THE DEVOLUTION OF FUNCTIONAL ANALYSIS

Workpaper

26th October 1982

Ken Meyer
British Gas
and
John Doughty
Honeywell Information Systems, Ltd.

The opinions expressed in this paper are those of the authors and do not represent the opinions, statements and policies of their respective employers and of the DDSWP.

1. INTRODUCTION

In recent years the completion of a comprehensive functional analysis has been omitted from the work of the Analyst(1). There has been an emphasis on data specification in system design rather than on the user's functional requirements. This resulted from the needs to design data bases carefully from a performance point-of-view because of the way in which data base management systems structured the data base. Consequently highly "structured" data bases followed from this and these were not flexible in handling changes in system requirements.

Systems building software tools are now available allowing a more flexible approach to system specification such that there is less emphasis on data definition and more on functional definition. Data base software in the future will provide access paths to data rather than leaving the determination of access paths to the systems designer. Work will still need to be done on the provision of requirements information enabling the data base software to hold access path determination criteria.

Stemming from the work of the ANSI SPARC Committee,(2) the BCS Data Dictionary Working Party (DDSWP) in 1976 outlined a comprehensive description of a meta data dictionary, its benefits and uses.(3) The DDSWP followed this report by concentrating its efforts on a debate of concepts to be included in the dictionary and recording points of agreement in a Journal of Development.(4) This Journal of Development has been enhanced for publication this winter.(5) Earlier concepts have been implemented in one Data Dictionary System.(6) Most of the DDSWP work on concepts dealt only with the concepts of entity, attributes and relationships mapped to elements, properties, triggers and constraints

Users have tended to use packaged dictionaries by a description of the more essential records-data-sets at the implementation level followed by the description of the real world (of the business) in terms of entities-attributes-relationships (EAR) which match (and map to) the implementation level.

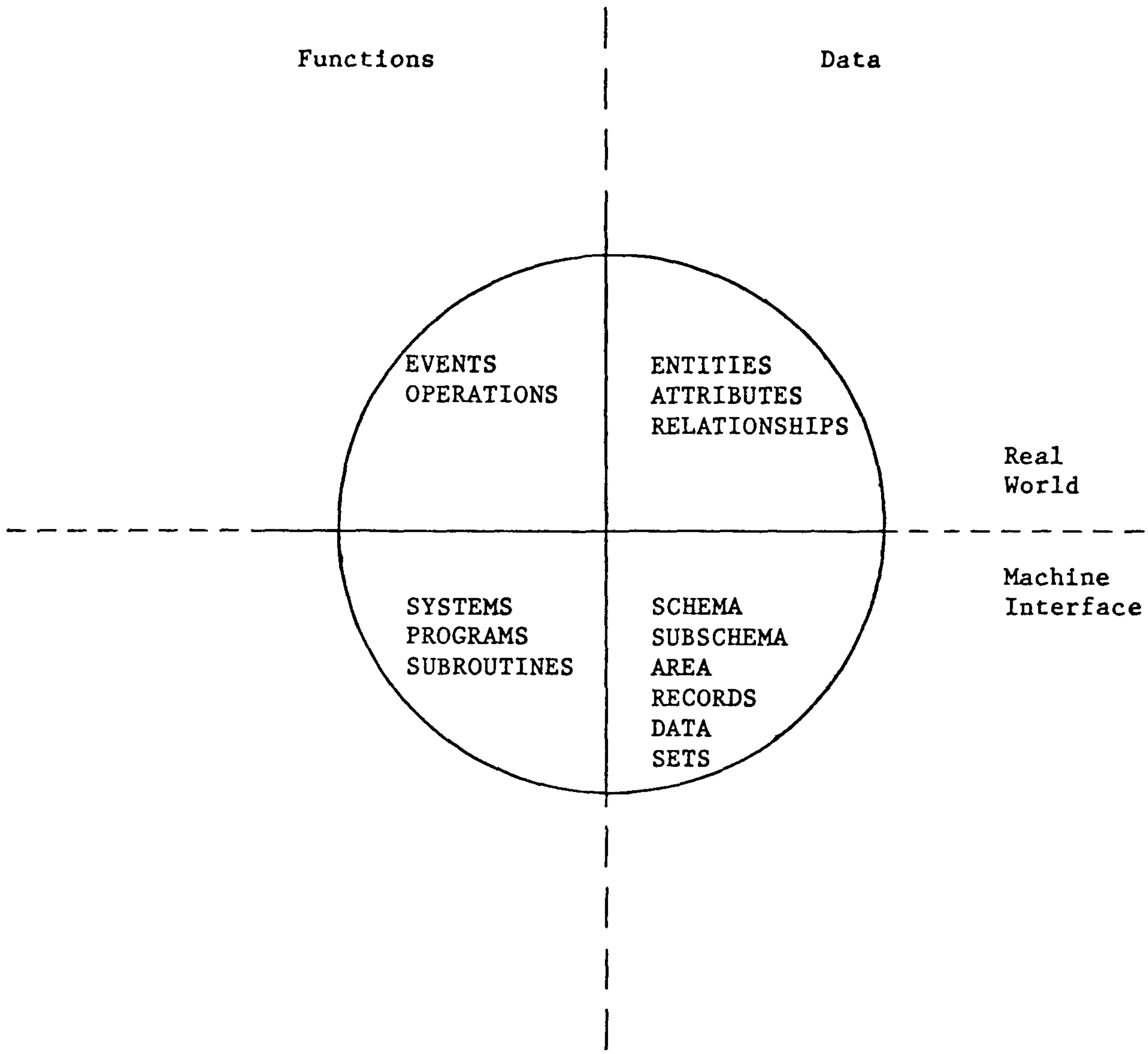


Figure 1 - The Data Dictionary Schematic

As a result of the emphasis on data definition, two points have been made; (1) the data descriptions convertible to COBOL and structured databases were entered in dictionaries. Implementation level data definition language (DDL) could be transcribed easily and a common reference cell of machine translatable statements could be readily available. (2) The conceptual level could be built-up into a glossary and justified by the successful implementation of a structured database. Thus the conceptual entries have become more or less mirror images of the implementation of databases, rather than a true description of the actual user requirements.

