Entity Model Clustering: Structuring A Data Model By Abstraction

P. FELDMAN* AND D. MILLER**†

*James Martin Associates, Spa House, 17-19 Worple Rd, Wimbledon, London SW19 4JS and School of Industrial and Business Studies, University of Warwick, Coventry

** Whitbread & Co Plc, Business Services Dept, Whitbread Court, Letcombe St, Reading

†Now Strategy Manager, British Rail Board, Information Systems and Technology, Churnet House, Carrington Street, Derby

Entity models are increasingly being used in the development of computer systems as an aid to the comprehension and documentation of the data in an organisation. Entity modelling can aid the understanding of an organisation's data, both computerised and non-computerised, for the strategic benefit of the organisation and as an aid to communications within and across its boundaries.

In the past there have been many problems with the use of entity relationship diagrams (the diagrams which result from entity modelling) either due to a lack of detail, or through too much detail in too small a space. Thus the models have not been entirely suitable for their intended purpose or for most other purposes. This paper discusses an approach to structuring an entity model as an aid to information management within an organisation. The technique, called entity model clustering, was developed jointly by Whitbread & Co Plc and Thames Polytechnic. It is simple in concept, has a sound basis and has been applied on a large scale in Whitbread since June 1983.

The benefits of entity model clustering to the organisation, for end-user computing, to the information systems department, and to the entity modelling process are discussed.

Received June 1984

1. INTRODUCTION

1.1 Data modelling

Data Models are increasingly being used in the development and maintenance of computer systems. Tsichritzis and Lockovsky[1] have described many different types of data model and their uses. In this paper we are concerned solely with entity modelling and entity relationship diagrams.[2, 3, 4] In entity relationship diagrams entity types, representing data, are shown as rectangles and relationships between entity types are shown as connections between the rectangles (see Fig. 2). However, even though we are only dealing with entity models here, the discussion can apply equally well to a number of other forms of data model.

1.2 Entity modelling

Typically entity modelling is undertaken in both the planning and the analysis stages of system development.[5] Entity modelling is carried out so that systems development can take place based on the form of data rather than on an organisation's processes, current systems or structure. It is thought that any resulting information systems should be more resilient as a result.

In the analysis phase, entity models are most often used for the comprehension and documentation of an organisation's data. These are crucial for the good practice of systems development, especially where information systems are concerned. Entity modelling has been widely applied in this area and its application well documented.[2, 3, 4, 6, 7, 8, 9, 10, 11] Entity modelling's most important benefits are accrued through the communication properties of the models formed; inexperienced personnel have been found to be able to understand and work with small models very easily. The communication properties are vital for the validation of a model and are useful as an aid to the refinement of existing models. Due to the communication, more confidence can be placed in a model as the basis for further development work and the resulting systems have more acceptance due to active user participation in their development.

Entity relationship diagrams are a method of diagrammatically representing an entity model. For the following it is important to understand the difference between a model and its representation. For example, a photograph of a model of a bridge is not the same as the model of the bridge. The photograph can be enlarged, reduced, added to another photograph, and so on; however, no matter how the photograph is manipulated, the model stays static and with the same meaning. The photograph is a representation of the model which can be changed to enable the model to be more easily comprehended, without changing the meaning or content of the model. As with the photograph, entity relationship diagrams can be manipulated in various ways to enable the underlying entity model to be more easily comprehended. This is the basis of the concepts discussed in this paper.

1.3 The problems of current entity modelling techniques

It is widely proclaimed that diagrams convey more meaning than either textual specifications ('a picture speaks a thousand words'), or data dictionary output, this being essentially 'structured' text anyway. However, the usefulness of any diagram is inversely proportional to the size of the model depicted. We consider any diagram with more than about 30 entity types to be reaching the limits of easy comprehension, depending on the number of relationships – the more relationships the less comprehension is possible due to the accompanying increase in complexity.

Large diagrams tend to be spread over very large pieces of paper with many long (possibly tortuous) relationships, many crossing lines and the introduction of many connectors. An example is where diagrams are displayed on walls, with the problems that may ensue in transportation, copying and presentation. The net result is a diagram which is very difficult to understand, present and reproduce. This, combined with the difficulty of implementing necessary changes, introduces an element of undesirable instability into models which are otherwise very valuable. Diagrams are often artificially constrained to the display medium in an attempt to overcome the problems of large diagrams, but this has a consequential loss of valuable detail; for example, overview diagrams are often constructed to fit on a sheet of paper by ignoring a lot of important details. The effect of all these complications is to reduce the value of entity relationship diagrams.

These inadequacies suggest that entity relationship diagrams are not used to their full potential in non-trivial situations. What is needed is a method that allows entity models to be usefully applied on a large scale in such a way that representations of them are easy to maintain, readily comprehensible, stable and yet provide adequate detail for development and planning.

1.4 A new technique to enhance entity modelling

As discussed above, a technique is needed to enable entity models to be applied on a large scale with no erosion of their usefulness; this paper describes such a technique, called entity model clustering. This technique was developed jointly by Whitbread & Co Plc and Thames Polytechnic. The result of entity model clustering is a clustered entity model.

Clustered entity models are easy to maintain and comprehend whilst being resilient to change, and provide as much, or little, detail as required. The technique has its foundations in tried and tested structuring techniques combined with a full set of diagramming conventions (see section 2). When applied to an entity model, the technique allows entity types to be viewed at various levels of complexity independent of organisational constraints, but reflecting the business context. The structure is based on the relationships of the contained entity types, as opposed to the way the entity types are used or perceived. However, inasmuch as use and perception actually affect the relationships shown between entity types, use and perception are reflected in a clustered entity model.

The following describes entity model clustering in detail. Section 2 describes the constituents of a clustered entity model and a method of formation. Section 3 gives a fictitious example of entity model clustering and Section 4 discusses some of the benefits which accrue from the use of entity model clustering.

2. ENTITY MODEL CLUSTERING

2.1 Constituents

The concepts of abstraction[12, 13] and linking diagrams by decomposition (e.g. Ref. 14) are well known and accepted. Their novel use in combination forms the basis for entity model clustering.

Basically, a clustered entity model is a hierarchy of successively more detailed entity relationship diagrams, with a lower-level diagram appearing as a single entity type on the next higher-level diagram (cf. structured data flow diagrams where functions are decomposed into more detailed self-contained diagrams.[14] This results in a single model consisting of a 'tree' of entity relationship diagrams.

