The use of decision tables within Systematics

By C. B. B. Grindley*

Systematics embraces a range of techniques for designing and describing information systems. One of these techniques, used to construct a model of an information system, is shown to use special case of decision table. The form of this table and certain advantages gained from using it are described in this paper.

(First received in January 1967, and in revised form in January 1968)

A range of techniques under the general title of Systematics is being developed for designing and describing information systems. The main feature of these techniques is that they allow the user to concentrate on the design and description of the information system without having to consider the computer strategy problems concerned with how the system is to be implemented. Some of these have been described elsewhere (Grindley, 1966). One of the basic techniques in this earlier paper was called an AND/OR Matrix. As King (1967a) has shown this AND/OR Matrix is a form of decision table. During the field work which has since taken place, this particular technique has undergone some modification. The present form of the decision table and the way in which it is used is described in this paper.

General decision tables

A full description of decision table techniques is given in King (1967b). The general decision table takes the form shown in **Fig. 1**.

Condition	Stub	Entry
Action	Stub	Entry

Fig. 1

In its extended entry form the condition stub contains the names of items whose state governs the actions to be taken. In the condition entry columns the different combinations of the states of these items are explored. Within the action stub, the names of the items to be derived are shown. In the action entry columns the different actions required to perform the derivations are given under the appropriate combination of states. See **Fig. 2**.

Sex	Male	Female . . . . . . etc.
Length of Service	<5 years	<5 years . . . . .
Pension Contribution	2s.	1s. 6d. . . . . . .
Holiday Entitlement	16 days	15 days . . . . . .

Fig. 2

This means, when sex is male and length of service is less than 5 years, pension contribution is 2s. and holiday entitlement is 16 days, etc.

Furthermore, the next step to be taken for each set of conditions may be indicated as in **Fig. 3**.

Sex	Male	Female . . . . . . .etc.
Length of Service	<5 years	<5 years . . . . .
Pension Contribution	2s.	1s. 6d. . . . . .
Holiday Entitlement	16 days	15 days . . . . . .
Go to	Table 14	Table 15 . . . . .

Fig. 3

Each set of conditions, together with the appropriate action is called a *rule*. See **Fig. 4**.

	Rule 1	Rule 2 . . . etc.
Sex	Male	Female . . . . . .
Length of Service	<5 years	<5 years . . . . .
Pension Contribution	2s.	1s. 6d. . . . . . .
Holiday Entitlement	16 days	15 days . . . . . .
Go to	Table 14	Table 15 . . . . .

Fig. 4

Decision tables in Systematics

In Systematics, the basic derivation statement is called an *element*. The element is in fact a special case of decision table in that it displays three special features. The form of the decision table just illustrated will be modified in order to illustrate each of these features.

The first feature of the element is that it is confined to providing the rules for the derivation of one derivative only. For example, length of service may influence pension contribution and holiday entitlement. A general decision table might explore all relevant states of 'length of service'. The element explores only those which affect one derivation, say, pension contribution. A separate element explores those which affect holiday entitlement. If some states affect both then these states appear in both elements. All the rules for the derivation

* *Urwick Diebold Limited, St. Andrews House,* 40, *Broadway, London, S.W.*1.

Pension Contribution

Sex Length of Service	Male <5 years	Female . . . etc. <5 years . . .
Go to	2s. Table 15	1s. 6d. . . . . . Table 16 . . .

Holiday Entitlement

Sex Length of Service	Male <5 years	Female . . . etc. <5 years . . .
Go to	16 days Table 15	15 days . . . . Table 16 . . .

Fig. 5

of the derivative are given in the one element. We thus identify a shift of emphasis from *fully exploring a set of conditions*, to *fully exploring the derivation of a single item*. There is, therefore, one separate element for each derivative. The element is named after its derivative, as in **Fig. 5.**

The second feature is to introduce a fresh set of conditions known as *primary conditions*. These conditions determine when the element is performed. They are distinct from the normal decision table conditions (called *secondary conditions*) which determine the particular derivation to apply. Primary conditions trigger the element as a whole. Secondary conditions govern the individual rules. For example, the element 'holiday entitlement' may be performed for each person on the payroll. Entitlement in a particular case may depend upon sex and length of service. A change of employee's number would therefore be the primary condition, sex and length of service would be the secondary conditions. This example illustrates an interesting difference usually found between primary and secondary conditions. Secondary conditions explore different states of an item, e.g. less than five years, five–ten years, over ten years, etc. Primary conditions usually explore the mere change of state of the item, irrespective of what its new state is, e.g. each employee's number. Thus we now have two sets of conditions as shown in **Fig. 6.**

Holiday Entitlement

Primary conditions	Employee's Number	For each	
Secondary conditions	Sex Length of Service	Male <5 years	Female . etc. <5 years . . .
	Go to	16 days Table 15	15 days . . . . . Table 16 . . . .

