Keizersgracht 557
1017 DR AMSTERDAM
Netherlands
(020) 224243

August 1981

DATA ANALYSIS - The answer to successful implementation of IMAGE

by RICHARD IRWIN

Presentation to the HP3000 International
Users Group

1981 Berlin International
Meeting Germany

ABSTRACT

IMAGE is now considered to be technically one of the most successful DBMS's. However, as with other DBMS's it suffers badly when poor design tools and methods are used. Data Analysis is an essential part of a concept that is growing rapidly in Europe and America.

This paper describes the Data Analysis process; how to begin. Defining data areas and data resources. The concept of entities attributes and relationships. The degree of relationships, one-to-one, one-to-many, many-to-many. Types of relationship, e.g. optional, involuted, multiple. The use of the data model as a learning aid to the analyst and to communicate with non-dp users. Life cycle, sub-type and time dependent entity roles. The interactive part of data modelling. Mapping to the logical database. Distinction between conceptual, logical and physical stages of development. Overcoming general structural limitations of IMAGE.

This paper is written and presented by Richard Irwin, a Senior Consultant with Database Consultants Europe BV who has spent the past five years analysing, designing and implementing IMAGE/3000 systems.

1. INTRODUCTION

This paper is primarily concerned with the system development cycle in a business environment. This does not mean however that data analysis should not be performed in other environments, e.g. - scientific - but in order to demonstrate its usefulness, a specific area has been chosen. The paper has also concentrated on the logical construction of IMAGE and using an HP3000, but many aspects can be seen to apply to other file systems and ranges of hardware.

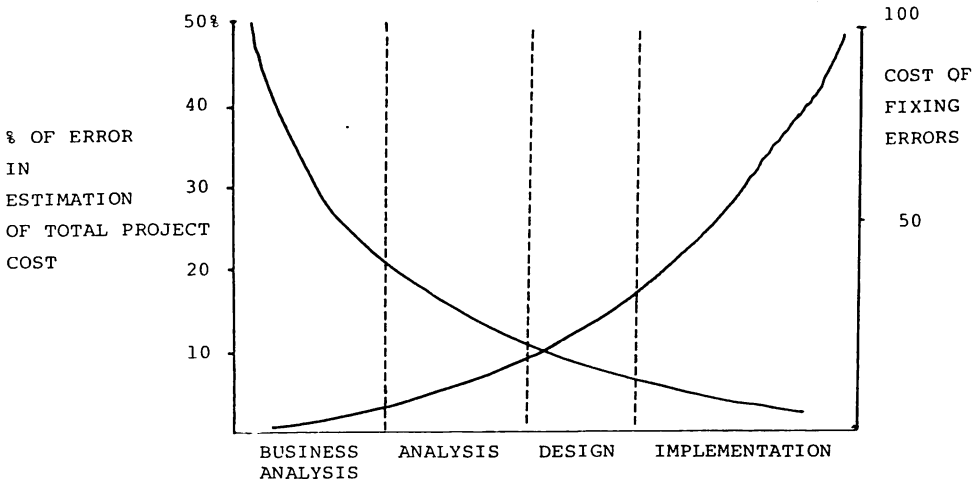
2. WHY USE A METHODOLOGY?

A 'method' is a procedure for carrying out a certain task. A 'methodology' is an integrated set of procedures, founded on consistent basic principles, which provide a complete framework within which a given task can be performed. A methodology is used to perform these basic functions.

2. 1. Highlighting of problems at an early stage.
The development process is structured to allow critical management, user and technical decisions to be taken at the right time, i.e., as early as possible. (see Figure 1 over)

Figure 1

(C.G. Davis "Requirements Problems in large real-time systems development")



2. 2. Providing a means of communication.
Check point facilities provide a means of communication between all levels of personnel concerned with the project.

2. 3. Proof of progress.
DP management is under constant pressure to show results. Without a methodology all we do is push for early system completion, thus instead of the project being time-shared as in Figure 2 (over),

we save time in the analysis, resulting in Figure 3.

