IBM Data Processing Division
Thomas J. Watson Research Center
P. O. Box 218
Yorktown Heights, New York

June 23, 1961

Memorandum to:

Subject: Tabular Techniques Development
Distribution #3

This is the third release of material concerning the development of tabular techniques for systems and programming description. Enclosed are three reports:

(1) A report by Sutherland Company describing a method of recording management decision rules and other information necessary to adapt an information system to an automatic medium of data processing.

(2) A report by Burton Grad, IBM, describing two techniques of representing the decision logic of an insurance company file maintenance problem; namely, traditional flow charts and tabular form.

(3) A paper given by Burton Grad at the 12th GUIDE International meeting in Montreal, Canada, June 1, 1961 describing the general concept of tabular techniques.

Burton Grad th.

SYSTEMS
ENGINEERING
SERVICES

CLEARINGHOUSE REPORT

INFORMATION PROCESSING SYSTEM
ANALYSIS

June 5, 1961
Ref. No. 1F2

Sutherland Co.

INFORMATION PROCESSING SYSTEM ANALYSIS INSTRUCTIONS

1. **Purpose.** To provide a standard method of recording the management rules (arithmetic and decision processes) and other information necessary to adapt an Information System to a mechanical or other medium of processing.

2. **General.** The method described in the following instructions eliminates the need for lengthy narrative with its inherent disadvantages of misinterpretation by the reader and difficulty of organization by the writer. This method also eliminates the need for the system analyst to prepare detailed flow charts to convey to a processing specialist the processing required to obtain the desired results of the Information Processing System. The method of documentation is general enough to allow the Information System to be adapted to any medium of processing, but detailed enough to permit the application of the Information System to electronic machine processing by a machine specialist who has no prior knowledge of the Information System.

A. **Documentation Preparation.** The documentation will be prepared by the system analyst and forwarded to the processing specialist. The processing specialist may be a punched card equipment specialist, an electronic equipment processing specialist or a manual and standard office equipment processing specialist. In many instances, the manual and standard office equipment processing specialist will be the system analyst.

B. **Content of Documentation.** The documentation prepared by the system analyst will include the following:

(1) General System Chart including the inputs to the system and the sources of the inputs, the outputs of the system and the disposition of the outputs and the data to be retained by the system.

(2) A general narrative description of the Information System which will include the purpose and scope of the Information System and any other pertinent information that may be helpful to the Processing Specialist.

(3) **Description Sheets**

a. Input Description
b. Process Description and Process Description Continuation
c. Output Description

3. Input Description Sheet.

A. General. An Input Description Sheet is used to describe the content of Action Sets and Retained Data Sets which are input to the information system.

B. Headings.

(1) In the upper left-hand corner, place the two-character "System Identification" for the system being described.

(2) Below the "System Identification", place the "Set Identification" for the Input Set being described. If the input is an Action Set, use the identification of the Action Set. If the input is a Retained Data Set, use the unique Retained Data Set identification assigned to the set.

The first two characters of the Retained Data Set identification are the System Code, the next two characters will be "RD". The next character (s) is used to identify uniquely each Retained Data Set. For example:

04RD1
Billing — 04 ... 1 — Assigned to the Customer File

04RD2
Billing — 04 ... 2 — Assigned to the Price List

(3) Indicate in the space provided for "Frequency of Processing" the most frequent period in which this set is to be input to the system.

(4) Process. Indicate in the space provided the name of the process being documented. In most instances the process will directly correspond to what is described by the System Identification. Occasionally the System Identification is not definitive of the process being documented and the actual process name should be indicated. For example:

System Code 20 is assigned to Salary Payroll which includes: Pay Check Preparation, Personnel Reports, Labor Distribution, Tax Reports and Annuity Reports. In this example, the System Code would be 20, but the process would be Pay Check Preparation, Labor Distribution, etc. depending on

(7) In the space provided indicate the Form Type for the set. Examples of "Form Type" are: "Manual", "Punched Card", "Magnetic Tape", and "Paper Tape".

(8) For "Source System I.D." indicate the two-character System Code of the System which processes the set immediately prior to this system. If the Input Set is a Retained Data Set which is added to in more than one system, indicate the system from which the Retained Data Set will be received.

(9) In the upper right-hand corner indicate the page number, the name of the person preparing the Input Description Sheet, and the date of preparation.

C. Management Rule Numbers. For Action Sets AND RETAINED DATA SETS indicate in the spaces provided across the top of the sheet the three-digit numbers of the Management Rules (other than Validation Rules) which must be executed if this set is present. If there is not sufficient space on one Input Description Sheet for all the rules, use additional sheets.

D. Element Name.

(1) In this column enter the "Element Names" assigned to the elements that are contained in the Input Set. For an input, regardless of whether or not space is provided for an element, no entry should be made for the element, if it is always blank.

(2) Additional information on each element is placed to the right of the element name.

E. Element Code. In this column place the seven-character element code number corresponding to each element name.

F. Element Code - Suffix.

(1) An element in a set may be used differently or prepared differently depending on what other elements identify it. An example is the Element "Quantity on Hand Total". This element may appear twice on a set. In one place, it may be the total "Quantities on Hand" for each "Stock Number" at each "Location". In the other place, it may be the total of all "Quantities on Hand" for each "Stock Number" at all "Locations". In the first instance,

H. Element Description - Numeric. If the element described by the element name contains any numeric characters, enter an "N" in this column. Otherwise, leave the column blank.

