FROM DATA ANALYSIS TO TurboIMAGE DESIGN

Jos Witteveen, Glenn Pereira
Database Consultants europe BV.
Amsterdam, Netherlands

ABSTRACT

Because information is one of the company's most valuable assets in any organisation and the availability of information one of the highest priorities, good Data Analysis and proper Database Design have become critical success factors in the system development environment today.
This paper will describe how Data Analysis, Conceptual Access Path Analysis and Logical TurboIMAGE Database Design should be carried out in order to meet the information requirements of an organisation.

INTRODUCTION

TurboIMAGE/3000 has been developed to replace IMAGE/3000 in order to provide increased performance and functionality.
The database architecture remains as a two-level network structure with data set relationships between owners and members. It enables the HP/3000 DP professional, working in an environment in which data is shared across many people, departments and applications, to develop databases which can offer a number of significant advantages, such as :

- Data integrity in an online environment
- Central control of data
- Scope for future development
- Recovery facilities
- Easy-to-use application development tools

These advantages, however, can only be realised if the database design is carried out correctly with due regard to the natural structure of the data and the way it has to be accessed and maintained. The risk of getting it wrong is high and the costs of putting it right afterwards can be enormous.

ENVIRONMENT

A DATABASE is simply defined as a 'COMMON POOL OF SHARED DATA'.
Recognition of the fact that the organisation exists in a shared-data environment is the pre-requisite for building a database. In a shared-data environment the following principles apply:

1. Data exists in its own right and has relationships with other data.
2. Data is not exclusively owned by any department or application area, it must be available to any authorised person or function within the organisation.
3. Data has to be organised within the computer in a manner that reflects its real-world existence.
4. The objective is to hold each piece of data only once in the database files with exceptions allowed only for good reason and subject to appropriate control.

5. Applications are built with due regard to the need for control and synchronisation of data.

Figure 1 shows the System Development Cycle required in a Shared-Data, Interactive environment. The right side of the diagram shows the steps which are commonly used in Computer System Development ie. Activity Analysis, Application System Design and Structured Program Design. The left side shows a similar cycle of events for analysing the data (Data Analysis), designing the data structure within the limitations of the DBMS structuring rules (Logical Database Design) and the final physical design of the database (Physical Database Design). The centre of the diagram demonstrates the steps required to analyse how the activities (and eventual computer processes) need to use the data.

All design begins with proper and thorough analysis. It is necessary to collect the information using structured analysis techniques and feed the results into a structured, systematic design procedure. Before database design we are concerned with:

- the structure of the data, Data Analysis
- the way this data is used, Access Path Analysis.

DATA ANALYSIS

Trying to understand the structure and the characteristics of data has become a science in itself. It is widely practised today using a number of different techniques such as Data Analysis, Information Engineering and Normalisation, the latter being, perhaps, the best known.

In essence, analysis of data consists of recognising all the entities and attributes that exist within the scope of your project and, subsequently, the relationships that exist between entities.

An ENTITY is defined as anything of interest and about which data can be kept. Typically, an entity can be an object (building), person (employee), place (country) or event (flight).

An ATTRIBUTE is defined as an element that helps to describe the occurrences of one or more entities. Typically, attributes can be codes, names, amounts, prices etc.

A RELATIONSHIP is defined as an association between entities.
Whereas many types of relationship can exist, the two most common are the 'one-to-many' relationship and the 'many-to-many' relationship. Each relationship is drawn as a straight line between two entities with some symbol (arrow, crows-foot or trident) to show the 'many' of a relationship.