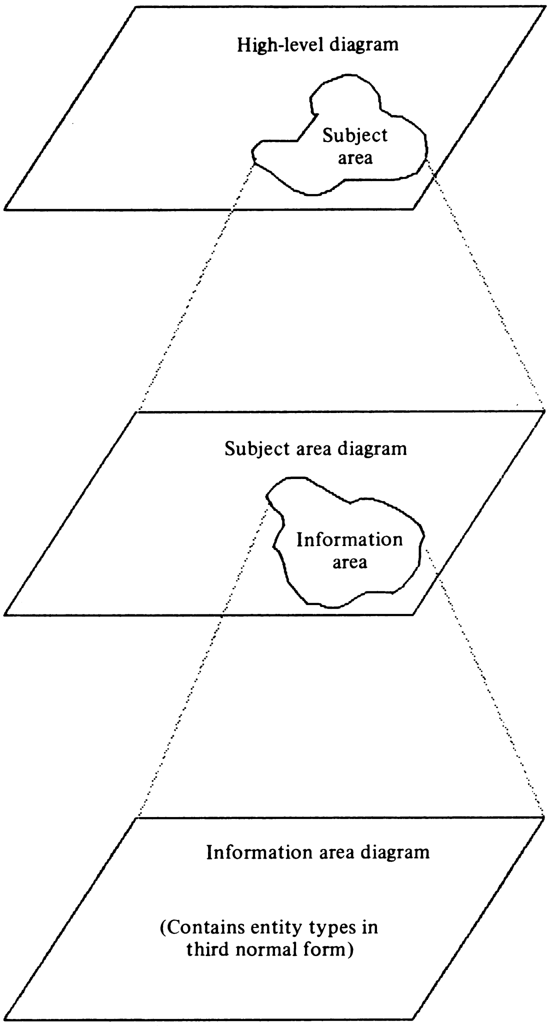
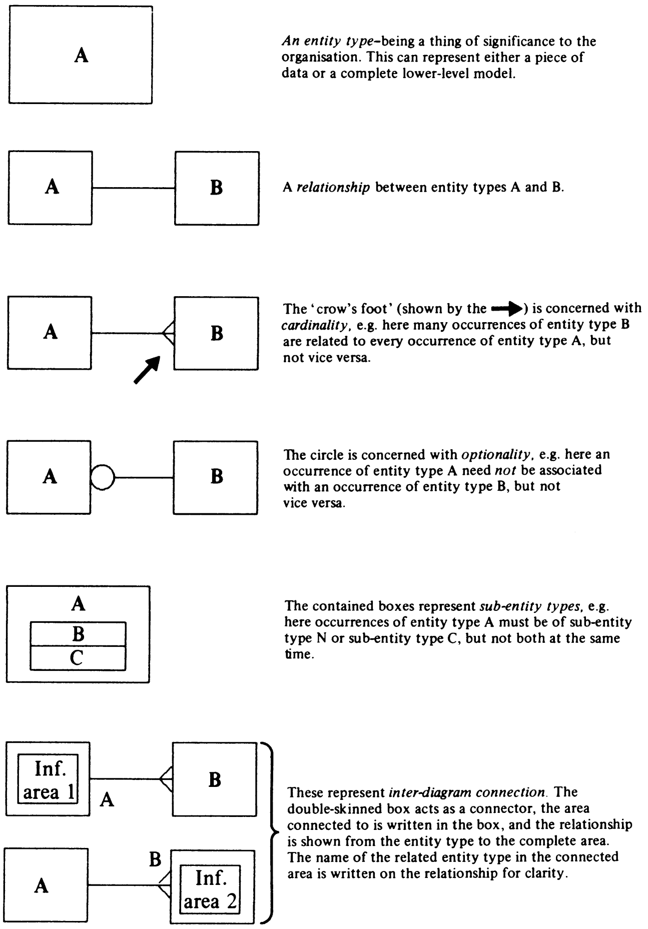


Figure 1. Hierarchy of successively more detailed entity relationship diagrams.

Experience showed that it was desirable to duplicate certain entity types in a number of branches of the tree; examples of these are Product, Supplier and Customer. These duplicated entity types were found to be fundamental to a modelled organisation and hence have been termed 'major' entity types. As a broad generalisation, major entity types tend to relate to data which would appear as '*master*' files in batch computer systems (also known as '*standing*' data). Entity types which are not major are called 'minor'; these tend to relate to batch '*transaction*' file data. Both major entity types and minor entity types should be of about the same level as Third Normal form relations, see.[15] This is the same in clustered entity models and conventional entity models. The combination of major entity types, minor entity types and their interrelationships form the totality of a conventional entity relationship diagram.

In practice, three levels of diagram have been found to be useful; a high-level diagram, decompositions of this called 'subject areas', and further decompositions of this



An entity type–being a thing of significance to the organisation. This can represent either a piece of data or a complete lower-level model.

A *relationship* between entity types A and B.

The 'crow's foot' (shown by the ➡) is concerned with *cardinality*, e.g. here many occurrences of entity type B are related to every occurrence of entity type A, but not vice versa.

The circle is concerned with *optionality*, e.g. here an occurrence of entity type A need *not* be associated with an occurrence of entity type B, but not vice versa.

The contained boxes represent *sub-entity types*, e.g. here occurrences of entity type A must be of sub-entity type N or sub-entity type C, but not both at the same time.

These represent *inter-diagram connection*. The double-skinned box acts as a connector, the area connected to is written in the box, and the relationship is shown from the entity type to the complete area. The name of the related entity type in the connected area is written on the relationship for clarity.

Figure 2. Diagrammatic conventions of entity relationship diagrams and clustered entity models

called 'information areas' (see Fig. 1). The number of levels is determined by the diversity and complexity of an organisation (see 2.2.3).

Information areas, the lowest level, consist of major entity types, the appropriate minor entity types and their interrelationships. Similarly, subject areas consist of major entity types, the appropriate information areas (appearing as entity type boxes) and their interrelationships. The high-level diagram consists of all the major entity types, all the subject areas (appearing as entity type boxes) and their interrelationships. So a clustered entity model is a set of abstractions of a conventional entity relationship diagram, with each level being a decomposition of the higher levels. The lower levels enable the desired subject or information areas to be concentrated on without the distraction of extraneous detail. Major entity types appear at every level to act as 'signposts' to the detail and to provide the context to interpret the detail (see 2.2.5 for further explanation).

As for the number of levels, the actual constituency of an area is flexible and depends very much on an analysts's opinion of the best way to depict a given situation. For example, where a major entity type only applies to a single subject area it has been known to show that major entity type in the subject area and its constituent information areas, but not on the high-level diagram. In this case the major entity type is only fundamental to a restricted functional area of the organisation, this being reflected in the diagrams formed.

One of the guiding aims in the development of entity model clustering was to ease maintenance, i.e. the upkeep and change of diagrams. As there are many fewer components on any single diagram, it is much easier to draw the different, smaller diagrams than when all entity types appeared on a single diagram with a large number of relationships. However, the danger of proliferating and duplicating entity types had to be avoided. This was achieved by constraining each minor entity type, information area and subject area to appear in only one place but be referenced as many times as necessary by means of inter-diagram connectors (see 2.2.1). The inter-diagram connectors are duplicated at source and destination to enable all connectors to a component to be found easily without complication.

2.2 Methods of formation/derivation

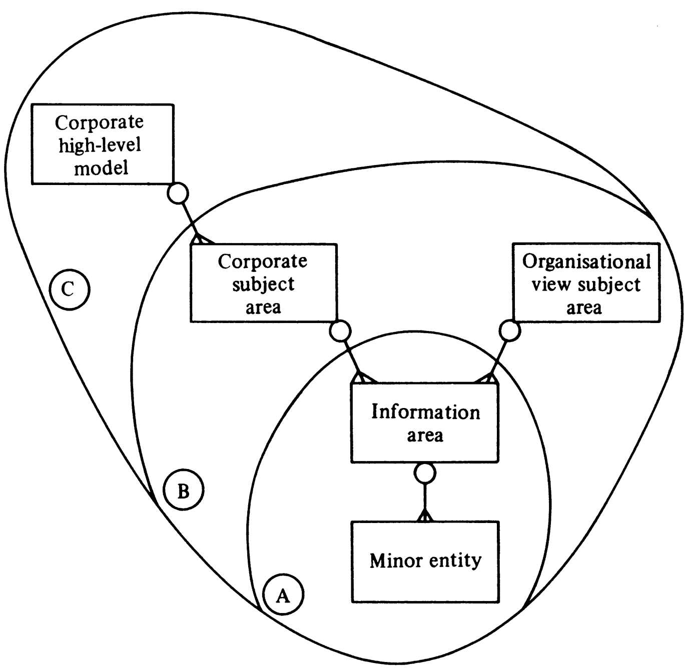
The following considers the most effective method of entity model clustering. This method is iterative and the results are adaptable to an organisation and its information requirements. The most likely application for the method is to structure an existing conventional model, or set of models. However, it has also been used to guide the development of an entity model. Only the former is considered in this paper. The method is empirical, but based on the tried and tested concepts of abstraction and decomposition. There are two aspects to the formation of a clustered entity model, algorithmic and interpretative. The interpretative aspects arise from the peculiarites inherent in any model of human activity and derive either from the activity or from the modeller's *view* of it. The method described here does not just apply to single models but has been used on a related set of models, with a single, clustered model being synthesised.