Fig. 6

The third feature is to substitute a 'use' list for the 'go to' information. 'Go to' provides a processing sequence. Systematics is not concerned with sequence outside that needed to derive one isolated derivative. To go further is to be concerned with computer processing strategy which is deliberately outside the scope of Systematics. On the other hand, it is extremely useful for the designer of the information system to know what part the derivative plays within the total system. The names of any other elements which use the derivative are therefore given, as in **Fig. 7.**

It is difficult to produce a complete statement of use manually. A computer can perform this function quite simply.

Holiday Entitlement

Employee's Number	For each	
Sex Length of Service	Male 5 years	Female . . . . . etc. 5 years . . . . .
	16 days	15 days . . . . .

Use: Deductions, holiday pay.

Fig. 7

Advantages

What are the main advantages expected to be obtained from using this special form of decision table?

More manageable

In the first place, the tables themselves become very much more manageable. The various combinations of conditions can often amount to a large number of entries in the table. If these entries are limited to those giving rise to the derivation of one item only, then the table becomes easier to construct.

Output orientated

This 'one element per item' approach enables the design method to be entirely output orientated. This was one of the early objectives of Systematics—that the designer could work back step by step from the output wanted, and thus discover what rules and supporting information were required as he went along. For example, let us consider a possible approach to information systems design. In this approach an existing system is not presumed; the analyst sets out to design one. He identifies a basic output requirement, say an amount owing or 'invoice total'. He then asks is this item given to the system or derived by it? If given, he merely records the fact. If derived, he attempts to analyse its derivation using the element technique described. He then lists all the items referred to in this element and classifies them 'given' or 'derived'. Again the 'givens' are merely recorded, but the 'derived' items are each further explored in a similar manner. The designer thus develops a cascade until he reaches a level at which all items are given. See **Fig. 8.**

If an element were not confined to the derivation of one item, this approach would be inhibited. For example, considering level 3, it is possible that certain customer types affect the customer's credit limit as well as, or instead of, discount. But to explore credit limit at this time would be to divert attention from the scheme of thought being developed.

DERIVATIONS		ORIGINS Item	Given	Derived
Level 1	Invoice total	Item totals		√
Level 2	Item totals	Quantity	√	
		Price	√	
		Discount		√
Level 3	Discount	Customer type	√	
		Quantity ordered this year		√
Level 4	Quantity ordered this year	Quantity	√	
		Customer number	√	
		Date	√	

Fig. 8

Avoids processing sequence

It is important to distinguish between two sequence problems involved in computer systems. Firstly, there are those inherent in the derivation. For example, where

Deductions = Tax + Graduated Pension

it is essential to calculate Tax and Graduated Pension before calculating deductions. Secondly, there is the superimposed sequence enforced by the 'one job at a time' attitude of the computer; e.g. the sequence shown in **Fig. 9.**

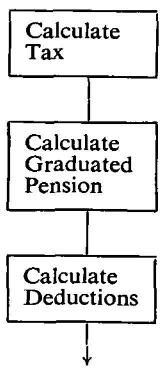


Fig. 9

It is to be noted that the sequence shown in **Fig. 10** is an equally valid solution to the problem. It may, for

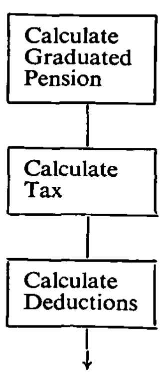


Fig. 10

some reason, be a less efficient solution in terms of computer processing, however. The strategy adopted within the computer system is of vital importance. But it is confusing to have to consider it at the same time as the information derivation statements are being designed and specified. It also makes such specifications unnecessarily complicated to read. Furthermore, it is premature to consider computer strategy until the requirements have been specified.

The removal of the 'go to' feature allows the analyst to avoid non essential statements of sequence. Sequence essential to the solution of the derivation problem is catered for in two ways:

1. The element is self triggering. That is, its primary conditions state when it should be performed.
2. Elements which should have been performed previously are named. See **Fig. 11** for an example.

Deductions

Employee's Number (given)	For each	(primary conditions)
Nil		(secondary conditions)
	Tax (derived) + Graduated Pension (derived)	

Fig. 11

Being self triggering the element does not automatically follow a previous element (processing sequence) but is done for each employee, i.e. each time employee's number changes. The elements for any derived items referred to, however—tax and graduated pension—must be performed first.