In doing so, the Analyst has not permitted the dictionary process to do what it should do best, normalise data at the conceptual level.

The lack of an automatic translation of conceptual level requirements into the implementation level has prevented the description of events and operations to the extent planned by the DDSWP.

There are structural prohibitions in the first generation of DD systems, although additional properties may be compiled into the system.(7) The structures of EAR have themselves created a fixed block to development, and to useful ends. In this instance standardisation occurred too early by premature implementation.(8)

It is possible to define a data dictionary using new concepts. In this sense the older dictionary may be used to define a new type of dictionary system, which is probably best called an "information resource manager".

2. A FUNCTIONAL APPROACH

This paper deals with a Functional Analysis model. A further paper will deal with Attribute-Role designations and the possibility of entity-relationship generation on the basis of automatic normalisation. Both explore the structures needed to describe a meta meta dictionary which fit the logic of the respective papers.

Throughout the system requirements/specification phase the Analyst has interviewed user staff to accumulate a complete and comprehensive picture of the workings of the business. More than likely this is accomplished on an applications level, an area of interest to the firm and a "bit beyond" which covers the obvious interfaces.

From this information the Analyst is expected to sift, define and designate a description of the workings of the business with allowances for variations, alternative paths, differences in practices and contingencies. In other words, the Analyst should have determined the system requirements within the organisation and written it into a specification which is used for writing programs.

A pure functional model is "a description of the meaning and structure of the uses of data in the business, unconstrained by hardware, software or application system requirements."

3. STATE AS A STARTING POINT

A State is "a measurable and momentarily stable position in which an enterprise finds itself."

States of affairs, or simply States, established a fact of existence possibly and logically, as opposed to chance. Each State is composed of constituent objects at any given time. An approximate State is one that reduces the number of constituents to the number of internal properties of each constituent object, more or less, because some constituents are not germaine, others have invariant values and others have such ranges of values as to present no pattern.

Entities may be the objects involved in forming the State, but each one of these Entities are composed of Attributes. Attributes have a Value or a range of Values which determine collectively the State. Entities have an identifying Value, a key (unique or not), but this is not sufficient in detail by itself to determine a State.

4. A STIMULUS DISTURBS THE STATE

A stimulus or trigger "is an occurrence which attempts to change a State."

The stimulus is the cause of a change in State whenever this exists. Some stimuli will not change a state, only because the mechanism is single State (recorded at the entity level) and therefore has no complex Attributes with Values. On the other hand, some complex pattern of Attributes have fixed Values (short of a calamity).

On the other hand, States can exist and several States can exist and by application of stimulus the States change. This is typical of a machine; for example, a reciprocating pump. The stroke of the piston is in one of four principal positions; at the top end of the stroke; the bottom end of the stroke; and the two positions which are dependent on the direction of the thrust.

A computer may be said to operate in States and the simplest of these is a binary counter. Changes of States are initiated by stimulii and it is immaterial in which State the machine is in when the stimulus is applied. The change in State cannot be interupted regardless of how long it takes. Each State is a "wait" state and is stable pending arrival of the stimulus.(9)

States therefore are required to be stable and the change in States may be of any length. A system may not change State without receiving a trigger which causes the change to begin.

Trigger has been used as the basic primitive to explain changes in State by the DDSWP, but this has been mapped to Event and the difference is semantic and was coined by DDSWP to avoid direct reference to any implementation. Event is more widely accepted.

5. EVENT AS THE STIMULUS

An Event is "whatever comes to pass, occurs or takes place which may initiate, change or terminate a function."

An Event, then, begins a function and a function is a series of Operations. This will inevitably change the State of the organisation. The Event definition is broad enough to include the trigger by inclusion of the option inherent in the trigger of doing nothing and yet to name the Operations collectively as functions.

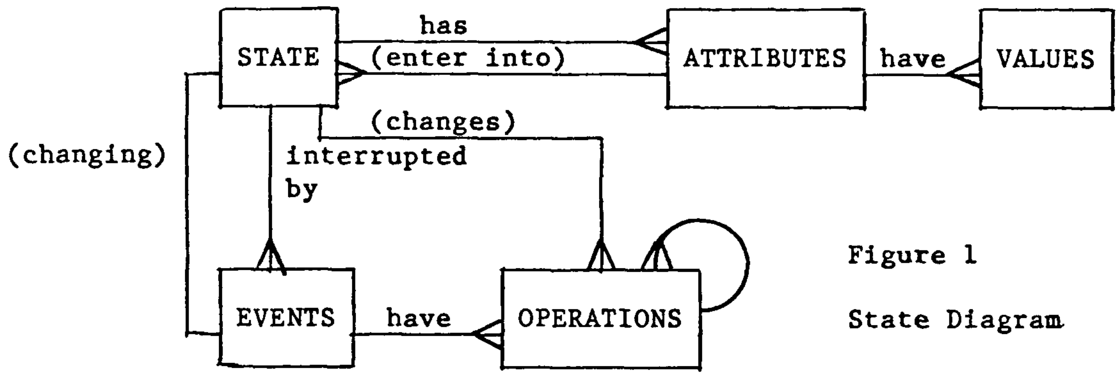


Figure 1

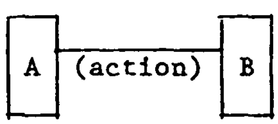
State Diagram

Figure 1 shows the relationship between State and Attributes (with Values) and between State and Events duplicated by the Operations. Perhaps it would be better to show that an Event causes a function which changes State through Operations.