Figure 2 IDEAL

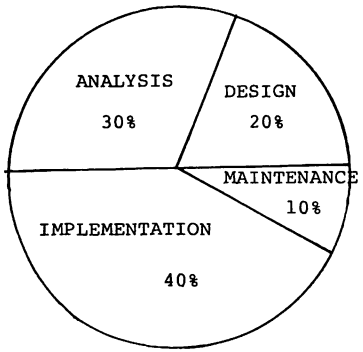
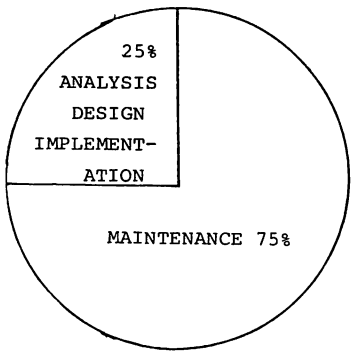


Figure 3 REAL



With a methodology however, we can prove our progress at each step by producing checkpoint documents. A methodology must therefore provide:

- guidelines which ensure that we don't overlook things (not rules as they are too inflexible)
- an approach which is top-down or outside-in and modular
- easily understood diagrams for communication
- standards for use and documentation

3. DATA - A VALUABLE RESOURCE

For years the value of data was grossly under-estimated. This meant that the emphasis was on the application approach, where, for each application, the data would be defined again resulting in the following difficulties:

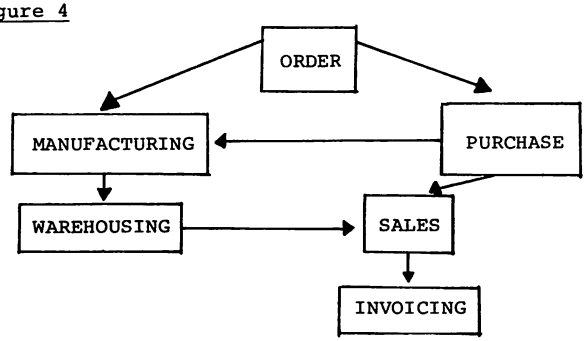
3. 1. Duplication of data
  - inconsistencies of value, timeliness and meaning
  - cost of storage

3. 2. Consolidation across applications
  - file collation
  - proliferation of work files
  - integration of processing

3. 3. Lack of control
  - satisfying new application requirements
  - availability and use of data

When the introduction of the database philosophy came, it was not necessarily a philosophy centred around Database Management Systems but more the acceptance of the need to share data. Data is the basis of information flow across the functional boundaries within an enterprise as represented in Figure 4.

Figure 4



The definition of a database should be:
an organised, integrated collection of data which
- is structured to reflect the real world of the enterprise

- is stored independently of programs which use it
- satisfies the requirements of multiple user application

or all of the above may be summarized quite simply by defining a database as "a common pool of shared data".

However, new problems soon became apparent to the designs of early database systems:

- lack of procedures to make and document critical design decisions
- development of single-application oriented databases
- adoption of a bottom-up approach to data analysis meant that the designs became inflexible
- failure to fully exploit the rôle of data dictionary/directories
- no allowance (or design) for database recovery or re-organisation
- the database project was usually seen as a file conversion exercise