Localises attention

Perhaps the chief advantage when designing a system, and certainly when amending it later, will be the facility to localise attention to any degree defined by the analyst or the problem, and to ignore the rest of the system. Consider the elements which might be involved in the cascade in Fig. 8. These are shown in **Fig. 12.**

Any one of the four elements could be considered separately. This is because we know that the whole story of a particular derivation and also its place in the overall system is described in the one element. Taking element 3, we can be sure that no alternative method of deriving discount is described in another element. We also know that it is to be derived for each product ordered and that the 'quantity ordered this year' is to be derived first.

Similarity to nervous system

It is of interest to note that the form of decision table described bears some relationship to the neuron. The animal nervous system appears to rely upon signals traversing a network of neuron cells largely of the type shown in **Fig. 13.**

Invoice Total (1)

Order No.	For each
	Sum of item totals

Item Totals (2)

Product No.	For each
	Quantity (price — discount)

Discount (3)

Product No.	For each		
Customer Type	Home	Home	Overseas
Quantity ordered this year	>100	Else	—
	$\frac{1}{10}$ × Price	0	$\frac{1}{5}$ × Price

Quantity ordered this year (4)

Product No., Customer No., Year	For each
	Sum of quantity

Fig. 12

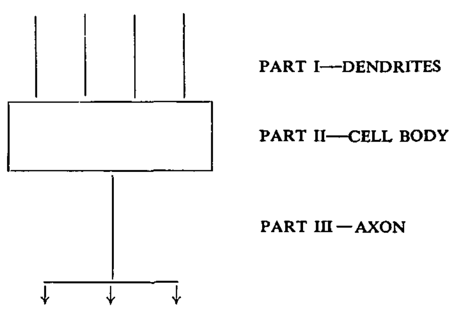
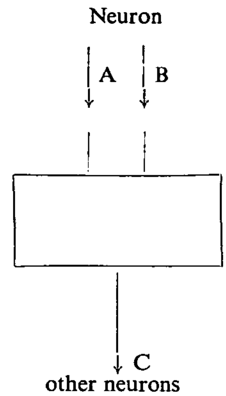


Fig. 13

Part I, the dendrites, consist of devices for interpreting signals from other neurons. Part 2, the body, generates the output signal. The strength* of the output signal appears to depend upon the interpretation, by the dendrites, of the strengths of the input signals and upon the physical properties of the particular neuron body. Part III, the axon, carries the output signal to the input areas (dendrites) of other neurons. For a readable further description of the nervous system, see Wooldridge (1963).

The element is the basic building brick within Systematics. The information model designed consists of a number of these elements. Their similarity to nature's basic building bricks is illustrated in the simple example shown in **Fig. 14.**

* Signals within the nervous system are electrical. Their strength varies according to frequency of pulse rather than voltage of each pulse. This distinction is ignored here.



Element

A	*p*	*q*	*q*
B	√	√	*x*
C	7	10	4

Use: other elements

Fig. 14

The neuron illustrated receives signals from the axons of other neurons A and B. Various combinations of strengths of these signals are interpreted by the dendrites; these interpretations stimulate the generation of signals of various strengths in the body of the cell which, in turn, form the output C. C is distributed by the axon as a stimulant to other neurons.

The element shown similarly receives signals from the output of other elements A and B. Various combinations of strengths of these signals are explored. The strength of the output C varies according to these combinations; e.g. If A = *p*, and B = √, then C = 7, etc. The value of C may then stimulate action in other elements.

Syntactic description

A syntactic description of the Systematics language is given in **Fig. 15.** This description is largely in Backus–Naur form. A difficulty arises, however, since Backus–Naur form provides a linear description, whereas Systematics takes essentially a two dimensional or tabular form. Rather than attempt to produce an exact description therefore, those separators and terminators which are implied by the tabular form have been left out.

