aspects of the systems and what kinds of ways of looking at the system can best be conveyed by any one tool.

8. An example

I shall conclude by describing one example of the way in which union participation has affected the system properties in one particular development.

It is the story of one new system which is being proposed in Norway but it is still not ready. It is a public data base for poisonous chemicals which will be available for access by people using these chemicals in their work place. The Chemical Workers' Union has sent a letter to the authority dealing with this data base making three points:

1. In this data base the union insists that at least one version of the description of the properties of the chemicals should be in a language understandable to the data shop stewards (as well as doctors and chemical engineers)—it follows that data could be held in a number of different language descriptions.
2. The union wants it to be possible to insert into the data base the subjective experience of using the chemicals in work place situations—for example the unions would wish to be able to record the medical impact on workers using those chemicals.
3. The union has asked to be included with each chemical a list of union representatives in organisations where there is experience in using that chemical so that if a firm begins to use a new chemical then workers within the firm know where they can go for information about how to handle the new chemical (such a system as the one above provides cross-contact, which is something which is most important to trade unionists).

The term cross-contact is not a dictionary term. It is one which has been developed by the Norwegian trade unions as a consequence of their first project. This leads to the final question: 'Is it right that trade unionists should need to be defining new terms as part of their involvement in the design process?'

A natural language data base interface to the user

P. H. Prowse* and R. G. Johnson†

One of the functions of the systems analyst is to obtain a requirement from a user and produce an outline design for the system. This design is described in a document often called a user specification. The user is then asked to approve this document. This approval often represents the last moment that the user can change his requirements without the risk of an expensive redesign. Consequently it is imperative that any omissions or misunderstandings in it are identified and corrected. The user specification must, therefore, be expressed as clearly as possible in terms the user can easily understand.

With a small set of batch programs the user is dealing with a relatively straightforward document describing a small range of functions. However, in a complex data base system in the very large data base (VLDB) class with several thousand data elements and many hundred relations with significant inter-relation dependencies it is improbable that user management will be able to fully understand the details of the proposed design. The details of the design are obtained from the user and, although the designer can ensure that the information is self consistent, only the user can ultimately check the design.

The existing approach to the documentation of the design maintains very detailed definitions of each data element in the data base and sometimes a brief description of the relations. These definitions generally ignore the fact that there is an underlying domain to the data element and the various roles which the domain plays in different relations may be difficult to determine. Consequently new methods are required to improve the clarity of the description of the proposed design. In particular the constraints imposed by the design on the user, in terms of possible real world relationships which cannot be handled, need to be made very clear.

This paper suggests an approach in two stages to making the relationships clear. Firstly it stresses the importance of third normal form relations as a model of the business system required. From these are derived a set of formal statements about the inter-relation dependencies, or 'business rules'. The second stage uses the business rules and the relations to generate English language sentences describing the system.

Modelling the system

Since Codd wrote his seminal paper (Codd, 1970), much research work has been carried out on all aspects of the use of relations. In the design area, most authors are now agreed that an essential component of the design process is the creation of a third normal form model.

The properties of third normal form models are well described in Wiederhold (1977). For design, one of the main advantages is that a list of data items is generated together with an explicit statement of the functional relationship of elements to each other. From this model data base implementations such as Codasyl networks can be simply created.

The relational model created by the designer contains a great deal of information that the user should agree in the user specification. The designer has identified a large number of distinct data items, each with a unique name, for which he can supply a definition. In addition, for each data item, he has

*Esso Europe Inc, 5 Hanover Square, London W1.
†School of Mathematics, Statistics and Computing, Thames Polytechnic, Wellington Street, London SE18 6PF.

decided that it is functionally dependent on other domains in the relation in his third normal form model and that other domains may be functionally dependent upon it. This information, which contains the fundamental relationships between data items in the design, is frequently not explicitly explained to the user. It is present only implicitly in the system description and the proposed relations.

