the data collected on the models above are distinctly better than others. Documenting the attributes in detail, and classifying them according to "primary", "secondary" and "tertiary" is quite useful. However, the edit criteria was not nearly as helpful. As I have reported in Chapter 5, Section E of this thesis, the data collection process using the protocol as set out is a labour-intensive one. In the case study, my familiarity with the application area made the process much faster than for applications where I have had less experience. It is clear from my experience with this research that the meaning and interpretation that can be made from a given model, from its technical implementation, and from its data is much a function of the undocumented semantics present in the organisation. Due to a factor which I call variable semantic determination, the full consequences of these undocumented understandings are seldom available in one place or with one person.

My experience with the protocol and the analysis of the results which it generates suggests that there are areas of data collection that do not contribute much to the direct determination of data model stability. Whether the attributes are indexed, are foreign keys, or are derived do not appear to contribute anything to the analysis. Documenting the edit rules which are part of the tacit understanding of the organisation is also a time consuming, difficult, and ineffective part of the protocol. The protocol does require a careful review of the complete design of the data model, conceptually and as implemented. One of the side effects of this research was the discovery of a number of errors in database design at the level of the data specification to the DBMS. For example, where I classified an attribute as primary to the entity, I expected that the DBMS edit routine would require this data to be entered before storing the data. Many of the fields which ought to have been designed as mandatory were classified as optional to the DBMS.

From a value perspective, I found the protocol to be worthwhile from a research perspective and from a commercial systems analysis point of view. Applying this tool has provided me with some additional insights into the prospects for improving the practice of data modelling, and possibly its theoretical base. It has also forced the user communities of some of the application models into a clearer articulation of the meaning intended by their models, and has initiated subsequent refinements of the model in some applications.

In summary, the stability measurement protocol demonstrates some weaknesses when compared to the usual standards of a measurement effectiveness. However, it is the best measurement tool to have been developed to date. The results described above successfully demonstrated that it is possible to design and apply a method for measuring the stability of data models. The analysis of the models under scrutiny demonstrated that most of the changes in these data models were primarily related to shifts and changes in the users' understanding and construction of the world and secondarily to modelling errors. Some semantic errors and confusions led to changes in the models as well.

PART III
TOWARDS A CRITICAL THEORY OF DATA MODELLING

CHAPTER 7
DATA MODELLING STABILITY, INSTABILITY, AND EVOLVABILITY

Part III of this thesis is directed at exploring data modelling from the specific issue of stability measurement to the general questions about the theoretical underpinning of data analysis. Chapter 7 addresses the questions of stability, instability and evolvability of data models in the context of the evidence generated by the stability measurement tool. Chapter 8 discusses the implications of these results for data modelling generally, and for the broader topic of systems analysis. Finally, Chapter 9 presents the conclusion of this research and identifies a number of further research initiatives which are suggested, wholly or in part, by the results of the stability measurement protocol.

As the survey of the data modelling literature in Chapter 2 and 3 demonstrated, there are numerous claims for data model stability, notwithstanding the fact that there is no well articulated and agreed upon definition of what a data model is, or what the "stability" of a data model might mean. Prior to this research, there has been no attempt to measure the stability of data models. With the protocol that has been developed here, we now have an operational way to determine the indices of data model stability.

A. GENERAL OBSERVATIONS FROM APPLYING THE MEASUREMENT PROTOCOL

Some general observations which came out of the measurement protocol are in order before the specific issue of stability is addressed. In the protocol, the classification of attributes according to whether they are "primary", "secondary" or "tertiary" is fundamental to the assessment of data model stability. The classification of attributes in this way was done by one researcher, after coming to the best understanding of the application, the entity, and the attributes in the time available, with the user community participation. Any researcher using this process must carefully consider the accuracy of the classifications, especially where they change from one version of the model to another. This classification process is useful from two important perspectives. First, it provides a coarse assessment of the degree of data model stability over time by determining how many of the original primary attributes remain the same, are changed slightly, are changed radically, are added as substitutes for previous primary attributes, or are dropped entirely. Second, it provides a coarse index of the change in application functionality, by the analysis of the number of primary attributes which have been added for reasons other than substitution.

A potential objection to this process is that the attribute classification process looks suspiciously like a reversion to an objective reality orientation. In other words, it is easy to suggest that choosing a set of attributes which are thus classified as "primary" and "fundamental" to the application seems to run in the direction of "one correct interpretation", akin to the "objective reality" position. The judgment of the researcher is extremely important in this classification process. However, the goal for the researcher is to make this judgment taking into account the norms of the particular user community for whom the application is primarily intended to serve. Secondly, the purpose of the application likely has an enormous impact on which of the attributes will be classified "primary". In fact, the extent to which new attributes are classified as "primary" is an index of the shifting focus of the application functionality.

One of the important peripheral lessons from the stability analysis is that there is much more to a data model as far as it relates to end user interpretation, than simply naming the entities, naming the attributes, documenting the relationships, and normalising the records. The process of applying the stability measurement process, and having to infer the meaning of a data model without an intensive understanding of the applications, helps to emphasise the importance of extra-model elements for the meaning of the data. One will quickly recognise that most of the meaning which can be inferred from a model and its data is not based on the evidence of the model by itself, but on the informal semantics of the user community in conjunction with the model. The degree to which the semantic understanding of a particular user matches the intended semantics of the original data modeller, as they were originally designed, and as they have been revised in practice by the user community, is probably a significant index of the likelihood that the user will be able to "put the data into context".

One of the outstanding questions generated by this research is the question of whether the inclusion criteria is a necessary element of a model, either at the conceptual or logical level. There is little discussion in the literature about the inclusion criteria. For example, in Flavin (1981, 39) the idea of "defining properties" is used instead of inclusion criteria. A defining property is "the essential, observable features of the object that are associated with all occurrences of the object". The process by which one actually identifies these defining properties is not described in detail. Everest (1986, 247) discussed the importance of the inclusion criteria, primarily for classifying the entities, but the usual database user does not have access to this classification process when he tries to interpret the results of a query to the database. In the end, the property which might "define" a given object might not be an observable feature of the object itself, as in the case of GI Joe. GI Joe is a "doll" and not a "toy soldier" primarily because the United States Supreme Court said he is. Frequently government agencies have the power to deem that a given state of affairs exists, apart from whether any particular person, or company, agrees with the decision. Actually, GI Joe is probably both, "doll" for the purpose of calculating and remitting customs duty, and "toy soldier" for marketing purposes.

Applying the idea of the inclusion criteria at the level of the entity, the attribute, and the code level within the attribute is not clear in the literature. Indeed, this research method into data model stability has approached the definition of the inclusion criteria strictly from the perspective of the entity, and has documented it in prose form only. As we have noted, DBMS's have no facility to track, document, and manage the inclusion criteria as an important element for meaning and hermeneutic support for the data model and the database. The research demonstrated cases where the inclusion criteria may not have anything to do with the data attributes which are worth storing.

In the stability analysis, the concept of inclusion criteria should apply at a level beyond the entity. The idea of inclusion criteria can be applied at the level of the attribute as well. In the typical data model, the semantics of the inclusion rules for classifying a given entity using a given attribute set are not documented. For example, who is responsible, under what conditions, for what purpose to change the data in the space database which describes a building space as "committed" to "under construction"? When the contract with the construction company has been signed? When they take possession of the keys to the floor they will work on? When they begin demolition of the existing dividers, as required? When they begin to re-build the space according to the specification? After the first inspection report which confirms that some or all of the above activities have begun or have been completed? The criteria which are used in these classification processes are frequently not specified in a detailed way, and often depend on the semantic understanding and formality of the person collecting and communicating the data to the database system.

It is clear from the models which have been analysed here that the inclusion criteria as applied at every stage of data model specification are an important factor for data model stability. The absence of any extensive discussion of inclusion criteria as it affects the interpretation of a data model as well as its stability must be viewed as a serious weakness in data modelling theory.

B. WHY DO MODELS APPEAR STABLE AT ALL?

The method for monitoring data model stability did not anticipate the importance of the inclusion criteria beyond the entity. The stability measurement process documents changes at the level of the entity only, and does not measure the degree of change which the criteria has experienced. However, we can say there is definite evidence that it does change, and that it does so without having any direct impact on the data structures or on any of the processing. In due course, the users of the data may notice a change in the membership of the entity set in question and will have to re-negotiate the meaning, to clarify the impact on processing and on their understanding of the entity specifically and the data model generally.

Insofar as the "primary" attributes and their change over time are an index of the relative stability of a data model, the early evidence was that primary attributes persist in relatively stable models. All of the models which were assessed resulted in additional primary attributes as well as secondary ones. Changes to the range and domains of the attributes in general demonstrated finer and finer gradations of attribute distinctions, usually resulting in a gradual extension of the codes which are permitted for a given attribute. This is probably a result of the empirical evidence successfully challenging the existing classification system, thus requiring the user to refine the coding scheme. There might be some generalisation which could be discovered about the rate at which this phenomenon occurs, and whether the rate of including new codes into the classification scheme for the entities in the data model might eventually fall off and approach zero asymptotically.

The traditional view of data models, especially normalised entity relationship ones, is that they are stable. At the outset of this research, I had limited expectations (one way or the other) for the stability of data models. The theoretical foundations upon which data modelling rests are remarkably incomplete, when considered critically. As has been noted in the summary to the review of the existing thinking on data modelling, the principles for building a data model as set out in the literature are contradictory, unclear, unspecific, and/or incomplete. What might then account for the impression the literature gives that data models are stable? The answer to this question may be related to the name of the primary entities.

The users have the name of the entity to help them with their interpretation of the data model. Like the names of the attributes, the names of the entities also vary in the user environment. In the case of the personnel system, the individuals who worked in the organisation were variously, and interchangeably, referred to as "staff", "employees", "personnel", and even "workers" occasionally. But within the database schema, for each entity there was one single name which appears to have persisted over long periods of time. This is an important point. I suspect that the persistence of the **name** for a given entity, even when the primary attributes of the entity change, or when the inclusion criterion change dramatically over time, may be the most important contributor to the perception that data models are stable over time.

Another reason why data models do not change more often may be related to the labour intensity of the modelling process itself. The effort necessary to develop, validate and document data models, whatever tool and methods are used, is significant. The survey of data modellers and data administrators across a wide variety of commercial and public sector applications documented the pressure they are under to come to terms with the problems raised by data modelling. Their users want them to deal quickly with the continuing backlog of application needs, such that there simply is not the time and energy available to re-visit previous modelling efforts. The initial results of this research indicate that user groups may be having difficulty getting the attention which would be necessary for ongoing maintenance of the data model until there is a pressing functional reason to revise the application as a whole, or until some other factor such as organisational or functional integration forces an application reconsideration.

In such circumstances, it would seem that the data models should remain highly stable, if there are not the resources to invest in ongoing maintenance. In other words, where data models are stable, they may be stable for the wrong reasons. In fact, Everest (1986, 416-17) pointed out that changes to the data model result in a restructuring conversion process on the logical structure of the database, potential changing processes which act upon the revised database, (including user application programs, catalogued queries, stored report definitions, stored transaction definitions). Everest wrote that it is necessary to change people's understanding of the database concept. This process might involve revising user manuals, sending out change notices, and conducting training sessions. With this amount of effort, there is the potential for a built-in resistance to making changes. Thus, the stability of a given database may not have anything to do with the quality of the data model.

The evidence of this research is that the structure of data models and the database content contribute rather less to the interpretations which are made of them by users groups, than a wide number of other factors, such as the user documentation, the generally understood inclusion criteria for entities and attributes, the word of mouth explanations passed from user to user, and other forms of tacit understanding throughout the user community. The initial evidence of this research is that the data model and the database may be stable partly because the other meaning constituents of the environment are far more powerful and more flexible even in the context of the relative inflexibility resulting from data modelling structures, semantic integrity constraints, update rules, and other technological elements. In other word, even a poorly designed database which deserves to be unstable might appear stable specifically because the users have a rich repertoire of meaning constructs and coping mechanisms which can compensate for the weaknesses and limitations of a database and its modelling constructs, thereby making technical changes unnecessary.

This suggests that if the data modelling constructs become more effective, more thorough, and more flexible in describing the user "reality", the apparent stability of the modelling process may actually decrease. This phenomenon may occur for the following reasons. Most of the "reality" which is managed in part by automated databases is socially constructed, and is thus as exposed to socially-driven change as any other social construct. The notions of "employee", "student", "asset", "account", and even such a tangible idea as "building" as we will see, are all a function of the norms of the organisation which seeks to record data about these things. Thus as the opportunity or need to modify a given norm for "employee", or "building space user" arises, the degree to which a data model will have to change could well be a function of how thoroughly the given modelling process describes the norms. The other lesson in this is there is no need to resolve the age-old question of whether there is an objective reality or not, since most (if not all) organisations and people do not behave as if there is. They simply operate with group norms that shift over time.

In the case study, for example, the entity "Real Estate Coordinator" began quickly to shift its meaning after the last iteration of the data model. The user community decided that it would be far more useful to limit membership in this entity of the database to the relatively senior executives to whom the real estate organisation wanted to focus the responsibility for space usage. This was in marked contrast to the previous business policy of having a network of coordinators distributed much more broadly throughout company. The undocumented inclusion criterion for this entity had changed dramatically, yet nothing in the data model reflected this important shift in meaning. The model appeared to be "stable". The existing data structures and relationships afforded an acceptable data management process in support of a conscious policy decision in part of the user community. In fact, the systems professional who created the model was unaware of the change.

How stable are models, especially when compared to the claims of modelling literature? Even taking the most simplistic understanding of what constitutes a data model, they are not really stable at all. Only one of the seven models analysed showed more than 50% of the primary attributes unchanged after model modification. The stability of data models is more apparent than real, chiefly because the name of the entity type is relatively stable, even though its primary attributes, its secondary attributes, and its inclusion criteria might be changing dramatically. It is unreasonable to claim that the entity type is stable because its name is the same, when it now means something quite different from what it did, through a change in consensus of meaning.

The main reason that data models appear to be so stable may also be that they are so poor at "capturing the meaning", and people are so effective at intuitively accommodating these weaknesses through projecting meaning **onto** the data structures, rather than abstracting meaning **out of** them.

C. STABILITY VERSUS EVOLVABILITY IN DATA MODELS

The evidence of the data models included in this research indicates that models typically grow rather than shrink. This observation applies to the number of entities, the number of attributes, and the average number of classifications within the coding schemes. However, the names of the entities tend to persist, even where their inclusion criteria change. The extension of the data model was usually attributable to extending the functionality of the application and not particularly because of modelling errors at the outset. The question of the teleology and affordances associated with each entity and attribute are also significant elements of the "meaning" for each. The measurement method did not document these meaning elements of the models. Furthermore, if the relationship of the teleology and affordances are fundamental meaning primitives to the correct interpretation of a data model and its database, then the fundamental notion of data independence from processing is flawed in an important way.

In all of the cases analysed, the models got bigger not smaller even though many of the attributes get dropped from model to model. The net effect in all of these cases is models which are bigger, not smaller. There must be some limit to this phenomenon. Historically, the limit has probably been technological, but if there were no limit here, presumably there would be a limit to the amount of data which the organisation could capture, process, and interpret in a useful way. This would imply some kind of theoretical limit to the size of databases, but not to their instability, since the changes which one can make to the classifications and definitions are practically limitless.

We have seen that instability in data models was caused by a variety of factors in the seven study applications, including poor modelling practices, incomplete functional analysis, changing application requirements, and occasionally, semantic errors. The costs of changes to the schema are real, especially where new primary attributes are added to the schema, and where the new primary attributes are not available in any machine readable form. For example, if an organisation decided to change its compensation practices to take into account the economic conditions of the place where the employee lived, such as the prevailing rent for the type of accommodation which the employee chose, the data model would have to be extended in a number of different ways which might include attributes related to employee preferred housing arrangements. The payroll database might have some information related to their employees' marital and dependent information, but it is unlikely to record the number of bedrooms his current accommodations have. The cost of extending the database structure to include such data might be relatively modest, from the perspective of redefining the data model and reloading the database. However, the cost of collecting, validating, entering, verifying, and integrating the data into the routine processes when this data changes is likely to be substantial.