It will be noted that a GIVEN item may have many SUBMISSIONS. For example:

⟨NAME⟩ = ⟨SUBMISSIONS⟩
Quantity
Ordered = 7, 14, 83, 6,

being the different quantities ordered for different products or by different customers. The question arises as to which submission is relevant to a particular calculation. The rule is:

TERM, the latest submission at the time the element is triggered.
SERIES, all submissions since the element was last triggered.

e.g. Fig. 12, element 2 quantity is the amount submitted for a particular value of product number.

```
⟨MODEL⟩ ::=                 ⟨GIVENS⟩, ⟨DERIVEDS⟩
  ⟨GIVENS⟩ ::=              ⟨GIVEN⟩ | ⟨GIVENS⟩, ⟨GIVEN⟩
  ⟨DERIVEDS⟩ ::=            ⟨DERIVED⟩ | ⟨DERIVEDS⟩, ⟨DERIVED⟩
    ⟨GIVEN⟩ ::=             ⟨NAME⟩ = ⟨SUBMISSIONS⟩
      ⟨SUBMISSIONS⟩ ::=     ⟨SUBMISSION⟩ | ⟨SUBMISSIONS⟩, ⟨SUBMISSION⟩
        ⟨SUBMISSION⟩ ::=    ⟨IDENTIFIER⟩ | ⟨NUMBER⟩|⟨LOGIC⟩
```

Where: IDENTIFIER is an alphanumeric string which attempts to identify uniquely a member of a class, e.g. a customer's number.

and: NUMBER is a number subject to the laws of arithmetic (in any specified radix), e.g. the value of age or gross pay.

and: LOGIC is an alphanumeric string identifying an attribute of class members, e.g. a member's sex or tax code.

and: NAME is an alphanumeric string uniquely identifying each GIVEN and each DERIVED item, e.g. Holiday entitlement.

```
    ⟨DERIVED⟩ ::=           ⟨NAME⟩ = ⟨ELEMENT⟩
      ⟨ELEMENT⟩ ::=         ⟨TRIGGERS⟩ ⟨DERIVATION⟩
                           |⟨TRIGGERS⟩ ⟨ALT DERIVATIONS⟩
                           |⟨TRIGGERS⟩ ⟨LTD DERIVATION⟩

      ⟨DERIVATION⟩ ::=   ⟨TERM⟩ | ⟨SERIES⟩ | ⟨DERIVATION⟩ ⟨OPERATOR⟩ ⟨DERIVATION⟩
        ⟨TERM⟩ ::=       ⟨NAME⟩ | ⟨LITERAL⟩
Where: LITERAL           is a number or an alphanumeric string.
          ⟨SERIES⟩ ::=   ⟨FUNCTION⟩ | ⟨NAME⟩
            ⟨FUNCTION⟩ ::= S | MAX | MIN | AV | LIST | F
          ⟨OPERATOR⟩ ::= + | − | × | /
      ⟨TRIGGERS⟩ ::=  ⟨TRIGGER⟩ | ⟨TRIGGERS⟩, ⟨TRIGGER⟩
        ⟨TRIGGER⟩ ::=  ⟨DERIVATIONS⟩ ⟨EVENT⟩
          ⟨DERIVATIONS⟩ ::=  ⟨DERIVATION⟩ | ⟨DERIVATIONS⟩, ⟨DERIVATION⟩
          ⟨EVENT⟩ ::=  ⟨STATES⟩ | FOR EACH
            ⟨STATES⟩ ::= ⟨STATE⟩ | ⟨STATES⟩, ⟨STATE⟩
              ⟨STATE⟩ ::=  ⟨RELATIONAL OPERATOR⟩ ⟨DERIVATION⟩
                ⟨RELATIONAL OPERATOR⟩ ::= > | < | ≤ | EX | =

      ⟨ALT DERIVATIONS⟩ ::=  ⟨ALT DERIVATION⟩ ⟨ALT DERIVATION⟩
                            |⟨ALT DERIVATIONS⟩, ⟨ALT DERIVATION⟩
        ⟨ALT DERIVATION⟩ ::=  ⟨CONDITIONS⟩ ⟨DERIVATION⟩
          ⟨CONDITIONS⟩ ::=  ⟨CONDITION⟩ | ⟨CONDITIONS⟩, ⟨CONDITION⟩
            ⟨CONDITION⟩ ::=  ⟨DERIVATION⟩ ⟨VALUE⟩
              ⟨VALUE⟩ ::=  ⟨STATES⟩ |—| ELSE
      ⟨LTD DERIVATION⟩ ::=  ⟨CONDITIONS⟩ ⟨SERIES⟩ | ⟨CONDITIONS⟩
                            ⟨DERIVATION⟩ ⟨OPERATOR⟩ ⟨SERIES⟩
```