Business rules

The business rules specify the relationships between the relations. These determine limits on changes in the real world which the system can support without modification to the data structures. The method of producing business rules is illustrated by an example. The example chosen is a practical one which has been streamlined so that it can be used as a case study on a logical data base design course. The example taken for a pilot study consisted of 57 data elements, of which 43 were unique, and 10 relations. The relations, which form the interim solution to the case study, are in a raw third normal form. They are refined during the logical data base design course but are taken in this case in their raw form in order to see what effect the description of concepts might have on the normalisation process.

The relations are given in **Fig. 1.** Each relation has a name: for example, purchase order. The attributes applicable to each relation are contained within the brackets and the key to the relation is underlined. The method for producing the business rule starts by creating a key table, which cross references the key domains against the relations. This is shown in **Fig. 2.** In the table, K represents the key domains for each relation, and F represents a domain that is a key domain in a relation, but not in the particular relation, i.e. a foreign key.

From the key table it is possible to determine the relationships between the individual relations. For each relation in turn, the 'K' entries are matched against the entries in the other columns in the table. If the key domains in one relation are the same as those in another relation then the relationship is said to be 1:1. This implies that for one occurrence of the first relation there can be at most one occurrence of the second. If all the key domains, 'K' entries, of one relation are included in another relation then the relationship is said to be 1:N. For each occurrence of the first relation there can be more than one occurrence of the second. If neither of the above conditions

1. PURCHASE ORDER	(P-ORDER No., SUPPLIER No., BUYER CODE, DATE PLACED, USER DEPT., USER LOCATION, PAYMENT TERMS, REQUESTOR NAME, AUTHORISER NAME, ATTENTION NAME)
2. PART	(PART No., DESCRIPTION)
3. PURCHASE ORDER PART	(P-ORDER No., PART No., UNIT CODE, AVERAGE PRICE, TOTAL QUANTITY ORDERED)
4. PROMISED SHIPMENT	(P-ORDER No., PART No., SHIPMENT DATE, DELIVERY DATE, QUANTITY ORDERED, PRICE, DELIVERY STATUS)
5. ACTUAL RECEIPT	(P-ORDER No., PART No., SHIPMENT DATE, DELIVERY DATE, QUANTITY RECEIVED)
6. REJECT	(P-ORDER No., PART No., SHIPMENT DATE, REJECT DATE, REJECT REPORT No., QUANTITY REJECTED)
7. BUYER	(BUYER CODE, NAME, DEPT.)
8. SUPPLIER	(SUPPLIER No., NAME, ADDRESS LINE, CITY, STATE, ZIP CODE)
9. PAYMENT COMMITTED	(SUPPLIER No., BUYER, QUARTERLY COMMITMENT, ANNUAL COMMITMENT, QUARTERLY PAYMENT, ANNUAL PAYMENT)
10. PAYMENT	(SUPPLIER INVOICE No., P-ORDER No., INVOICE DATE, DUE DATE, GROSS VALUE, NET VALUE, PAYMENT STATUS)

Fig. 1

A PURCHASE ORDER has many:	PURCHASE ORDER PARTS
A PURCHASE ORDER PART has many:	PROMISED SHIPMENTS ACTUAL RECEIPTS
An ACTUAL RECEIPT has many:	REJECTS
A BUYER has many:	PURCHASE ORDERS PAYMENTS COMMITTED
A SUPPLIER has many:	PURCHASE ORDERS PAYMENTS COMMITTED
A PAYMENT COMMITTED has many:	PURCHASE ORDERS

Fig. 3

Key table

Key domains	*Relations* 1	2	3	4	5	6	7	8	9	10
P-ORDER No.	K		K	K	K	F				K
PART No.		K	K	K	K	F				
DATE				K	K	F				
REJECT REPORT No.						K				
BUYER CODE	F						K		K	
SUPPLIER No.	F							K	K	
INVOICE No.										K

Fig. 2

hold, then the relationship is M:N, that is there is no relationship between the occurrences of the two relations.

The business rules generated from the table are shown in **Fig. 3.** The relationships between all pairs of relations not listed are M:N. The importance of these relationships is that they specify bounds on the model which has been created. For example, the relationship of relation PURCHASE ORDER PART to relation PROMISED SHIPMENT is 1:N. In the example the promised shipment date is part of the key of the PROMISED SHIPMENT relation and allows the computer system to process the despatch of an item from an order in a number of separate loads.