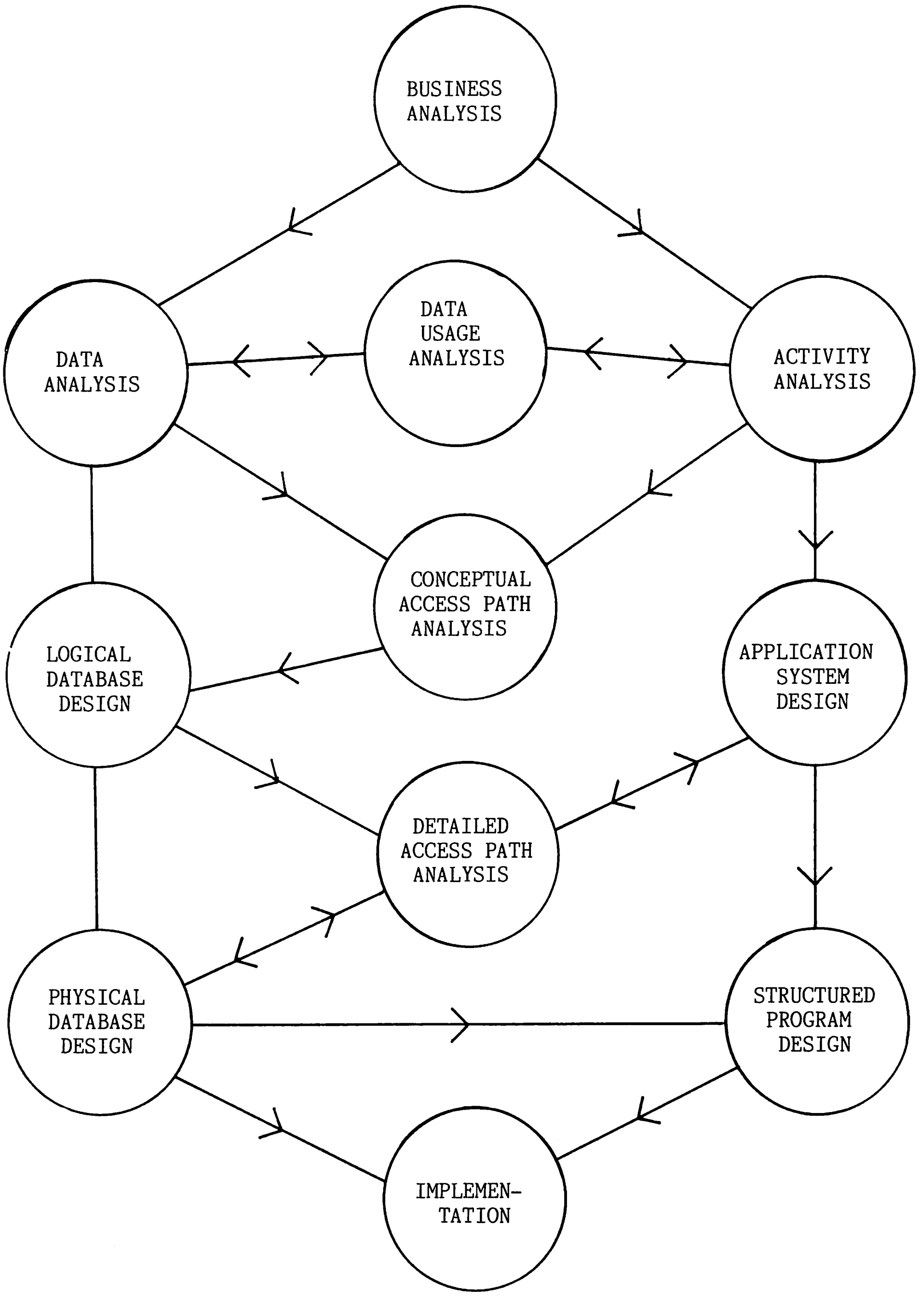

FIGURE 1 - The Systems Development Cycle in a Shared-Data, Interactive Environment.

If, for example, a customer can have many orders, but *one order can* be associated with only one customer, it may be represented as shown in Figure 2 below:

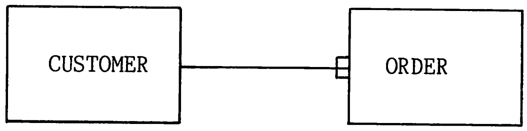

Figure 2 - The one-to-many relationship

If it was required to show a relationship between Supplier and Product where one supplier can supply many products and one product can be supplied by many suppliers, it may be represented as shown in Figure 3 below:

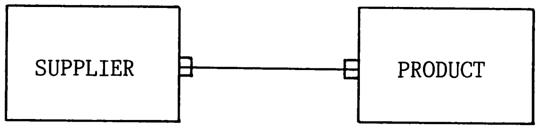

Figure 3 - The many-to-many relationship

A variation, which is commonly used, is the optional relationship. This shows that a relationship may hold in some cases, but not all. If, for example, a product may or may not be supplied by a supplier (ie. some products may be manufactured internally), the above straight line would be dotted at the product end, as shown in Figure 4 below:

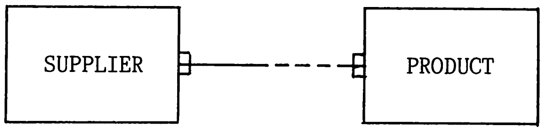

Figure 4 - Optional relationship

By identifying all the entities and relationships within the scope of your project, it is possible to build up a picture of the data in that area. This picture, which is normally called a Data Model, is a conceptual representation of the data - the real world - and it contains all natural associations.In addition to building the Data Model, all entities, attributes and relationships should be documented. This documentation, would contain precise descriptions, volumes, characteristics, special conditions etc. The attribute documentation should become, essentially, the Data Dictionary and should comply with any previous dictionaries set up within the organisation.