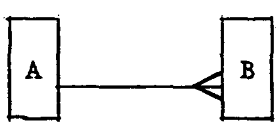
For those not familiar with the notation used

represents an element. The name identifies the element uniquely.

A single line joining two boxes represents a 1:1 linkage between two elements, A and B. Action represents the relation of B in respect to A.



Same as above, but parenthesis around action represents the relation of A in respect to B.



This convention represents A and B, such that a 1 to many linkage exists.

Events are generally featured on forms or screens sometimes called "Information Interview Displays" as a starting points of a sequence of operations done by an employee. Sometimes these operations are numbered with a short description of each operation with the Attributes affected. This is a principal means of tracing the creation, alteration, utilisation and deletion of data.

6. FUNCTION STEMS FROM EVENT

Function is a logical sequence of Operations which stem from an Event. All Information Interview Displays will contain a functional description of the operations performed on Attributes, including intermediate clerical tasks and delays to these tasks in completing the function. Some of the steps will not employ data. Others will record data at its origin (reading a meter) or perform monitoring functions until a change in state is recorded.

Much of the discipline at writing these functional descriptions has had its origins in O&M work.(10)

Function is defined as "the sum total of the allowable combination of steps to resolve or satisfy the change in State stemming from an Event". This definition, however, suffers from the convenience of fitting the purpose of this exposition.

Function may also be defined as "a process performed which causes a change of State." It becomes a way of describing a high level change of State and then sub-defines process into sub-processes. Therefore, the first definition is cleaner, because it does not depend on further breakdowns.

There are two cases which are less troublesome with the first definition. First, the case of an Event which immediately is concluded, as with a button which is depressed to light a light and, when depressed again, has no effect on the light. It is akin to pressing the doorbell when it is already ringing. A sub-function under those circumstances has less meaning than a null operation.

Secondly, the case of an event exists which has no effect at all, as with a telephone whose wires are no longer connected to the system or is unattended.

7. OPERATIONS SATISFY FUNCTIONS

Operations, taken together, represent the complete set of activities to satisfy the functional requirements. Data operations, a subset of operations, represent the complete set of activities originating, changing, accessing or deleting data within the (function of the) business.

Operations need not involve data at all. Some may have no relevance to an information system. The opening of a file cabinet drawer or the logging on at the terminal have no effect on data, but these are largely auxilliary or storage or waiting operations. Second, it is difficult to visualise the mechancial aspect of a running machine not being termed an operation.

An Operation "records an action, deed or thought which completes a step of work." Of course, this can be kept at a miniscule level in describing the changing state of a binary counter, but the recording is at a grosser level within information analysis.

In the data operation is the act, deed or thought originating, changing, accessing or deleting data within the function.

In the simplest sense, one Operation follows another Operation through a complete cycle or across a complete plane, until the function is completed and released when the state is altered (or a return is made to the original state, as with a null operation).

In the simplest sense, one Operation invariably follows another Operation each and every time. There are no variations based on individual practices, work policies, availability of materials (books, lists, references and previous history), completion of other tasks and so forth.

Events and Operations are a complex network, since neither of the two simple propositions are true.

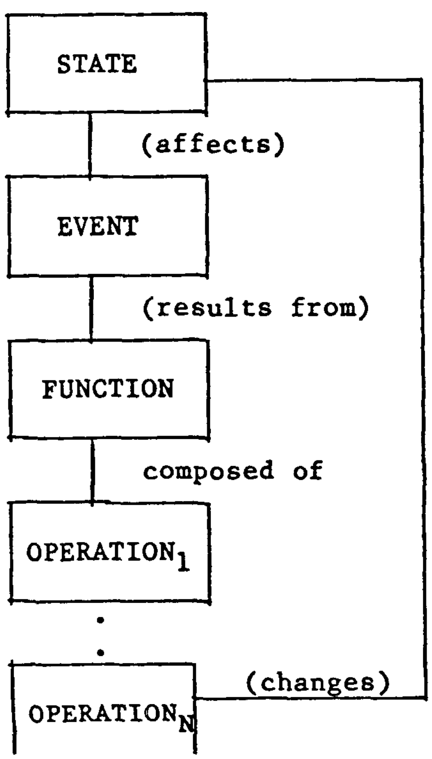


Figure 2

Changing State based on an occurrence of an Event

Figure 2 shows the sequence of elements necessary to satisfy the simple propositions. Note that the Operations which fulfill a function are not interrupted. Everything operates the same way every time. This cannot be true.

8. SEQUENCES DEPENDENT ON THE SAME OR OTHER ATTRIBUTES

In the Linkage Determinant, as an Event occurs and before and during any Operation is permitted to occur, other states of the business can affect pathways between State and Event or between Event and Operation and/or between one Operation and another. The Attribute tested may be set for interogation, but the values of the occurrence may alter pathways.