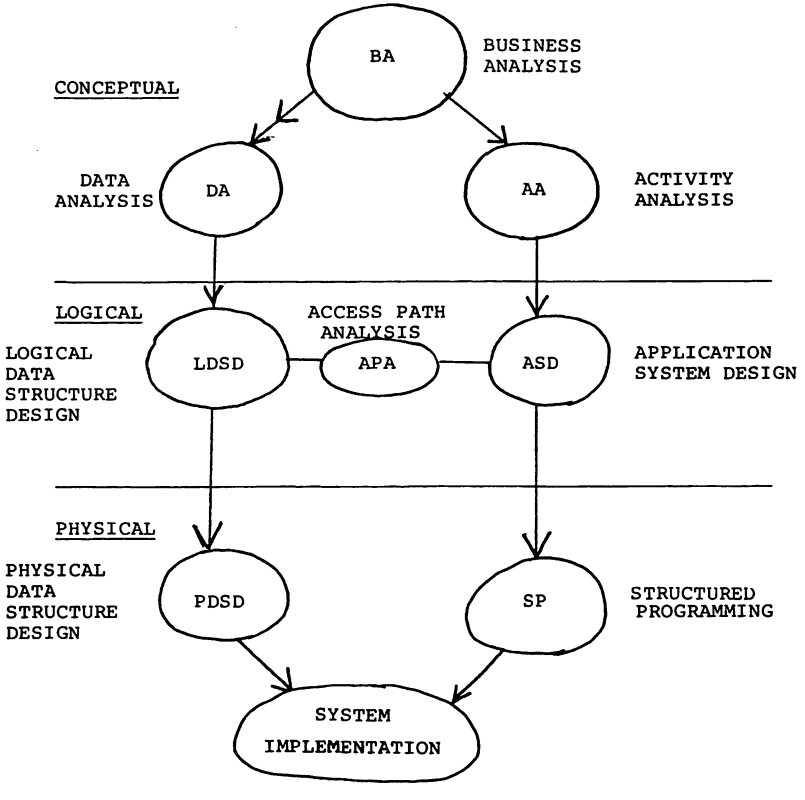
To summarise - the database philosophy evolved from a need to share data, but whilst there are many benefits for the use of a database there are also pitfalls in the development stages.

4. WHAT IS DATA ANALYSIS?

Data analysis is part of the systems development cycle as shown in Figure 5.

Figure 5

The Systems development cycle



The objective behind each part is as follows:

METHOD	OBJECTIVE
Business Analysis (BA)	To define the business area boundaries for analysis needs, at a high level
Data Analysis (DA)	To analyse the data resources
Activity Analysis (AA)	To define the users' information handling processes
Logical data structure design (LDSD)	To map the data model to the logical data structure
Application system design (ASD)	To translate the user information handling processes into a technical application system design

The most important factor to note is the early split between data and functions. Most of the emphasis in other system development methodologies has been on the functional side, typically on the programming effort. Although programming errors are one direct cause of costly and inflexible systems many of the errors can be traced back to errors in the analysis and design stage.

What is fundamentally wrong with many approaches is that no method exists for analysing and describing in a concise, user-oriented way, the business data and how it operates, divorced from any considerations of how the system will eventually be designed.

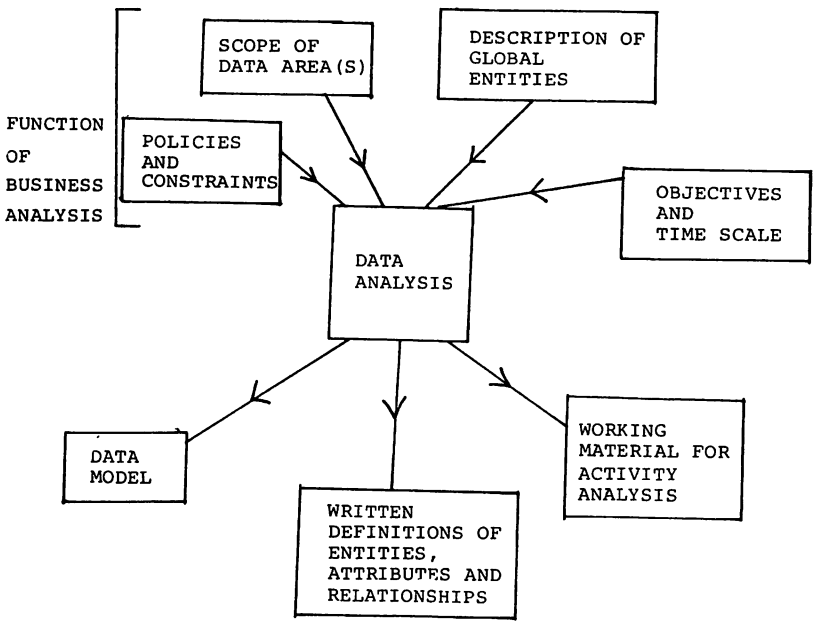
It is often desirable, and should be possible to analyse a business without any prior constraints on how parts of the business are to be computerised and which business functions will form the basis of computer systems.