Figure 5 shows a Data Model for the Order Processing System.

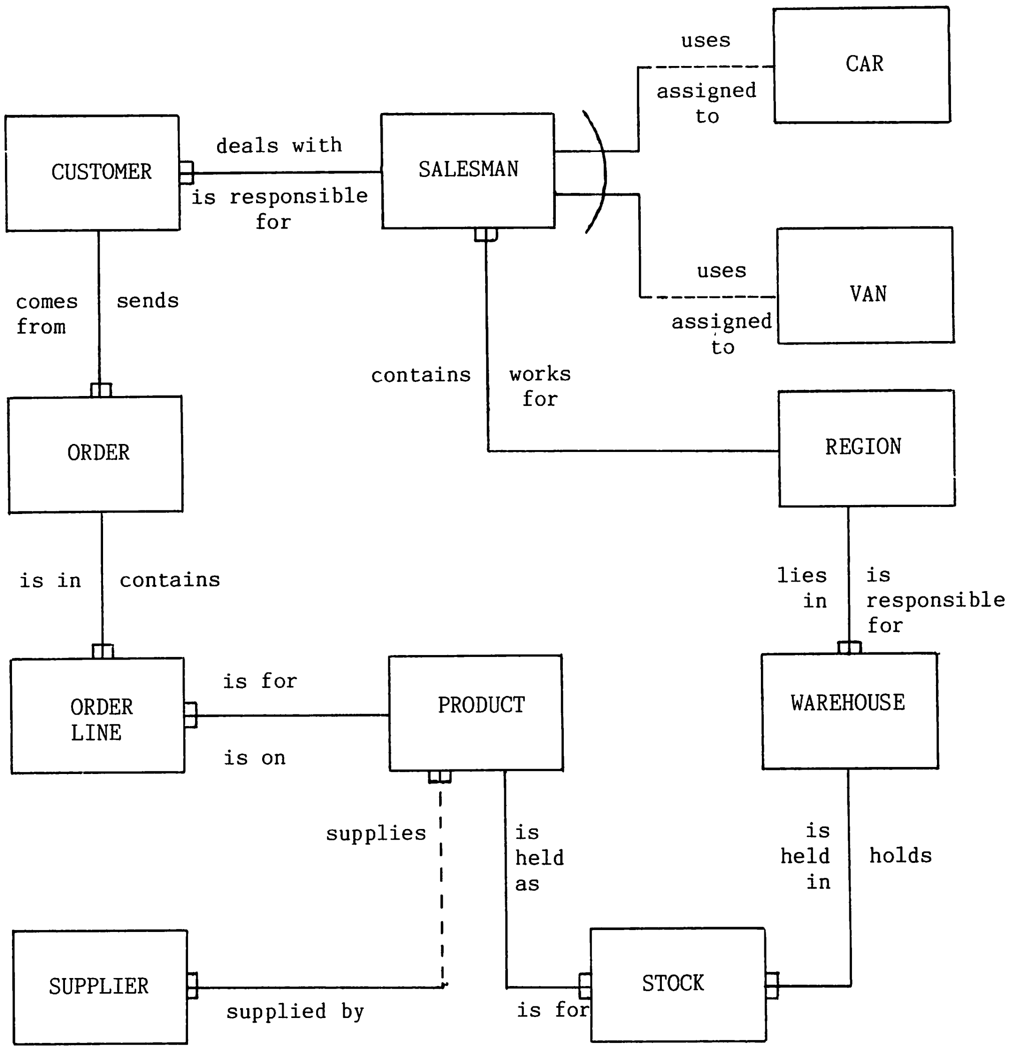

Figure 5 - Data Model.

ACCESS PATH ANALYSIS

Access Path Analysis is a technique used to identify how the data is accessed. It is carried out by drawing Access Profiles for all processes identified during the Activity (or functional) analysis.
Processes can be categorised as follows:

- RETRIEVAL Processes which, by definition, need to access data.