Business rules can play an important role when amendments are being made to an existing system. For example, in a business system we may have details of departments and managers. If each department has a manager and each manager looks after one department, this is a 1:1 relationship of departments and managers.

It is now possible to choose one of the two following sets of relations for our model.

Model 1

MANAGER (MANAGER No., NAME, EMPLOYMENT DATE, BIRTH DATE)
DEPT (DEPT No., NAME, MANAGER No., TOTAL EMPLOYEES)

Model 2

MANAGER (MANAGER No., NAME, DEPT No., EMPLOYMENT DATE, BIRTH DATE)
DEPT (DEPT No., NAME, TOTAL EMPLOYEES)

The key tables for the two models are shown in **Fig. 4**(*a*) and (*b*). From Fig. 4(*a*) the relations MGR and DEPT are in relationship 1:N, while in Fig. 4(*b*) the relationship is N:1.

	MANAGER	DEPT
MANAGER No.	K	
DEPT No.	F	K

Fig. 4(*a*)

	MANAGER	DEPT
MANAGER No.	K	F
DEPT No.		K

Fig. 4(*b*)

Suppose that in the business modelled by this system, a department expands until it is necessary to appoint a second manager within this department. It is now necessary to amend our computer system to allow for this change. In effect the previous real world relationship of manager to department was 1:1 and has been replaced by a more general one of N:1.

If we had used the model in Fig. 4(*b*) we would see that the data model will already support this change. It may be necessary to amend report layouts and similar details but the data structures will support the new relationships. However if we had used the model in Fig. 4(*a*) a more fundamental change would have been necessary, since the existing data model must be changed to support the new relationship.

The authors believe that business rules can be agreed by the user with guidance from the analyst. The analyst will be able to bring to the user's attention any business rules which seem to him to be in possible conflict with the user's wishes as previously determined. It should be noted that the analyst need only concern himself with business rules that restrict the user, for example 1:N when the user wishes M:N, since generalisation while possibly inefficient does not restrict the ultimate system.

Narrative

The authors believe that it is vital that the relationships between data are highlighted for the user, when obtaining his acceptance of them. The objective is to develop a good description to explain a data base to users in a friendly fashion that does not require any understanding of data base technology. The key to this is understanding the concepts behind each relation and documenting them. These concepts are implicit in the data base design and sometimes may be inferred from the existing data dictionary definition data. The concepts help unravel a data base. The user should be encouraged to read the description of those parts of the data base pertinent to his business function and to comment on their accuracy.

The description of each relation should be in a narrative style with good flow. Technical jargon should be avoided, but if it adds to the clarity and is well known to the user, then it may be used if it is explained in a footnote. Every data element should be included, but in a natural way using everyday terms rather than the precise class words and attributes used for defining data elements. Background information should be added where it seems most appropriate. We again use the example in Fig. 1. One paragraph is allocated to the explanation of each relation. The method employed is to convert each relation into a series of sentences, describing the relationships between the data items in the relation. The sentences are readily understandable to the user and he can detect the presence of any errors.

The process of producing narrative text is easy if the work is done by an analyst fully conversant with the area. Equally well a knowledgeable user might be trained to produce the narrative text. The time taken to produce the text in the example was approximately two hours or 12 minutes for each relation. In addition the narrative text includes many of the business rules. An area for future investigation is the possibility of machine generation of the text, possibly prior to editing by the analyst.

The solution shows the final text. A list of all concepts defined is also given. The concepts are really 'business primitives' and need to be reported in the logical design along with the relations.

Natural language text

1. *Relation name: purchase order*

A buyer, who is identified by a code, places a purchase order on a supplier. This purchase order is allocated one number and issued on one date. It is placed for one department and one delivery location. The material is authorised by one manager and the purchase order is for the attention of one person in the supplier's organisation. The purchase order specifies payment terms which are represented by a code indicating payment by a particular date.

2. *Relation name: part*

A unique number is assigned to each part which is purchased. There is one description for each part.

3. *Relation name: purchase order/part*

Many parts may be ordered on one purchase order. Each part has:

(*a*) one unit of measurement

(*b*) total quantity ordered on all shipments

(*c*) an average price for all shipments.