More than one value of an Attribute may be tested. More than one Attribute may be tested.

Values may be of any type or range specified in the definition of the Attribute or can be a State of an Event (state-event), logically portrayed by a semaphore, another Attribute.

Each test has a truebranch or, in multiples, a single truebranch. The default is a falsebranch.

Figure 3 shows how the testing of Attributes can determine sequences of Operations.

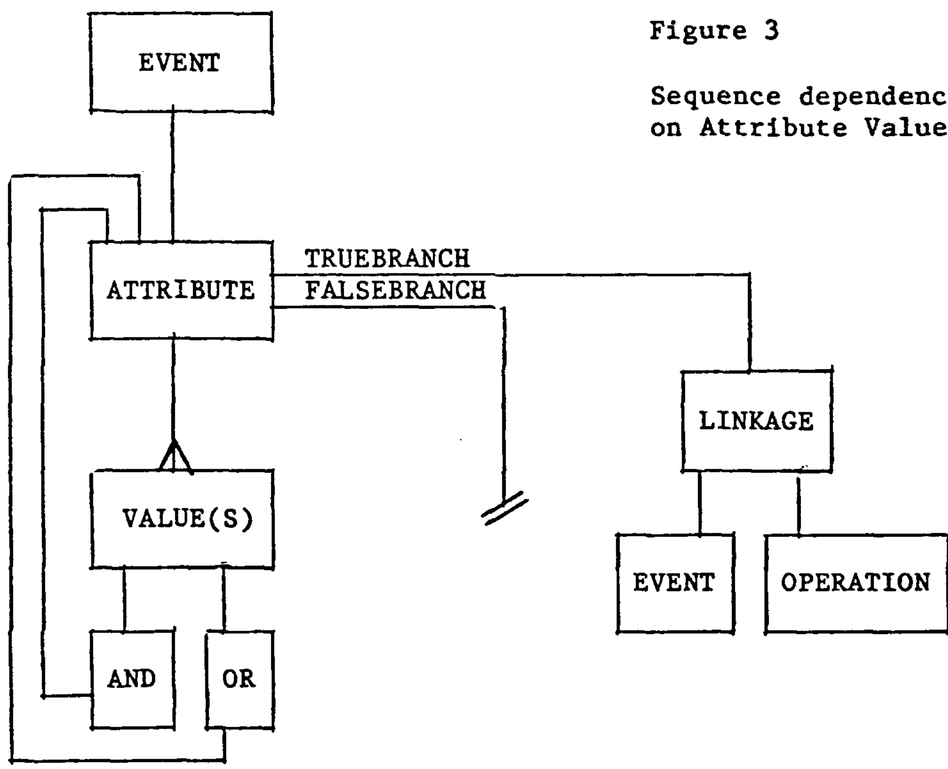


Figure 3

Sequence dependency on Attribute Values

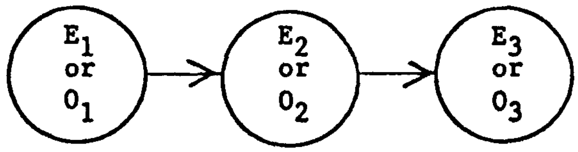
When truebranch is determined, a linkage may be made to one or more Events or Operations.

In reality, the falsebranch may be an active linkage. Similarly, conditional-expressions and negatives of the values of attributes or their sign or zero-value may be tested.

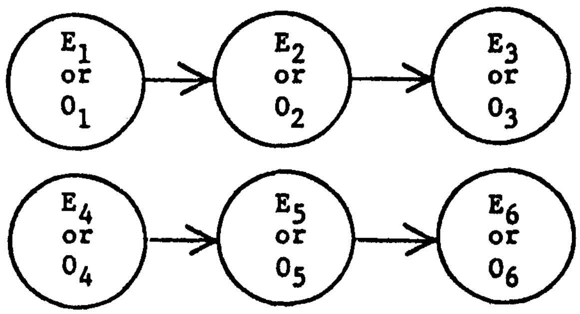
9. THE CONNECTION BETWEEN EVENTS AND OPERATIONS

Events are connected to other Events or Operations in one of five modes defined below.

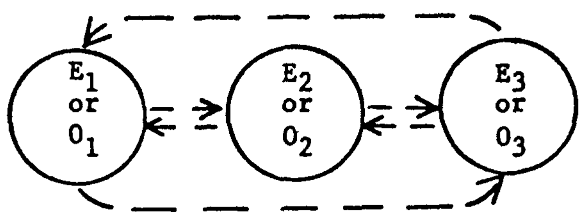
(1) Directional
One Event or Operation follows or causes another Event or Operation



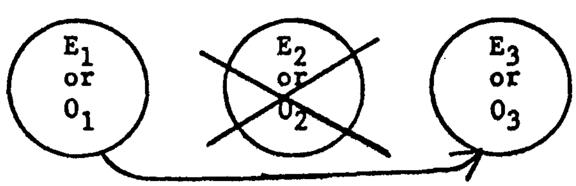
(2) Simultaneous
While one Event or Operation follows or causes another Event or Operation, another Event or Operation follows or causes another Event or Operation, at the same time.



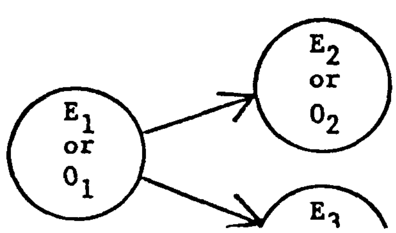
(3) Indifferent
While one Event or Operation takes place, another Event or Operation may take place, but it makes no difference which occurs first.