2.2.1 *Diagramming conventions*

Entity model clustering only introduces one diagrammatic convention on top of entity relationship diagramming conventions, this being a convention to deal with inter-diagram connection. The conventions are summarised in Fig. 2. Entity types are represented by a rectangle (box), relationships between entity types by connections between the boxes, cardinality is represented by 'crow's-feet' and optionality by circles on the relationship lines. Some entity types can be partitioned into sub-entity types, which divide the set of the entity occurrences into mutually exclusive subsets. Sub-entity types are represented as boxes within a box. An area diagram is represented as an entity type on a higher-level diagram and expands into a single diagram, thus entity types can either be conventional entity types or can represent lower-level diagrams. Inter-diagram connectors are shown as double-skinned boxes and can easily be distinguished from entity types containing sub-entity types, as there must be at least two sub-entity types if there are any.[3]

2.2.2 *Finding major entity types*

First the major entity types of the modelled organisation must be extracted. Based on the concept of logical horizon[8] (see Fig. 3), occurrences of a major entity type should be uniquely identifiable from any related entity types, i.e. all of a major entity type's relationships should



The logical horizon of the entity shows the entities which can be uniquely identified from that entity. It can be established from the 1:N(N ⩾ 1) relationships emanating from the entity, e.g. each minor entity only appears on a single, uniquely known information area. Successive logical horizons can be found by treating them as abstractions of the contained entities.

Ⓐ can be thought of as an abstraction of information area and minor entity, and the process can be repeated so that for each Ⓐ group there is only one organisational view Subject area (for a given division of the company) and only one corporate Subject area.

Figure 3. Logical horizons explained through a simplified model of entity model clustering.

be 1:MANY 'outwards' (major entity:related entity). For example, in Fig. 4 all of the relationships to entity type Product are 1:MANY with the 1 on an edge of the Product box. Furthermore, a major entity type should be of fundamental importance to more than one functional area of the organisation, i.e. should appear in more than one information area. For example in Fig. 5 Customer has relationships to two information areas (these examples are expanded upon and their derivation explained in section 3). So major entity types are fundamental, shared data. They are often characterised by having an inherent stability and an easily discernible, though often complex 'life-cycle'.

So major entity types can be elicited by analysing a model to discover those entity types which have only 1:MANY outward relationships, or whose MANY:1 inward relationships are only to major entity types. Care must be taken where MANY:MANY relationships appear on a diagram. For the purposes of discovering major entity types, MANY:MANY relationships can be treated as 1:MANY outward relationships, as the majority of such relationships can be notionally broken down into 1:MANY relationships with 'intersection data'; it is a notional breakdown because it does not actually need to be done.

The set of major entity types formed from this process will be modified by removing any entity types which are only related to single information areas (see 2.2.3, and see section 3 for an example). Other entity types can be added to this set if they are found to be of significant interest to a number of information areas. The two aspects of entity model clustering can be seen here; the relationships are made use of in an algorithmic fashion and the results are subsequently interpreted to deal with inbuilt anomalies in the base entity models.

2.2.3 *Clustering entity types*

Subject areas and their information areas can be thought of as decompositions of the relationships between major entity types. This bears a correlation to a method used to analyse information requirements, which is to find a high-level 'view' and to continuously decompose the relationships between the entity types found. In conventional models the constituents of this decomposition can only be documented as a non-hierarchical diagram. Clustered entity models are mainly formed by clustering these non-hierarchical diagrams back into a hierarchy by the use of abstraction; the clustering process is guided by the major entity types previously found.

Information areas are formed by first abstracting minor entities into a logical horizon (see Fig. 3) and then successively abstracting the logical horizons until there are only logical horizons and majority entity types. These abstracted logical horizons are the 'first-cut' information areas. More often than not, this process actually results in more than one information area relating to the same group of major entity types. These similar information areas are then abstracted to form a subject area. The high-level diagram documents the subject areas found and the non-decomposed relationships between major entity types, relationships which will only appear on subject areas and information areas when they enhance the meaning of those areas. For example, in the Customer Representation information area on Fig. 5 it is meaningful to depict the relationship between Customer and Region. To summarise, information areas are formed by the abstraction of a logical horizon (cluster of 1:MANY outward relationships) between some major entity types, and subject areas are the abstraction of a cluster of information areas relating to broadly the same group of major entity types (Fig. 5).

It may be necessary to continue this process to higher levels of abstraction for an organisation with very complex and/or diverse information requirements. There is no restriction placed on the number of levels which can be used. It has been found that the size and complexity of information requirements bears a close relationship to the diversity of an environment. This is because different entity types and their relationships are needed to cope with the diversity of information modelled. Thus for simple environments only two levels of clustered entity model may be appropriate, while for very diverse environments four or more levels may be appropriate. The number of levels should not be pre-determined, but should be allowed to grow with the model during the analysis process. The description above is based on the results of the study at Whitbread, where it was felt that three levels were a manageable number which did not cause overcrowding of diagrams.

Relationships can exist between information areas and between subject areas, i.e. across area boundaries. These relationships arise when the clustering process results in more than one cluster between major entity types. For example, in Fig. 4 Order appears in two logical horizons

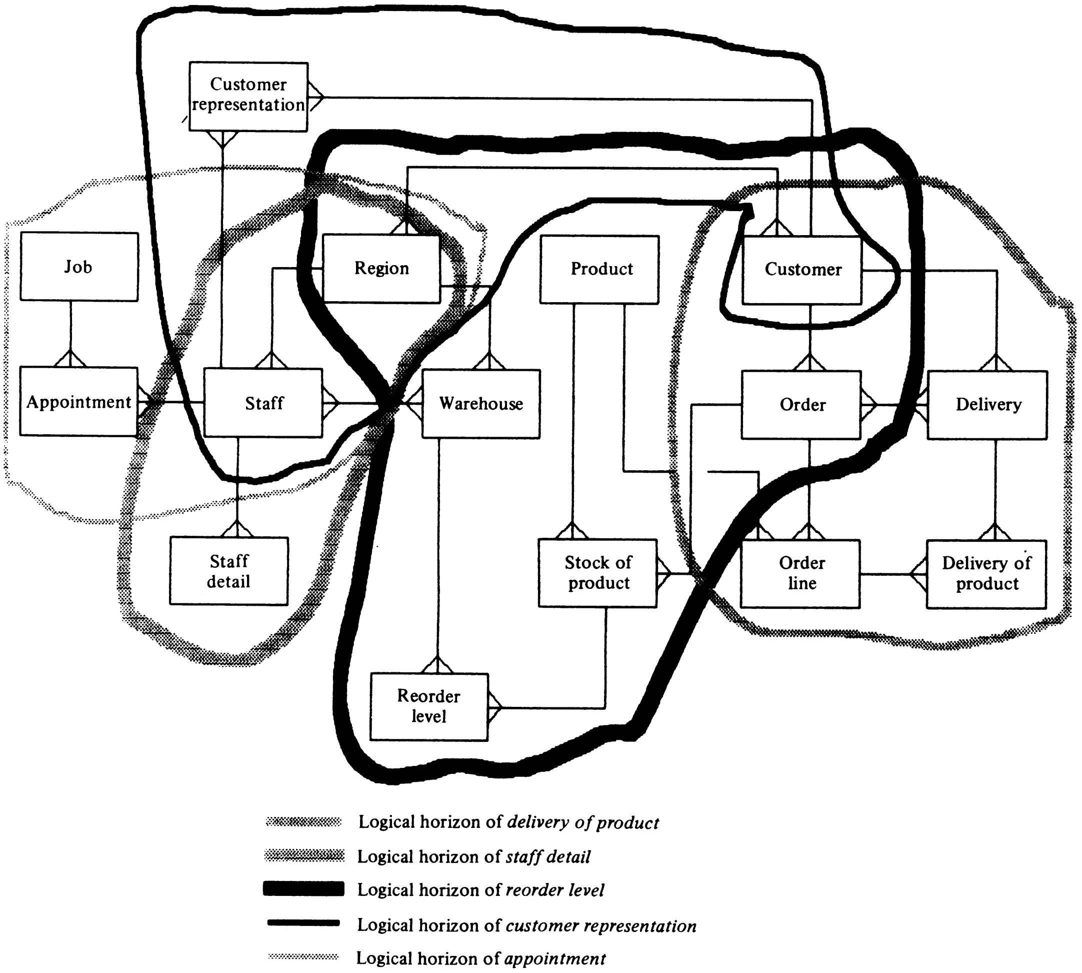

Customer representation
Job
Region
Product
Customer
Appointment
Staff
Warehouse
Order
Delivery
Staff detail
Stock of product
Order line
Delivery of product
Reorder level
Logical horizon of *delivery of product*
Logical horizon of *staff detail*
Logical horizon of *reorder level*
Logical horizon of *customer representation*
Logical horizon of *appointment*