I. Element Description - Characters - Total. Place in this column a maximum of two digits to describe the maximum number of characters that the element may contain. Do not include in the total number of characters, punctuation marks in numeric fields which are used for arithmetic processes.

J. Element Description - Characters - Decimal. This entry is made only for all numeric elements which may be used in arithmetic computation. Enter in this column the number of digits that appear to the right of the implied decimal point.

K. Element Classification (Class.). Depending on whether the element described by the element name is a Recognition, Identification, Action, Action Modifier, Information, or Superfluous Element, enter an "X" in the appropriate column. See the definitions for the different Element Classifications in Appendix I. Generally, the different classifications are mutually exclusive. However, any element may be described by more than one classification other than "Information" and "Superfluous". For retained Data Sets only Recognition and Identifying Elements need be indicated.

L. Number of Times an Entry May Appear on This Set. Place in this column a maximum of three characters to indicate the "Average" and a maximum of three characters to indicate the "Peak" number of times an entry may appear for this element on this set. If the number exceeds 999, use "C" for hundreds and "M" for thousands.

M. Validation Rule (s). In this column list the three-digit Rule Numbers for the Management Rules which must be executed to validate the element described by the element name. Use as many lines as are necessary for each element name.

N. Identifying Element Codes.

(1) For Identifying Elements that are used to identify an element on the Input Set, the Identifying Element Codes are listed vertically in the spaces provided. If more space is required, use additional Input Description Sheets.

(3) The first two lines of Figure I illustrates the example described in paragraph 3, F, preceding. Quantity on Hand with Suffix "A" is for each Stock Number at each location. Consequently an "X" appears under both 9300100 and 7976050, the Element Codes for Stock Number and Location respectively. Quantity on Hand with Suffix "B" is for each stock number at all locations. An "X" only appears under 9300100, the Element Code for Stock Number. In this case, Stock Number alone is the Identifying Element for Quantity on Hand. The third line of Figure 1 indicates that the entry (s) for location is identified by an entry for Stock Number.

System Identification ______ Process ______

Set Identification ______ Set Name ______

Frequency of Processing ______ Volume: Average

Form

Source System I. D.

MANAGEMENT RULE NOS.

ELEMENT NAME	ELEMENT CODE	ELEMENT DESCRIPTION Suffix	Alpha	Numeric	Char. Total	IDENTIFYING ELEMENT CO 9300100	7976050
QTY ON HAND TOT	8768150	A				X	X
	8768150	B				X	
LOCATION	7976050					X	
STOCK NO	9300100						

O. Reference Note (Ref. Note). If there is a need for a reference note, place a check mark (✓) in the column. Cross-reference the note with the System Identification, Process, Set Identification, and if required the Element Code and Suffix.

P. Remarks. This column may be used for any additional information believed necessary by the analyst preparing the Input Description Sheet.

4. Process Description Sheet.

A. General.

(1) A Process Description Sheet is used to describe Management Rules used in processing information within a system.

(2) Rules for Validation are shown on separate sheets from all other processing rules. It is assumed by the analyst that all Validation processing is to be accomplished before other processing is begun.

B. Headings.

(1) In the upper left-hand corner place the two digit "System Identification" for the Analysis System.

(2) In the space provided for "Process", indicate the name assigned to the process being described. (Refer to paragraph 3, B, (4) preceding).

(3) If the processes described by the Management Rules on the sheet are Validation Processes, place an "X" in the "Validation" Box.

(4) In the right-hand part of the heading, enter in the spaces provided: the page number, the name of the person preparing the sheet, and the date of preparation.

C. Line Number. On each line in this column, place a four-character line number. It is suggested that the right-most digit always be blank in case there is a later need for insertion of additional lines. Line numbers will be uniquely assigned to all lines within the description of a particular ~~process for a system.~~ MANAGEMENT RULE. ~~Thus, if the last line on Page 1 for a process is line number 022, the first line~~

D. Condition/Action Indicator (C/A).

(1) If a condition is expressed on this line, place a "C" in this column; if the line is used to express an action, place an "A" in this column. If what has been placed in this column for an immediately previous line is true for a line that follows, no entry need be made for the line that follows.

E. Management Rule - Current. In this column on the first line for each Management Rule place a three-digit number for the Management Rule. The numbers of all Management Rules will be uniquely assigned for all rules within a Process for a System.

F. Management Rule - Prior. In this column list the three-digit numbers of all of the Management Rules which must be considered before the rule specified in the "Management Rule - Current" column is considered. Generally, a rule is prior to another rule only if it specifies the creation of elements of data necessary for the processing of the current rule. Management Rules for Validation of elements will not be shown as prior rules for non-validation Management Rules.

G. Source - Element Name, Prior Result or Actual Value.

(1) If one source for a condition or action is an element, place the name assigned to the element in this column. If the source is an actual value (Literal or Descriptive constants - See Appendix I) place the actual value in this column. If the source is the result of an action in any rule, place the designation of the result in this column. (Results of an action are designated as "Result X", where "X" is any character A to Z or 0 - 9. The first result of a rule is designated as "Result A", the second as "Result B", etc. Unique designations of prior results are only necessary within each rule. Two different rules may each have an intermediate result designated as Result A.