- MAINTENANCE Processes which need to insert, modify or delete data but, in doing so, would normally need also to access data (eg. validate that customer exists and credit status is OK before inserting an order).

The essential information required for logical database design is the access requirements ie. the ENTRY POINTS into the data structure and the NAVIGATION PATHS around the data structure. The overhead of updating the database is considered during Physical Design, when update performance will be evaluated.

An ENTRY POINT represents an initial entry into the data structure. It can consist of a search for one occurrence of an entity (normally using the unique identifier, eg. Customer Code) or a search for several occurrences (eg. all customers called Smith). In the latter case, the occurrences may be required in a particular sequence (eg. all customers called Smith in customer code sequence). Figure 6, 7 and 8 below, show how initial entry can be represented on an Access Profile. The single arrow indicates that one occurrence is accessed and the double arrow indicates that many occurrences are accessed. An 'S' next to the double arrow shows that the occurrences are accessed in sorted sequence:

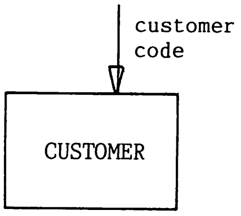

Figure 6:
Single Record
Access

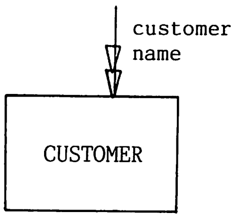

Figure 7:
Multiple Record
Access

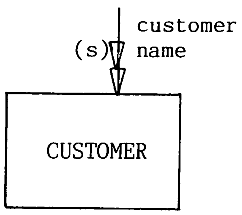

Figure 8:
Multiple Record
Access in Sorted
Sequence

A NAVIGATION PATH consists of one or more relationships that need to be used, by a process, to obtain all the data that it requires. Figure 9 shows the navigation path for a process that needs to enquire on orders, by customer, and display all the order lines in sequence.

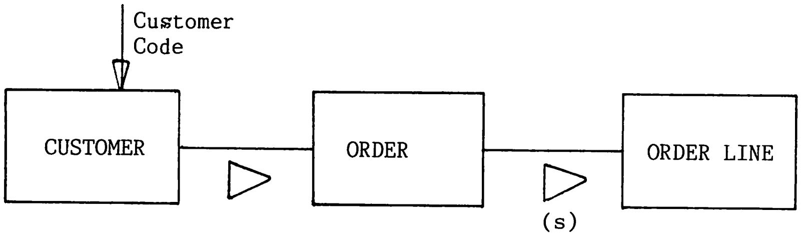

Figure 9 - Access Profile for Order Enquiry

In addition to drawing the access profiles, documentation forms should be completed for each process showing frequencies, volumes of data accessed, specific attribute usage within an entity, whether it is run in batch or on-line (if known at that stage), response time or run time constraints etc.

From this documentation it is necessary to compile a number of statistics and matrices which show the essential information required. Typically, these are as follows:

Usage Statistics	-	showing how frequently entities and relationships are used.
Usage Matrices	-	showing in which processes the entities and relationships are used and/or created.
Attribute Usage Matrix	-	per entity, showing where attributes are used and/or created.
Summary of Retrieval Access Requirements	-	showing all access paths required into and around the data structure.

LOGICAL DATABASE DESIGN

In order to separate the functional and structural design decisions from the physical design decisions, it is necessary to conduct an initial logical database design which produces a logical data structure that:

1. Is built in accordance with the TurboIMAGE structuring rules.
2. Contains all needed data.
3. Supports all access requirements.
4. Will be used as the input for physical database design.

Logical database design is carried out in two steps:

I - Refining the data model
II - Mapping to TurboIMAGE

REFINING THE DATA MODEL

The main objective of the refining process is to determine what is required of the data model, using the results of conceptual access path analysis.

During Data Analysis all relevant information is collected with regard to the entities, attributes and relationships defined within the scope of the analysis. The resultant data model should be reasonably close to the structure that needs to be mapped on to the database but would, normally, need to be refined in a number of ways as follows:

1. Eliminate many-to-many relationships from the data structure because TurboIMAGE does not support them. (In fact, very few Database Management Systems do).

   They are eliminated by the introduction of a new "in-between" entity (sometimes called a junction entity) and the creation of two one-to-many relationships in place of the many-to-many.