Figure 4. Example of an entity relationship diagram showing logical horizons.

and, hence, information areas; as Order can only appear in one information area, some of its relationships will have to cross the boundary to the other information area. Boundary relationships may occur if clusters are formed between groups of major entity types, where some of these major entity types are duplicated across the groups; for example, Customer and Product appear in the two logical horizons which Order appears in.

If a boundary relationship is found, a decision will need to be taken as to where to draw the boundary between the clusters, i.e. which minor entity types/information areas will appear in which information areas/subject areas respectively. Clusters broadly relate to the use made of the entity types contained in them, as this is reflected in a large number of the relationships. Also, entity types which are related due to 'structure', e.g. 'order consists of many order lines', tend to be strongly bound together and made use of in the same way in most processes. This is not so with entity types related because of other causes, e.g. appearance in a common process as in 'order results in delivery'. As it is sensible to keep structurally related entity types together wherever possible, the boundaries between areas should be drawn to keep structural relationships in the same area. If there are no structural relationships then a boundary should be drawn to put entity types in the area to which they are most suited by functionality. For example, a Delivery entity type should be put in a distribution-related area as opposed to a production-related area. If the result of the decision is not obvious, the boundary may hide an as yet unfound major entity type.

As for major entity types, some manipulation of the results may be necessary to accurately reflect the information usage and requirements of an organisation. This can result in more boundary relationships. However, in these cases the above guidelines should still be followed.

The use which the clusters reflect is not that of detailed processes but of broad functional areas of the organisation. These functional areas are not necessarily related to the organisation's structure – quite often they cut completely across existing structures – but they do relate to the main purpose and activities of the organisation. For example, an organisation with a purchasing function may not have a specific purchasing department, but may have purchasing distributed across all the organisation's departments. If there is a purchasing function, there will probably be a purchasing subject or information area which clusters the data used by the purchasing function. There is unlikely to be say, a 'Buy Equipment' subject area, or a 'Decide on Supplier Suitability' information area – both these processes are part of a Purchasing function and would use data in, or related to a purchasing subject or information area.

A knowledge of the functional requirements of an organisation is useful to decide on boundary relationships and to name the clusters formed. The information and subject areas are named to reflect the functional area they

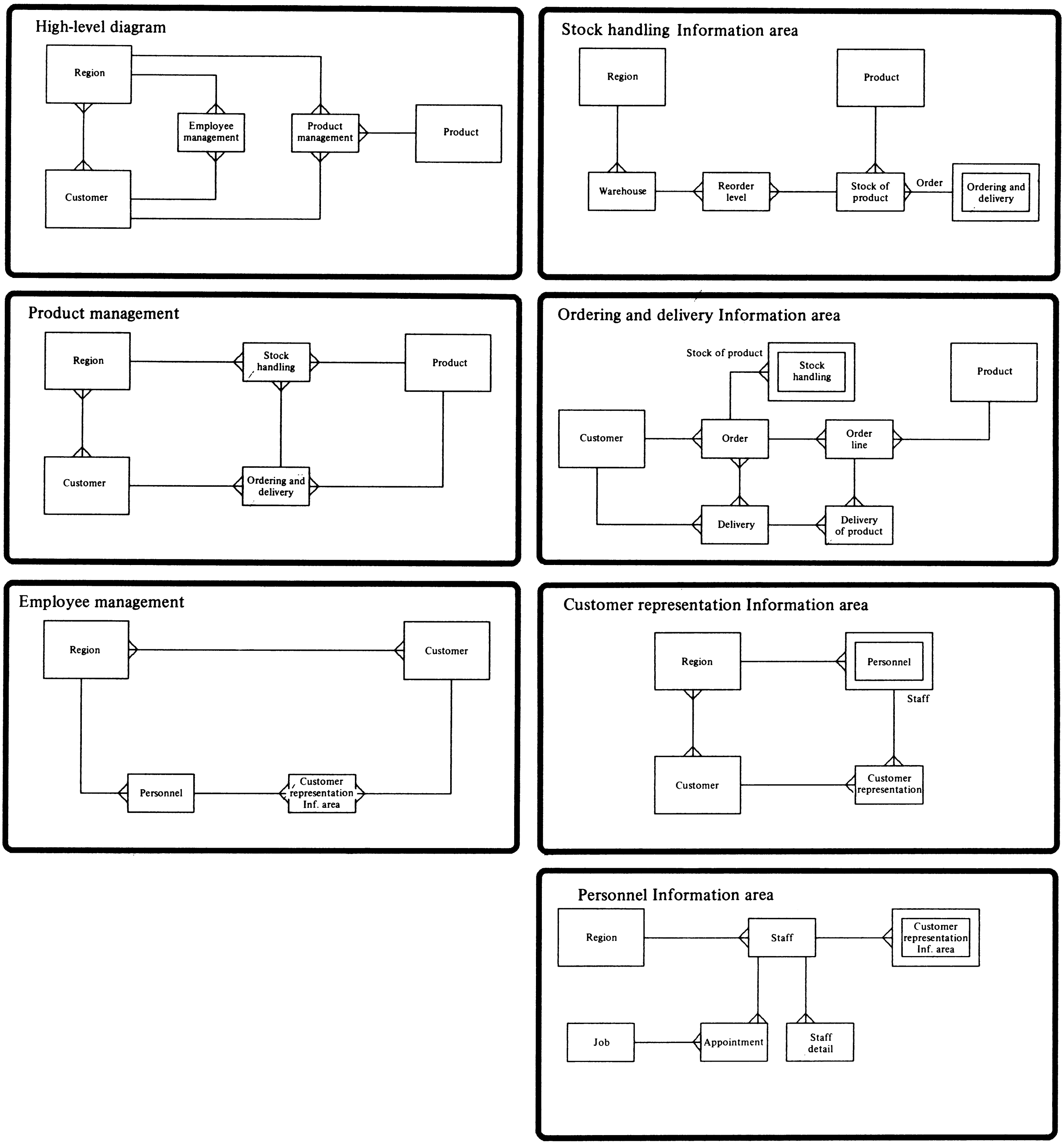


Figure 5. Example of a clustered entity model based on Fig. 4.

relate to (e.g. Personnel) or to reflect groupings of data (e.g. Ordering and Delivery). Sometimes clusters are mainly concerned with a major entity type, in which case they are named appropriately.

2.2.4 *Generalisation and aggregation abstraction*

Smith and Smith[13] isolated two forms of abstraction to form a hierarchy, generalisation and aggregation. A generalisation hierarchy is one formed by the generalisation of its components, e.g. Car and Moped generalise to Road Vehicle and Road Vehicle, Train, Boat and Plane generalise to Vehicle. An aggregation hierarchy is one formed by the aggregation of its components, e.g. a set of Order Lines and an Order Total aggregate to form an Order. These are represented in a number of ways in entity modelling and entity model clustering.