Fig. 15

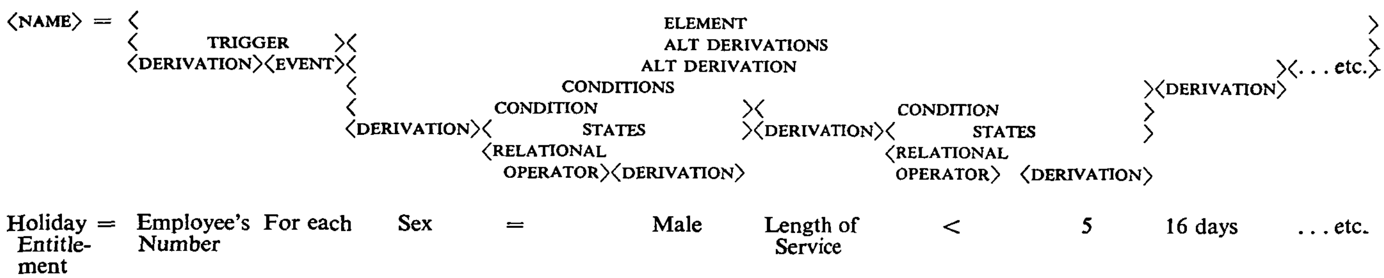


Fig. 16

Fig. 12, element 4 quantities are all those submitted for a particular product, customer and year.

The element is a statement in two dimensional form. Taking the example in Fig. 7, the element fits into the linear Backus–Naur description as shown in **Fig. 15.**

Notes

(1) Special Symbols:
under ⟨FUNCTION⟩ S = Sum
F = Frequency
under ⟨RELATIONAL OPERATOR⟩ Ex = Except
under ⟨VALUE⟩ — = Not applicable

(2) The equals sign is assumed in the tabular form.

(3) Where more than one TERM or SERIES is present under primary or secondary conditions OR logic applies to a horizontal list and AND logic to a vertical list, e.g.

Customer No., Year	For each

means for each change in Customer No. OR in Year and

Sex	Male ....
Length of Service	<5 years ....

means where Sex is Male AND where Length of Service is <5 years. The order in which the TERMS or SERIES are written is arbitrary and of no consequence.

References

GRINDLEY, C. B. B. (1966). Systematics—a non-programming language for designing and specifying commercial systems for computers, *Computer Journal*, Vol. 9, p. 124.
KING, P. J. H. (1967a). Some Comments on Systematics, *Computer Journal*, Vol. 10, p. 116.
KING, P. J. H. (1967b). Decision tables, *Computer Journal*, Vol. 10, p. 135.
WOOLDRIDGE, D. E. (1963). *The Machinery of the Brain*, McGraw-Hill Book Company.

Book Review

Prediction Analysis, by JOHN R. WOLBERG, 1967; 291 pages. (London: *D. van Nostrand Co. Ltd.*, 86s.)

In case others are as puzzled as I was by the title of this book, let me explain what Prediction Analysis is about. A variable y is related to other variables $x_1, x_2, \dots x_m$ by an equation $y = f(x_1, x_2, \dots x_m; \alpha_1, \alpha_2, \dots \alpha_p)$ where the function f is known but the parameters α_i are not. Observations subject to errors of known variance are made on y and on the associated xs. Using the method of least squares it is possible to obtain estimates α_i of the parameters, together with their standard errors. Prediction analysis is concerned with how many and what type of observations should be made in order to achieve prescribed precision in determining the parameters. The author claims that most experiments are of this form and that prediction analysis is the basis of the planning of experiments.

The book has a short chapter on the statistical background, a longer chapter on least squares and a similar chapter on the general theory of prediction analysis. The remainder of the book consists of five chapters dealing with special cases of f (polynomial, exponential, sine, gaussian) with a single x ($m = 1$) and one case with three independent variables.

The statistical material is badly presented with inadequate definitions, a series of incorrect statements about an unbiased estimate of a standard deviation and complete confusion between 'independent' and 'uncorrelated'. The least squares material appears to miss the reason for weighting the squares inversely proportional to the variances. The derivation of the general least squares solution (§3.6) is incomprehensible to me, as is the subsequent result for the standard errors. It is hard to see whether the final result is correct since the resulting computational procedure is iterative and it may well be that the author's method, although different from the usual Newton–Raphson approach, still converges to the least squares values. This appears to need more discussion, using modern numerical analysis, than the author provides. To satisfy myself that the author's method is not wildly wrong I did succeed in obtaining his results by a different method. Anyone who similarly finds himself in difficulties is welcome to write for my notes—which may well be equally obscure, though not, I think, to a statistician. There is considerable discussion of the computational problems, including flow diagrams.

The method of prediction analysis is important and it is a pity that a better description of it has not been provided. It is puzzling to see no mention of preposterior analysis, which has exactly the same ends.

D. V. LINDLEY (London)