This research demonstrates that the role of the different elements of a model is important to the issue of stability and evolvability. For data analysts who are interested in the question of consciously managing change in data models to preserve meaning appropriately over time, these model elements should be carefully considered. Attempting to design data models specifically to be stable along all of the documented dimensions might also force the user to design with more of an eye to the future. The focus of evolvability and stability should be on managing the interpretation of data and the preservation of meaning over time.

Parts of the literature recommend that the enterprise model be designed to contain only static data. The implication is that the enterprise model would not concern itself with dynamic information issues. If this is the modelling bias, then stability in a data model should be entirely predictable. The practical reality is that organisations ignore this advice and live with the consequences of changing data requirements. There may even be a parallel between the notion of data definition turnover and staff turnover - zero turnover may be worse than high turnover. For a given enterprise there may be an optimum rate of change for data models which balances off the need to respond to changes in the business environment while respecting the organisation's need for continuity.

If an organisation really wanted data model stability, it could build the data model and implement it with an extremely inflexible database management system that could only be changed with the greatest effort. Flexibility in the tools one uses has a significant impact on the changes which one is prepared to consider undertaking. For example, buildings are stable because the cost of changing them is so great. How "stable" would a building be if it were as easy to re-configure it as say a structure made out of LEGO bricks, with all of the electrical, mechanical, telecommunication, plumbing infrastructure plug-compatible? I expect that buildings would change frequently if this were possible.

Thus we are led to the general question of evolvability, and the question of consciously managing change in the data models appropriately to preserve meaning over time, which may be much more important than aiming for data model stability. Data modelling theory and practice should contribute to the effective management of change, not the creation of data structure inflexibility. Working towards functionally appropriate stability must be the objective, not maximum stability.

There may be some things to improve the likelihood of this appropriate stability. Given the evidence of the seven models analysed above, the most frequent contributor to model change was changing user requirements. It was not clear from the analysis what part of these changes might have been anticipated in advance. One would think that the focus would have to be on the clarification of the business issues, and the articulation of current and future data requirements. Traditional ER modelling does not specifically address this question. There are two non-traditional approaches which may contribute something to this discussion. First would be the NORMA analysis approach (Backhouse, 1991). The second would be the semantic normal form approach set out by Stamper (1985). Making the assumption that the stability measurement process outlined above can be improved to track changes which might have been anticipated versus those which are imposed by external demands, these two non-traditional approaches deserve to be tested to see if they reduce the incidence of preventable change.

CHAPTER 8

DATA MODELLING THEORY, PRACTICE, AND PROSPECTS

Data modelling and process modelling are two of the most important techniques in systems analysis. This research has generated results relevant to data modelling issues which go well beyond the narrow concerns of stability measurement. The general objective of this chapter is to explore some of these broader questions. Specifically, Chapter 8 discusses:

- an example from the stability analysis cases of how data modelling is actually practised;
- the lessons inherent in this example; and,
- semantics, hermeneutics, and data model meaning.

A. DATA MODELLING - AN EXAMPLE OF HOW IT IS PRACTISED

We have seen by the survey on data modelling practices discussed in Chapter 3 that the information systems community does not model data in ways which are consistent with the theoretical requirements. Researchers in other areas, such as Galliers (1986), have noted similar kinds of results. Galliers reported that approximately 10% of organisations surveyed indicated any relationship between information systems planning and the business objectives. Yet the vast majority of survey respondents report that they use varying forms of data modelling methods, even though the articulation of requisite business objectives is absent. If data modelling theory is weak to the extent where it is not practised in a way consistent with the theory, why is it used at all?

There might be a number of answers to this question. Perhaps it is used simply because there is nothing better available. Data modelling does provide some kind of structure to a process which can be difficult and frustrating at the best of times. More importantly, data modelling is a process which serves the needs of the database administration group in that it is directed towards the specific problem of record design.

While the review of the literature has been helpful in demonstrating some of the problems of data modelling which contribute to problems in schema instability, and more importantly in schema interpretation, the case study provides an example of the semantic difficulties which emerge in the creation of any data model. There are some useful lessons about the practice of data modelling if we return to the case study and carefully examine the problems inherent in creating just one element of the model. Let us turn back to the example presented in Chapter 5, and consider the question of what a building is. The question emerged because of the normalisation and technical redevelopment of the original system.

The user organisation owned buildings, wanted to keep track of them, and leased out the space to internal clients and paying customers. Rental practices had been well established in the community for a long time, so there was no major problem defining what a lease was and how the business used the idea. As for a building, it would appear to be intuitively clear and obvious what a building was. The analyst's initial assessment of the situation was that the basic data element building blocks and categories were straightforward, and that the data modelling project would focus primarily on questions of collecting and managing data, not on defining and clarifying it.

Recalling the case study, the user community was a relatively large company employing over 10,000 people, in the telecommunication industry in Canada. Telephone companies in Canada are organised on a regional basis for the most part, for geographic and political reasons which are not directly relevant to the study. The specific user of record was the real estate department. It had three major areas of responsibility: 1) real estate planning, allocation, acquisition and disposal, 2) building design and construction, and 3) facilities management, by way of property maintenance, security, cleaning, utility provisioning, and other services.

The organisation had been using a mainframe based data processing system which was beginning to show its age. The questionable quality of the data and the clumsiness of the system in combination with the apparent importance of the application led the organisation to conclude that this system needed to be replaced quickly. The global objectives of the replacement system were to track the buildings which the real estate department owned or leased and to record who occupied the space in the buildings for what purpose.

The modelling process began using the classical "top-down" approach of establishing the business policies operating in the environment, through a series of interviews with the senior members of the management team. The obvious question first question was "how many buildings do you have?" The user did not know. When pressed for an estimate, he finally conceded that the number would be somewhere between 1 000 and 1 100. A thousand buildings seemed like a huge number at first blush. He quickly explained that the real estate department was responsible for all of the buildings which housed equipment and / or staff throughout a geographic area slightly smaller than the combined areas of England and France.

The fact that the organisation could narrow its estimate only to within 100 buildings was also surprising. On the face of it, the idea of misplacing a building seemed farfetched. The idea of misplacing 100 buildings seemed unbelievable. As it turned out, misplacing a building was not difficult and happened with some frequency for good reason to be discussed later.

Intuitively, the team began clarifying the idea of "building" by what the linguists such as Aitchison (1987) would classify as the prototype approach. This was done quite informally and without the specific intent of developing a prototypical structure. It was done along the lines of listing the most important and most obvious examples of "building" of concern to the users and the team thus began to define the term from there. They were sitting in a multi-story office complex at the time; it was obviously a building. Another good example of a building was the major switching equipment buildings which the company owned. This too seemed straightforward.

As the team thinking moved away from the prototypical building, the discussion began to get more complicated. The organisation owned a large number of relatively small portable equipment offices. Built using the technology of a mobile home, these buildings are often mounted on skids and were transported by truck to remote sites where they form a significant part of the distributed network of telecommunication switches. While most people probably think that the notion of building necessarily implies the existence of a foundation, the description of these portable premises as "buildings" seemed to be reasonable. Each of these structures was portable, but most of the time the organisation left it in one location for long periods of time, often measured in decades. As most experienced systems analysts would recognise, the key words in this sentence were "usually" and "often". These key words were a signal to look for exceptions to the stated practice.

The question was posed: If it were mounted on wheels, instead of skids, would it still qualify as "building"? As far as the user was concerned, it did. Furthermore, this portability accounted for the fact that buildings were misplaced. For example, it was relatively easy to arrange the move of one of the portable buildings from the storage site to one of the remote locations. If someone did not record that fact, the building was easily "misplaced" in the sense that the system did not know where it was. This circumstance raised another notion entirely foreign to a common sense understanding of "building", that of building-in-storage. Once one has made the conceptual shift from "building" as a permanently-fixed-structure-in-one-location to portable structure, one will find it difficult to deny the idea of building-in-storage, especially when presented with a large storage compound populated with over 20 buildings awaiting further disposition.

The question of the portable buildings posed questions that needed further, more thorough thinking at a later point, so the team proceeded to explore the list of other buildings that the real estate department was involved in. The organisation owned residences, which they rented at reduced rates to employees in remote locations. The real estate department wanted to keep track of these, too. Calling up the image of a company-owned residence in a remote location, most Canadians would imagine a home in a rural setting with a garage to park the building service vehicle. Would the garage count as a separate building, to be recorded as distinct from the residence? Did it matter whether the garage was attached to the house?

Developing a system to track building data required some kind of collective understanding of what "building" might mean. The systems literature has some articles which talk about the general techniques of semantic analysis. However, there is little in the way of tested, formal tools or structured methods which might provide specific direction on how this should proceed.

1. An Initial Definition of Building Was Prepared

If one wants to determine what a given word means, one often begins by consulting a dictionary. The Webster's New World Dictionary College Edition (Guralnik and Friend, 1966, 191) provided the following entry for building: "anything that is built, as a house, factory, etc.; structure." This rather general definition is interesting from a number of perspectives. First, note that the lexicographer has provided two exemplars or prototypes i.e. house and factory, much in the same way the user did when trying to define "building". The dictionary went on to provide a brief distinction between the idea of building and structure as follows:

> "**building** is the general term applied to a fixed construction with walls and a roof, as a house, factory, institution, etc...
>
> **structure** also suggests an imposing building, but has special application when the manner or material of construction is being stressed (a steel *structure*);" [bold and italics in the original].

This entry is not really very helpful. The definition and the following clarification suggests that some of the criteria for classifying a building are: 1) fixed construction; 2) walls, and; 3) a roof. The idea of fixed construction would exclude a tent, but might include a truck. "Walls and a roof" is a small benefit, since it directs our attention to the possibility of a structure with a roof but no walls, such as semi-protected storage shed or parking facility.

As it turns out, a dictionary is not a good place to start looking for the meaning of anything. This is probably not obvious to the average systems person. But Chomsky (1988, 27-28) offers the following instructive opinion:

> Anyone who has attempted to define a word precisely knows that this is an extremely difficult matter, involving intricate and complex properties. Ordinary definitions in monolingual or bilingual dictionaries do not even come close to characterizing the meaning of the word nor need they do so, because the dictionary maker can assume that the user of the dictionary already possesses the linguistic competence incorporated within the language faculty of the mind / brain.

Of course, defining the words to populate a data dictionary is exactly the task of data modelling. To paraphrase Jardine (1985, 73), using a data dictionary as the foundation for organisational understanding might conquer syntax, but true meaning would still be terra incognita.

There were many questions which came out of the discussion of building ontology. As an interim step, the following ideas were agreed. A building is a structure with walls and a roof which is intended to be left at one location relatively permanently. Roofed structures without walls are specifically excluded. Residences (houses) were included, but the garages to these houses were excluded. Parking garages for vehicles belonging to the organisation were included. Buildings did not require a foundation, a basement, or a permanent location to be included. They could also be moved, and put into storage.

With this tentative definition in hand, a list of the attributes of likely interest was developed. For example, it seemed intuitively obvious that buildings had numbered floors, and occupied spaces about which the user would want to store data. The American National Standards Institute definitions of relevant building terms, such as usable area, rentable space, and construction area helped somewhat. These standards are generally known as the Building Owners and Managers Association (BOMA) standards. The analyst then began to apply these notional constructs to the buildings as the user understood them to exist.

2. Testing the Idea of Building, and Its Attributes

One of the major advantages of an idea like "building" is that it is a tangible entity, that you can actually inspect while trying to define the term. There are many other business entities which are more conceptually abstract and therefore rather more difficult to come to terms with, so to speak. For example, "contract", "employee", "inventory", "facility", "productivity", "performance", "attainment" each has sticky abstract qualities which challenge the systems analyst.

The first building looked at in testing the inclusion criteria and the list of related entities and attributes was a large commercial office tower. The building was one of three separate structures joined underground by a series of wide passage ways large enough to accommodate a moderately sized shopping mall. The question which immediately arose - "where does one building stop and the next one start?" This question was never really resolved, except in an entirely arbitrary way.

A brief tour of the building raised a number of other questions.

- How should a covered area such as a patio be treated? Porch? Underground parking facility?
- How should an upward extension to the building for the purpose of mechanical gear, such as heating, ventilation, and air conditioning be treated?

- Should the space occupied by a telecommunication transmission tower on the roof be included?
- How do the floors below ground level get numbered?
- How is the space on a given floor measured? To the inside of the glass? To the inside of the interior wall? To the exterior of the outside construction including any protruding construction members?
- How is space used in common by a number of different groups recorded?

There were other trivial semantic issues which are imbedded in the above linguistic structures. For example, the ground floor in North America is understood by convention as the first floor, while in Europe the first floor is the first floor **above** ground level. In older buildings, it was customary to exclude the number 13 from the naming of the floors for reasons of enduring superstition. Some buildings have a thirteenth floor and some do not. Most people are not conscious of the fact that floors use the number as a name and not really in the sense of a quantifier. It is not possible to say with any confidence for buildings as a class that the number of the top floor is the number of floors in the building. While this may seem to be a silly point, one of the first information demands from the subsequent automated system was a list of all buildings over a given number of floors.

There were other complications. The data reconciliation team responsible for reviewing the data being collected on buildings tried to come to grips with a drawing like the one in Figure 5 below. The structure in question was constructed in three distinct stages. The part labelled Section A was constructed to house equipment in 1958. In 1969, the eastern wall was taken down and the structure was enlarged to include Section B. In 1985, a further addition was made to accommodate an administrative group with a door between the final addition and Section B.

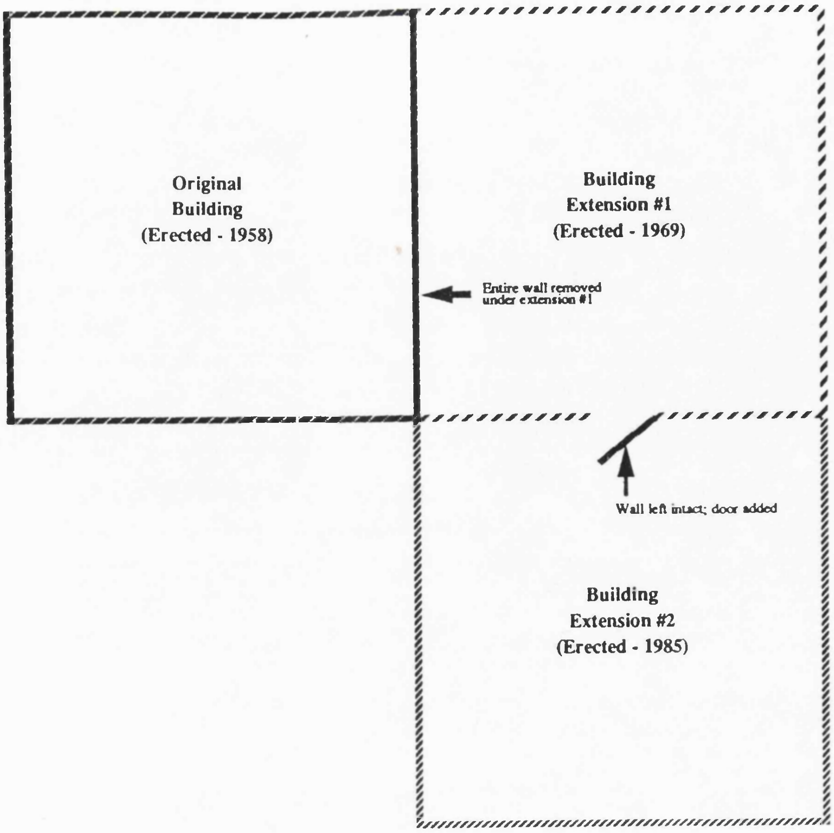


Figure 5

When the accounting group was contacted to get an idea of their perspective, they indicated that their rule was to depreciate buildings over the expected life time of 50 years, and as far as they were concerned, the drawing represented three separate buildings. When the planners responsible for assigning space to people were asked, they viewed the above diagram as one building. The door access and adjacency situation makes it entirely reasonable to conceive this building as one contiguous space. The architect looked at the plan and offered the opinion that according to the local building code, this particular structural configuration would probably be classed as two buildings. The question remained: did the drawing represent one building, two buildings, or three buildings? The unequivocal answer - yes.