In many approaches other than Data Analysis, the emphasis is placed on determining and analysing the "output required", (i.e. listings, reports, computer files, etc. - a dangerous practice in itself, as information requirements are never static) and then expressing the results in terms of computer files, English narrative descriptions (often long and complex) of the "processes required", and technical flowcharts of the flow of data through the system. The results of this approach are apparent - inflexible systems which are not resilient to change and whose development is often unco-ordinated and fraught with problems. The underlying cause is that the business was never fully understood before the design stage.

One approach which seeks to remedy this lack of an analysis methodology is known as data analysis. Database Consultants Europe BV (DCE) has sucessfully used this technique for a number of years during which time the initial concept has been developed into a complete analysis and design methodology.

Data analysis is a method used to understand and document a complex environment in terms of its data resources. The results of data analysis are summarised in a diagram known as a data model. Detailed results are documented on specially designed forms. The input and output of Data Analysis can be summarised as in Figure 6.

Figure 6



5. WHERE TO BEGIN?

To initiate the task of Data Analysis the following tasks should be performed:

5. 1. Gaining support
- from management
- data processing staff
- user departments involved

5. 2. Definitions
- of the objectives
- of the timescale
- terms of reference

5. 3. Design and acceptance
- Data Analysis standards as compared with existing standards
- Documentation

5. 4. Anticipation of possible unfortunate discoveries
- irreconcilable coding systems
- inconsistent existing data
- incorrectly interpreted reports

5. 5. Education and training
- theory and methodology
- detailed procedures.

6. DATA AREAS AND RESOURCES

When choosing the data area for analysis, it should be small enough to be manageable and not too complex that an overview cannot be attained. There should be clear cut definable boundaries with a minimum of interaction with other areas. It should be independent of, or well defined in terms of, specific applications. There should also be a business requirement for applications to be implemented in that area, or for improvements to be made to existing application. The data resources can be categorised in the following way:

- Personal knowledge and ideas
- Clerical records
- Manually produced reports
- Correspondence
- Computer files
- Other computer readable data
- Computer produced reports

7. DEFINITION OF ENTITIES, ATTRIBUTES AND RELATIONSHIPS

Within data analysis, there are three major components:

7. 1. Entities

An entity is something of fundamental importance to a company. It is thus something about which

data will probably be kept in an information handling system, e.g. objects, people, places or abstractions such as events.

7. 2. Attributes

An attribute is a basic unit of information which describes an entity. Within the company environment, an attribute cannot usefully be sub-divided into other units of information. An entity must have attributes if it is of interest to the company, e.g. the entity "insurance policy" could have the attributes policy number, date policy started, person's name, person's date of birth.

7. 3. Relationships

A relationship is an association between entities, e.g. the entity 'order' is related to the entity 'order line', the entity 'car' is related to the entity 'part'.

There are no theoretical rules which can be applied to decide when an object is worth being an entity or attribute until the company environment is known to the analyst. If the object is not of fundamental importance to the enterprise then it is not worth keeping information about it. For example, can a building be an entity? In the case of a company which simply exists in one building the answer could be 'no'. However, in the case of a construction company or an electrical installation company the answer would most definitely be 'yes'.

A similar example for attributes is a 'person's weight'. In the case of a vacuum cleaner sales company the answer would be that a 'person's weight' would not be an attribute but for a hospital it would.

8. ANALYSIS OF ENTITIES

In order to recognise entities it is important to ask what data or objects are within the chosen area(s). A first pass of definitions of entities and preferably their distinguishing or key attributes is made. The key point to remember is that the focus is on entities not processes. However it is important to ask what events take place (e.g. job offers) in order to identify entities which are abstractions. It is advisable to check all input and output reports for other possible report entities.