Figure 10 below, shows how the many-to-many relationships between Product and Supplier is expressed as two one-to-many relationships:

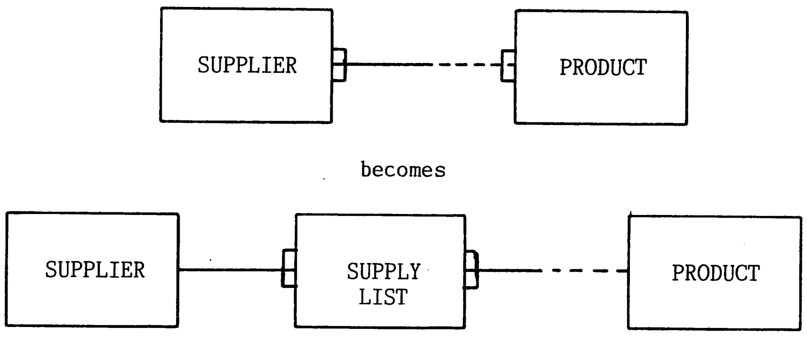

Figure 10 - Resolving Many-to-Many Relationships

Note:
The Access Paths for supplier - product need to be re-analysed.

2. Eliminate any entities, attributes and relationships which, although they belong to the conceptual Data Model, are not required for use by the computer application(s). This decision has to be made carefully, taking into consideration the following points:

   - all entities, attributes and relationships that are put into the database are likely to need maintenance to some degree.

   - due consideration must be given to future application development plans (say, next two years) and possible information requirements (new reports, ad-hoc enquiries etc.) that are not fully covered by the Terms of Reference for the application(s) currently being developed.

3. Add new relationships to the data structure if the access path analysis shows traversal between entities that are not directly related.

   For example, in the Data Model (Figure 5) there is no direct relationship between SALESMAN and ORDER (only an indirect relationship via CUSTOMER) but should there be a requirement to access ORDER from SALESMAN a new relationship could be added between SALESMAN and ORDER.

4. It may be possible to combine entities. Some judgement, based on experience, is required but two rules of thumb would be as follows:

   - where two entities are normally accessed together and their relationship, although one-to-many, is nearly one-to-one. This, typically, occurs where an order (or invoice) could have many lines or a customer could have many debtor numbers but, in practice, they almost always have one.

   - where two entities are very similar in terms of their attributes and they have the same relationships. For example, in our Data Model, Car and Van could be combined into one entity called Vehicle.