Perhaps the answer to this data modelling dilemma was the possibility that the idea of "building" is too abstract, and not specific enough. Would the model be more successful and less confused if we were to move down the abstraction hierarchy and keep track of information at the level of the individual floor? As it turns out, there are numerous cases of semantic judgments and interpretations which were necessary when looking at individual floors. For example, how should mezzanine floors be treated? Crawl spaces between floors used for storage or equipment? Floors below ground level? Floors below ground level which extend beyond the external walls of the building, whatever a building might be? Similar problems of data definition and judgment exist when collecting data about individual spaces.

All of these difficulties were identified within the real estate department. Of course, the real estate department is not the only part of the organisation which was interested in building related data. Real estate resources accounted for something in the order of £200 - 250 millions (CAN $400-500 millions Canadian) of long term assets on the balance sheet. The organisation beyond the real estate department depended on information about building and other asset information to generate revenues. The nature of long distance services in Canada often means that a call will originate in one jurisdiction, and terminate in another. In the past, there had been a question about how the revenue from such a call should be shared. A national organisation called Telecom Canada was created with representatives of all of the telephone companies. They have established complex rules to share revenues, but the revenues are calculated net of applied expenses. Therefore each jurisdiction must report on the revenues collected and costs incurred generating the revenues.

These conventions can work only where each jurisdiction uses the same rules for identifying costs and revenues. To achieve the goal of consistent practice across the country, a national common language bureau has been established. In this way, the telecommunication companies minimise variations in language use which might effect the sharing of net revenues. Thus, for telecommunication purposes in Canada, there already existed a standard definition for "building". Unfortunately, it includes Environmentally Controlled Manholes (ECM), among other things. An environmentally controlled manhole is typically a concrete structure constructed underground and accessed from the top by a manhole cover. The structure might contain switching and communication equipment, and might be heated.

A definition of building which includes ECM's poses a special problem for the real estate department which has no responsibility for, or interest in, these structures. This introduced extra complexity when the discussion about the number of buildings took place outside the real estate department. In order to answer what seems to be a simple question - "how many buildings does the organisation have?" - we must ask a number of qualifying questions:

- Do you mean buildings owned or building leased from someone else, or both?
- Do you mean buildings for people, equipment, shared between people and equipment or all of them?
- Do you mean buildings for the purposes of the real estate function, the depreciation calculation, or for Telecom Canada's need?
- Do you mean buildings as contiguous space, or buildings as defined by the building code?

At the outset, the notion of "building" appeared to be simple and straightforward. In the end, it was complicated and inconsistent across the organisation. In other words, the different linguistic communities in the organisation puts the notion of "building" to different uses.

B. LEARNING FROM THE PROBLEMS OF DATA MODELLING

This single example of one entity out of the case study demonstrates many of the problems inherent with data modelling. This research has demonstrated a simple case where there are distinct linguistic communities within an organisation. These differing communities are able to communicate and work together because they have a general understanding of the context(s) of the other linguistic communities. Where there are disagreements, these are negotiated for clarity and occasionally for agreement. The current theory and practice of data modelling does not take these different linguistic communities into serious consideration. Most of the authors in the literature expect that where differences in usage for a given term exist, a consensus must be negotiated.

The theory ignores the possibility that such linguistic differences might be entirely appropriate and actually functional. In the rest of organisational life, there is no expectations that lawyers and accountants should use language in the same way. The linguist Benjamin Lee Whorf (1956, 246-7) offered these thoughts:

> The term "space", for instance, does not and CANNOT mean the same thing to a psychologist as to a physicist. Even if psychologists should firmly resolve, come hell or high water, to use "space" only with the physicist's meaning, they could not do so...
>
> Now this does not simply breed confusions of mere detail that an expert translator could perhaps resolve. It does something much more perplexing. Every language and every well-knit technical sublanguage incorporates certain points of view and certain patterned resistances to widely divergent points of view.

The usual theoretical database answer to this is to approach the record design from the perspective of a meaning-makers construction kit. The database schema permits different "views" of the data. These different views are constructed by presenting different subsets of the schema attributes for a given entity to different users. The inadequacy of this approach is manifold. First, it does not permit inconsistency where such inconsistency is appropriate and useful, as is the case with the building example. Second, it implies that meaning is created from a database through the relation of attributes that constitutes the record design of a given entity type. As we have discussed in Chapter 7, the data model and its subsequent schema provide little of the context and semantics of the situation to help the end user in understanding the meaning implied by the designers and creators of the data. The context is missing as it relates to the entity and attribute affordances which are not part of the data model or the schema. If the coherence of ideas as as important as Schank and Leake (1986, 336) claimed, presenting views by selecting sub-sets of the attributes is not an adequate way of representing the contex. Typically, the inclusion criteria for entities, attributes, and relationships are not specified. The agency of the data model and the purpose of its sum and its parts are also excluded from the model. Attempts to force a "reconciliation" between related definitions in different data models being used for different purposes might actually contribute to data model instability.

In the survey response of Chapter 3, many of the respondents talked about the difficulty of reconciling data definitions once systems were redeveloped across organisation boundaries. A number of data administrators commented on these difficulties. The current theory of data analysis is that these differences need to be reconciled away, resolved so that the data definitions are consistent across the organisation as a whole. This strategy probably results in the highest level of convenience for the data administration group, but not necessarily for the user community. In the case study, the user community had divergent ideas about "building". For the purposes of the real estate department given their current mandate, environmentally controlled manholes are not buildings. For the purpose of revenue and expense sharing nationally, they are included. The inconsistency is clear, functional, appropriate. As long as the inconsistency is understood, why should it be eliminated, apart from the fact that database management systems and modelling theories do not easily tolerate such an approach? Other authors have written about the need for managing views which conflict even though both may be correct (Wiederhold, 1983, 347). As noted in Chapter 2 of this thesis, the parable which is used to justify the requirement for reconciliation of data definitions is that of the six blind men and the elephant. However, there is little evidence to support the idea that data modellers have a broad enough view of the enterprise to integrate these views, especially given the typical organisation's lack of commitment to business and systems planning.

When it comes to paradox, each of us lives with representations of the world which would be inconsistent when put side by side. For example, a manager might view his organisation variously as a hierarchy, a team, or a group of highly competitive components, depending on the circumstance. These representations are mutually inconsistent for the average model to take into account. Persisting in one or another of these views, to the exclusion of the others, might have serious consequences. The objective here is not to eliminate these paradoxical representations by forcing a manager to choose one of them and live with it. These separate metaphors and models for the organisations serve different and complementary uses even if they are paradoxical.

Forced intra-organisational data and systems integration may have other unintended consequences. In Canada, a relatively large government department was created by the integration of a ministry of mines and minerals, and a ministry of land and forests. The new organisation, now responsible for energy and natural resources brought a number of disparate accounting systems together and created a large mainframe-oriented DBMS-based accounts receivable system. This one system tracked the oil and gas royalty payments due to the Crown as well as individual permits to chop down Christmas trees. Eventually, the wheel of cabinet organisation turned once more, and the politicians elected to separate the mines and minerals from the administration of lands and forests. Unfortunately, the integrated system (data definitions, database, collection procedures, data) which managed the accounting function was so enmeshed, that it was not feasible to separate the accounting data management of these two ministries. As a consequence, the government has had to establish a bridging administrative group which attempts to serve the needs of two masters in two different ministries. In effect, a significantly sized administrative group serves two ministries, as a consequence of systems design choices made eight years earlier.

One of the biases of data modelling specifically and of information engineering generally is the focus of information for "decision making" in the sense of discrete, separate, one-off's instead of action, in the sense of continuous behaviour, such as sending out bills, statements, changing reports of the state of the organisation. Action and behaviour are much broader abstractions, and may be just as important to the enterprise as specific discrete decisions.

The theoretical orientation of the data modelling literature is that formal tools are necessary, and almost sufficient. Why do the theoreticians focus on "provable constructs", predicate logic, mathematical modelling? It may be because these notions are much tidier and easier to manage than linguistic issues. For example, Zadeh (1982, 26) said that an assertion in fuzzy systems theory is not normally a proposition that has a "high degree of truth and, in addition, is informative in relation to a stated question." This bears careful thought, since the implication is that there are degrees of truth just as there are degrees of precision. Entity relation modelling and the subsequent application of predicate logic do not admit to the idea of "degrees of truth". The idea of "informative to the stated question" poses serious problems for database query, since the database and its rules are unable to make a judgment about whether the question and its answer might be relevant to a given issue. This is a human judgment (contrasted with a machine choice), which can be quite sophisticated.

Finally, and most importantly, the notion of presenting different user "views" through different subsets of the entity attributes, has built into it a number of un-examined assumptions.

- There is homogeneity among the existing and potential user communities in their selection of attributes to describe a given entity.
- The affordances of the entities and attributes are consistently shared throughout the community.
- The inclusion criteria and classification processes are predictable and consistent for entities, attributes, and relationships.

Data modelling as a process might be a relatively unbiased way of raising language issues in a given community in order to develop a consensus that might have been absent, thus creating the necessary illusion of common purpose through common language. There is some doubt about how sensitive data analysts are to the idea that the act of modelling or data analysis changes the understanding the user has of his own environment. A data model is a form of linguistic abstraction which emphasises some things, while suppressing others, much like the function of metaphors. It highlights some elements of the organisation, while hiding others. The data model is used as the foundation of the data dictionary. That dictionary might constrain or limit the language an organisation will use in thinking about itself.

Data modelling in combination with process modelling forms a subset of Information Engineering. According to Avison and Fitzgerald (1988), the information engineering methods have four levels: planning, analysis, design, and construction. This division is somewhat artificial since it is clear that the planning phases must include some form of analysis. There is also a significant amount of design related work as early as the planning stage. The entities that one chooses in the planning phase is a kind of design, if by design we mean to make a choice among alternatives that improves the likelihood of success.

Data planning, analysis, design and construction necessarily require some form of semantic analysis, whether intuitively done as in the case of the building example, or whether done using more formal tools and processes of semantic analysis. We must turn our attention to the questions of semantic analysis and its contribution to data modelling.

C. HERMENEUTICS AND SEMANTIC ANALYSIS - COMING TO THE MEANING OF THINGS

One of the most important assumptions underlying data modelling theory is that meaning is captured and stored in words or data elements. In marked contrast, this research began with the question of how users interpret data models, not what meaning might be "captured" in them.

At the beginning of the development of a method to measure data model stability, the question was asked: what in a data model contributes to how a user might infer meaning from the evidence presented by the data model structures and its associated data content? In other words, what are the real hermeneutic elements of a database? The database schema has certain structural characteristics which could be used to deduce the meaning of a given database and the data within it. For the effective, "correct", interpretation of an information system to occur, one would expect that the data and the structural characteristics of its presentation would affect the interpretation which might be made of the model. A number of elements of a schema might be valuable to the end user of a system in his attempts to make sense of the system at all levels, including its inputs, outputs, and the process in between. The next section of this thesis will review how the evidence supports the meaning primitives which were predicted to be important to the schema.

1. Hermeneutics - The Process of Interpretation

Firstly, the stability measurement method tracked the names of the individual attributes based on the thinking that the name of the attribute contributes to the interpretation which can be made of the data stored under that category. In the case study, the evidence is that while the attribute name clearly plays some function, there is substantial variation in the specific name that users will apply to a given category. For example, the business community used a variety of expressions for the supervisor responsible for managing the maintenance of a given property. Sometimes he was referred to as the "supervisor", the "properties supervisor", the "foreman", the "properties first level". The original data model called this data attribute "F/M". Occasionally, the data dictionary actually gets the name of the attribute wrong as in the case of "measurement of quality" which is understood by the user community actually to refer to "quality of measurement", which is an entirely different idea. From this evidence we can conclude that the name of an attribute is of some consequence to its interpretation, but users may tolerate significant variations in language usage for a specific attribute. This may in part account for Kent's (1983, 9) observation that "in general practice, there is no systematic discipline for the naming of fields and records." Having said this, the choice of entity name can make a difference to the ease and accuracy of data interpretation.

The second major factor in interpreting a given data model and its data is the range and domain of the data element. The evidence from the data models examined in this research was that the range and domain were consequential for interpretation, but the situation is complicated by a number of factors. One of the problems in domain code interpretation is that the criteria for assigning a particular classification appears rarely to be documented. In the case of *marital status* for example, what criteria is applied to distinguish between those employees who might be co-habiting with a person of the opposite sex, as opposed to those who have established a "common-law" marriage, complete with contract documentation? In the case of the *space use code* in the real estate example, the coding structure of the database allows for "parking" on the one hand, and "garage" on the other. Nothing in the documentation, in the database, or in discussions with the user community led to clarification of how these categories are discriminated. Most of the time, classification rules appear to be left to the native language skills of the individual who codes and interprets the information. When the obvious classification rules break down, the users can ask to extend the classification system to account for a newly found discrepancy, or they can modify their own understanding or interpretation of the classification criteria inherent in the existing coding scheme.

In the above research, we have seen examples where the classification system implicit in a range of codes breaks down. One of the ways in which such breakdowns can be prevented in structuring coding conventions is to consider whether it might be possible to use more than one of the proposed codes at the same time to classify a given circumstance as was demonstrated in the case study example of the *space use code* which originally included occupancy information as well. Sometimes there are cases where similar, or apparently synonymous terms actually refer to distinctly different ideas.

For example, you might expect that the classification of a space as vacant might mean just about the same as classifying it as surplus. However, there might be an important distinction between these two descriptions. "Surplus" might be used to describe the portion of a space in use which is surplus to the current requirements of the organisation given the number of people who occupy the space, and given the space standards which the organisation has established for those staff. A space could also be classified as vacant but not surplus, in the case where the organisation has a specific plan for using the space. In this situation, the attributes "surplus" and "vacant" could not be part of the coding domain of one attribute. Two separate attributes would have to be created. Thus we can make a suggestion for data modellers to improve the techniques of attribute code development: test the possibility that an entity might be fairly described by two separate codes within the same attribute. If so, the attribute should probably be split.

The role which edit rules play in contributing to data hermeneutics, whether these edit rules were specified in the software, or whether they were informal was considered. Users for the most part do not know what the formal editing rules for each of the attributes are. Sophisticated integrity constraints are difficult to design, and can be expensive from a technology basis (Wong, 1982, 242) and therefore have to be traded off against other considerations, such as system performance. The constraints which are applied informally by end users appear to be more pragmatic, and common sense oriented. For example, in the case of the area which a floor occupies, the user community did not know, or care, what the system constraints for these data might be. From experience, they "knew", or expected, that floor size was a function of its geographical area, i.e. that one does not expect large buildings in small towns. They "know", or expected, that no floor would be greater than 3,000 square metres. When questioned about the issue of expectations, most users describe complex rules of thumb which condition their expectations about the data which they would find associated with a given data element. Where such expectations were not met, an individual might question a co-worker for clarification, that is to say, turn to someone else to negotiate the meaning of this computer mediated data. This process probably varies according to the curiosity, competence, and experience of the individual involved.