In entity modelling, generalisation is dealt with by the use of sub-entity types. For instance, in the example above Car and Moped would be subtypes of a Road Vehicle entity type and Road Vehicle, Train, Boat and Plane would be subtypes of a Vehicle entity type. The subtype convention was introduced to represent generalisation and does so reasonably well. Generalisation can also be shown by the use of relationships, for example 'Road Vehicle is Car or Road Vehicle is Moped'. This tends to be used mainly where subtypes cause too complex a diagram for easy communication; however, there are no restrictions on the use of either method.

Aggregation is also dealt with by two methods in entity modelling. One method is the assignation of attributes to an entity type, e.g. Person has Name, Sex and Address attributes all of which are aggregated to form the Person entity type. The second method is the use of relationships. As an example, Order Total would be an attribute of an Order entity type and Order Line would be an entity type with a relationship to Order. The choice between attribute and relationship reflects the semantics and significance of the data, the choice usually being made intuitively.

Entity model clustering does not impose any restrictions on the way that abstraction is dealt with by entity modelling. In fact the clustering process is a controlled form of aggregation abstraction – all the entity types within an area are aggregated to form that area. In entity model clustering, minor entity type subtypes are dealt with in exactly the same manner as for conventional entity modelling, but a degree of flexibility is introduced in the depiction of subtypes of major entity types. Due to their nature, major entity types tend to have a complex structure. This is often shown by the use of involuted relationships and subtypes, for example an Organisation major entity type would have a hierarchy, represented by an involuted relationship, and consist of many mutually exclusive parts, represented by subtypes. Involuted relationships can be shown throughout the model, however, subtypes only need to be depicted on the high-level diagram and in those areas which they apply to. For example, assume major entity type Product consists of Bought Product, Sold Product and Intermediate Product subtypes. Product would be shown however necessary, however the subtypes would only be shown on those areas where they are applicable, for instance Bought Product would be shown on a purchasing area and Sold Product in a sales area. In these cases the inapplicable subtypes could be generalised to Other Product. Thus the view of a major entity type can change, depending on the area under consideration. The basic meaning of a major entity type remains static throughout the model; it can just be interpreted in different ways in different contexts.

Generalisation by relationships is unaffected by the use of entity model clustering. However, generalisation hierarchies would be used in the determination of area boundaries as it is highly desirable to keep generalisation hierarchies intact. This is achieved because generalisation relationships have a structural nature with the components of the hierarchy invariably used in the same way (see 2.2.3). So a generalisation hierarchy would tend to appear in a single area. The same argument applies to aggregation.

2.2.5 *Cartographical analogy to the use of a clustered entity model*

The approach we propose here for the management of large entity models is analogous to the cartographical method of locating addresses within a country (see Fig. 6). A traveller unfamiliar with the geography of a country would first use an overview map to locate the desired area, making use of major cities as landmarks. The traveller would then resort to a more detailed map to locate a town in relation to the cities and to a street map to actually pinpoint a required address. Roads and rail networks correspond to relationships between cities, towns and so on.

If a street map were viewed without previously consulting smaller scale maps then the beneficial context of the town would not be available; for instance, the best route to take and the geographical nature of the area, e.g. whether in north or south, in desert or rain forest, cannot easily be found from street maps. Also, without this context there is always the possibility of using the wrong map, e.g. a map of Washington, England or Washington State, USA instead of Washington DC, USA.

The analogy to clustered entity models is that a map of a country is equivalent to a high-level diagram, a map of a country or state to a subject area and town plans to information areas. Town landmarks correspond to entity types, cities to major entity types, town streets to relationships between the landmarks, inter-town roads/rail to inter-diagram connections and town/city connections to relationships between minor and major entity types.

3. AN EXAMPLE OF ENTITY MODEL CLUSTERING

This section considers an example of forming a clustered entity model from a conventional entity model. The conventional entity relationship diagram which is the basis for this clustering is shown on Fig. 4. This diagram covers ordering, distribution, stock handling and staff handling in a very simplified form to enable the example to be more easily understood. The diagram bears no relation to any actual enterprise. For the purposes of this example some logical horizons have been delineated on the diagram, these would not normally appear. The results of the clustering process are shown in Fig. 5. The algorithmic and interpretative aspects have not been separated because they are too interdependent.

3.1 Find major entity types

The main criterion for a major entity type is an entity type which has only 1:MANY outward relationships, i.e. is at the 'top' of a logical horizon. From the logical horizons marked on Fig. 4, we can see that the entity types Customer, Job, Product and Region are candidates for major entityship (the MANY:MANY between Customer and Region being notionally decomposed into two 1:MANY outward relationships). On consideration of these entity types, they appear to be a reasonable set of fundamental entity types. If we consider the criteria that a major entity type can have 1:MANY inward relationships from other major entity types, most other entity types would appear to be candidates; this is a result of the simplified example. Intuitively, the only other likely fundamental entity types are Staff and Warehouse.

Some people would expect Order and Delivery to be fundamental. However, these entity types are based **in** the functionality of an organisation; this suggests that they are not fundamental as an organisation's functions tend to be based **around** major entity types, for instance

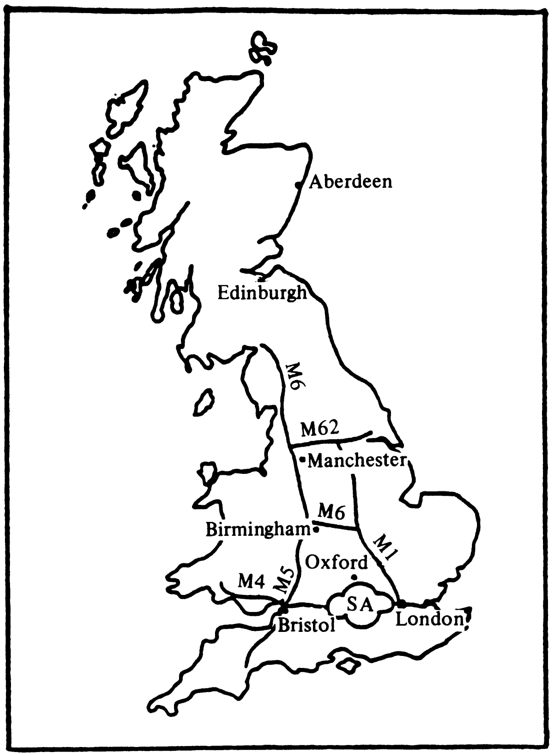


Figure 6.1. High-level diagram of Great Britain.

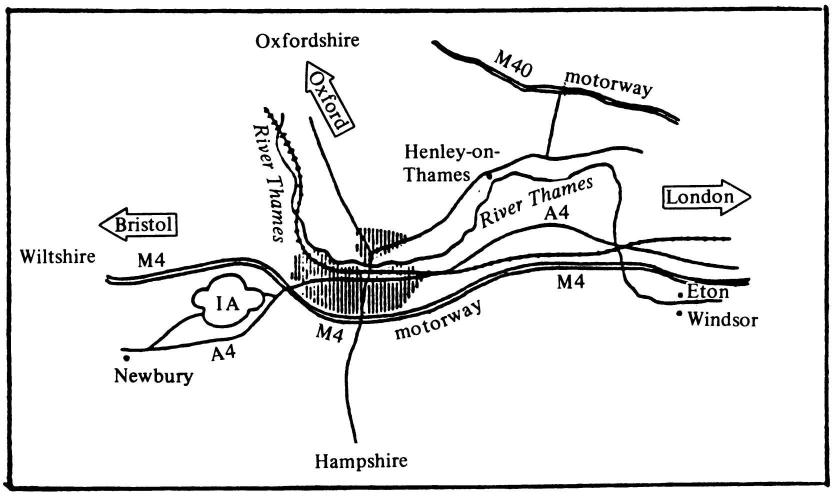


Figure 6.2. 'Subject area' diagram of Berkshire.