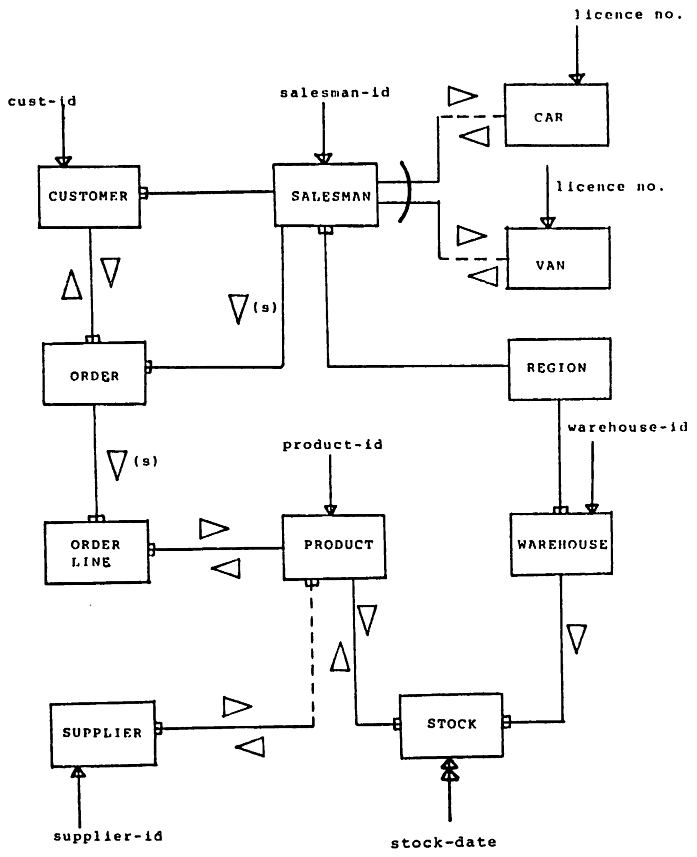

Figure 11 - Summary of Retrieval Access Requirements

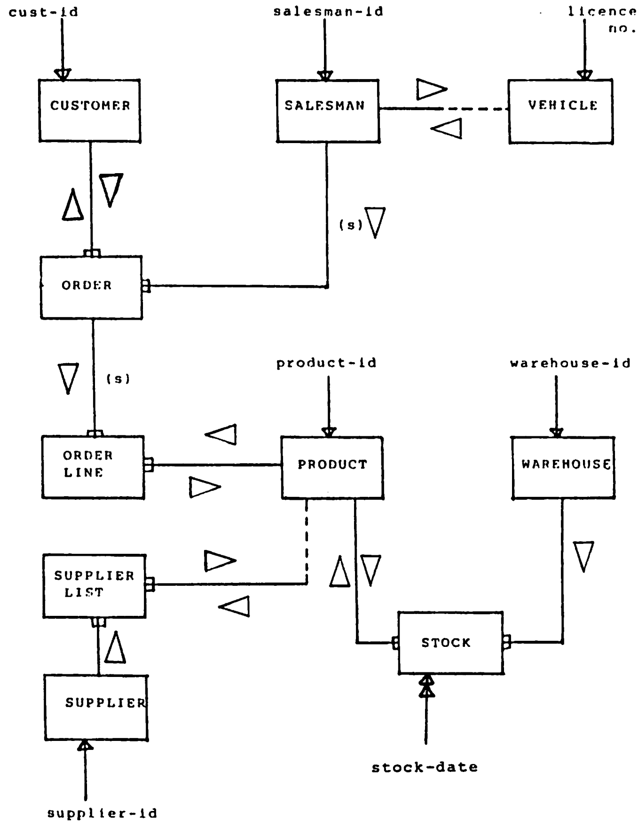

Figure 12 - Refined Data Model +
Summary of Access Requirements

In order to carry out the steps, outlined above, the information from the Data and Access Path Analysis is required.

Figure 11 shows a Summary of Retrieval Access Requirements for the Order Processing Data Model.

Figure 12 shows the Refined Data Model which has resulted from applying the above steps.

- The Supplier/Product relationship has been expanded into two one-to-many relationships.
- The Region entity has been eliminated.
- The relationships Region/Salesman, Customer/Salesman and Region/Warehouse have been eliminated.
- The relationship Salesman/Order has been added.
- The entities Car and Van have been combined to give a new entity called Vehicle.

MAPPING TO TurboIMAGE

The objectives of mapping are:

1. To translate the refined data model into a logical data structure that complies with the TurboIMAGE structuring rules.
2. To implement the access requirements into the logical data structure using the TurboIMAGE options.

Before the mapping is performed, it is essential to understand, very clearly, the structuring rules for TurboIMAGE.

1. STRUCTURING RULES

Within TurboIMAGE, two types of data sets can be supported, master and detail.
Relationships are supported between master and detail data sets.

A data set is a collection of records with the same record format. It is equivalent to what, in conventional terms, would be called a file with one type of record.

1.1. Master

A master data set must have a unique data field, the search field. TurboIMAGE supports calculated (hashed) access to a master data set using the value of the search field. A master data set may be linked to up to 16 detail data sets.
Two types of master data sets are supported, the manual master set and the automatic master set.

1.1.1. Manual

The manual master may contain more data fields in addition to the search field. A manual master set must be maintained by the application. If an entry needs to be added to an associated detail data set using a value of the search field that does not exist in the manual master, then the master record must first be created. Deletion of all detail entries will not cause the removal of the manual master entry.

1.1.2. Automatic

The automatic master set may only contain one data field, the search field. An automatic master is maintained by TurboIMAGE. When a data record is added to an associated detail data set using a value of the search field that does not exist in the automatic master, TurboIMAGE automatically creates a new entry in the automatic master for that value. Similarly when detail entries are deleted TurboIMAGE automatically removes corresponding entries from the automatic master. Typically, an automatic master set would be created to allow more efficient access to a detail data set for a particular search field eg. ORDER-DATE.

1.2. Detail

The detail data set can contain several data fields. A detail data set can be linked to a maximum of 16 master sets.