A second example, a newly trained nosologist (one who classifies diseases, typically to record the cause of death for registrars of vital events) might question a death certificate which listed the cause of death in a man as cancer of the breast. In the case of a man under the treatment of oestrogen, such a development might be possible. It is also possible in the case of a normal male, in rare circumstances. Here, the edit criteria would be less important than the negotiated meaning norms. Therefore, I conclude from this research that edit criteria contribute little to meaning interpretation, and hence are of limited use in monitoring data model stability, since they appear to have extremely limited use in applied data hermeneutics.

As noted in Chapter 7, the question of the entity inclusion criteria is also an important one. These criteria are important in data collection and in data model interpretation. What something is understood to be is fundamentally bound up with these criteria, whether we believe reality is objective or subjective. Yet, none of the data modelling methods which analysts report using makes any explicit reference to the entity-attribute-relationship inclusion criteria. None of the models which were examined in this research had a clear and unequivocal statement of the inclusion criteria, and in a number of cases, these criteria were the subject of ongoing conflict, especially where there were attempts to reconcile differences across organisational units. This problem was commonly reported in the survey phase of this research. DBMS products typically have no specific provision for documenting the inclusion criteria or for coming to terms with enforcing these criteria at any of the entity, attribute, or relationship levels. Most of the entities that have been carefully examined in this research have significant inclusion criteria which are not schema attributes. For example, the real estate personnel had a rule-of-thumb definition of building that went something like this: a building is any structure that has four walls, a roof, and a door large enough for a person to get through. None of the building schemas keep track of whether the building actually has a door that is large enough to get through. In a sense, recording these facts would be superfluous, literally by definition. It is assumed if the data has been put in the database, the object must have satisfied the criteria.

Note that some modelling theorists like Flavin (1981, 58) suggested that "the information content of an object type can be identified with the set of its attributes". He suggested that the set of attributes which are associated with a given object type (i.e. entity, or relationship, when data is being stored about the relationship), constitutes the specification of the object. There was no recognition that the inclusion criteria separate from entity attributes play any role in the object specification and subsequent user interpretation. Keesing (1981, 83) said "the intersection of features seems far too simple a model to deal with the human ability to recognise patterns." This "intersection of features" sounds exactly like the relational data modelling (i.e. entity attribute analysis) proposed by Flavin (1981).

The relationship data in a data model appeared to contribute little to the end user interpretation of the model. It appeared that the relationship aspect of data models is primarily one of data management and less one of major importance in data interpretation. For the most part, users are intuitively aware of the relationships between entities, but this understanding is not a function of the data model which has been created. Their understanding is more directly related to the natural language assumptions that they make, than through their understanding of the formal relationships which are an explicit part of the database schema.

One thing which the data model cannot communicate is the way that the processing consequences of a data element might influence the understanding a user has of a data element. Data models provide little in the way of cues and clues for database schema and data interpretation, especially from the perspective of the processing consequences. We must once more return to the quote from Mijares and Peebles (1976, 27). "In order to describe a semantic view the user of the relational model must carry the semantics of the relations around in his head." This is especially important when one considers that the purpose of a database often involves an intent to communicate. Communication necessarily involves data hermeneutics and meaning. The problem with data models is that they preserve the context of the database inadequately, and most importantly, it is impossible to negotiate meaning with a database in the event of inevitable misunderstanding, or to negotiate different interpretations.

2. The Prospects for Extending Semantic Analysis and Data Modelling

It is clear from the survey of data modelling practices that organisations make major commitments of effort and resources to develop data models. These models are often based on the legal, social, and commercial fictions which have poorly circumscribed, differentiated referents in many cases. In part, the objective of data modelling is to determine how to exploit these fictions, to create new ones, with the support of information systems, in a way which contributes productively to the organisational progress.

The best data modelling methodologies offer little help in identifying and reconciling data definition differences which emerge in the process. For example, consider relational data design. As has been noted by a number of authors (Kent, 1979; Everest, 1986), relational techniques are mostly about the question of record design whether the sticky problems of data meaning have already been sorted out or not. These techniques are unable to provide consistent rules for selecting entities, attributes or relationships from the problem space. This may be partly to do with the attempt to separate data definitions from processing considerations.

One major complicating factor to data analysis is the data modelling dialectic, the interpenetration of data and processing. It is a myth that data can be independent from processing. We can separate them structurally by using a DBMS. But understanding the complete meaning of a data element requires an understanding of how that data element affects others, how it is processed, and what outputs it influences. How a data element affects the routine and exceptional behaviour of the firm might also change one's understanding of the meaning of a term. The separation of data modelling from processing introduces problems from the perspective of "meaning" since language-in-use is the foundation to many theories of linguistic meaning. The language-in-use principle for the data element of a data model necessarily implies understanding, implicitly or explicitly, the processing use.

During the process of creating a data model, the systems analyst leads the user group though a process of developing a conceptual framework constructed out of a set of abstractions. An abstraction is a way to suppress some details about a circumstance while highlighting those which are "relevant" (Smith and Smith, 1977a, 1977b). There are a number of abstraction techniques which the literature review of Chapter 2 identified, such as generalisation, aggregation, functional differentiation, and characterisation. However, none of the theoretical discussions of these techniques takes into account the practical linguistic realities of differing language sub-communities. These abstraction processes are a matter of design choice, yet he have little in the way of empirical data which might suggest design principles to guide the choice in these areas. One of the problems with the idea of "relevance" as it relates to the abstraction process is that the term requires a judgment by someone. Relevance is a complex idea. This process of abstraction invites problems by suppressing details which are irrelevant today, but which might be vital tomorrow.

The theory of data modelling says little about how semantic analysis might apply to these aggregation processes. When we do semantic analysis, we might take the approach of asking our clients to rate exemplars of a given idea according to how well the exemplar represents the concept. In the end, the client must come to the point where he will judge that a particular example is not close enough to the basic meaning which he intends when he uses a given word to be included into the category. This is a way of using the extremes to test the classification structures. In the building example above, we can continue to ask whether specific examples qualify until we get a distinction on what passes the category inclusion criteria and what does not: office, warehouse, residence, garage facility might all pass the test. Garage associated with a residence, tent, and truck might all fail. Basically, this is akin to the process of measurement where we will measure to the degree of precision which is useful and not much further. There is an infinite degree of precision which is possible, but there is also a real threshold beyond which nobody cares. Likewise there is an infinite degree of semantic analysis possible, but eventually we will get to the point of what the lawyers call "distinctions without differences." As Siu (1957, 61) expressed it, "it is always a question of how much error one can afford in the circumstance."

We identified an important implication in the case study presented above, that among English speakers in a given organisation, there are fundamentally different linguistic communities: lawyers, accountants, real estate negotiators, facility managers, architects, personnel managers. While it may appear that they are using the same vocabulary, careful assessment of the intent and meaning of this vocabulary may demonstrate fundamental differences of understanding. This is not just a question of a difference in view, as the data modellers might say, but in fundamentally different understanding.

Having said this, it seems intuitively clear that there has to be some degree of commonality of meaning among the shared terms in a given organisation. As Brown (1991, 134) indicated, management needs terms which are "common and relatively invariant across all of the company's businesses." The key word here is "relatively", a notion which is poorly tolerated by current database administration techniques. In fact Brown himself went on to say that uniformity can be taken too far. "Different businesses with different strategies require different information." A relatively inflexible tool like a data dictionary may be part of the problem in a business, not part of the solution.

Much of the literature recognises the importance of front end analysis, prior to overt data modelling to clarify these differing information needs. Some authors such as Flavin (1981, 8) referred to it as "policy research and analysis" which is "basic to the modelling process". Unfortunately, how such policy research and analysis is to be conducted was not presented. This analysis is in part the search for classification structures which will serve the needs of the organisation. However, most classification systems eventually break down, as is demonstrated by the debate about what a building is. The classification problem is everywhere. Bloor (1978) noted in his paper exploring language and mathematics, "the redrawing of classificatory boundaries is an integral part of mathematical reasoning". The key word in this sentence is the word "redrawing"; that is, we should not expect mathematical boundaries as they exist at one point to be fixed forever. Actually, it is the responsibility of the mathematician to test the boundaries of the classifications. We could extend the notion of boundary exploration to include business and administrative reasoning, as well. Perhaps we should be taking a pragmatic view of classification systems as a whole and focus on the behaviours which they enable or afford. The focus on the classification system would then be on whether it is useful, not whether it is true.

The rigid application of a classification system can result in some bizarre practices, as was evidenced by the British Post Office's differentiation between a package and a packet i.e. a packet is wrapped in paper and tied with string, presumably so it can be opened for inspection. However the definition of string is somewhat more haphazard, given the packet/package which arrived at our door with the string (and the bow where it was "tied" together) neatly drawn with a crayon.

As Leech (1981, 26) pointed out, conceptual boundaries often vary from language to language in a way "which defies principled explanation." Thus classification and establishment of boundaries can only be done on a case by case basis, respecting the norms of the varying communities. We might consider a question which comes out of this: are cultural or linguistic norms invariant, or even stable over time? One could consider the impact of shifting cultural norms, and their impact on data modelling stability. Leech wrote that the motivation for classifications is supplied by cultural norms, rather than by external reality. Perhaps we should be teaching data analysis as a form of organisational anthropology.

Given that all data analysis in the end uses language, we must take current linguistic research into consideration when we undertake this kind of analysis. For example, Armstrong, Gleitman, and Gleitman (1983) published an article entitled "What some concepts might not be". They documented the results of disturbing research into how language is used by people in quite surprising ways. They discussed how English speaking subjects willingly classify feminine nouns, such as mother, sister, aunt, waitress according to their degree of "femaleness". Respondents consistently classified "mother" as being more female than "comedienne". These researchers made similar discoveries regarding odd and even numbers. Most respondents assessed the number 4 as being more even than the number 18, although this formulation is mathematically nonsensical. Data modelling theory makes no allowances for these natural linguistic behaviours. Might the idea of "degree of ..." have some application in an accounts receivable application? Are some accounts more "receivable" than others? This is certainly true for those involved in the accounts collection process.

One of the fundamentals of data analysis that deserves to be emphasised is that semantic analysis always precedes the mathematisation of the record design. This semantic analysis is in part a process to explore and establish boundaries for classification. One of the first lessons is that semantic analysis is a fundamental part of all systems analysis, whether it is done intuitively as presented in the above example, or in a more organised and formal fashion. The process of exploring the data structures most suitable for an organisation is often complex, even when it appears to be straight forward.

What principles of semantic analysis might be applied in our work as data analysts? As one of its primary goals, semantic analysis has the objective of establishing boundaries. As described above, this is about what something is and what it is not, i.e. its ontology. One of the ways of addressing this is to consider the entity's life cycle. What was it, before it became what it is? Before something is a building, what is it? A construction project? And what is the transition rule which changes a construction project into a building? An inspection report? What is it after it is a building? Would this be a demolition project? A sale of the building? An accidental destruction? The time element in the classification process e.g. construction management processes may need to classify something as a "building" project before the rest of the organisation is prepared to agree to this, may account for some of the discrepancies in definition which occur across organisational boundaries. The time element is clearly a significant issue for data analysis.

Data analysis should probably owe more to the field of linguistics than it does to mathematics. Unfortunately, data analysis fails to take into account many of the findings of the linguistic community. For example, Barnwell (1980, 37) wrote about techniques for translating from one natural language to another. She referred to the technique of contrast, that is by comparing a word in each of its senses with other words within the general area of meaning. This suggests the idea that the words one does not use in a given context may be as important as the words one does use. De Saussure (1959) talked to this exact point when he said that the meaning that a hearer creates from a particular word is often inferred from all of the words which were not used, but might have been used in the context. For example, imagine you are introduced to a couple: "This is Jack and his friend Jill". You will likely unconsciously infer that Jack and Jill are not married from the use of the word "friend". This is not because husbands and wives are typically unfriendly toward each other, but because the person doing the introduction could have used the word "wife" but did not. Similarly, the use of the word "office" instead of the word "building" is a sign to the hearer which delimits or shapes the interpretations which will be made.

We noted in the above example of clarifying what a building was, how a dictionary was of limited benefit. If the dictionary was not much help, one could turn to a thesaurus for help. The thesaurus should be able to provide a list of related words which will vary from the notion of "building" in ways which might help us to find a more satisfactory description of the very idea of "building" itself. If anything, a thesaurus might help us to decide what a building is **not**. What something is not is perhaps as important to understanding an idea, as what it is.

Returning to the example of the building entity, what does the New Roget's Thesaurus (Lewis, 53) have to say about "building"?

> **BUILDING**--N. **building,** structure, edifice, pile, skyscraper, office building, arcade, rotunda, mausoleum, shed, lean-to; public building, hall, palace, capitol, casino, castle, chateau, alcazar; annex, extension, wing, superstructure, dome, cupola; aerie.
>
> **tower,** steeple, church tower, bell tower, belfry, beacon, donjon, turret.
>
> **porch,** patio, piazza, terrace, lanal, veranda, gallery, loggia, stoop, portico, porte-cochere. [emphasis in the original].

This list contains many entries which have low relevancy to the topic of concern to the application. On the other hand, it did point to some areas which required further discussion. Shed, lean-to, annex, extension, wing, superstructure, dome, porch, patio, terrace are all ideas related to building which were useful in clarify the systems requirements of the building system. For those developers working on the idea of the data repository, perhaps they should consider adding a data thesaurus as a parallel reference to the data dictionary. The data thesaurus might help to clarify the differences and distinctions among the various elements of the data dictionary, especially in known areas of confusion.

Barnwell (1980, 173) wrote that it is necessary to think below the semantic surface constantly, to clarify the semantic relations more precisely. How might this be done? One way is to take a thesaurus approach to data analysis rather than the usual dictionary practice. Another way might be the use of the semantic normal form (Stamper, 1979) which takes an approach more directed at knowledge elicitation than it is at DBMS record design. A third way would be to use the semantic analysis approach of NORMA (Backhouse, 1991) which goes some distance to dealing with a number of the modelling problems noted above. Finally, further thought is necessary on a theory of meaning which focuses on the how data changes the action of the organisation, as well as how data might shape the attitudes of an organisation.

One aspect to language and meaning which semantic analysis must confront is the question of variable semantic determination. Osgood, Suci and Tannenbaum (1957, 321) observed in their research on the question of how to measure meaning "our data are replete with cases where individuals differ in their semantic differential profiles for the same sign-vehicles." Individuals will have widely varying understandings of the terms that they use in the course of everyday business. On the one hand, a certified accountant will have a much more highly detailed semantic determination of an accounts payable record, how an individual accounts payable record relates to the accounts payable ledger, what a liability is and how that relates to accounts payable, the differences between a "liability" and "accrued liability" and a "contingent liability" and their individual relationship to the accounts payable ledger, and other elements of the accounting systems. On the other hand, a junior accounting clerk would likely have a much more limited understanding of the idea of accounts payable. Notwithstanding the limited understanding, the clerk might still be able to use the term quite appropriately and effectively in some contexts.

This would not be a surprise to linguists who have long recognised the difference between language performance (the way language is used) and competence (the internalised set of rules which govern the production of language) (e.g. Aitchison, 1987). This poses something of a problem for the average systems analyst who is looking for someone to articulate the set of rules for an organisation, when many of the individuals may only be in a position to explain how they behave, without a complete idea of the consequences of their behaviour.

If we want to be clear in our data analysis, we might consider the following tactics.