(2) Deletion of an Element. The deletion of an element from a set is indicated by placing the Descriptive Literal "/BLANK/" in this column, entering a check mark in the "Set Equal To" column, and entering the Element Name and Set Identification for the element to be deleted in the appropriate spaces in the "Source/Disposition" column.

(3) Deletion of a Set. The deletion of a set is indicated by placing

which was assigned is entered in this column. Otherwise, the column is left blank.

I. Source - Set Identification.

(1) If the entry made in the "Source - Element Name, Prior Result or Actual Value" column was an Element Name, enter in this column the seven-character set designation for the set of which the element is a part. If an element for a rule may appear in one Input Set or another, depending on which set is present, more than one Set Identification may be listed in this column as a source for the element described by the Element Name. If the entry in the "Element Name" column is the designation of a Result of an action in this rule or another rule, enter the three-digit number for the source rule within parentheses. This column is left blank if the entry made in the "Source" column is an entry for an actual value. Examples of entries that may be made in this column are:

24165A = Set
(152) = Management Rule
152 = Set

(2) The addition or insertion of a set into an Output Set or Retained Data Set may be indicated by placing the Set Identification of the set to be added or inserted in the "Source Set Identification" column, the Set Identification of the Output Set or Retained Data Set in the "Source/Disposition Set Identification" column and a check mark in the "Set Equal To" column. The element columns of both the Source and Source Disposition will be left blank. This procedure will only be used if all the Elements of the Output Set or Retained Data Set are contained in the Input Set.

J. Condition (Cond.). If a condition is expressed on a line, it is "Greater Than", "Less Than", or "Equal To". Place a check mark (✓) in the appropriate column (s) to indicate the relationship between the first and the second Source Elements or Actual Values. The relationship between the three conditions is a logical "or" condition. More than one column may be checked for a line. In reading, "or" is inserted between each condition checked.

For example, if "AMT SALARY" is the first Source Element, (O) is symbols in the column:

+	for addition
-	for subtraction
x	for multiplication
/	for division
Σ	for sum

(2) Explanation of Operation Symbols.

a. An entry of "+" in this column indicates that the first source entry is to be added to the second source entry.

b. An entry of "-" in this column indicates that the second source entry is to be subtracted from the first source entry.

c. An entry of "x" in this column indicates that the second source entry is to be multiplied by the first.

d. An entry of "/" in this column indicates that the first source entry is to be divided by the second.

e. An entry of "Σ" (Greek letter "Sigma") in this column indicates that all entries for the first specified element are to be summed.

L. Set Equal To. If the element or result specified in the "Source/Disposition" column is to be "Set Equal To" another element, actual value or prior result, or is to be "Set Equal To" the result of an arithmetic action, place a check mark in this column. The last line of any action within a rule will have a check mark in the "Set Equal To" column.

M. Source/Disposition - Element - Name Result, Prior Result or Actual Value.

(1) If the entry to be made in this column is for a source for a condition or an action, the way to make the entry is described in paragraph 4, G, (1).

(2) If this column is used to indicate disposition for a result of an action, enter the appropriate element name or prior result designation. (See

O. Source/Disposition - Set Identification. If the entry made for the "Source/Disposition" column is an entry for a Source, see paragraph I. If the entry is a Disposition entry for an element, enter in the "Set Identification" column the Set Identification for the set or sets in which the Element is to be placed. If the entry is a Disposition entry for an intermediate Result, leave the "Set Identification" column blank.

P. Operation.

(1) To relate arithmetically an entry in the "Source/Disposition" column on one line with an entry in the "Source" column on the next line, indicate the arithmetic operation in this "Operation" column using one of the following symbols:

+ for addition
- for subtraction
/ for division
x for multiplication

(2) Explanation of Operation Symbols.

a. An entry of "+" in this column indicates that the "Source" entry on the next line is to be added to the "Source/Disposition" entry on the line where the "+" appears.

b. An entry of "-" in this column indicates that the "Source" entry on the next line is to be subtracted from the "Source/Disposition" entry on the line where the "-" appears.

c. An entry of "/" in this column indicates that the "Source/Disposition" entry on the same line is to be divided by the "Source" entry on the next line.

d. An entry of "x" in this column indicates that the "Source/Disposition" entry on the same line is to be multiplied by the "Source" entry on the next line.

Q. Note Reference (Note Ref. (✓)). If a note or remarks are necessary and/or advisable to explain further a condition or an action, place a check mark suffixes are necessary for a rule, Process Description Continuation Sheets should be used.

(2) In describing a Management Rule, all the conditions which must be considered at any one time will be listed on a Process Description Sheet. Following the conditions, all the actions which may be executed for the conditions of the rule will be listed on the same Process Description Sheets (insofar as possible). Management Rule Suffixes are used to relate a combination of positive and/or negative results for one or more conditions to the execution of one or more actions within a rule.

(3) Unless a Management Rule describes an unconditional action (action taken regardless of the results of any conditions), an action is taken only when the results of certain conditions are positive ("Y") and/or negative ("N"). In describing a Management Rule, all the pertinent possible combinations of condition results must be related to the actions for the rule.