1.3. Relationships

A relationship or path is a means of connecting records, within the TurboIMAGE database, which relate to each other. A relationship can only exist between a master and a detail data set. One of the data fields within the record format of a master data set, the search field, must be included in the associated detail data set record format. The value of the search field must be unique within the master data set, but that value can be present in several record occurrences in the detail data set. TurboIMAGE will maintain a chain of pointers embedded within the data records for all entries which contain the same search field value. This will enable access to all related data records of the detail data set for an occurrence of a data record of the master data set. The particular master and associated detail data records are referred to as a chain. The detail records in a chain may be sorted by a data field.

2. TurboIMAGE ACCESS METHODS

Although several access methods are supported by TurboIMAGE (TurboIMAGE ref. manual 32215-90050), the principal ones to consider, during the Logical Design stage, are as follows :

Access on master sets :	I -	serial, in the sequence in which they are physically stored.
	II -	calculated, using a search field value to find a unique record.
Access on detail sets :	I -	serial, in the sequence in which they are physically stored.
	II -	chained, all detail entries for one master record. The sequence of the detail records in a chain may be controlled by defining a sort field for the chain.

3. TurboIMAGE SPECIFICATIONS

These specifications have been copied from the DATA MANAGEMENT specifications guide (32215-95002) :

- max. data item names per database : 1023
- max. data items per data entry (record) : 255
- max. data sets per database : 199
- max. detail data sets per master set : 16
- max. master sets per detail data set : 16
- max. search items per detail data set : 16
- max. entry size : 4094 bytes
- max. entries per data set : 2 billion (blk. factor 255)
- max. entries per chain : 2 billion
- max. characters per database name : 6
- max. characters per password : 8
- max. characters per data set name : 16
- max. characters per data item name : 16

The logical design stage is concerned with the facilities offered by the DBMS physical limitations (eg. CPU, disc-I/O etc.) will be taken into account during physical database design.
On-line access is the most important criterium when we have to consider alternative ways of implementing access possibilities.

4. MAPPING GUIDELINES

1 - Select all entities from the refined data model that only have 1:1 or 1:M (many) relationships with other entities and map them to a manual master data set.

2 - For each manual master select the search field.

3 - Map all remaining entities to detail data sets.

4 - For every detail set, implement the M:1 relationship that exists in the refined data model with the entities that were mapped to manual masters in step 1 and include the search field in the detail data record.

5 - Select all entities that were mapped to detail data sets in step 3 but have 1:1 or 1:M (many) relationships with other entities in the refined data model. Choose a suitable search field for these entities.

6 - Create automatic masters containing the selected search field for the detail data sets selected in step 5.

7 - Implement the 1:M relationships that exist in the refined data model between two entities that were mapped to detail data sets by relating them to the automatic masters created in step 6. Include the search field in the detail data record.

8 - Implement a relationship between manual master sets (1:1 relationship in refined data model) by creating a detail data set containing only the search items of the manual masters.

9 - Consider extra automatic masters for details on which direct entry is required or access via the existing master sets is clumsy.

10 - Remove all relationships (not the search fields) when the access profiles show that you only go from detail to master and never from master to detail.

11 - If more than one search item on a màster set is required, consider mapping the master to a detail data set with automatic masters for the needed extra search fields. The original manual master will remain with only one field, the search field.

12 - Determine which master to detail paths should be sorted.

NOTES

If a detail data set has an optional relationship with an associated master, prepare a dummy (empty) master entry including a value for the search item because TurboIMAGE will not allow non-owned detail entries.

The logical data structure shown in Figure 13 was produced by applying the mapping guidelines on the refined data model and summary of access requirements in Figure 12. For each data set and relationship the appropriate guideline, that led to its determination, is indicated.

WHAT COMES NEXT ?

The logical data structure will subsequently be input to the physical database design process which is outside the scope of this paper. However, in summary, the physical design will be concerned with :

- performance optimisation
- considerations with regard to the environment (eg. CPU, disc-I/O, disc space)
- user design constraints (eg. response time, run time)