(4) Negative
One Event or Operation prevents another Event or Operation from being caused (or followed from another).



(5) Choice
One Event or Operation or another Event or Operation occurs as an alternative.



The logical expressions are shown as follows;

Directional	$P \rightarrow Q$
Simultaneous	$P \& Q$
Indifferent	$(P \rightarrow Q)$ OR $(Q \rightarrow P)$
Negative	$P \rightarrow - Q$
Choice	P OR Q

It may be projected into complex expressions and can result in new unique situations which should be recorded in the list above. For example, the simultaneous operation assumes two Directional, parallel functions in category (2), but one or the other or both may be Indifferent.

10. THE RECORDING OF AN EVENT

As a logical consequence of each Event or Operation, a notation is made of the passage of a State to the Event mentally by the clerk or person doing the job. The mirror image required, however, must demand a registry within the pool of Attributes used in processing the function stemming from Events as a whole. If this were called a global constant representing the transition from State to Event this concept would also take into account the convenient partitioning into functional areas.

Therefore, the semaphore system attached to the occurrence of Event is recorded globally and included wherever and whenever it is used. The process is automatic.

The Operation in sequence does not need similar recording, because of the mechanism in pt. 11 and also because an Operation never occurs in isolation without a preceding Event, a following Operation and either a return to the State or to stimulate an event-trigger.

11. LOGICAL CONSEQUENCE OF EVENTS AND OPERATIONS

Each Event and Operation is described in detail within the Information Interview Display and in complete and consolidated detail drawing from several forms and mapping back to the Events by the name and Operations by number. The reconciliation of diversity is expected, when the Formal Event Definition and Formal Operation Definition is set.

With the Formal Definition a number of unique operations are usually set for the definition. The number, however, is reduced by the automatic rendering of the condition attached to the event. The criteria for Operations are rigid;

(1) The Operation must always occur.
(2) The Operation may not access data
(3) The Operation assumes data is present when sufficient and necessary semaphores are set.
(4) Only global semaphores may be set.
(5) No other Events or Operations may be tested.
(6) The Operation must be unique and un-duplicated.
(7) The name of the procedure recording the Operation relates to the name of the Event and Operation.

Events are consistent, abstract, unique, unduplicated triggers attaching solely to the function that is selected as a consequence of the Event. More than one function may be triggered by an Event. If the State is "man standing under a tree" then the Event "tree is struck by lightning" can yield several functions.

Event has the tendency to look outwards from the firm, but Operations may be either inwardly or outwardly directed. Among the former, some Operations are minute in effect;

(1) The establishment of a counter.
(2) The adding of one number to another.
(3) The setting of a switch.
(4) The opening of a valve.
(5) The mechanical act of positioning the page within a printer.
(6) The answering of a telephone.
(7) The physical transmission of an ACK or NAK.
(8) A physical movement.

12. LOGICAL CONSEQUENCE OF THE LINKAGE BETWEEN EVENTS AND/OR OPERATIONS

While the logical consequence of an Event or Operation is restrained, the logical consequence of the passage from an Event to Operation, Event to Event or Operation to Operation also exists and is less restrained. These are of two magnitudes, simple and embeded.

The general nature of the linkage is to record all actions, deeds or thoughts recorded which occur before the transfer to the consequent Event or Operation. A simple consequence is one that refers to Attributes (any) as operands and contains operators and qualifiers to be used on them. the structure of the simple consequence is any standard high-level language. Most are named as procedures.

Many of the logical consequences will not have been completed at submission-time for the linkages forming the network. For this purpose, the codes ACCESS, UPDATE, CREATE, DELETE, ARITH, RECALL, CHANGE, REMEMBER, FORGET or their business analogue equivalents are permitted at this stage.

The procedures may be enumerated at the later stage. In this event the first line of the procedures must be user-specified with a procedure-name. Any COBOL construct specified as a user-named-object may be written in the procedure.

Complex consequences or embeded consequences may be recorded within the procedure by reference to an independent procedure. An independent procedure is a self-contained function which has a limited scope or outlook and which deals only with one function and group of attributes invariantly. While these are single in their purpose, they have an Event and an outcome in Event, but the Operation (only) as defined by the procedure is usable.

An illustration of the above is a page change on a printer for a named report, in which the operation specifies a skip-to-new-page, print line one of header, print line two of header, restore line count and skip two lines. The Event is the printing of the report-x and the page change is an Operation page-head-x which is tied to the Event by measurement of the variable line count against a constant.

The page-head-x may be forced at anytime by reference to the procedure only by PERFORM, not by invocation of printing of report-x.

13. THE NETWORK RELATED TO ATTRIBUTES

When the network is completed between Events and Operations as outlined above, the Attributes accessed by the Events and Operations have been referenced. Reference is made to Attributes;

(1) In an operation attached to an Event or Operation Formal Definition.
(2) In the interogation of a condition or value to determine linkage.
(3) In a procedure as a logical consequence of the linkage.