(4) A simple example is shown in Figure 2. In this sample Management Rule there are only three conditions shown on lines 001 to 003. One set of results for the conditions are listed under Suffix A; i.e., if the result of the conditions on lines 001 and 002 are positive the action specified on line 004 should be taken. Under Suffix C, if the results of the conditions on lines 001 and 003 are positive and the result of that on line 002 is negative, the action specified on line 004 should be taken.

PROCESS DESCRIPTION

Page ______ Of ______

Prepared by ____________________

Date ____________________

☐ Validation

SOURCE: ...NT — ...r Result / ...alue	SUF.	Set Ident. or Rule No. for Prior Result	COND.: Greater Than	Less Than	Equal To	Oper (+ - / X Σ)	Set = To (V)	SOURCE/DISPOSITION: ELEMENT — Name, Result, Prior Result or Actual Value	SUF.	Set Ident. or Rule No. for Prior Result	Oper (+ - / X)	Note Ref. (V)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
HAND		67491	✓		✓			QUAN ORDERED		6089			Y	N	Y	Y	N													
RATING		40A			✓			(A)					Y	Y	N	N	N													
...DERED		6089		✓	✓			CUSTOMER MAX		40A					Y	N														
													8	0	1	0	0													
													0	5	0	4	1													
...DERED		6089					✓	QUAN SHIPPED		409257			X		X															
HAND		67491					✓							X																
							✓									X	X													

Management Rule Suffix and Frequency — Cond: Y = Condition Is Satisfied; N = Condition Is Not Satisfied; Blank = Condition Not Applicable. Action: X = Action to Be Taken; Blank = Action Not to Be Taken.

...2. EXAMPLE OF USE OF MANAGEMENT RULE SUFFIX AND FREQUENCY

(5) As is evident from the example, pertinent results for conditions are indicated for a suffix using "Y" for "Yes" and "N" for "No". Under each suffix an indication of an action to be taken is shown with an "X" on the line, ("Set Equal" line if more than one) on which the action is described. If neither "Y" nor "N" is placed on the line for a Condition under a given Suffix, it indicates that for the combination of results shown under the suffix, the result of this condition is immaterial; the result can be positive, negative, or undetermined.

(6) For a Management Rule, on the line (s) following the last line describing the conditions, the analyst will indicate the probable Frequency of Occurrence as a percentage for the results of the conditions listed under each suffix. The total Frequencies of Occurrence for all suffixes within a Rule should be 100 percent. For any frequencies less than 1%, use "1". In Figure 2 the Frequency of Occurrence is indicated between lines 003 and 004. In this example the probability of the conditions of rule 001A prevailing is 80%, written "$\substack{8\\0}$". For rule 001D, the probability of occurrence is 4%, written "$\substack{0\\4}$".

5. <u>Process Description Continuation Sheet.</u>

A. <u>General.</u> A Process Description Continuation Sheet is used only if, for a Management Rule, there were insufficient suffixes on the Process Description Sheet to depict all the combinations of results for the conditions described on it.

B. <u>Headings.</u> The instructions for completing the heading information are the same as shown for the Process Description Sheet, paragraph 4, B, preceding.

C. <u>Line Number</u>. In the line number column post the line numbers from the Process Description Sheet that this sheet is a continuation of. Use exactly the same spacing and relative positioning of the line numbers as appears on the Process Description Sheet. This will enable the user to lay a completed Continuation Sheet next to the sheet it is a continuation of to have effectively a single sheet of paper.

D. <u>Management Rule Suffix and Frequency.</u>

(1) In the blank heading blocks, place one-or two-character suffix

6. Output Description Sheet.

A. General. An Output Description Sheet is used to describe the content of output from an Information System.

B. Headings.

(1) Enter the "System Identification" for the system being described in the space provided.

(2) Enter the "Set Identification" for the Output Set in the space provided.

(3) Indicate in the space provided the name of the process being documented. (Refer to paragraph 3, B, (4) preceding).

(4) In the space provided for "Number Copies" indicate the number of copies that are required for this Output Set.

(5) Place the "Set Name" in the space provided.

(6) Indicate in the space provided for "Volume" the "Average" and "Peak" number of sets that will be prepared.

(7) Form Type. Indicate the Form Type for the set. For example, Standard Print, Punched Card, Multilith Mat, etc.

(8) Special Form I. D. If the set is to be prepared on a special form, indicate the identification of the special form in the space provided.

(9) In the upper right-hand corner enter the Page number, the name of the person preparing the sheet, and the date of preparation.

C. Element Name.

(1) In this column enter the Element Name for each of the elements which may appear in this set.

(2) Additional information on each element is placed to the right of

E. Element Code - Suffix.

(1) If an Element Code Suffix is required (See paragraph 3, F, Input Description Sheet), enter a one-character alphabetic designation for the suffix in this column.

F. Element Description - Alpha.

(1) If the Element described by the Element Name contains any non-numeric characters, enter an "A" in this column. Otherwise, leave the column blank.

G. Element Description - Numeric.

(1) If the Element described by the Element Name contains any numeric characters, enter an "N" in this column. Otherwise, leave the column blank.

H. Characters - Total.

(1) Enter in this column a maximum of two digits to describe the maximum number of characters that the Element may contain.

I. Characters - Decimal.

(1) An entry is made in this column only for all numeric Elements which are a result of or may be used in arithmetic computations. Enter in this column the number of digits that should appear to the right of the implied decimal point.