9. ANALYSIS OF ATTRIBUTES

Once the major entities have been identified, determination of relevant attributes is performed by examining:

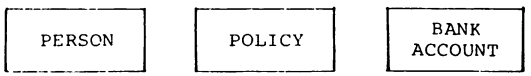
- Manual files
- Documents
- Decision criteria (to identify implicit selection or distinguishing attributes)
- Computer files

For new entities it will be necessary to ask when data is needed to be kept about that entity.

10. PICTORIAL REPRESENTATION

10. 1. Entities

Entities are represented by a rectangular box with the name of the entity written inside the box.



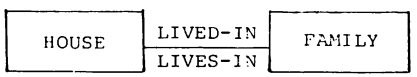
In the early global data analysis phase, only the entity name is written inside the rectangle. However, when doing the detailed data analysis it is sometimes convenient to include also the names of the identifying attributes.

10. 2. Relationships

Relationships are shown by drawing a line between the entities, also showing the degree of the relationship. An abbreviated name of the relationship can be written alongside the line. (NB. entity names and relationship names are read in a clockwise direction).

10. 2. 1. One to One (1:1)

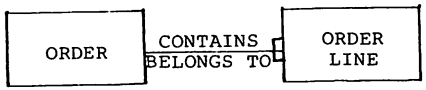
The One to One degree of relationship is represented by



i.e. One house is lived in by one family and one family lives in one house.

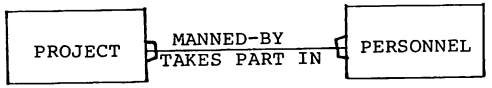
10. 2. 2. One to Many (1:n)

One to many degree of relationship is represented by



i.e. One order contains many order lines and one order line belongs to one order.

10. 2. 3. Many to Many (n:n)

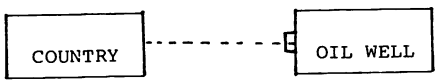


One project is manned by many personnel and one person can take part in many projects.

11. FORMS OF RELATIONSHIP

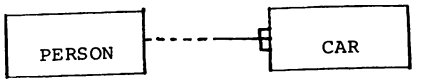
Relationships can take many forms

11. 1. Optional Relationships



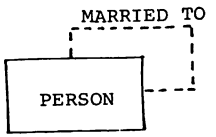
i.e. A country may or may not contain oil wells.

11. 2. Partially optional



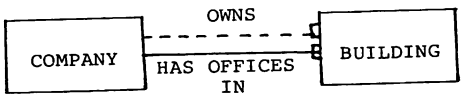
i.e. A person may own none, one or many cars but a car must be owned by a person.

11. 3. Involuted relationships



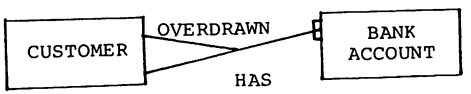
An involuted relationship is a relationship between occurrences of the same entity type, e.g. a person may be married to another person.

11. 4. Multiple relationships



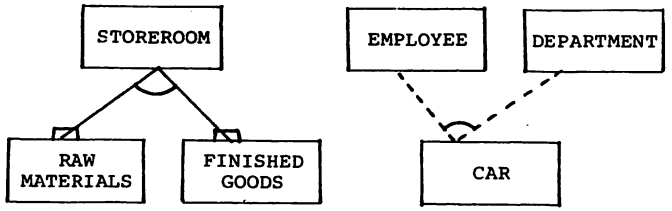
i.e. A company may own many buildings and must have offices in many buildings.

11. 5. Inclusive relationships



An entity can participate in the one relationship - "overdrawn" only if it also participates in the other relationship - "has".

11. 6. Exclusive relationships



An entity occurrence may participate in any one, but not more than one, of a number of alternative or exclusive relationships, e.g. a car must be owned by either an employee or a department, not both.