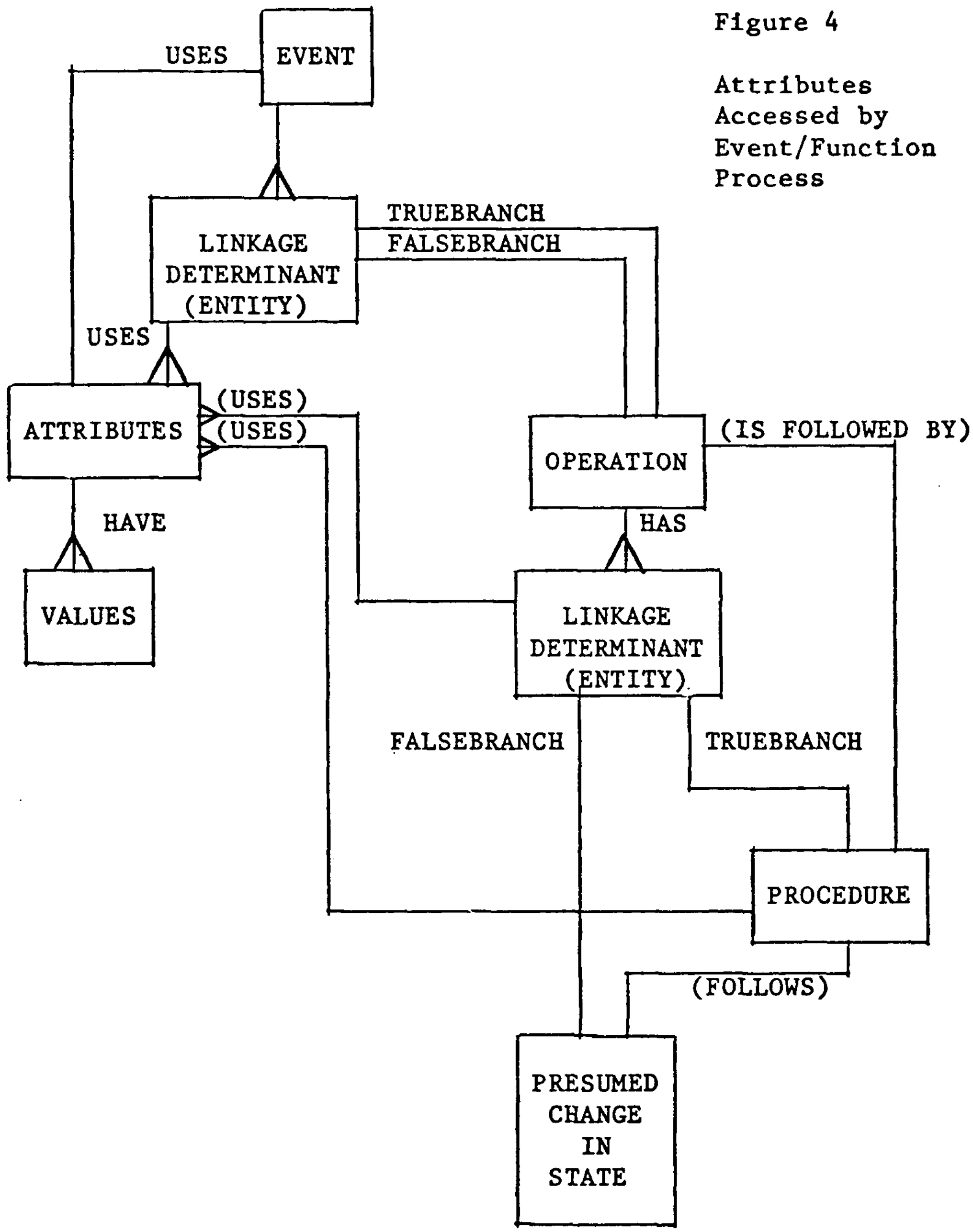


Figure 4

Attributes Accessed by Event/Function Process

Values accompany all attributes and must be noted specifically in the references drawn by linkage determinants.

Figure 4 shows how attributes are accessed in determination of linkages and in the procedures which stem from this determination. Observe that Attributes have Values which are subjected to strict integrity checks as to format, content, value and relationship to other Attributes (through Roles and Values) which are subject to integrity checks.(11)

The Linkage Determinant is not unique within structures. Entity represents any thing, object or concept and its existence is dependent upon how many internal characteristics are available and measurable. These internal characteristics are termed attributes and their measurements are termed values.

It is obvious that the Linkage Determinant is the Entity which accompanies any Event which disturbs State. It follows that as Linkage Determinants change, entity occurrences also change. The composition of entities may also be unstable, because their existence in a State is dependent on measurable attributes, subject to rules of evidence.

There are at least three types of entity and to distinguish this type of entity, it is called a Functional-Entity.

It is suggested that the Attributes arranged (as referenced) form a suitable framework for functionalisation or, if you prefer, a transaction analysis.

The same analysis can be rendered by taking Events and then Operations and setting out a record for each Attribute referenced as follows;

Subject Attribute Join Object Attribute Reference		
Event Linkage Operation	Event Linkage Event	Operation Linkage Operation

From this combination, a Transaction Analysis may be made and charted in the same way as a model of data can be constructed.