J. Element Classification.

(1) Depending on whether the Element described by the Element Name is a "Recognition", "Identification", or "Other" classification of Element, enter an "X" in the appropriate column.

K. Number of Times an Entry May Appear on This Set.

(1) Enter in this column a maximum number of three characters to

L. Source - Set Type.

(1) If the Source for the element described by the Element Name is other than "Direct Recording" from an Action Set or Retained Data Set, place an "X" in the column headed "Process (X)".

(2) If the element described by the Element Name is to be placed on the Output Set as a result of a Direct Recording from a Retained Data Set after all posting to the Retained Data Set has been accomplished, enter an "A" in the column headed "Ret'd (A, B, or X)".

(3) If the element described by the Element Name is to be placed on the Output Set as a result of a Direct Recording from a Retained Data Set before any posting to the Retained Data Set has been accomplished, enter a "B" in the column with the heading "Ret'd (A, B, or X)".

(4) If the element described by the Element Name may be placed on the Output Set as a result of a Direct Recording from a Retained Data Set, either before or after posting to the Retained Data Set has been accomplished, enter an "X" in the column headed "Ret'd (A, B, or X)".

(5) If the element described by the Element Name is placed on the Output Set as a result of a Direct Recording from an Action Set, enter an "X" in the column headed "Action (X)".

M. Source - Source Set Identification for Direct Recording.

(1) If the element described by the Element Name is to be placed on the Output Set as a result of Direct Recording from a Retained Data Set or an Action Set, enter in this column a maximum of seven characters for the Set Identification of each source set. If there are more than three sources, use additional lines.

N. Identifying Element Codes.

(1) For Identifying Elements that are used to identify Elements on the Output Set, the Identifying Element Codes are listed vertically in the spaces provided. If more space is required, use additional Output Description sheets.

APPENDIX I - DEFINITIONS

1. Action Element

An element within an Action Set, the entry for which is the value to be inserted or replaced, or the value of the adjustment to be made via a Recording Action or Actions or arithmetic computation.

2. Action Modifier Element

An element within an Action Set which alters the Recording Action or Actions in some manner.

3. Action Set

An Input Set for a system whose presence may require the execution of specific Management Rules. Input other than Retained Data Set.

4. Constant Value

A value, which does not appear as an element in either a Retained Data or Action Set, used as a source for an element or elements in an Output Set.

A. Descriptive Constant

An entry which designates between two slashes (/) the commonly understood name of a constant value.

Examples are :

/ Blank /	- Designates one or more blanks.
/ Current Year /	- Designates 1962, if that is the current year.
/ ANNN /	- Designates a field in which the first character is non-numeric and the rest are numeric.

B. Literal Constant

5. Direct Recording

The unconditional transferring of an element from an Action Set or Retained Data Set to an Output Set. No prior processing other than validation is required for the element in the Action Set or Retained Data Set. The recording is dependent on the presence of the Action Set or Retained Data Set and the requirement to produce the Output Set.

6. Frequency of Occurrence

A number which indicates, as a percentage, the probability a particular result, or combination of results of a condition or conditions, will prevail.

7. Identification Element

An element within an Action Set which permits the segregation of a particular set from others containing the same Recognition Element values; it is used to associate the set with other sets containing different Recognition Element values and to indicate how elements within the set are recorded and identified.

8. Information Element

An element within an Action Set, which does not influence the Recording Action nor is it recorded in this system. It may be subject to validation for the purpose of an overall system check and is required for processing in subsequent systems.

9. Management Rule

The action or actions and generally an associated condition or conditions which indicate the decisions and processes required to operate an Information Processing System.

10. Output Set

A set created by an Information Processing System for the use of another Information Processing System or by the same Information Processing System, but using a different medium to accomplish its processes.

12. Recognition Element

An element within an Action Set which identifies the function of the set. The Set Identification is a Recognition Element unless otherwise stated.

13. Retained Data Set

A set which is used to maintain elements which are required to accomplish the preparation of the Output Sets and may not be available on the Action Sets. The Retained Data Set will include the elements required to validate the Action Sets.

14. Set

A meaningful grouping of more than one element of data.