12. THE DATA MODEL AS A COMMUNICATION TOOL

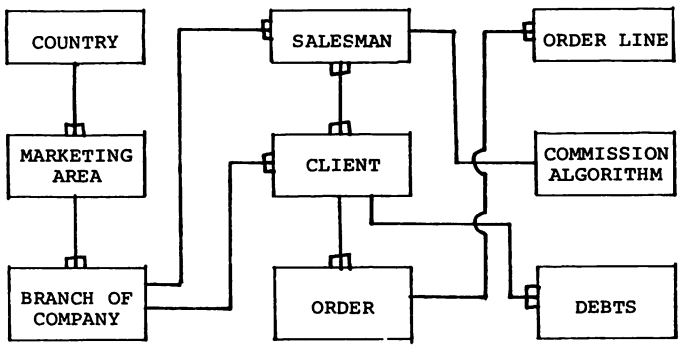
It is imperative for any methodology to be able to use the analysis results for communication purposes. Since communication must take place between the user of the information and the technician who is performing the analysis, it is also necessary to keep the pictorial representations as simple as possible to avoid misunderstandings and to enhance co-operation. No user has the time or inclination to sit down and discuss 50 or 60 pages of documentation. It should also be remembered, therefore, that only those parts of the picture which pertain to the user's direct area of interest should be brought to him for discussion.

At all times the picture should represent the real world of data and its relationships.

A situation which must be avoided is the tendency to think in computer terms instead of user terms. It is also important to keep the representation in the simplest possible form.

Figure 7

Example data model - first pass



It should be noted that a hierarchical structure has been deliberately avoided so that relationships can be thought about more easily. Obviously models become messy and have to be redrawn when too many lines of relationships are involved.

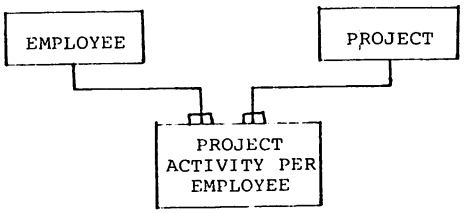
13. PRACTICAL PROBLEMS

13. 1. Many to many relationships

When a many to many relationship occurs it usually means that another entity can be identified between the two entities.

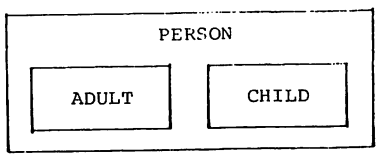
Example

i.e. An employee works on many different projects and projects have many different employees. Because it is useful to have the attribute 'length of time of employee on project' we are obliged to create the following situation:



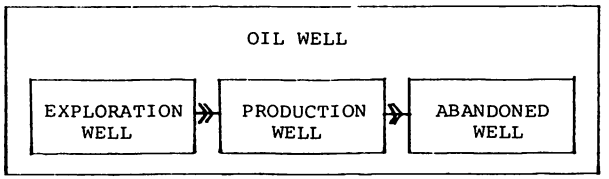
13. 2. Entity rôles

Entities which are similar but which have slightly different attributes depending on the value(s) of certain classifying attribute(s), are probably entity 'roles'. An entity rôle is a sub division of an entity type which is difficult to separate from the entity type with which it is associated. E.g. A 'person' entity may be subdivided into 'ADULT' or 'CHILD'. This would be represented in the following way:



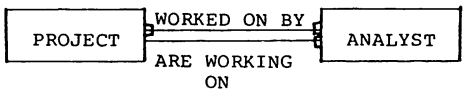
13. 3. Life-cycle rôles

Life-cycle rôles are special cases of sub-type rôles in which a sequence exists between the sub-types. This sequence is shown by a double arrow on the connecting line.



13. 4. The Problem of Time

Most time problems can be represented simply by showing the multiple relationships of present and past, e.g.



14. ACCESS PATH ANALYSIS

Access path analysis could easily be a large enough topic for discussion within a separate paper. In this paper the subject is summarised to demonstrate its relevance. To perform access path analysis the following tasks are performed.