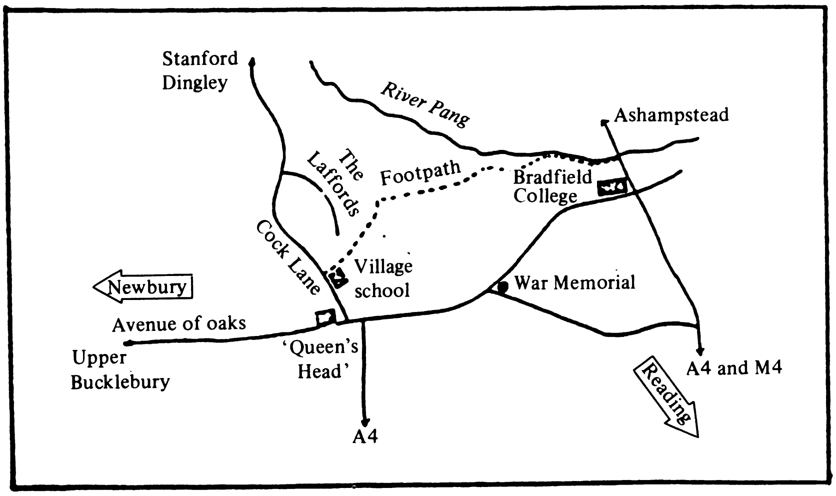


Figure 6.3. 'Information area' diagram of Bradfield, Berks.

Figure 6. Cartographical analogy to a clustered entity model.

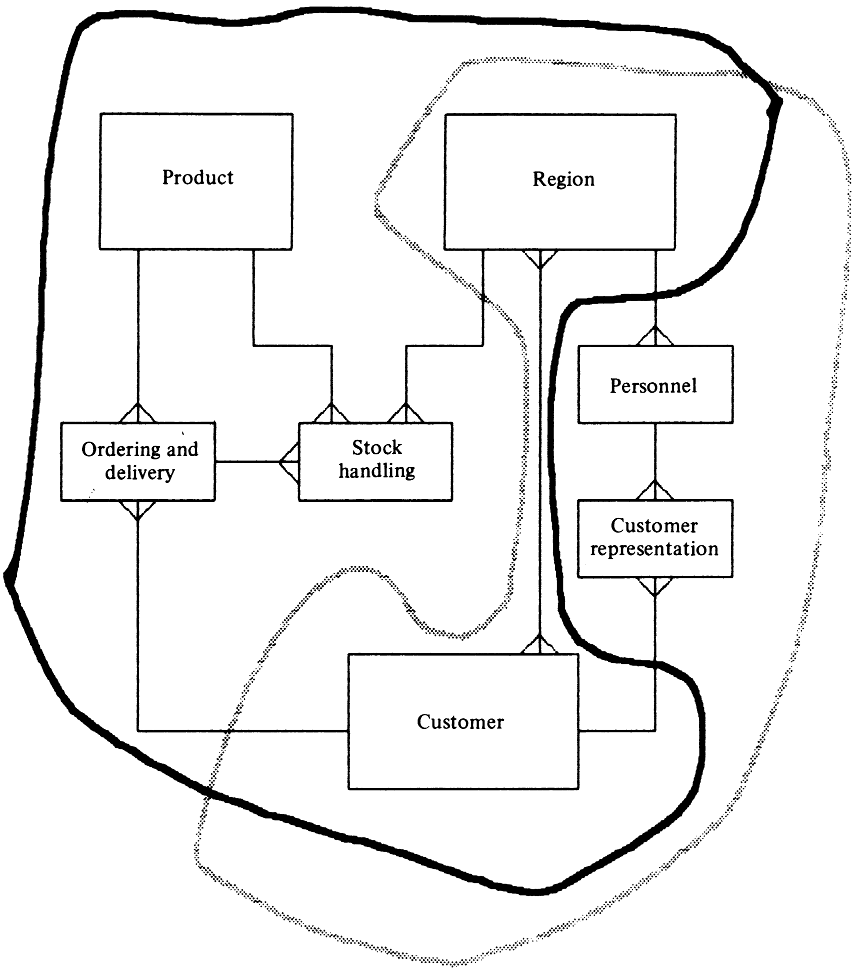


Figure 7. Intermediate situation in the clustering process.

Customer Handling and Product Production, both of which result in Orders and Deliveries. Customers and Products can exist without Orders and Deliveries, for example new Customers and Products, but the reverse does not hold. This is not to imply that Orders and Deliveries are not important (in many ways they are more important than Customers and Products) they are just not fundamental.

Therefore the initial set of major entity types is Customer, Job, Product and Region, with the possible inclusion of Staff and Warehouse.

3.2 Find information areas

To find information areas we will use the logical horizons delineated on Fig. 4 as the basis of abstraction and we will use the major entity types found above.

There are a number of start points for abstraction, all of them are the 'lowest' point of a logical horizon.

Take Delivery of Product. At the top level there is Product and Customer; Order, Order Line and Delivery are in between. This is one information area which, guided by the contained entity types, we shall call Ordering and Delivery.

Take Reorder Level. Customer, Product and Region are at the top level; Warehouse, Stock of Product and Order are in between; we shall name this area Stock Handling. As can be seen, there is a boundary dispute between Ordering and Delivery and this area. The relationship between Order and Stock of Product is 'fulfils', i.e. is functional. The relationship between Warehouse and Reorder Level is 'has a', suggesting a structural relationship. The relationship between Stock of Product and Reorder level is also 'has a', suggesting aggregation, i.e. also structural. Therefore there should be a boundary drawn across the functional relationship between Order and Stock of Product.

Take Appointment. At the top level there are Job and Region, with Staff in between. The result of taking Staff Detail is a very similar set of entity types. This suggests that Appointment and Staff Detail should be in the same information area; this we shall call Personnel. The logical horizon based around Customer Representation has commonalities to this group in Staff and Region; this might suggest inclusion in Personnel, but because Customer Representation is related to Customer (a different major entity type) it probably needs to be an information area on its own. The correctness of this decision can be seen if the type of processes which would apply to the different information areas are considered. For example, Personnel would have employment-related processes such as Hiring, which would not directly affect Customer Representation, whereas Territory Selling processes applying to Customer Representation have little relevance to Personnel.

The MANY:MANY relationships between Staff and Warehouse, and Order and Delivery need to be thought about. The MANY:MANY between Customer and

Region is a relationship between major entity types and so does not affect the constituency of the information areas. Order and Delivery only occur in the same information area, therefore this relationship does not cause any difficulties. On the other hand Staff and Warehouse appear in different information areas, so the relationship between them must be considered as a 'boundary dispute'. The question is whether or not the information area containing Warehouse is the same as the information area containing Staff. The resolution of the MANY:MANY relationships is the staff working at a given warehouse for a particular period of time. This suggests that Staff and Warehouse are not concerned with the same information groupings, so belong in different information areas. This can be seen to be a reasonable split if the broad functions represented by the respective information areas are considered – Personnel and Stock Handling should be separate groupings. This latter, intuitive process has value because if the areas had a lot of common functionality, the resolution of the MANY:MANY would probably have reflected this.

Thus we have four information areas: Personnel, Stock Handling, Ordering and Delivery, and Customer Representation.

3.3 Major entity type iteration

We must now examine the major entity types in the light of the information areas discovered. An entity type can only be major if it is shared between information areas. Job only affects a single information area, so is not major. Product, Region and Customer all affect at least two of the information areas, so are major. Of the two 'possibles', Warehouse only affects a single information area, so cannot be major whereas Staff affects two information areas, so can be major. Whether to include Staff as major or to have a relationship between the Personnel and Customer Representation information areas is an interpretative decision. The latter is better for this simplified example; in a more complex example, where three or more information areas are concerned, the former would probably be more useful.