- Explore a thesaurus and less importantly a dictionary, to help identify what it might be and what it might not be.
- Consider the alternative linguistic formulations possible for the environment, and the consequences for the user community, of choosing one in favour of another.
- Determine in a practical way the relationship between the intended meanings in the user community and the likely interpreted meaning (Leech, 1981, 21).
- Analyse the entity life cycle, in order to be clear about the transition points between entities (e.g. when and how does a proposal become a contract, when and how does a contract result in a liability, when does a liability result in a payable, when is the contract considered complete).
- Create a list of potential and actual affordances as a way of beginning to document the purpose(s) of each of the attributes and entities.
- Clarify and document who the agent of the classifications is and under what authority.
- Clarify the ontological antecedent to each of the entities.
- Explore the inclusion criteria for each of the relationships, the attribute codes, and especially for the entities.

- Explore the assumptions of relevance, potential transfers of authority or power, unanticipated consequences, and sources of responsibility for action (Winograd and Flores, 1986).
- Document where such inclusion criteria and definitions might be inconsistent with similar entities, attributes, and relationships in other, related models.

Functional dependency theory should also help to avoid some forms of semantic confusion, such as that of the space user code case. The problem with functional dependency analysis is that one of these errors is a lot more obvious after it has been implemented than before.

In summary, the theory of data analysis is heavily influenced by the set theoretic schools of thought. These approaches benefit little from the results of language research put forward by researchers in the linguistics, who have documented many of the paradoxes of the relationship of language and meaning. The actual practice of data modelling by information systems professionals does not reflect the theoretical requirements of data modelling, however weak these may be, but follows the data modelling methods only generally. The linguistic issues are clearly absent from both the theory and practise of data modelling. In the absence of a coherent integration of linguistic research findings into the practice of modelling, there are a number of short term tactics available to the analyst which might help to limit the number of changes a model is subjected to.

In the end, however, the practical demands of the organisational environment will require changes to data structures as the business itself evolves. These shifting requirements may be driven by changing relationships with customers, supplier, employees, or government agencies. The objective for data modelling cannot be to maintain a stable data model, but a data model which is flexible in responding to these demands, and which preserves the maximum degree of meaningful data over time.

CHAPTER 9
CONCLUSIONS AND FURTHER RESEARCH CONSIDERATIONS

The research effort which has been the subject of this thesis has focused on questions related to the stability of data modelling. Data modelling is usually applied in the context of the traditional IS view that information is best seen as an organisational resource without ownership boundaries. According to the literature, the best way to achieve this state is through top-down data planning, modelling, and coordinated data structuring. Whether termed data analysis, data design, or information modelling, data modelling practitioners and theoreticians have made a number of claims about the efficacy and efficiency of their diverse approaches to the topic. One of the most common claims is that of schema stability. The question of schema stability is quite important since the rationale for the conceptual separation of data from processing is ostensibly because data requirements are fundamentally more stable than processing requirements. This claim is made notwithstanding the fact that there is no generally accepted definition of stability and, prior to this research, no method to measure degrees of stability.

A. GENERAL SUMMARY OF THE RESEARCH

The research project began with a review of the foundations to data modelling, and the theoretical weaknesses of these formulations generally. Recognising that there were widely varying ways in which user information requirements were modelled, we undertook a survey of data modelling practices in Canada, with a view to determining what the state of the art was, and what practices were most common. This survey was also designed to determine if there were any existing methods to monitor database schema stability which were used commercially. The survey reported a number of interesting aspects of data modelling and the problems associated with trying to reconcile data definitions across organisational boundaries. The survey also demonstrated unequivocally that the information systems community does not practice data modelling in a way which is highly consistent with the theory.

The research identified two serious impediments to developing this stability measure: 1) the literature is unspecific about the requirements for each of the usual phases of data model development (conceptual, logical, and physical); and, 2) the literature documents many different techniques and approaches to building data models. However, this research developed a method whereby the major elements of a data model can be consistently represented, whatever process was originally used in the modelling process. This was achieved through the concept of reconstructing a logical relational schema from the record design. The reconstruction process attempted to identify the primary meaning primitives of a database and its data model in order to track changes to them in different iterations of the model.

The most common data modelling tools in use are variations of the entity-relation or entity-attribute-relationship forms, in combination with data flow diagramming. However, there is a high degree of variation in the use and application of these tools. The research had to resolve the question of how to monitor changes in data models from one version to another when each version might have been created by a different modelling tool.

The question of database hermeneutics was also considered. What part of the database schema is useful to the end user of an application? This research has made an initial attempt to answer this question in developing a method to track changes from one model to another. Having identified the various elements that had a likely impact on the end user's capability to interpret a data model correctly, we assumed that it was proper to consider these elements as a legitimate part of what constitutes a model, and were therefore elements which might change. Changes in these characteristics of the data model might change the way in which the model was interpreted.

The original measurement protocol was reviewed by experienced systems professionals who offered suggestions and criticisms for the process. Based on this feedback, the method was revised, and the technique for gathering the data applied in detail to an initial series of test data models. The case study series of data models was examined carefully in order to identify problems with the method itself, and to demonstrate some of the likely reasons data models change over time. A larger group of models were examined and the results analysed, in order to provide a more global assessment of the tool in use, and to provide a preliminary assessment of data model stability. This data also provided data to support a critique of the measurement protocol itself.

The early evidence indicated that data model instability has its roots in errors in modelling, errors in the semantic analysis (whether done consciously or intuitively), and in changes to the requirements brought on by changes to the "reality". The group of applications which has been examined in this research suggested that some of the elements of a data model are significantly more important than others. The primary attributes of a given entity are of more consequence to the subsequent interpretation than the semantic integrity constraints of the model and their software implementation. The results of the application of the protocol demonstrated that data model instability is also caused by both modelling error and semantic difficulties.

The results of applying the measurement protocol demonstrated some surprising results. The stability of data models is often more imagined than real. This early result deserves extensive replication and testing.

The research necessarily demands some thought about the way in which data models are actually developed. As an outcome of the research, we have also come to new understanding of the problems associated with the transformation of natural language into the constraints of data dictionaries. The limitations of a lexicographic dictionary in helping systems analysts come to understand a given application domain apply equally if not more so to a data dictionary when it comes to making meaning. Based on the lessons learned about the separation of data dictionaries from the context(s) of their development, we have explored some of the potential strategies which an analyst can use to improve the process of semantic analysis. Finally, we have outlined below a number of further research projects which naturally follow from the results to date.

The research has also documented an unequivocal case where an organisation should not reconcile the data definitions across all organisational boundaries, because such an approach would not be optimal. This was an example of where the different elements of the company should agree to disagree.

There also appears to be strong evidence of variable semantic determination for fundamental terms in the organisation. This suggests the need to modify the data modelling theory to take this into account, especially from the perspective of providing useful cues and clues to end users to assist in the correct interpretation of the corporate data resources. Bertrand Russell (1948, 52) might have been anticipating the invention of database management systems when he wrote "there is no obvious limit to the invention of ingenious apparatus capable of deceiving the unwary."

As a consequence of clarifying the meaning associated with the models in this research, I conclude that users are well able to tolerate significant differences in semantic regimes, where such differences are functional and relatively clear. For example, in the case of the personnel database, the expression "Jacob works for the city" in one context would be understood as intending "employee of the city" and therefore expected to be part of this personnel database. In another context the same expression might not imply such a membership at all, as in the case of a contract employee. In a third situation, two participants might have completely different understandings of Jacob's employment situation at the city, until some clue to the misunderstanding emerges. The effects of language misunderstandings is often quite well tolerated, since the context usually makes the meaning relatively clear. When it is unclear, a quick negotiation takes place which eliminates, reduces, or at least clarifies this misunderstanding. Data modelling in the design of automated information systems provides no such capability.

There are a number of specific examples of the phenomenon of meaning clarification for original words and expressions which have been identified in this research. The buildings case study provides two such occurrences, one which is identified by the stability measure protocol, and one which is not. The protocol identified the semantic shift which occurred with the notion of "occupancy code", but it did not identify the shift in the organisational norm for the inclusion criteria regard real estate coordinator records. In the software marketing example, the user community had to reconsider the ideas of "customer" contrasted with the idea of "dealer". The personnel system demonstrated the highly variable use of the fundamental idea of "shift".

In the end, it appears that effective modelling should not necessarily seek to achieve data schema stability. Organisations use data models as a way of managing a social reality which affords the people in the organisation a variety of ways of acting upon their world. What the particular elements of a data model may afford in the way of organisational action will change over time, whether this is the result of external factors which force change, or whether these factors are internal.

An example of an externally imposed change might be the case of changed legislation which redefines what constitutes "employment". Legislation of this kind is sometimes an extreme form of the practice of "deeming" a reality, notwithstanding the facts. An example of an internally generated "change in reality" would be the shift from a traditional inventory control system to just-in-time materials management which seeks to eliminate the idea of production input inventory entirely. The idea of "inventory" is a socially determined one that has not been imposed as a natural part of the universe. It is not a question that God would not have given us warehouses if he had not intended us to have inventory. This research suggests that there should be greater research done on the question of data model evolvability, and the appropriate preservation of meaning across model versions. Data model stability is not necessarily the best objective.

Information management seeks to support organisational thinking and behaviour or action through the provision of meaningful information in a cost effective way. Semantic analysis can contribute directly through the clarification of what is meaningful, what is to emphasised and what is to be suppressed. Semantic analysis can also help by sharpening existing IS tools and methods.

B. OTHER RESEARCH QUESTIONS WHICH EMERGE FROM THIS WORK

As is the case with much research, this thesis raises more questions than it answers. There are many new research opportunities suggested by this work. First, there is more work required in collecting better data on the investment effort made into data modelling to support the subsidiary speculation that there is an optimum effort in data modelling which ought to be made.

Given the time consuming and difficult task of making sense out of data models which were implemented 10 years ago, and seriously modified 5 years ago, it is strongly recommended that this research be done longitudinally forward not backward. In other words, the data collection should be done with the implementation of new systems, where the researcher would have much better chance of improving the quality of the initial data. This approach might also reduce the labour intensity of the whole exercise by having knowledgeable users to explain the context of the data model. The researcher might then revisit the data model on a periodic basis to determine what changes had occurred, tracking the reasons for the changes closer to the time of implementation. It is possible that some changes are implemented and then reversed, once the organisation understands more fully the consequences of it.

It is also possible that such a programme of interim reviews would have a positive effect on the data model. One of the benefits of this research to those who participated is that the researcher spotted areas of the data models which needed revision e.g. indexing, edit criteria (mandatory or optional), redundancies. There are other demonstrated benefits of the tool:

- It might sensitise users to the question of stability, and encourage them to take the longer view of the model.
- It might help to shape expectations on the part of the user about planning for change.

Interpreting data models without the benefit of the original designer being present is often a difficult task. There is an urgent need to extend the research into the pragmatic ways of developing cues to improve the user interpretation of data structures and data content. This might be supported by some research to determine the degree of semantic variability and its significance in an organisation. There is evidence from the models which have been examined in this research and from the general linguistic research relating to semantic under-determination, that some workers in a given environment may not ever completely understand the full meaning of task relevant words, nor need they.

The whole area of business hermeneutics deserves much greater attention. What actual principles, practices and procedures can we set out for analysts to use in clarifying the meaning and its making in an organisation? Some brief ideas beyond the traditional data model, data flow diagramming approach have been suggested. More extensive work, such as the LEGOL/NORMA project (Backhouse, 1991; Stamper, 1985) needs to be explored. There is the outstanding question of what impact designing for stability over some planning horizon might make to the design of schemas. What is a reasonable and practical planning horizon for a data modeller? Does it depend on the application? Does it depend on the industry?

If we were to apply the measurement tool to a large number of models, would the index of stability vary according to size of organisation, type of application, and/or type of industry? Are some forms of structured approach to the articulation of user needs more likely to reduce instability? What impact does the data modeller and the methodology itself have on the resulting model, not to mention on the user's understanding of his own meaning?

This research has produced some unexpected results on the question of model degradation over time. What variables affect the degradation rate? How many of those models which depend so heavily on an "adequate" analysis of business policy, actually have business policy overtly set out as part of the framework of the data model? There may be other techniques which we can apply to increase the likelihood of articulating unspoken business policy.

If someone were to undertake a parallel research project directed at measuring the indices of change for the processing part of the puzzle, we would be in a position to validate the claim that data is more stable than processing. I suspect that there is a high degree of correlation between changing data definitions and changing processes. It would be possible to use the data generated through the stability measurement protocol to generate other model metrics, perhaps of size and complexity.

It would be an interesting project to see how the presence of IS technology, specifically data modelling, and fourth generation languages have in changing business policy. The literature on modelling theory derives the model out of the policy, and does not really address how the policy is influenced by the model and its data.

Finally, it would be interesting to see how databases shape the thinking of organisations if such research could be possible. Chua (1986, 608) set out a provocative discussion on the way the assumptions built into accounting practice limit the way that an organisation can think about itself. He claimed that "there is a tight linkage between explanation, prediction, and technical control." Explanation, prediction and technical control through forced integration of data definitions as a result of the need to share data throughout an organisation may be part of the hidden agenda of data modelling.

William Hazlitt said in *On Taste*, "rules and models destroy genius and art." Research into the empirical effects of the rules and models of information modelling and database management technologies may be an important factor in preserving the potential of genius and art in the modern organisation.

BIBLIOGRAPHY

Agosti, M. and R. G. Johnson. 1984. A framework of reference for database design. *Data Base* 15:4 (Summer).

Aitchison, Jean. 1985. Cognitive clouds and semantic shadows. *Language and Communication* 5:2.

—— 1987. *Words in the Mind: An Introduction to the Mental Lexicon.* Basil Blackwell, Oxford.

Allwood, J., L. Andersson, and D. Osten. 1977. *Logic in Linguistics.* Cambridge University Press.

Ariav, Gad. 1986. A temporally oriented data model. *ACM Transaction on Database Systems* Vol. 11, No. 4.

Armstrong, S.L., L. R. Gleitman, and H. Gleitman. 1983. What some concepts might not be. *Cognition* 13: pp. 263-308.

Arnold, J. and T. Hope. 1983. *Accounting for Management Decisions.* London, Prentice Hall.

Arner, Douglas. 1959. On knowing. *The Philosophical Review*, LXVIII, No. 1 (in Yolton, John W. (ed.) 1965. *Theory of Knowledge.* The MacMillan Company, New York).

Austin, J. L. 1975. *How To Do Things With Words (2nd Edition).* Oxford University Press, Oxford.

Avison, D.E. 1985. *Information Systems Development: A Data Base Approach.* Blackwell Scientific, Oxford.

Avison, D.E. and G. Fitzgerald. 1988. *Information Systems Development: Methodologies, Techniques, and Tools.* Blackwell Scientific, Oxford.

Bachman, Charles W. 1969. Data structure diagrams. *Data Base* (1:2).

Backhouse, J. 1991. *The use of semantic analysis in the development of information systems.* Unpublished Ph.D. Thesis. London School of Economics. London.

Baker, Geoff. 1985. Application development using data analysis. *Computer Bulletin* June.

Barnwell, Katherine. 1980. *An Introduction to Semantics & Translation.* Summer Institute of Linguistics, Berkeley, California.

Berger, P. L., and Luckman, T. 1971. *The Social Construction of Reality.* Penguin, Harmondsworth.

Berka, Karel. 1983. *Measurement: Its Concepts, Theories, and Problems.* D. Reidel Publishing, Dodrecht, Holland.

Biller, Horst and Erich J. Neuhold. 1977. Concepts for the conceptual schema (in Nijssen, G.M. (ed.) 1977a. *Architecture and Models in Database Management Systems,* North-Holland, Amsterdam.)

———— 1978. Semantics of data bases: the semantics of data models. *Information Systems.* Vol. 3, No. 1.

Bloor, David. 1978. Polyhedra and the abominations of Leviticus. *British Journal for the History of Science* Vol. 11, No 39.

Boland, R. and Hirscheim R. (eds.) 1987. *Critical Issues in Information Systems Research.* John Wiley, Maidenhead.