4. *Relation name: promised shipment*

A supplier may promise many shipments of a part on a pur-

chase order. Each shipment is for a specified quantity of parts to be shipped on one date and delivered on another date. The status of delivery is recorded by a code.

5. *Relation name: actual receipt*
A quantity of parts for an actual shipment made on one day is received on another day. This quantity may not be the same as that promised.

6. *Relation name: reject*
Quality control require that some parts be inspected. The quantity of parts which are rejected are recorded on a reject report together with the date of shipment, the part number and the purchase order number. A unique number is allocated to each reject report and a separate report is made out at the end of each working day for each shipment.

7. *Relation name: buyer*
Each buyer is allocated a code number. His name is recorded as well as the department that employs him.

8. *Relation name: supplier*
A supplier may have many addresses. A unique number is allocated to each address. A supplier's name is recorded with each address as well as street, city, state and zipcode.

9. *Relation name: payment committed*
A buyer is authorised to make commitments of money, to an agreed level, with particular suppliers. The amount of money committed by each buyer is recorded by quarterly and annual totals for each supplier. Payments to the supplier are also recorded in quarterly and annual totals for each buyer.

10. *Relation name: PAYMENT*
Payments are made against invoices received from suppliers. These invoices may refer to many purchase orders. The date of each payment is recorded together with the date of payment expected by the supplier, the gross amount payable and the net amount paid. The status of payment is indicated by a code.

	CONCEPTS
BUYER	AGREEMENT
PURCHASED	PAYMENT TERMS
IDENTIFIED	MONEY
CODE	TIME INTERVAL
PLACES	PART(S)
PURCHASE ORDER	DESCRIBED
AUTHORISATION	TEXT
MANAGER	ALLOCATED
ATTENTION	REFERRED
PERSON	EMPLOYED
NUMBER	UNIT OF MEASUREMENT
PLACED	TOTAL QUANTITY
SUPPLIER	AVERAGE PRICE
DATE	ACTUAL SHIPMENT
DEPARTMENT	RECEIVED
USES	INSPECTED
LOCATION	REJECTED
	RECORDED

Conclusion
The authors believe that greater efforts must be made to ensure that the user has a fuller understanding of the implications of the design produced by the analyst. Towards this end this paper has listed two complementary approaches. Firstly, the business rules which specify the relationships between relations and secondly the narrative text which describes each relation.

Both techniques allow the user to understand more fully the implications of a design. In consequence the user's approval of the outline design will be based on a clearer understanding. This should lead to fewer changes being made later due to errors in the design, with a resulting decline in development times and costs for new projects.

Acknowledgement
The authors wish to acknowledge the contribution of Ms. G. Butler of Exxon Corporation in the introduction of the business rules concept and the case study used in the logical data base design course. They also wish to thank Exxon Corporation for permission to publish the paper.

References
CODD, E. F. (1970). A relational model for large shared databanks, *CACM*, Vol. 13 No. 6, pp. 377-387.
WIEDERHOLD, G. (1977). *Database Design*, McGraw-Hill.

The user interface

E. B. James
Computer Centre, Imperial College, Exhibition Road, London SW7 2BX

Why we need a new approach
Our purpose here is to discuss new approaches to systems analysis and design which may help to alleviate a growing dissatisfaction with the performance of existing computing systems. These systems are under attack on two main charges. The first suggests that the systems do not do what is required by the users; there is a mismatch between the views of the designers and the users as to what was originally required of the system. The second charge is that it is far too difficult to make the computing system do anything useful. Other papers in this symposium concentrate on the first issue. We propose to concentrate on the second issue, which is concerned with the quality of the 'user interface' as it is called. We feel that this is particularly urgent because the type of user typical of a computing system is changing rapidly. The far less experienced user who will soon make up the principal proportion of all users will have, it appears, even greater dissatisfaction with existing computing systems and their dissatisfaction is likely to become rapidly more vocal.

In this discussion, we aim to define those qualities which contribute to ease of use and which seem to be missing from existing designs. We try to determine who is capable of defining these qualities in detail and we look at some previous attempts to satisfy these requirements. Then we consider the current