Figure 5

Attribute
References
by Function

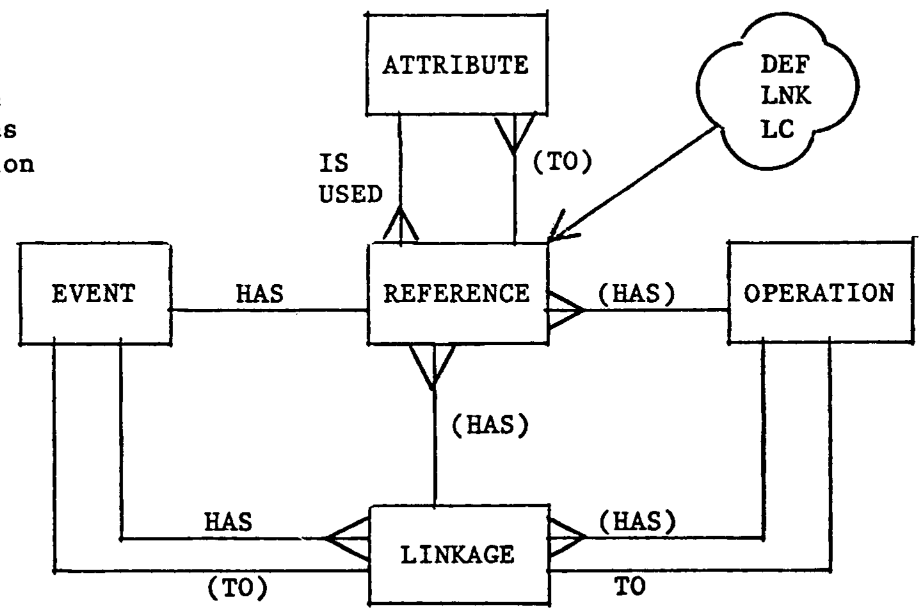


Figure 5 shows Attribute References by Function Stemming from an Event. This agrees with Figure 4, but is not an operational model. Neither does it provide for showing functionality of each Attribute.

Figure 6

Function
Attribute
Relationship

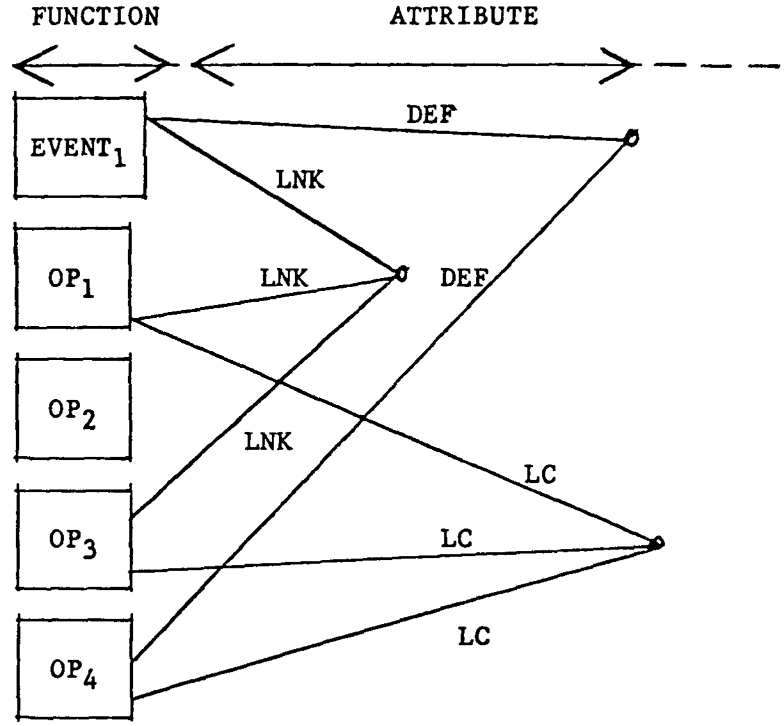


Figure 6 solves the problem of attribute identification, once the generlisation of figures 4 and 5 are understood.

The access paths may be clustered together by reference to linkage references and checked manually with the prime and alternative keys of the data analysis.

In the alternative the reference to the Attributes may be obtained by entering Roles between Attributes. Construction of a model of functions from Attributes and Roles may be identified in a hierarchy.

The pathways may be recorded as sections, individual accesses to records as procedures, and the entire transactional unit as a section or program.

14. THE TRANSCRIPTION OF EVENTS/OPERATIONS

The preceding sections have described the necessity, purpose and benefits from the accomplishment of a functional description and commitment to a data dictionary. This section describes how one method of transcription could be used to accomplish the tasks involved in a step-by-step programme. It may not necessarily mean that steps must be taken one after the other, but they must all be accomplished or found to be achieved within the firm.

(1) The Analyst prepares an outline of the study of a Functional Area and prepares a list of Events within this part of the business.

(2) The Analyst interviews employees about each Event and completes an Information Interview Display which shows the Operations performed by the Interviewee and the Attributes which we originates, uses, modifies or deletes.

(3) The Analyst completes a "Property Description" on each Attribute. In this fashion a checks is maintained between Events, Operations, Attributes and User-entities. Roles between Attributes are described.

(4) The Analyst proceeds to detail a formal definition of all Attributes and can extend the defintions of a Role as the study of the Attributes proceeds. Standard definitions are related back to synonyms.

(5) The Data Modelling phase establishes Entities and Relationships, assigning Attribute populations.

(6) The Analyst reviews the Information Interview Displays and groups these Displays by Event and by Events which, during execution of Operations tied to an Event, cause an Event. Identify all Events.

(7) Take an Event Definition Display and write a formal definition of the Event, tying the formal definition back to the original Event recordings.

(8) Take on Operation Definition Display and write a formal definition for each Operation.

(9) Relate the formal definition back to the original recordings (on the Information Interview Display) by interviewee, Event, Operation number and user-entity-serial-number (from the Property Description).

(10) For each Event, the linkage to Operations, and to other Events is studied. An Event Sequencing Display is prepared for each path stemming from an Event.