15. Superfluous Element

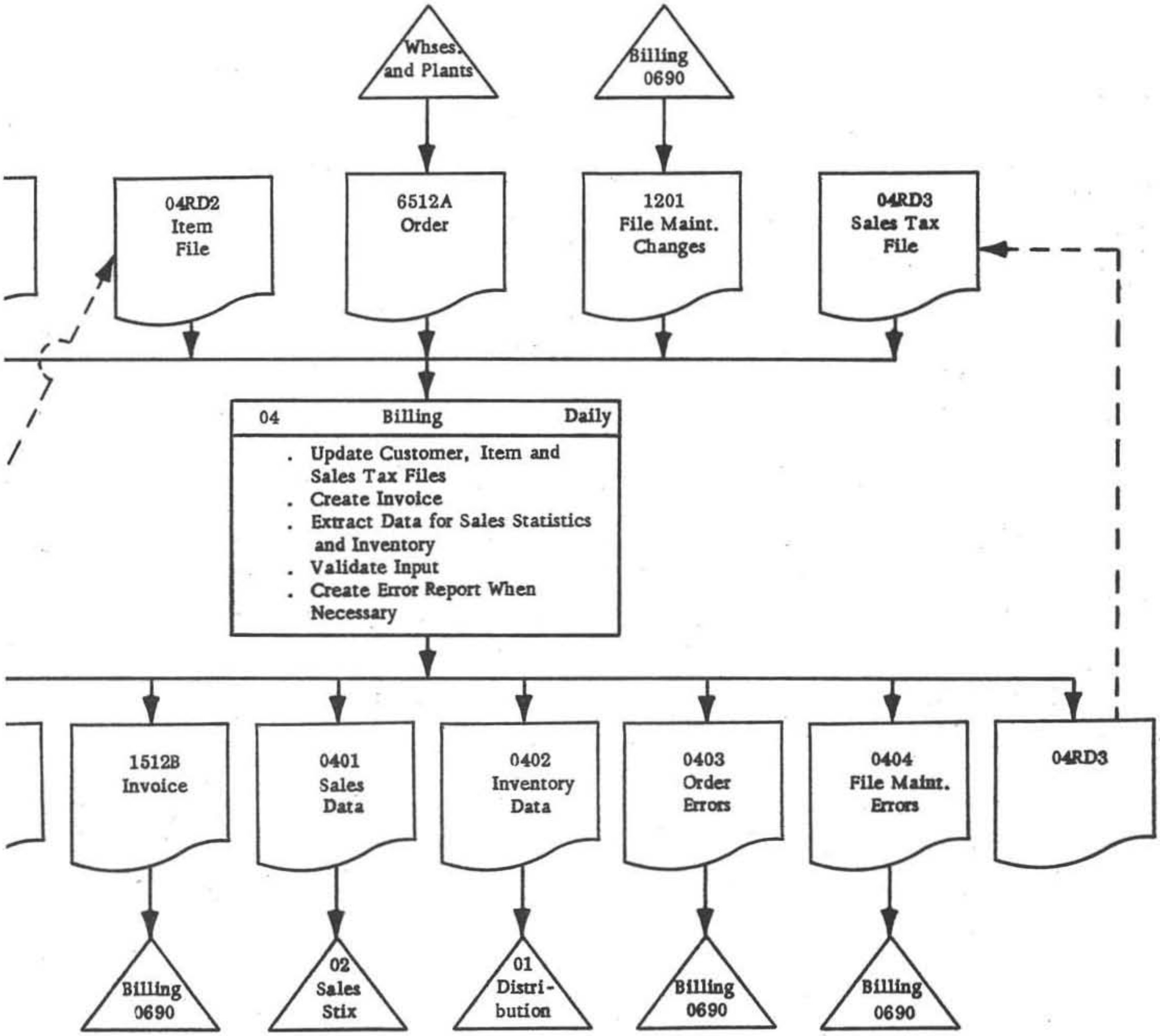
An element within an Action Set which is not required for processing in this or subsequent systems.

SYSTEM CODE 04 BILLING

1. Purpose: To develop an invoice from a copy of the order which indicates that shipment has been made to a customer from a warehouse or factory.

2. Scope: The system will include all debit billing to all customers.

3. Other Outputs:
   a. Selected data will be furnished to the Sales Statistics system for Sales Accounting and Sales History.
   b. Selected data relative to inventory will be furnished to the Distribution system for inventory adjustments.
   c. A record of input received that did not meet the criteria established (invalid) will be furnished to the Billing Department.


Whses. and Plants
Billing 0690
04RD2 Item File
6512A Order
1201 File Maint. Changes
04RD3 Sales Tax File
04 Billing Daily
. Update Customer, Item and Sales Tax Files
. Create Invoice
. Extract Data for Sales Statistics and Inventory
. Validate Input
. Create Error Report When Necessary
1512B Invoice
0401 Sales Data
0402 Inventory Data
0403 Order Errors
0404 File Maint. Errors
04RD3
Billing 0690
02 Sales Stix
01 Distri-bution
Billing 0690
Billing 0690

stomer Code — 6813250

Disc. % — 6920130

Change Code — 8760100

Sold To

6813200

s Sold To

6813210

omer Code, Name Sold To or Address Sold To changes, create
"Delete" for Old Customer Code
"Add" for New Customer Code, Disc. %, Name Sold To and Address Sold To
e Codes: 1 = Delete 2 = Add 3 = Change Disc. %