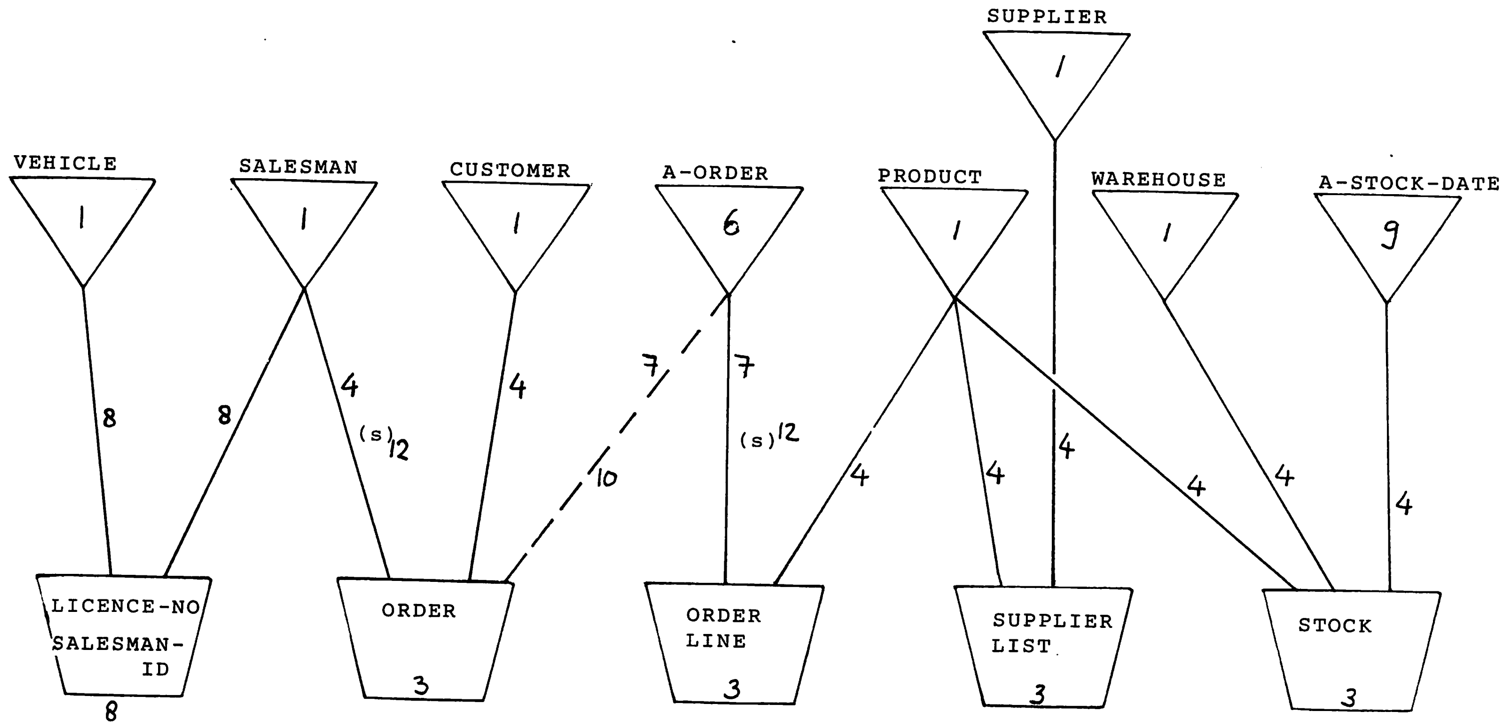

Figure 13 - Logical Data Structure for TurboIMAGE

JOS WITTEVEEN

Jos Witteveen has been working with database systems for the last six years, mainly on mini-computers. During the last three years, while working as a consultant for Database Consultants europe BV, he has been applying structured analysis and design techniques, particulary in the area of mapping conceptual data structures to a logical database design. His hardware/software experience includes Hewlett-Packard 3000 series and Digital VAX. With his previous employer he worked as a system manager providing technical consultancy and training to Hewlett-Packard installations.

GLENN PEREIRA

Glenn Pereira started working within EDP twelve years ago. During the last four years he has been applying structured analysis and design techniques in developing database systems.
His knowledge and experience includes: IMS/DL1, IDMS, IMAGE/3000, IBM/38 and VAX-11 DBMS.

BIBLIOGRAPHY

[Wierenga 1984] Wierenga, Hans, Physical Database Design, Amsterdam, Proceedings Decus, 1984.

[Pereira 1984] Pereira, Glenn, Logical Database Design, Amsterdam, Proceedings Decus, 1984.

[Green 1984] Green, Rego, White, Greer, Heidner, The IMAGE/3000 Handbook, Wordware, 1894 ISBN 0-914243-00-4

[Atre 1980] Atre, S., Database, Structured Techniques for Design, Performance and Management, Wiley, 1980

[Martin 1976] Martin, James, Principles of Database Management, Prentice-Hall Inc., ISBN 0-13-708917-1