Brachman, Ronald J. 1977. What's in a concept: structural foundations for semantic networks. *International Journal of Man-Machine Studies* Vol. 9, 127-152.

Bravoco, R.R. and Surya B. Yadav. 1985. A methodology to model the information structure of an organization. *The Journal of Systems and Software* (5).

Brodie, M., ed. 1981. Proceedings of workshop on data abstractions, and databases. *Special Issue: ACM Sigart/Sigmod/Sigplan Notices.* Pingree Park, Colorado.

Brown, J. S. 1991. Research that reinvents the corporation. *Harvard Business Review,* January-February.

Brown, and Fraser. 1979. Speech as a marker of situation. (in K.R. Scherer, and Giles, H., eds. *Social Markers in Speech*). Cambridge University Press.

Bubenko, J. A. 1979. Data models and their semantics. *Proceedings of the 2nd State of Art Conference on Data Design.* Infotech, London

Carden, J. 1986. The structural stability of corporate and devolved databases. *The Computer Journal* Vol. 29, No. 4.

Carlson, C. Robert, M. M. Carlson, and K. Arora Adarsh. 1982. The application of functional dependency theory to relational databases. *The Computer Journal* Vol. 25, No. 1.

Casanova, Marco A. and Jose E. Amaral de Sa. 1984. Mapping uninterpreted schemas into entity-relationship diagrams. *IBM Journal of Research and Development* Vol. 28, No 1.

Chase, Stuart. 1937. *The Tyranny of Words.* London, Methuen.

Checkland, Peter B. 1981. *Systems Thinking, Systems Practice.* John Wiley and Sons, Chichester.

Chen, P.P. 1986. The time dimension in the entity-relationship model. *Information Science 86.* North-Holland, Amsterdam.

——————— 1976. The entity-relationship model - toward a unified view of data, *ACM Transactions on Database Systems* Vol. 1, No 1.

Chomsky, Noam. 1988. *Language and Problems of Knowledge.* MIT Press, Cambridge, MA.

Chua, Wai Fong. 1986. Radical developments in accounting thought. *The Accounting Review* Vol. LXI, No. 4, 601-632.

Churchman, C.W. and Ratoosh, P. (eds.) 1959. *Measurement: Definitions and Theories.* Wiley, New York.

Claybrook, B.G., Claybrook, A. and Williams, J. 1985. Defining Database Views as Data Abstractions. *IEEE Transactions on Software Engineering*, January 1985.

Codd, E. F. 1970. A relational model of data for large shared data banks. *Communications of the ACM*, Vol. 13, No. 6.

——————— 1979. Extending the database relational model to capture more meaning. *ACM Transactions on Database Systems*, Vol. 4, No. 4.

——————— 1982. Relational database: a practical accommodation for productivity. *Communications of the ACM*, Vol. 25, No. 2, February.

Cohen, L. J. 1962. *The Diversity of Meaning.* Methuen, London.

Condon, Jr., J. C. 1975. *Semantics and Communication.* (2nd Edition) MacMillan, New York.

Cook, S. and Stamper, R. K. 1980. LEGOL as a tool for the study of bureaucracy. (in *The Information Systems Environment.* Lucas, H., Land, F., Lincoln, T. and Supper, K. (eds.) North-Holland, Amsterdam.)

Curtice, R. M. and Jones, P.E. 1982. *Logical data base design.* Van Nostrand Reinhold Company, New York.

Dahl, O. J., Dijkstra, E.W., and Hoare, C.A.R. 1972. *Structured Programming.* Academic Press, London.

Date, Chris J. 1986. *An Introduction to Database Systems Vol 1 (4th Edition)* Addison-Wesley. Reading, MA

Davenport, R.A. 1978. Data analysis for database design. *The Australian Computer Journal*, Vol. 10, No. 4.

De, Prabuddha., Sen, Arun., and Gudes, Ehud, 1982. A new model for data base abstraction. *Information Systems*, Vol. 7, No. 1.

Deen, S.M. 1985. *Principles and Practices of Data Base Systems.* MacMillan Publishers Ltd., London

Dowty, David R. 1979. *Word Meaning and Montague Grammar.* D. Reidel Publishing Co., Dodrecht, Holland

Edgar, John A. 1986. *The Design of a Unified Data Model.* Unpublished Ph. D. thesis, University of Aberdeen.

Ellis, Brian 1966. *Basic Concepts of Measurement.* Cambridge University Press, Cambridge.

Everest, Gordon C. 1986. *Database Management: Objectives, System Functions & Administration.* McGraw-Hill, Singapore.

Falkenberg, E. 1976. *Concepts for Modelling Information.* North-Holland, Amsterdam.

Fann, K.T. 1970. *Peirce's Theory of Abduction.* Martinusnijhoff, The Hague.

Feldman, P. and Fitzgerald, G. 1985. Action modelling: a symmetry of data and behaviour modelling. (in Grundy, A. F. (ed.) 1985. *Proceedings of the Fourth British National Conference on Database.* CUP, Cambridge)

Feldman, P. & Miller, D. 1986. Entity model clustering: structuring a data model by abstraction. *The Computer Journal*, Vol. 29, No. 3.

Fillmore, Charles J. 1977. *Scenes-and-frames semantics.* North-Holland, Amsterdam.

Flavin, Matt. 1981. *Fundamental Concepts of Information Modelling* Yourdon Press, Inc., New York.

Floyd, Christiane. 1986. A comparative evaluation of system development methods (in Olle, T.W., Sol, H.G. and Verrijn-Stuart (eds.) 1986. *Information Systems Design Methodologies: Improving the Practice.* North-Holland, Amsterdam)

Frost, R. A. 1986 Formalising the notion of semantic integrity in database and knowledge base systems work. (in Oxborrow, E.A. 1986. *Proceedings of the Fifth British National Conference on Databases.* Cambridge University Press.)

———— 1985. Using semantic concepts to characterise various knowledge representation formalisms. *The Computer Journal* Vol. 28, No. 2.

Frost R. A. and Whittaker, S. 1988. A step towards the automatic maintenance of semantic integrity. *The Computer Journal,* Vol. 26, No. 2.

Fung, K. T. 1984. A reorganization model based on the database entropy concept. *The Computer Journal,* Vol. 27, No.1.

Galliers, R. 1987. *Information Systems Planning.* Unpublished Ph.D thesis. London School of Economics.

Gane, C. and Sarson, T. 1979. *Structured Systems Analysis: Tools and Techniques.* Prentice Hall, New York.

Gardiner, M. 1968. On the meaning of randomness and some ways of achieving it. *Scientific American.* July.

Ghiselli, E. E., Campbell, J. P., and Zedeck, S. 1976. *Measurement Theory for the Social Sciences.* W. H. Freeman, New York.

Gorman, Michael M. 1984. *Managing Database: Four Critical Factors.* QED Information Sciences Inc., Wellesley, MA

Grundy, A. F. (ed.) 1985. *Proceedings of the Fourth British National Conference on Database.* Cambridge University Press, Cambridge.

Guralnik, D.B., and Friend, J.H. (eds.) 1966. *Webster's New World Dictionary.* Nelson, Foster, and Scott, Toronto, Canada.

Gvishiani, J.M. 1985. *Systems Research - Methodological Problems.* Pergamon Press, Oxford.

Hammer, Michael and McLeod, Dennis. 1981. Database description with SDM: A semantic database model. *ACM Transactions on Database Systems,* Vol. 6, No. 3.

Hedberg, Bo & Sten Jonsson. 1978. Designing semi-confusing information systems for organizations in a changing environment. *Accounting, Organizations and Society,* Vol. 3 No. 1.

Jackson, Michael. 1983. *System Development.* Prentice-Hall International Inc., Englewood Cliffs, NJ

Jardine, D.A. 1985. Semantic agreement and the communication of knowledge. (in Steel, Jr., T.B. & Meersman, R. (eds.) 1985. *Data Semantics*, North-Holland, Amsterdam)

Jones, J. A. 1986. *Databases in Theory and Practice* Kogan Page Ltd., London.

Kahn, Beverley K. 1983. Some realities of data administration. *Communications of the ACM*. Vol. 26, No. 10.

Keesing, Roger M. 1981. *Cultural Anthropology: A Contemporary Perspective (2nd edition).* Holt, Rinehart and Winston, New York.

Kent, W. 1978. *Data and Reality.* North-Holland, Amsterdam .

——————— 1979. Data model theory meets a practical application. *Seventh International Conference on VLDB*, 1981, pp 13-22.

——————— 1979. Limitations of record-based information models. *ACM Transactions on Database Systems* Vol. 4, No. 1, March.

——————— 1983 A simple guide to five normal forms in relational data base theory. *Communications of ACM*, Vol. 26, No. 2, pp. 120-125.

——————— 1985. The realities of data: basic properties of data reconsidered. (In Steel, Jr., T.B. & Meersman, R. (eds.) 1985. *Data Semantics*, North-Holland, Amsterdam)

Kerschberg, L. Klug, A, and Tsichritzis, D. 1976. A taxonomy of data models, North-Holland. (in Lockemann, P.C. and Neuhold, F. J, eds.) Amsterdam

Kimura, T.D., Gillett, W.D., and Cox, J.R. 1985. A design of a data model based on abstraction of symbols. *The Computer Journal*, Vol. 28, No. 3.

Klein, H. K. and Hirschheim, R. 1985. Fundamental issues of DSS: A consequentialist perspective. *Journal of Decision Support Systems*, Vol. 1, 5-24.

——————— 1987. A comparative framework of data modelling paradigms and approaches. *The Computer Journal*, Vol. 30, No. 1.

Kling, Rob. 1987. Defining the boundaries of computing across complex organizations. (in Boland, R. and Hirschheim, R. (eds.) 1987. *Critical Issues in Information Systems Research.* John Wiley, Maidenhead.)

Kyburg, Jr., Henry E. 1984. *Theory and Measurement.* Cambridge University Press, Cambridge.

Lakoff, G. and Johnson, M. 1980. *Metaphors We Live By*. University of Chicago Press, Chicago.

Langefors, Borje. 1979. Infological models and information user views. *Information Systems*, Vol. 5.

Langefors, Borge, and Samuelson, Kjell. 1976. *Information and Data in Systems*. Mason/Charter Publishers, New York.

Larousse Encyclopedia of Animal Life. 1967: The Hamlyn Publishing Group, London.

Ledgard, Henry F., and Taylor, Robert W. 1977. Two views of data abstraction. *Communications of the ACM*. Vol. 20, No. 6.

Leech, Geoffrey. 1981. *Semantics - The Study of Meaning (2nd Edition)*. Pelican, Harmondsworth.

Lescanne, Pierre. 1983. Behavioural categoricity of abstract data type specifications. *The Computer Journal*, Vol. 26, No. 4.

Lewis, Norman (ed.) 1961. *The New Roget's Thesaurus in Dictionary Form*. G. P. Putnam's Sons Inc., New York.

Lockemann, P.C,, Mayr, H.C., Weil, W.H., and Wohlleber, W.H. 1979. Data abstractions for database systems. *ACM Transactions on Database Systems*.

Lundeberg, Mats, Goldkuhl, G., Nilssen, A. 1979. A systematic approach to information systems development. *Information Systems*, Vol 4.

Lyons, John. 1979. *Semantics (Volumes 1 and 2)* Cambridge University Press, Cambridge.

MacKay, S.M. 1969. *Information Mechanism and Meaning*. MIT, Cambridge, MA

Marche, M.M. *EDP and apprenticeship administrations*. Unpublished M. Ed thesis, University of Alberta, Edmonton Canada 1978.

Marche, Sunny. 1991. On what a building might not be - a case study. *International Journal of Information Management*, **11**, (pp. 55-66).

Martin, Daniel. 1986. *Advanced Database Techniques*. MIT Press, Cambridge, MA

Martin, James. 1976. *Principles of Data Base Management*. Prentice-Hall, Englewood Cliffs, NJ.

————— 1981. *An End-User's Guide to Data Base.* Prentice-Hall, Englewood Cliffs, NJ.

————— 1983. *Managing the Database Environment.* Prentice-Hall, Englewoods Cliff, NJ.

————— 1985. *Systems Design from Provably Correct Constructs.* Prentice-Hall International, London.

Martin, James, and McClure, Carma. 1985. *Diagramming Techniques for Analysts and Programmers.* Prentice-Hall, Englewood Cliffs, New Jersey

Mason, R.O., and Swanson, B.B. 1982. Measurement as an MIS foundation. *Database,* Fall 1972.

McCorkle, Jr., Chester O. 1977. Information for institutional decision making. *New Directions for Institutional Research,* Vol. 15, Autumn, 1977.

Meadows, D. H. and Robinson, J. M. 1985. *The Electronic Oracle: Computer Models and Social Decisions.* John Wiley and Sons, New York.

Mijares, I. and Peebles, R. 1976. *A Methodology for the Design of Logical Data Base Systems.* North-Holland, Amsterdam.

Miller, G.A., and Johnson-Laird, P.N. 1976. *Perception and Language.* Belknap Press of Harvard University Press, Cambridge, MA.

Model Systems Ltd. 1983. *The Structured Systems Analysis and Design Methodology (SSADM).* Central Computer and Telecommunications Agency, London.

Navathe, S. and Kerschberg, L. 1986. Role of data dictionaries in information resource management. *Information & Management,* (10), North-Holland.

Nelson, Katherine. 1985. *Making Sense: the Acquisition of Shared Meaning.* Academic Press Inc., New York.

Nijssen, G.M. (ed.) 1976. *Modelling in Data Base Management Systems.* North-Holland, Amsterdam.

————— 1977a. *Architecture and Models in Database Management Systems.* North-Holland Publishing, Amsterdam.

————— 1977b. Current issues in conceptual schema concepts. (in Nijssen, G.M. (ed.) 1977a. *Architecture and Models in Database Management Systems.* North-Holland, Amsterdam.)

Ogden, C. K. and Richards, I.A. 1923. *The Meaning of Meaning.* Routledge, London.

Olivé, Antoni. 1985. *Conceptual Languages for Information Systems Modelling.* North-Holland IFIP, Amsterdam.

Olle, T.W., Sol, H.G. and Verrijn-Stuart (eds.) 1986. *Information Systems Design Methodologies: Improving the Practice.* North-Holland, Amsterdam.

Ortony, Andrew. (ed.) 1979. *Metaphor and Thought.* Cambridge University Press.

Osgood, C.P., Suci, G.J., Tannenbaum, P.H. 1957. *The Measurement of Meaning.* University of Illinois Press, Urbana.

Oxborrow, E.A. (ed.) 1986. *Proceedings of the Fifth British National Conference on Databases.* Cambridge University Press, Cambridge.

Rock-Evans, Rosemary. 1981. *Data Analysis.* IPC Electrical-Electronic Press Ltd. Surrey, England.

Parkin, A. 1982. Data analysis and system design by entity relationship modelling. *The Computer Journal.* Vol. 25, No. 4.

Peirce, C. S. 1958. *Charles S. Peirce: Selected Writings.* (Weiner, P.P. ed.) 1958 Dover Publications, New York.

Pichler, Franz. 1982. *On the Use of Structured Methodologies in General Systems Research.* Kluwer-Nijhoff Publishing, Boston .

Popper, Karl. 1972. *Conjectures and Refutations. Fourth Edition.* Routledge, London.

Prakash, Naveen. 1984. *Understanding Database Management.* Tata McGraw-Hill, New Delhi.

Raskin, Victor. 1985. *Semantic Mechanisms of Humour.* D. Reidel Publishing Company, Dodrecht, Holland.

Reddy, Michael J. 1979. The conduit metaphor. Cambridge University Press, Cambridge. (in Ortony, A. (ed.) *Metaphor and Thought*, Cambridge University Press)

Roberts, Fred S. 1979. *Measurement Theory with Applications to Decision Making.* Addison-Wesley, Reading, Mass.