CUSTOMER CHANGE ERRORS

Current
Month Day Year
xx - xx - xx

6813250 — xxxx bb

8760100 → X

bb

7000100 — xxxxxxxxx bb

6920130 — xx.x bb

6813200
xx ———————— xx
xx ———————— xx
6813210

Page Number xxxxx

Process BILLING **INPUT DESCRIPTION**

et Name ORDER

Form Type MANUAL

Prepared by: HCD

Volume: Average 500 Peak 1000

Source System I. D. 01

Date 6/15/60

007	008	009	010	011	012															

	ELEMENT DESCRIPTION					ELEMENT CLASS. (X)						NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		VALIDATION RULE (S)	IDENTIFYING ELEMENT CODES (X)										Ref. Note (✓)	REMARKS
	Suffix	Alpha	Numeric	Char. Total	Char. Decimal	Recog.	Ident.	Action	Action Mod.	Info.	Superfluous	Average	Peak		8368100		9300100									
0		A		13						X		1	1		X											
0			N	02	0					X		1	1		X											
0			N	02	0					X		1	1		X											
0			N	02	0					X		1	1		X											
1		A	N	09			X					1	1													
0			N	02	0					X		1	1	001	X											
0			N	02	0					X		1	1	001	X											
0			N	02	0					X		1	1	001	X											
0		A	N	09						X		1	1		X											
0		A		70							X	1	1		X											
0		A	N	70							X	1	1		X											
0		A		70						X		1	1		X											
0		A	N	70						X		1	1		X											
0		A	N	12						X		1	1		X											
0			N	05	0		X					1	1	003	X											
0			N	04	0		X					1	1	003	X											
0		A	N	23						X		1	1		X											
		A		23						X		1	1		X											
0		A		24						X		1	1		X											
1		A		13						X		1	1		X											
0		A	N	05			X					1	1	002	X											
0		A	N	08						X		1	1		X											
0		A	N	21							X	3	15		X		X									
0		A	N	25							X	3	15		X		X									

Process BILLING

INPUT DESCRIPTION

Page 2 of 23

...t Name ORDER

Form Type MANUAL

Prepared by: HCJ

Volume: Average 500 Peak 1000

Source System I. D. 01

Date 6/15/60

007 | 008 | 009 | 010 | 011 | 012

	ELEMENT DESCRIPTION					ELEMENT CLASS. (X)						NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		VALIDATION RULE (S)	IDENTIFYING ELEMENT CODES (X)			Ref. Note (✓)	REMARKS
	Suffix	Alpha	Numeric	Char. Total	Char. Decimal	Recog.	Ident.	Action	Action Mod.	Info.	Superfluous	Average	Peak		8368100		9300100		
ɔ		A	N	07			X					3	15	004	X				
o			N	05	0			X				3	15	005	X		X		
o			N	05	0			X				3	15	005	X		X		
ɔ			N	02	1			X				1	1		X				

Process BILLING

INPUT DESCRIPTION

Set Name CUSTOMER FILE

Form Type MAG T.

Prepared by: HCU

Volume: Average 20,000 Peak 30,000

Source System I. D. 04

Date 6/15/60

Suffix	Alpha	Numeric	Char. Total	Char. Decimal	Recog.	Ident.	Action	Action Mod.	Info.	Superfluous	Average	Peak	VALIDATION RULE (S)	8981100	6813250		Ref. Note (✓)	REMARKS
		N	05	0		X					1	1					✓	
		N	04	0		X					1	1		X				
	A		70								1	1		X		X		
	A	N	70								1	1		X		X		
		N	02	1							1	1		X		X		

Column groups: ELEMENT DESCRIPTION (Suffix through Decimal); ELEMENT CLASS. (X) (Recog. through Superfluous); NO. TIMES AN ENTRY MAY APPEAR ON THIS SET (Average, Peak); IDENTIFYING ELEMENT CODES (X).

Process BILLING

INPUT DESCRIPTION

Name ITEM FILE

Form Type MAG T.

Prepared by: HCL

Date 6/15/60

Volume: Average 3000 Peak 4500

Source System I.D. 04

ELEMENT DESCRIPTION					ELEMENT CLASS. (X)						NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		VALIDATION RULE (S)	IDENTIFYING ELEMENT CODES (X)										Ref. Note (✓)	REMARKS
Suffix	Alpha	Numeric	Char. Total	Char. Decimal	Recog.	Ident.	Action	Action Mod.	Info.	Superfluous	Average	Peak		9300/00											
	A	N	07			X					1	1													
	A	N	21								1	1		X											
	A	N	25								1	1		X											
		N	06	2							1	1		X											
		N	05	2							1	1		X											

Process BILLING

INPUT DESCRIPTION

et Name SALES TAX FILE

Form Type MAG.T.

Volume: Average 120 Peak 200

Source System I. D. 04

Prepared by: HCD

Date 6/15/60

	ELEMENT DESCRIPTION					ELEMENT CLASS. (X)						NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		VALIDATION RULE (S)	IDENTIFYING ELEMENT CODES (X)										Ref. Note (✓)	REMARKS
	Suffix	Alpha	Numeric	Char. Total	Char. Decimal	Recog.	Ident.	Action	Action Mod.	Info.	Superfluous	Average	Peak		9464/60											
0		A	N	05			X					1	1													
2			N	03	2							1	1		X											

Process *BILLING* INPUT DESCRIPTION

…t Name *CUST FILE CHANGE*

Form Type *MANUAL*

Prepared by: *HCJ*

Volume: Average *10* Peak *50*

Source System I. D. *04*

Date *6/15/60*

	ELEMENT DESCRIPTION				ELEMENT CLASS. (X)						NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		VALIDATION RULE (S)	IDENTIFYING ELEMENT CODES (X)										Ref. Note (✓)	REMARKS
	Suffix	Alpha	Numeric	Char. Total	Char. Decimal	Recog.	Ident.	Action	Action Mod.	Info.	Superfluous	Average	Peak		*898/100*		*68/3250*									
[illegible]			*N*	*05*	*0*		*X*					*/*	*/*													
[illegible]			*N*	*04*	*0*		*X*					*/*	*/*		*X*											
[illegible]			*N*	*02*	*/*			*X*				*/*	*/*		*X*		*X*									
[illegible]		*A*		*70*				*X*				*/*	*/*		*X*		*X*									
[illegible]		*A*	*N*	*70*				*X*				*/*	*/*		*X*		*X*									
[illegible]			*N*	*01*	*0*				*X*			*/*	*/*		*X*		*X*									