14. 1. For each access path, the entity types are listed in the order they are needed for a particular function.

14. 2. The selection criteria are recorded in terms of relationships and attributes.

14. 3. It has to be recorded whether each entity attribute type is retrieved, modified, created or deleted.

14. 4. It is also necessary to record any relationships created, modified or deleted.

Example of Access Path Analysis.

ACTIVITY ORDER ENTRY

ACTIVITY DESCRIPTION

Function 1 An order is received by telephone. The depot that will make the delivery is selected depending on whether the goods are bulk or packaged. The order is recorded, and related to the delivery point and the depot.

Function 2 The goods specified in each order line are validated. The order lines are recorded, linked to the goods and to the order or back-order as appropriate.

RESULT

Function 1
First entry point Delivery point- retrieved, selected by delivery point name.
Depot - retrieved, selected by bulk or package relationship.
Order - Stored, related to delivery point and depot.

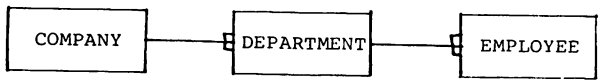
Function 2
Second entry point Product - retrieved, selected by product code.
Stock - retrieved, selected by relationship with product and depot and updated.
Order line - stored, related to order and product.

15. MAPPING TO IMAGE

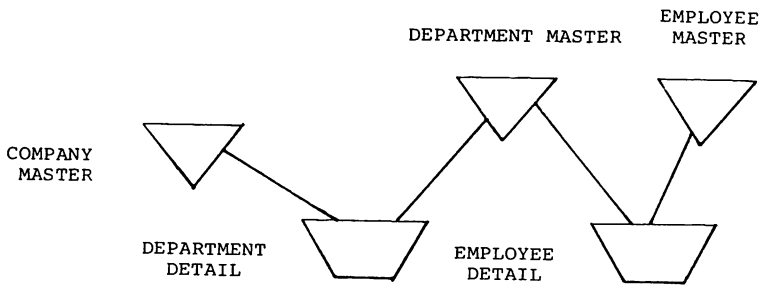
The activities of data modelling and activity analysis are performed as far as possible without reference to implementation techniques available in IMAGE. The process of access path analysis does, however, identify alternative methods of achieving the required result, besides showing that occasionally the data model is deficient or inconvenient for handling some functions. At this point it helps to understand the range of logical structures available, in order to consider the alternative methods of representing entities in the Database, together with their attributes and relationships. IMAGE provides the capability to implement relationships through the use of PATHS though in practice there can be many restrictions on using that PATH. The simplest way to convert entities to logical records is to create one record type for each entity type, i.e. creating a DATA SET or a number of DATA SETS for an entity. Attributes become DATA ITEMS. It may be necessary to divide attributes across more than one DATA SET to provide efficient access to the most frequently used attributes or to combine several entities into one logical record. Any such changes should be checked against the data model.

16. TYPICAL STRUCTURAL PROBLEM

A typical example of a problem presented by IMAGE is the following.



This would have to be represented in IMAGE by



Although it is an inconvenience to have to adjust the design to fit into these kind of circumstances, IMAGE has proved to be very simple for interpretation when looking to the ease of the DBSCHEMA.

17. PERFORMANCE CONSIDERATIONS

Performance considerations are quite often outside the scope of data analysis particularly as performance is usually dependent upon volume. In a high transaction volume system it is worth considering the possibility of splitting entities into sub-entities to reduce the volume in each set and perhaps even reducing the necessity for an access PATH.

18. CONCLUSION

Data analysis provides a good communication tool between the user wishing to understand his system and the analyst wishing to understand the user data. It provides a methodology which is flexible enough to adjust to new environments, not a checklist of standards which are inflexible to change.

IMAGE is successful mainly because it is simple to understand. With an easy-to-understand methodology and DBMS, implementation of systems becomes a smoother process with involved, motivated users and a database ready to cope with future demands.