Robinson, K.A. 1979. An entity/event data modelling method. *The Computer Journal*, Vol. 22, No. 3.

Ross, Ronald. G. 1986. *Entity Modeling: Techniques and Applications.* Database Research Group Inc., Boston, Mass.

Ross, Douglas T. and Schoman Jr., K.E. 1977. Structured analysis for requirements definition. *IEEE Transactions on Software Engineering*, Vol. SE3 No 1.

Ruchti, J. 1976. *Data Descriptions Embedded in Context.* North-Holland, Amsterdam.

Russell, Bertrand. Knowledge of facts and knowledge of laws. (in Yolton, John W. (ed.). 1965. *Theory of Knowledge.* The MacMillan Company, New York)

Saussure, de, Ferdinand. 1959. *The Course in General Linguistics.* Philosophical Library, New York (original work, 1915).

Schank, Roger C. and Leake, David B. 1986. Computer understanding and creativity. *Information Processing 86.* North-Holland, Amsterdam.

Schon, Donald. 1983. *The Reflective Practitioner: How Professionals Think in Action.* Basic Books, New York.

Schweiger, D. M. S, William, Ragan, James. 1986. Group approaches for improving strategic decision making. *Academy of Management Journal*, Vol. 29 No. 1.

Senko, Michael E. 1975. Information systems: records, relations, sets, entities, and things. *Information Systems*, Vol 1.

Shannon, C. E., and Weaver, W. 1949. *The Mathematical Theory of Communication.* University of Illinois, Urbana.

Shave, M.J.R. 1981. Entities, functions and binary relations: steps to a conceptual schema. *The Computer Journal*, Vol. 24, No.1.

Siu, R.G.H. 1957. *The Tao of Science - An Essay on Western Knowledge and Eastern Wisdom.* The Technology Press MIT, Cambridge.

Smith, Henry C. 1985. Database design: composing fully normalized tables from a rigorous dependency diagram. *Communications of the ACM*, Vol. 28, No. 8.

Smith, J. M. and Smith, D. C. P. 1977a. Database abstractions: aggregation. *Communications of the ACM*, Vol. 20, No. 6.

———— 1977b. Database abstractions: aggregation and generalisation. *ACM Transactions on Database Systems.* Vol. 2, No. 2.

Solvberg, A., and Kung, C.H. 1986. On structural and behavioral modelling of reality. (in Steel, Jr., T.B. & Meersman, R. (eds.) 1985. *Data Semantics*, North-Holland. Amsterdam)

Softech, Inc. 1981. *Integrated Computer-Aided Manufacturing (ICAM) Architecture Part II, Volume IV - Functional Modeling Manual (IDEF$_0$).* Waltham, Mass.

Sperber, Dan and Wilson, Deirdre. 1986. *Relevance: Communication and Cognition.* Basil Blackwell, Oxford .

Stamper, R.K. 1973. *Information in Business and Administrative Systems.* John Wiley, London.

———— 1979. Towards a semantic normal form. *Proceedings of IFIP TC2 WC on Database Architecture.* Venice, Italy

———— 1985. A logic of social norms for the semantics of business information. *IFIP WG 2.6 WC on Database Semantics*, Hasselt, Belgium.

———— 1987. Research issues in information systems: semantics. (in Boland, R. and Hirscheim R. (eds.) 1987. *Critical Issues in Information Systems Research.* John Wiley, Maidenhead.)

Steel, Jr., T.B. & Meersman, R. (eds.) 1985. *Data Semantics*, North-Holland. Amsterdam.

Sundgren, Bo. 1975. *Theory of Data Bases.* Mason/Charter Publishers, New York.

Sweet, Frank. 1984. What, if anything, is a relational database? *Datamation*, July 15, pp 118-124.

Symons, C.R. and Tijsma, P. 1982. A systematic and practical approach to the definition of data. *The Computer Journal*, Vol 25, No. 4.

Teichroew, Daniel and David, Gabor (eds.) 1985. *System Description Methodologies.* North-Holland, Amsterdam.

Tsichritzis, Dionysios and Lochovsky, Frederick. 1982. *Data Models.* Prentice-Hall, Englewood Cliffs, NJ.

Veryard, R. 1984. *Pragmatic Data Analysis.* Blackwell Scientific Publications, London

Wiederhold, G. 1983. *Database Design.* McGraw Hill, New York.

Whorf, Benjamin Lee. 1956. *Language, Thought and Reality.* MIT, Cambridge, MA

Winograd, Terry and Flores, Fernando. 1986. *Understanding Computers and Cognition.* Ablex Publishing Corporation, Norwood, New Jersey.

Wong, Harry K.T. 1982. Semantics, dynamics tradeoff in relational database design. *Information Science*, Vol. 7, No. 3.

Wood-Harper, A.T. and Fitzgerald, G. 1982. A taxonomy of current approaches to systems analysis. *The Computer Journal*, Vol. 25, No. 1.

Yadav, Surya B. 1983. Determining an organization's information requirements. *Data Base*, Spring.

Yolton, John W. (ed.) 1965. *Theory of Knowledge.* The MacMillan Company, New York.

Yourdon, E. 1989. *Managing the Structured Techniques. (4th Edition)* Prentice-Hall, Englewood Cliffs, NJ.

Zadeh, Lofti A. 1982. *Fuzzy Systems Theory: A Framework for the Analysis of Humanistic Systems.* Kluwer-Nijhoff Publishing, Boston.

Zahran, F. S. 1982. Structured analysis of data dictionary systems and their environments. (in Deen, S.M. and Hammersley, P. 1982. *Second British National Conference on Databases.* Middlesex Polytechnic.)

———————— 1981. A universal data model holder for data dictionary systems, *The Computer Journal*, Vol. 24 No. 3.

Zeller, Richard A., & Carmines, Edward G. 1980. *Measurement in the Social Sciences.* Cambridge University Press, Cambridge.

Zito, George V. 1984. *Systems of Discourse.* Greenwood Press. Westport, Connecticut.

APPENDIX I - SURVEY OF MODELLING PRACTICES WITH BACKGROUND NOTES

Of the 70 organisations which replied to the survey of data modelling practices reported in Chapter 3, 24 organisations expressed interest in participating further. These organisations were given a copy of the data collection tools, and a document which set out the classification definitions used in the process. The only formal returns from this request were from people who expressed regrets that their resources did not permit participation at this level of detail. Telephone follow-up of database administrators revealed a professional group which felt to be seriously underfunded for the task they had in front of them. Therefore, any extensive participation in this work was impossible for them.

As an alternative, local members of the data processing community were contacted and were given detailed presentations on the options for participating in the research. The options varied from conducting the detailed analysis themselves, to simply providing the record description and baseline system documentation. This group also reported that their resources were under severe pressure. In the end, seven different applications were identified where there was sufficient data available, and adequate user willingness to participate.

There may be other explanations for the unwillingness of database administrators to participate. Many organisations have very limited interest in the historical development of their systems, and dispose of the documentation of previous developments soon after these systems have been replaced. The quality of the documentation at all levels in the system cycle appeared highly variable, with much of it quite poor. Some administrators may have been embarrassed at the prospect of an outside researcher carefully reviewing the quality of the systems documentation. This may be true especially in the applications where there has been a high degree of change over a short period of time. In these cases, the resources of the systems organisation are applied to addressing the technical implementation problems without a high regard for providing a coherent and complete documentation trail.

DATA MODELLING MANAGEMENT

All data in the following questionnaire will be kept in the strictest confidence.

CONTACT INFORMATION

Your Name: ____

Position Title: ____

COMPANY INFORMATION

1. Approximate Number of Employees ______________________________

2. Approximate Number of Full Time Computer Staff ____________________

3. Number of staff involved in data administration ______________________

4. Which DBMS do you use?

   Is it used for: cricle one for each

Packaged software?	Yes / No
s/w under development?	Yes / No
all software?	Yes / No

   When was this DBMS installed? ______________________________

5. In the practice of systems work in your organization is data modelling used for : (circle Yes or No for each)

the development of individual applications?	Yes / No
strategic data planning at the level of major organization boundaries such as divisions or departments?	Yes / No
strategic data planning at a fuly integrated level?	Yes / No

Has your organization ever undertaken business strategic planning in a formal way?

Circle one Yes / No

If yes, when was the last time this was done?

How often is this done?

Is the EDP group usually involved? Yes / No

Have you attempted and/or completed enterprise modelling, using BSP (Business Systems Planning) or equivalent?

Circle one Yes / No

Is there a formal, periodic planning of Information Technology development involving: (circle Yes or No for each question)

user management? Yes / No

EDP / MIS management? Yes / No

data administration? Yes / No

Do you use a formal data modelling methodology?

Circle Yes or No for each

data flow diagramming	Yes / No
HOS charts	Yes / No
HIPO diagramming	Yes / No
Nassi/Schneiderman	Yes / No
Warnier-Orr	Yes / No
Jackson System Devel	Yes / No
Action diagrams	Yes / No
Decision tables / trees	Yes / No
Data analysis diagrams	Yes / No
Entity relationship model	Yes / No
Entity attribute relationship	Yes / No
Data navigation diagrams	Yes / No

Other - please specify ____________________

Do you use any automated aids to data modeliing?

Circle one Yes / No

If yes, Please specify which. ____

Do you have established documentation standards for establishing and maintaining data elements?

Circle Yes or No for each of the following questions:

rigid applied	Yes / No
formally applied	Yes / No
informally applied	Yes / No
not enforced	Yes / No

Do you monitor the stability of your data models over time?

Circle one Yes / No

If yes, briefly describe how this is done, or comment:

13. Have you had difficulties resolving differences in data definitions between / among different application areas?

Circle one Yes / No

If Yes, briefly indicate whether difficulties were serious, and what effect this has had on data administraton.

14. What is the biggest challenge in data administraion which your currently face?

Comments:

15. Would you be willing to test a methodology to measure data modelling stability over time?

Circle One Yes / No

Person to Contact
Organization (if different from above):

Address:

Telephone Number:

16. Any additional comments:

APPENDIX II

BLANK DATA COLLECTION SHEETS

1. Model Name: ______________________________

2. Entity Name: ______________________________

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary Second Other	Foreign Key	Indexed	Derived/ Original

B. ENTITY DATA COLLECTION SHEET

	Original	Revised
1. Model name and date:		
2. Entity name:		
3. Number of attributes used for keys:		
4. Number of attributes which are foreign keys:		
5a. Number of primary attributes for this entity:		
b. Number of secondary attributes:		
c. Number of other attributes:		
6. Number of indexed attributes:		
7. Is this entity primary or secondary to the application?		

8. Describe any differences or disputes in definition or understanding across the organisation for this entity.

9. Criteria for inclusion (if known, or specified).

C. RELATIONSHIP DATA COLLECTION SHEET

1. Model Name:____

2. Relationship Name:____________________________________

3. Entity 1:____

   Entity 2:____

   Other entities:______________________________________

4. Describe the criteria for this relationship.

D. BACKGROUND AND SUMMARY OF MODEL DATA

1. Organisation: ______________________________

2. Model Names and dates: ______________________________

3. General application area: ______________________________

____

____

4. Date(s) of revision(s): ______________________________

5. Original reason for developing model: ______________________________

____

____

____

____

6. Systems in place prior to model: ______________________________

____

____

7. File layouts of previous automated application available?__________ If so, please attach.

8. Documentation standard and technique used: ______________________________

____

____

____

____

____

D. BACKGROUND AND SUMMARY OF MODEL DATA

16. Describe any update process to the original model:

____

____

____

____

17. Comment on the degree and quality of management participation.

____

____

____

____

18. Comment on the degree and quality of user participation.

____

____

____

____

____

____

____

D. BACKGROUND AND SUMMARY OF MODEL DATA

19. List any entities, attributes and/or relationships which have been the subject of marked differences in definition in the organisation and describe.

20. Has your organisation attempted to integrate data models or data bases from different parts of the organisation? If so, please note any difficulties which this process had to overcome.

21. Which areas in these models have been particularly unstable?

APPENDIX III

COMPLETED DATA COLLECTION SHEETS - DOFS

1. Model Name: DoFS.

2. Entity Name: ~~FLOOR~~/SPACE (page 1)

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary/Second Other	Foreign Key	Indexed	Derived/ Original
ASSIGNMENT #	Numeric, 4	Numeric		K	O	–	N	O
Account CODE	Numeric, 6	Numeric	part of chart of accounts	N	S	–	N	O
S.C. (settlement code)	1-7	Alnumeric.	per policy documentation	N	S	–	–	O
BLDG GRP CODE	4 Numeric.	none	per chart of codes in documentation.	N	S	–	–	O
House SERV. CODE	707000 604600 misc	Numeric	per corporate COA.	N	S?	–	–	O
O L R (owned, leased, R)	O, L, R	Alpha	O, L, R	N	P	–	–	O
Lease Rental Code.	Numeric, 6	Numeric	per Corp. COA.	N	S	.	–	O
% of Invoice	Numeric (integ 3 decimal)	Numeric	≤ 100	N	S	–	–	O
DIS (district)	Alpha, 1	Alpha N.	per internal code convention	N	S	–	–	O
F/M (Foreman)	Alpha, 2	Alpha N.	initials of foreman.	N	S	–	–	O

A. ATTRIBUTE DATA COLLECTION SHEET

1. Model Name: DoFS.

2. Entity Name: ~~Floor~~ / SPACE (page 2)

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary/Second Other	Foreign Key	Indexed	Derived/ Original
H.S (House Services)	not known	not known	?	N	S	—	N	O
RE/SUB (operating reg)	text AN. 2	none	per user region designation. chart	N	S	-	N	O
CITY	4 Text	none, but redundant with CLLI	↔	N	S	-	N	O
Location Code (Accounting)	4 Numeric	~~per~~ none	one to one corresp. to CLLI, per accounting.	N	S	-	N	O
CLLI Code	6 Alpha 5AN	none	consistent w/ corporate CLLI data.	K	P	-	N	O
Building Address	Text, 30	none	street address or lot/block / legal.	N	S	-	N	O
FL (floor)	AN	none	not lower than -3, not higher 30	K	P	-	N	O
Building Name	Text 25	none.	commonly accepted corporate name.	N	S.	-	N	O
Space Indent (Identificⁿ)	Text 15	none	miscellaneous. textdata	N	S	-	N	O
Actual Square Meters	Numeric	none	occupied area, as measured	P	P	-	N	O

1. Model Name: DOFS

2. Entity Name: ~~Floor~~/SPACE (page 3)

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary/Second Other	Foreign Key	Indexed	Derived/ Original
Eq FACT (equating factor)	numeric	none	per documentation 7/0 ≤ 1.50	N	O	—	N	O
Equiv. Square meters	numeric	—	calculated	N	O	—	N	D
Total Occupied Area (floor)	numeric	—	less than gross area.	N	P	—	N	O
Unoccupied Area	numeric	—	less than gross .	N	P	—	N	O
Total Assignable Area.	numeric.	—	less than gross .	N	P	—	N	D
Gross Area	numeric	—	< 3000 m^2	N	P	—	N	O
Non-assignable Area.	numeric	—	less than gross	N	S	—	N	O
Effect Date (effective)	date.	—	Today or earlier	N	O	—	N	O
New.	~~blank, N~~ 0, 1, 2, 3	—	0 - no change 1 - add 2 - delete 3 - vacant row.	N	O	—	N	O
Lessor.	20 Text	—	None	N	S	—	N	O

1. **Model Name:** DoFS.

2. **Entity Name:** F~~LOOR~~/SPACE (page 4).