(11) For each Operation the linkage to other Operations and to Events is studied, as an adjunct to (10). An Operations Sequencing Display is prepared for each path stemming from an Operation.

(12) Because of the commonality between the two functions in (10) and (11) and their forms, steps (13) through (21) are meant to apply jointly.

(13) Whenever a Functional-Entity selection is made after an Event or Operation takes place to determine the path to another Event or Operation, the Attribute tested is shown on the Event Sequencing Display or Operation Sequencing Display under the IF heading. All high-level language (COBOL) comparisons (except arithmetic-expression operands) can be used, together with conjunctions.

(14) Arithmetic-expressions needed can be constructed in a Logical Consequnece Procedure, or, if further reference is made to the operands used in an arithmetic-expression in any other logical consequence procedure or IF-statements, at the Event Definition or Operation Definition level.

(15) Events or Operations are followed by other Events or Operations. The linkage is shown between Events and/or Operations is shown in the Linkage Section. The linkage is shown as one of five types;

directional
simultaneous
indifferent
negative
choice

Definitions are shown in section 9.

(16) Note that several selections may be placed on the same Event or Operation Sequencing Display. If no conjunction is shown after a will be taken as a separate selection in the IF section.

(17) Note that a single linkage must be shown, except when an indifferent mode is specified. In this case, all of the linkages must be shown on all forms which may describe linkages involving the indifferent mode.

(18) The Logical Consequence of the selection of a linkage between an Event/Operation and an Event/Operation in actual terms may be shown under the logical consequence section. Procedural COBOL language is permitted without exception. A procedure-name must be shown attached to the first entry, but further lines may or may not be given separate procedure-name.

(19) The operands shown are largely Attributes (for references to data), although all Procedure Division references are allowable.

(20) Whenever details are incomplete, only a procedure-name needs to be shown.

(21) Whenever reference to Attributes is necessary, but COBOL detailed coding is not available, the Attribute may be referenced by reference-verbs; viz:-

ACCESS
UPDATE
CREATE
ARITH
DELETE

(22) Other coding may be indicated on the Event or Operation Definition forms, similar to the Logical Consequence section, but no external references are permitted which are not used within the function. See Section 11 for the qualifications (operations). No PERFORM or CALL verbs are permissible.

(23) Lapse of time should be recorded in terms of seconds, minutes, hours, days.... in the Logical Consequence section and any Attribute causing the delay should be noted. If an Operation or Event completion is the cause of delay, it should be noted as;

```
WAIT FOR XYZ
```

(24) An original observation should be recorded as

$$\text{ORIGIN} \quad \begin{Bmatrix} \text{attribute} \\ \text{event} \end{Bmatrix}$$

$$\text{FROM} \quad \text{object-name}$$

15. IMPLEMENTATION NOTE

It is noted that the "data dictionary systems" currently marketed cannot be used to describe the elements, properties and linkages expected in this paper. The system, however, is consistent with any implementation of the Database Facility proposed for COBOL as considered by the CODASYL X3.J4 Committee and with ICL IDMS 200 (2900), IDMS From Cullinane Corporation for IBM, IDS-II for Honeywell and DMS-1100 for Univac. Smaller pilot implementations might be considered with MDBS-III. Forms layouts marked "Displays" should be virtual screens of up to 240 characters, akin to Univac's MAPPER of 133 characters, mapped to a physical screen of 80 characters.

16. CONCLUSIONS

Completion of a comprehensive functional analysis has been omitted, because of the emphasis placed on the completion of the implementation view before the conceptual view in the data analysis. It is suggested that there is a natural relation between Events and Operations which both relate to relevant Attributes and their Values. This paper has been concerned with the definition of elements associated with this part of the data dictionary and outlines one possible programme for implementation.

The authors acknowledge the assistance of members of the DDSWP and Mike Harper of British Gas in preparing this paper for publication. Comments should be sent to the Chairman, BCS Data Dictionary Systems Working Party, c/o British Computer Society, 13 Mansfield Street, London W1.

REFERENCES

1. Information Analyst, Systems Analyst, Data Base Administrator, Data Base Manager and staffs. In general, the systems development function, as opposed to maintenance, within the firm at present.

2. ANSI/X3/SPARC, Study Group on Data Base Management Systems, Interim Report, Feb. 1975.

3. Report of the Data Dictionary Systems Working Party, British Computer Society, March 1977. - Reprinted in SIGMOD Record, Vol. 9 No. 4 (December 1977), Special Interest Group on Management of Data, jointly with DATABASE, Vol. 9 No. 2 (Fall 1977), Special Interest Group on Business Data Processing. Available from Association for Computing Machinery (ACM).

4. K.H. Meyer, DDSWP Journal of Development, September 1977, as amended, out-of-print.

5. C.B. Morse, DDSWP Journal of Development, in press.

6. ICL Data Dictionary Systems, TP 6504, International Computers Ltd., Putney, as amended.

7. Beginning with version 600 of ICL DDS.

8. Charles Bachman argues that establishment of standards creates a fixed plateau from which develoment can take place, while Edgar Codd argues that experimentation is necessary. See "The Great Data Base Debate", Computerworld, 13/9/82.

9. C.S. Brookes, Functions, Events and States, CSB 7808, BCS DDSWP Proposal, 16/8/78, private.

10. V. Lazzaro, Systems and Procedures, 2nd Edition, 1968, Prentice-Hall, Inc.

11. This is the subject of another paper in preparation on "Automatic Normalisation and Entity-Relationship Generation through Attributes and Roles".