3.4 Find subject areas

In reality, this example would be much too simple to justify subject areas, however some subject areas will be defined from the information areas found above to illustrate the entity model clustering concepts. The process is very similar to finding information areas, so only the broad reasoning will be given.

There are four information areas: Personnel, Stock Handling, Ordering and Delivery and Customer Representation, which are all interrelated – this intermediate situation is shown on Fig. 7. A logical horizon can be found which includes Stock Handling and Ordering and Delivery but excluding the other information areas, similarly for Personnel and Customer Representation. Thus there are two candidates for subject areas. The boundaries between these areas can be resolved by using exactly the same reasoning as for the formation into information areas. Furthermore, if the functional basis for the areas is considered, then the groupings appear reasonable. So we have found two subject areas, which will be named Staff Management and Product Management after their constituents.

4. BENEFITS OF ENTITY MODEL CLUSTERING

Entity model clustering solves the dilemma of choosing between a large unstructured diagram that lacks cohesion and a superficial overview diagram that has insufficient detail. Entity model clustering enhances conventional entity modelling techniques, enabling them to be applied to large scale and/or diverse problems without the difficulties described in section 1.

The technique of entity model clustering is based on tried and tested principles. A data architecture or entity relationship diagram structured in this way has many benefits; there are benefits for a modelled organisation, benefits for end-user computing, benefits for information system development and benefits for entity modelling in general.

4.1 Organisational benefits

A clustered entity model can be viewed at different levels of abstraction as desired, a view which can additionally be confined to an area of interest. Thus the people most concerned with the results of any analysis can comprehend the results more easily and completely, and also make better use of them.

Organisationally dependent views, e.g. divisional and departmental, can be easily assembled using information areas as fundamental building blocks whilst retaining the structural independence of the detail contained within the information areas. Thus benefits of an often organisationally oriented technique such as data-flow diagrams can be combined with those of an organisation-free technique such as entity modelling without losing the inherent benefits of either; if anything the benefits are more powerful in combination. These organisationally dependent views can provide the basis for organisational change due to a better modelling, and hence comprehension of the way that information is used within and across organisational boundaries.

4.2 End-user computing benefits

Rapid technological change has provided end-users with the power to meet many of their own development needs using personal computer or prototyping facilities. This evolution of information systems imposes a new requirement on information system development, namely to provide an information management service within and across organisational boundaries. One inhibiting factor is that system developers tend to express information in terms which reflect an underlying physical system, whereas end-users deal in the real world. The interpretation of the need to provide an information service within the business context and its subsequent translation to implementation terms is a task that has been recognised for end-user computing;[16] it is also the key to the successful application of information system technology in general.[1, 5, 9, 17, 18]

After a system has been developed, any entity relationship diagrams on which that system is based are often just used for maintenance purposes. However, it is possible to utilise the diagrams as an 'index' to the information content of an information system, whether computerised or manual. Some of the end-users of an information system will have taken part in the development process and would be conversant with the conventions used, conventions which are simple enough to allow easy training of other personnel. The use of an entity relationship diagram in this manner would be facilitated by the utilisation of data dictionaries. However, entity relationship diagrams which result from large or diverse developments are not suitable for end-user computing purposes, due to failings in the way that entity modelling is currently applied (see 1.3). For example, dictionary support requires the user to know the exact name of an entity type or one of its many synonyms; names which are either buried in complexity on inaccessible diagrams, or not shown at all on a superficial model. Current dictionary output also has to be subsequently interpreted in the context of the business by use of an entity relationship diagram anyway. Thus the environment which would benefit most from end-user computing facilities faces the greatest difficulties in the dissemination of the basic information available.

Entity model clustering deals with the problems of large or diverse developments, thus improving the use of end-user computing. For example, end-users can make better use of data dictionaries with a clustered entity model. The subject and information areas provide the context to an information request. An end-user does not have to know of the existence of an entity type before the request, but can be led to it through the succeeding levels of detail of a clustered entity model. at the outset of a request all an end-user needs to know is the rough location of information, e.g. 'something about products and customers roughly associated with ordering', without having to know the exact entity types concerned.

4.3 Benefits for information system development

To realise strategic, tactical and operational benefits of corporate data, its importance as a resource must be recognised.[19] The meeting of this need is hindered by problems in the application of entity modelling which are primarily caused by difficulties in coping with medium to large volumes of information. This is precisely the situation which entity model clustering was developed to deal with, so would give a realisable benefit to any large organisation undertaking corporate data modelling.

For the reasons discussed previously, planning studies often result in superficial overview models containing little useful planning and development detail. This is beneficial for steering committee and management board reviews, but is inapproriate for the planning activities necessary for developing information systems, and also for necessary validation of the models produced by the development process. Entity model clustering allows models of sufficient detail for information system planning and development activities to be built without the complexity normally associated with large models.

Subject databases are collections of entities which it is desirable to develop and use together.[20] The identification of subject databases and their interfaces is aided through entity model clustering. First-cut design of many subject databases can be gained from subject and information areas, as these are clusters of related information all concerned with the relevant major entity types and all likely to be used in the same way. For example, a Customer subject area roughly maps to a Customer subject database, as all the information relates to customers and will all be used in customer-oriented processing.

Project boundaries can be delineated through the use of a clustered entity model. An initial investigation will elicit broad areas of interest, the subject areas. These can be made the focus of detailed investigations which are logically directed towards a result which bears a relationship to the developed models. For example, a project based around an ordering and distribution subject area will fully define that area and its surroundings; the choice of this area would be due to its forming a logical group in a clustered entity model rather than from being a department in an organisation, or from being a known a major function of the organisation. The results from taking logical groups should be more stable and better directed than from other ways.

System boundaries may also be able to be delineated through use of a clustered entity model. The subject and information areas can be made the basis of information systems with the interfaces pre-defined by the clustered entity model. The interfaces can be investigated to provide system dependencies for use in development planning. It must be emphasised that a clustered entity model would not be used in isolation to define projects and systems, but would be used in combination with techniques such as cluster analysis (Murtagh gives a good review of cluster analysis methods in Ref. 21). However, it is important to note that entity model clustering makes the use of these other techniques much easier, as logical groups of data are already defined for them.

The automation of support for entity relationship diagrams is becoming more widespread, with prototypes (e.g. Ref. 22) and even commercial products available. However, with modern technology these suffer from even more acute problems than the traditional media due to the restrictions of the screen resolution, which is often the equivalent of an A4 sheet of paper. With automated diagrams, a complete diagram will not be able to be seen at one time. Even with very good resolution, there will be a point where things are so small that they will be incomprehensible. The alternative is to 'scroll' through a number of pages. The loss of context inherent in this process is bound to cause comprehension difficulties. Entity model clustering would allow complete diagrams to be viewed and the context of separate diagrams to be retained, thus easing restrictions and enabling entity model automation to be more successful. Clustered entity model automation would consist of ordinary automation (as in Ref. 22) combined with that of movement through a data-flow diagram hierarchy.

The use of data dictionaries has been mentioned frequently above. It should be noted that a data dictionary (Datamanager) has been used to record the results of entity model clustering. Thus the structure of a clustered entity model can be supported by existing data dictionaries.

4.4 Entity modelling benefits

The entity modelling process derives most benefit from entity model clustering, mainly because entity model clustering was developed specifically to help with entity modelling. The previous benefits were side-effects found when the technique was developed, but this does not detract from their validity. A number of entity modelling benefits have already been discussed in the preceding sections.

The most important benefit of all is that entity relationship diagrams become much more stable and correct with entity model clustering. Due to the structure of a clustered entity model and its maintenance properties (see section 2), even very large models become easy to communicate, validate and maintain.