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary/Second Other	Foreign Key	Indexed	Derived/ Original
11 columns data elements	unknown	none	unknown	N	O	–	–	D?
Actual in Sq. Ft.	Numeric	none	none	N	S	–	–	D
Locator Code (position code)	Numeric, 7	none	position code consistent w/ Corporate data	N	S	–	–	O

1. Model Name: DOFS

2. Entity Name: FLOOR (p. 1)

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary Second Other	Foreign Key	Indexed	Derived/ Original
ASSIGNMENT #	Numeric, 4	Numeric.		K	O	–	N	O
ACCOUNT CODE	null field.	–	blank.	–	–	–	–	-
S.C. settlement code	null field.	-	blank	–	–	–	–	–
BLDG GROUP CODE	~~null field~~ 4, NUMERIC	NumEric	chart of codes in documentation	N	S	N	N	O
HOUSE SERV CODE	null field.	–	blank.	–	–	–	–	–
OLR (owned, leased rented)	O, L, R	Alpha	O, L, R.	N	S	N	N	O
Lease Rental code	null	–	–	–	–	–	–	–
% OF INVOICE	null	–	–	–	–	–	–	–
DIS (District)	Alpha, 1	Alpha	per understood code convention	N	S	N	N	O
F/M (Foreman)	Alpha, 2	Alpha.	initials of foreman.	N	S	N	N	O

1. Model Name: DOFS.

2. Entity Name: FLOOR (p. 2)

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary Second Other	Foreign Key	Indexed	Derived/ Original
H. S.	Not Known	not known	–	–	–		N	O
RE/SUB	text AN 2	non.	per user region designation cht.	N	S	-	N	O
CITY	4 TEXT	none, but redundant with CLLI →		N	S		N	O
LOCATION CODE (Accounting)	4 Numeric	none	one to one corresp. to CLLI, per acctng	N	S	–	N	O
CLLI ~~CODE~~	6 Alpha Num.	none	consistent w/ corp. CLLI code	K	P	–	N	O
BUILDING ADDRESS	Text, 30	none	Street address, or lot/block/ legal	N	S	–	N	O
FL	AN	none	not lower than -3 not higher than 30	K	P	–	N	O
BUILDING NAME	Text 25	none	Commonly accepted corporate name	N	S	–	N	O
SPACE IDENT	Text 15	none	miscellaneous text	N	S	–	N	O
ACTUAL SQ METERS.	Numeric.	none.	occupied area, as measured.	N	P	–	N	O

1. Model Name: DOFS

2. Entity Name: FLOOR (p. 3)

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary Second Other	Foreign Key	Indexed	Derived/ Original
EQ FACT. (equating factor)	numeric	none	per documentation ≥ 0 ≤ 1.50	N	O	-	N	O
EQUIV SQUARE METERS	numeric	none	calculated	N	O	-	N	D
TOTAL OCCUPIED AREA (floor)	numeric	none	less than gross area	N	P	-	N	O
UNOCCUPIED AREA	numeric	none	less than gross area	N	P	-	N	O
TOTAL ASSIGNABLE AREA	numeric	none	less than gross area	N	P	-	N	D
GROSS AREA	numeric	none	$< 3000 \text{ m}^2$	N	P	-	N	O
NON ASSIGNABLE AREA	numeric	none	less than gross	N	P	-	N	O
EFFECT DATE	date.	none	today or earlier	N	O	-	N	O
NEW	0, 1, 2, 3	none	0 - no change 1 . add 2 - delete 3 - vacant row	N	O	-	N	O
LESSOR	Text, 20	none	none.	N	S	-	N	O

1. Model Name: DOFS.

2. Entity Name: FLOOR (Page 4)

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary Second Other	Foreign Key	Indexed	Derived/ Original
" columns data elements	unknown	none	unknown.	N	O	N	-	D?
ACTUAL in SQ. FT.	Numeric	none	none	N	S	N	-	D
LOCATOR CODE (position code)	Numeric, 7	none	employee position code consistent w/ corporate data.	N	S	N	-	O

APPENDIX IV

COMPLETED DATA COLLECTION SHEETS - SPARC(1)

1. **Model Name:** SPARC (October, 1988)

2. **Entity Name:** BUILDING (page 1)

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary/Second Other	Foreign Key	Indexed	Derived/ Original
CLLI	AN, 8 ~~Fax~~	None	Consistent w/ Corporate except bldgs in storage	K	P	Y	Y	O
Building Name	Text, 30	None	''	N	P	N	-	O
Building Address	Text, 30	None	''	N	P	N	-	O
Municipality	Text, 18	None	''	N	P	N	-	O
Building Use Code	A, 1 charact.	per look up table. choice	primary use	N	S	N	-	O
Building Group	Numeric, 4	none	per corporate list	N	S	N	Y	O
ALC (Accounting Location)	Numeric, 4	none	per corporate code	N	S	N	-	O
Properties District.	Alpha, 1	none	per choice from table.	N	P	N	Y	O
Property Supervisor	Alpha, 2	none	Initials of property foreman	N	P	N	Y	O
Date of Original Construction.	Date, 6 format	legal date		N	S	N	-	O

1. **Model Name:** SPARC (October, 1988)

2. **Entity Name:** BUILDING (page 2)

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary/Second Other	Foreign Key	Indexed	Derived/ Original
Date of Acquisition	Date 8 format	legal date	date property purchased	N	S	N	–	O
Effective Date.	Date 8 format.	legal date	none	N	O	W	–	O

1. Model Name: SPARC (October, 1988)

2. Entity Name: FLOOR

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary/Second Other	Foreign Key	Indexed	Derived/ Original
CLLI	AN, 8	~~None~~ Mandatory, unique	Consistent w/ Corporate. xcpt bldgs in storage	K	P	Y	Y	O
FLOOR	N. 3	~~none~~ mandatory, unique	~~8~~-3 ≤ 30	K	P	N	Y	O
Total Usable Area	N 6.1	None	Less than Construction	N	P	N	N	O
Total Rentable Area	N 6.1	None	More or equal to Rentable	N	S	N	N	O
Construction Area.	N 5.1	None	More or equal to Rentab	N	S	N	N	O
Effective Date	Date Format	Any legal date ~~None~~	Any legal date ~~between~~ 10/31/88 and today	N	O	N	N	O

1. Model Name: SPARC (October, 1988)

2. Entity Name: SPACE (~~page~~ 1)

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary/Second Other	Foreign Key	Indexed	Derived/ Original
CLLI	AN, 8	mandatory unique,	Consistent with Corporate xcept buildings in Acreage	K	P	Y	Y	O
FLOOR.	N, 2	mandatory ~~none~~ numeric unique	≥ 3 ≤ 30	K	P	Y	Y.	O
SPACE INDICATOR	A 2	mandatory ~~non~~ / unique	per plan cross reference.	K	P	N	Y.	O
DELETE/ADD/ MODIFY	A 6	D, A, M, or blank	delete, add, modify	N	O	N	N	O
SPACE CATEGORY	B, N	B, N. default B.	Boma or non-Boma	N	S	N	N	O
SPACE IDENTITY	A, 2	Alpha. none	-	N	P	N	N	O
OWNERSHIP CATEGORY	O, T, F	O, T, F choice	-	N	P	N	N	O
ACCOUNT CODE	6 Numeric	numeric.	per corporate acct codes.	N	S	N	N	O
BUSINESS UNIT CODE	2, AN	none	per Corporate acct codes	N	S	N	N	O
REGION/ DISTRICT	AN, 2	none	none	N	S	N	N	O

1. **Model Name:** STARC (October, 1988)

2. **Entity Name:** SPACE (page 2)

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary/Second Other	Foreign Key	Indexed	Derived/ Original
Usable Area	Numeric 6.1	numeric	≤ Rentable Area	N	P	N	N	o
Rentable Area	Numeric 6.1	numeric	≤ Construction Area	N	P	N	N	o
Equating Factor	Numeric 1.2	numeric	for accounting ≨0 ≤ 1.25	N	S	N	N	o
Surplus Area	Numeric. 7	numeric	less than usable.	N	S	N	N	o
Measure of Quality	C, D, M, E, L	Choice from options	–	N	S	N	N	o
Occupancy Code	V, R, G, W, G, C O, S, F, M, U T	Choice from options	–	N	P	N	N	o
Date of Measure	Legal date	legal date.	–	N	S	N	N	o
Services Code	~~Text~~ 6	Numeric or ~~to~~blank	707000, 604600.	N	S	N	N	o
Settlement Code	1 to 7 numeric	Choice.	Choice.	N	S	N	N	o
Lessor	AN Text	none.	name of leasing org.	N	S	N	N	o

1. **Model Name:** SPARC (october, 1988)

2. **Entity Name:** Space (page 3)

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary/Second Other	Foreign Key	Indexed	Derived/ Original
Lease Rental Code	numeric or blank.	none	legitimate acct code when present	N	S	N	N	O
% of Invoice	N [illegible]	≥ 0 ≤ 100	between 0 & 100% ∝ % space leased.	N	S	N	N	O
Position Code	Numeric	none	consist. w/ Corporate codes.	N	S	Y	Y	O
Effective Date	Legal Date	Legal date	between Oct 31/88 and present.	N	O	N	N	O

1. Model Name: SPARC (October, 1988)

2. Entity Name: REAL ESTATE CO-ORDINATOR

Attribute Name	Permitted Values	Specified Edit Rules	Understood Edit Rules	Key/ Non Key	Primary/Second Other	Foreign Key	Indexed	Derived/ Original
POSITION CODE	Numeric, 8	~~optional~~ mandatory unique	Consistent w/ corp. rules.	K	P	N	Y	O
LAST NAME	Text, 30	none	surname	N	P	N	N	O
FIRST NAME	Text, 10	none	-	N	P	N	N	O
ACCOUNT CODE HOLDER	~~Numeric~~, 30 Text	none	person responsible for budget	N	P	N	Y	O
CO-ORDINATOR LOCATION	Text, 10	none	-	N	S	N	N	O
COORD. ADDRESS	Text, 30	none	-	N	S	N	N	O
COORD. CITY	Text, 20	none	-	N	S	N	N	O
COORD. PHONE	~~Text~~ 8 Numeric	~~none~~ formatted string	-	N.	S	N	N.	O

B. ENTITY DATA COLLECTION SHEET

	Original	Revised
1. Model name and date:	DOFS	SPARC 1
2. Entity name:	—	BUILDING
3. Number of attributes used for keys:		1
4. Number of attributes which are foreign keys:		1
5a. Number of primary attributes for this entity:		6
b. Number of secondary attributes:		5
c. Number of other attributes:		1
6. Number of indexed attributes:		4
7. Is this entity primary or secondary to the application?		P

8. Describe any differences or disputes in definition or understanding across the organisation for this entity. Yes, significant variation with respect to the inclusion criteria: building, portable building, building in storag buried building. Real buildings vs. building on the books.

9. Criteria for inclusion (if known, or specified).

see above

B. ENTITY DATA COLLECTION SHEET

		Original	Revised
1.	Model name and date:	DOFS (1984)	SPARC1 (1988)
2.	Entity name:	FLOOR	FLOOR
3.	Number of attributes used for keys:	1	2
4.	Number of attributes which are foreign keys:	0	2
5a.	Number of primary attributes for this entity:	8	3
b.	Number of secondary attributes:	13	2
c.	Number of other attributes:	6	1
6.	Number of indexed attributes:	0	2
7.	Is this entity primary or secondary to the application?	P	P

8. Describe any differences or disputes in definition or understanding across the organisation for this entity.

9. Criteria for inclusion (if known, or specified).

Not specified but problematic in some areas such as mezzanines, crawl spaces, mechanical floors.

B. ENTITY DATA COLLECTION SHEET

	Original	Revised
1. Model name and date:	DOFS	STARC I
2. Entity name:	SPACE	SPACE
3. Number of attributes used for keys:	1	3
4. Number of attributes which are foreign keys:	0	2
5a. Number of primary attributes for this entity:	8	8
b. Number of secondary attributes:	19	13
c. Number of other attributes:	16	1
6. Number of indexed attributes:	0	4
7. Is this entity primary or secondary to the application?	P.	P

8. Describe any differences or disputes in definition or understanding across the organisation for this entity. ongoing debate about which kind of space should be coded and stored as a seperate record, or included in the rentable area.

9. Criteria for inclusion (if known, or specified).

see above. Some ongoing judgment on the part of staff re: how far to disaggregate the space

B. ENTITY DATA COLLECTION SHEET

	Original	Revised
1. Model name and date:	DOFS	SPARC(1)
2. Entity name:		REAL ESTATE COORDINATOR.
3. Number of attributes used for keys:		1
4. Number of attributes which are foreign keys:		0
5a. Number of primary attributes for this entity:		4
b. Number of secondary attributes:		4
c. Number of other attributes:		0
6. Number of indexed attributes:		2
7. Is this entity primary or secondary to the application?	–	S

8. Describe any differences or disputes in definition or understanding across the organisation for this entity. Previous function of the coordinator was lost during reorganization. Some doubt about how this entity serves the application.

9. Criteria for inclusion (if known, or specified).

Only loosely described, and likely to change.

C. RELATIONSHIP DATA COLLECTION SHEET

1. Model Name: SPARC 1

2. Relationship Name: Building Contains Floor.

3. Entity 1: BUILDING

   Entity 2: FLOOR.

   Other entities:

4. Describe the criteria for this relationship.

   All floors must belong to a building.

C. RELATIONSHIP DATA COLLECTION SHEET

1. Model Name: SPARC 1

2. Relationship Name: Floors Contain Space

3. Entity 1: FLOOR

   Entity 2: SPACE

   Other entities:

4. Describe the criteria for this relationship.

   All spaces belong to a floor in the building.

C. RELATIONSHIP DATA COLLECTION SHEET

1. Model Name: SPARC 1

2. Relationship Name: Real Estate Coordinators are Responsible for Space

3. Entity 1: SPACE

Entity 2: REAL ESTATE COORDINATOR.

Other entities:

4. Describe the criteria for this relationship.

Every space has a Real Estate Coordinator responsible for it.

D. BACKGROUND AND SUMMARY OF MODEL DATA

1. Organisation: AGT REAL ESTATE SPACE PLANNING/ADMIN.

2. Model Names and dates: DOFS (1984) - SPARC (1988)

3. General application area: Building space inventory, with implications for cost allocation, and revenue sharing.

4. Date(s) of revision(s): 1988.

5. Original reason for developing model: To track space allocated to different users. Secondary objectives - to support building operating cost allocation and revenue sharing.

6. Systems in place prior to model: DOFS - prior to DOFS not known.

7. File layouts of previous automated application available?________ If so, please attach.

8. Documentation standard and technique used: Narrative. No technical documentation. No diagrams.

D. BACKGROUND AND SUMMARY OF MODEL DATA

For the original model and its major revision complete the following, where possible:

	Original	Revised
9. Model author(s):	Unknown	MMM MANAGEMENT Assoc
10. Scope of modeling effort: (sub-application, application, divisional,or organisation wide)	unKnown	application
11. Estimate of modeling effort (in man days):	unKnown	est (3 - 4 weeks)
12a. Overall level of normalisation:	poor	reasonable
12b. Consistency of normalisation:		
13. Number of entities:	2	4
14. Total number of attributes:	70	48
15. Number of named relationships:	none	3.

D. BACKGROUND AND SUMMARY OF MODEL DATA

16. Describe any update process to the original model:

Update process done under extreme time pressures .

17. Comment on the degree and quality of management participation.

Limited management participation of average quality.

18. Comment on the degree and quality of user participation.

Moderate user participation of average quality.

D. BACKGROUND AND SUMMARY OF MODEL DATA

19. List any entities, attributes and/or relationships which have been the subject of marked differences in definition in the organisation and describe.

"Building."

20. Has your organisation attempted to integrate data models or data bases from different parts of the organisation? If so, please note any difficulties which this process had to overcome.

21. Which areas in these models have been particularly unstable?

Area definitions have changed most radically.