Our primary objective when developing entity model clustering was ease of use. We achieved this by allowing a major entity type to appear on any diagram where it is referenced by a relationship. This eliminates the cause of the majority of connections and reduces the number of models that need to be viewed for any particular purpose. The structure enables an entity type to be identified quickly in a top-down manner without any prior knowledge of its name/synonym. Our second objective in the development of entity model clustering was ease of maintenance. This has been achieved because the clusters minimise the impact of change (change is usually confined to one information area). Entity model clustering is thus a simple solution to simple design objectives.

Entity model clustering provides the context for the better analysis of functions. There is insufficient space in this article to consider this point, but there is an ongoing research project into this based at the University of Warwick.[23]

The fundamental entity types of an organisation are highlighted by entity model clustering, the major entity types. This knowledge is very beneficial for validation and communication. It is also very beneficial for design, as the fundamental entity types tend to require a lot of consideration when designing databases, e.g. for access paths. Invariably, major entity types are very complex with involuted relationships, a number of subtypes and complex key structures. The highlighting of major entity types during the analysis process gives the opportunity to identify their structure and for their keys to be fully analysed. As an example of the benefit of identifying major entity types, it may be found that a full study is needed into a Product Code, but it is unlikely that the key of an Order Line would require the same level of effort.

In the longer term, entity model clustering can be used as a basis for automating the construction of entity models by such means as 'template' models and 'blueprinting' (i.e. standard models which are elaborated to fit particular situations). When entity model clustering is implemented in conjunction with a data dictionary, the dimensions of a business context are provided without recourse to more complex solutions such as 'expert' data dictionaries.

5. CONCLUSION

Entity model clustering was developed in collaboration with Whitbread to address a recognised problem where there was no available solution. The technique does not alter the basic information content of an entity relationship diagram as there is no loss of information in the clustering process. If anything the opposite is true, because the technique allows information to be gleaned which is not easily extractable from conventional models.

The need for entity model clustering was described above, this being primarily to aid with the communication of large entity relationship diagrams. The technique is based on the abstraction of a conventional diagram to form a linked hierarchy of diagrams, and a largely algorithmic method was described to achieve this. However, human direction and intuition are still necessary due to the nature of the models which form the input to the method and due to the need to make the models communicable to humans. An example was given of entity model clustering which showed how the method would be applied and showed an end result of the method. Finally, a consideration was made of the many benefits which derive from entity model clustering over and above those which derive from the use of conventional entity models.

The technique was first applied to the Whitbread Corporate Data Architecture in June 1983 and has since had a wider application both within Whitbread and elsewhere, for example the Prudential Assurance Company. The technique has been proved to work and is easily applied. It builds on existing conventions, has been shown to be repeatable and potentially provides the basis for the automation of the production of entity relationship diagrams. Entity model clustering is still under investigation at the University of Warwick.

Acknowledgements

Part of this work was supported by the Science and Engineering Research Council (SERC) at Thames Polytechnic and CACI Inc International. G. Fitzgerald of the University of Warwick participated in the theoretical work and helped with this paper. I. Macdonald of James Martin Associates provided many useful comments on this paper.

REFERENCES

1. D. C. Tsichritzis and F. H. Lochovsky, *Data Models*, Prentice-Hall, Englewood Cliffs, New Jersey (1982).
2. I. Macdonald and I. Palmer, System development in a shared environment, in Ref. 18.
3. R. Rock-Evans, *Data Analysis*. IPC Business Press, East Grimstead, England (1980).
4. M. J. R. Shave, Entities, functions and binary relations: steps to a conceptual schema. *The Computer Journal* **24**, (1) (1981).
5. BCS Information Systems Analysis and Design Working Party of the BCS Database Specialist Group, *Information Systems Development: A Flexible Framework*, (1984).
6. P. P. Chen, The entity-relationship model: towards a unified view of data, *ACM Transactions on Database Systems* **1** (1976).
7. R. A. Davenport, The application of data analysis – experience with the entity relationship approach. In *Entity-Relationship Approach to Systems Analysis and Design*, edited P. P. Chen. North-Holland, Amsterdam (1980).
8. H. Ellis, A refined model for the definition of systems requirements. *Database Journal* **12** (8) (1982).
9. M. Flavin, *Fundamental Concepts of Information Modelling*. Yourdon Press, New York (1981).

10. A. Parkin, Data analysis and system design by entity-relationship modelling: a practical example. *The Computer Journal* **25** (4) (1982).
11. R. Veryard, *Pragmatic Data Analysis*. Blackwell, Oxford (1984).
12. M. Brodie and E. Silva, Active and passive component modelling: *ACM/PCM* in Ref. 14.
13. J. Smith and D. Smith, Database abstractions: aggregation and generalisation. *ACM Transactions On Database Systems* **2** (1977).
14. T. De Marco, *Structured Analysis and System Specification*, Yourdon Inc. New York (1978).
15. E. F. Codd, A relational model of data for large shared data banks. *Communications of the ACM* **13** (1970).
16. J. Martin. *An Information Systems Manifesto*, Prentice-Hall, Englewood Cliffs, New Jersey (1984).
17. BCS Data Dictionary Systems Working Party, The British Computer Society data dictionary systems working party report. *ACM SIGMOD Record* **9**, 4 (1977).
18. T. W. Olle, H. G. Sol and A. A. Verrijn-Stuart, *Information Systems Design Methodologies: a Comparative Review* North-Holland, Amsterdam (1982).
19. D. Sizer, Managing information as a corporate resource. *BCS Computer Bulletin* (September 1982).
20. J. Martin, Managing the database environment, Savant Institute (1981).
21. F. Murtagh, A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal* **26** (No. 4) (1983).
22. P. Feldman, A diagrammer for the automatic production of entity-type models. In *Proceedings of the Third National Conference on Databases* (*BNCOD3*), edited J. Longstaff, Cambridge University Press, Cambridge (1984).
23. P. Feldman and G. Fitzgerald, Action Modelling: a symmetry of data and behaviour modelling, in Proceedings of the Fourth National Conference in Databases (BNCOD 4), edited A. F. Grundy. Cambridge University Press, Cambridge, 1985.

Announcements

4–5 SEPTEMBER 1986

A Workshop on Medical Microcomputer Applications. The Middlesex Hospital, London W1. The subject area will cover a wide field in software development and hardware design that reflects current efforts to apply microcomputer technology in clinical practice and medical research.

For further information contact:

Mr P. D. Coleridge Smith F.R.C.S., Department of Surgical Studies, The Middlesex Hospital, Mortimer Street, London W1P 7PN (01-636 8333, ext. 7434/5).

In addition there will be a one-day course on **Multi-user microcomputer systems** at the Middlesex Hospital on 3 September 1986.

14–16 APRIL 1987

Automating Systems Development

Leicester Polytechnic, Leicester, England. An international conference on computer-based tools for information systems analysis, design and implementation.

The aim of the conference is to bring together researchers and practitioners in computer-aided systems development in order to exchange ideas and establish the state of the art. The conference will have two parallel streams – practitioner and researcher – and papers are invited in any area of automated systems design, for example:

- ● Automated data modelling
- ● Data dictionary systems
- ● Knowledge-based system design
- ● End-user computing
- ● System design tools
- ● Integrated project support environments
- ● Distributed information systems
- ● Adaptive systems

Key dates: 31 July 1986 (or as soon as possible thereafter) – abstracts of papers to be submitted; 30 September 1986 – requests for full versions of the papers issued; 31 January 1987 – papers submitted in full.

The conference proceedings will be published and issued to all delegates. Room will be available for the demonstration of appropriate systems by researchers and commercial organisations. Keynote papers have been invited from distinguished researchers and practitioners in the field.

Abstracts should be submitted to the conference organisers, David Benyon and Steve Skidmore at the School of Mathematics, Computing and Statistics, Leicester Polytechnic, P.O. Box 143, Leicester LE1 9BH.

For further details please contact

Short-course and Conference Unit, Leicester Polytechnic, P.O. Box 143, Leicester LE1 9BH