OUTPUT DESCRIPTION

Process BILLING   Set Name INVOICE   Form Type MAT MASTER

Number Copies 8   Volume: Average 500   Peak 1000   Special Form I. D. 1510

Prepared by: HCD

Date 6/15/60

	ELEMENT DESCRIPTION					Element Class. (X)			NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		SOURCE: Set Type			SOURCE	IDENTIFYING ELEMENT CODES (X)			
	Suffix	Alpha	Numeric	Char. Total	Char. Decimal	Recog.	Ident.	Other	Average	Peak	Process (X)	Retd. (A,B or X)	Action (X)	Source Set Identification for Direct Recording	8368100		9300100	Ref. Note (✓)
		A		13				X	1	1			X	1512 A	X			
o			N	02	0			X	1	1			X	1512A	X			
o			N	02	0			X	1	1			X	1512A	X			
o			N	02	0			X	1	1			X	1512A	X			
o		A	N	09			X		1	1			X	1512A				
o			N	02	0			X	1	1			X	1512A	X			
o			N	02	0			X	1	1			X	1512A	X			
o			N	02	0			X	1	1			X	1512A	X			
		A	N	09				X	1	1			X	1512A	X			
		A		70				X	1	1		X		04RD1	X			
		A	N	70				X	1	1		X		04RD1	X			
o		A		70				X	1	1			X	1512A	X			
o		A	N	70				X	1	1			X	1512A	X			
		A	N	12				X	1	1			X	1512A	X			
			N	05	0			X	1	1			X	1512A	X			
o			N	04	0			X	1	1			X	1512A	X			
o		A	N	23				X	1	1			X	1512A	X			
		A		23				X	1	1			X	1512A	X			
o		A		24				X	1	1			X	1512A	X			
		A		13				X	1	1			X	1512A	X			
		A	N	05				X	1	1			X	1512A	X			
		A	N	08				X	1	1			X	1512A	X			
		A	N	21				X	3	15		X		04RD2	X		X	
		A	N	25				X	3	15		X		04RD2	X		X	

OUTPUT DESCRIPTION

Process BILLING Set Name INVOICE Form Type MAT. MASTER

Number Copies 8 Volume: Average 500 Peak 1000 Special Form I. D. 1510

Prepared by: HCD

Date 6/15/60

ELEMENT DESCRIPTION					Element Class. (X)			NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		SOURCE				IDENTIFYING ELEMENT CODES (X)			
			Char.							Set Type							
Suffix	Alpha	Numeric	Total	Decimal	Recog.	Ident.	Other	Average	Peak	Process (X)	Retd. (A, B or X)	Action (X)	Source Set Identification for Direct Recording	8368/100		9300/00	Ref. Note (✓)
	A	N	07			X		3	15			X	1512 A	X			
		N	05	0			X	3	15			X	1512 A	X		X	
		N	05	0			X	3	15			X	1512 A	X		X	
		N	05	2			X	3	15		X		0ARD2	X		X	
		N	06	2			X	3	15	X				X		X	
		N	02	1			X	1	1			X	1512 A	X			
		N	05	2			X	1	1	X				X			
		N	07	2			X	3	15	X				X		X	
		N	08	2			X	1	1	X				X			
		N	07	2			X	1	1	X				X			
		N	08	2			X	1	1	X				X			
		N	03	2			X	1	1	X				X			
		N	04	2			X	1	1	X				X			

OUTPUT DESCRIPTION

Process BILLING Set Name SALES DATA Form Type MAG. TAPE

Number Copies 1 Volume: Average 2000 Peak 5000 Special Form I. D. ________

Prepared by: HCD

Date 6/15/60

	ELEMENT DESCRIPTION					Element Class. (X)			NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		SOURCE						IDENTIFYING ELEMENT CODES (X)										
				Char.							Set Type			Source Set Identification for Direct Recording													
	Suffix	Alpha	Numeric	Total	Decimal	Recog.	Ident.	Other	Average	Peak	Process (X)	Retd. (A, B or X)	Action (X)				9300/100										Ref. Note (✓)
		A	N	07					1	1			X	1512 A													
0			N	02	0				1	1			X	1512 A			X										
0			N	02	0				1	1			X	1512 A			X										
0			N	02	0				1	1			X	1512 A			X										
0			N	05	0				1	1			X	1512 A			X										
0			N	04	0				1	1			X	1512 A			X										
0			N	05	0				1	1			X	1512A			X										
			N	07	2				1	1	X						X										

OUTPUT DESCRIPTION

..ocess BILLING Set Name INVENTORY DATA Form Type MAG TAPE

Number Copies 1 Volume: Average 2000 Peak 5000 Special Form I. D. ______

Prepared by: HCD

Date 6/15/60

	ELEMENT DESCRIPTION					Element Class. (X)			NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		SOURCE				IDENTIFYING ELEMENT CODES (X)										
				Char.							Set Type														
	Suffix	Alpha	Numeric	Total	Decimal	Recog.	Ident.	Other	Average	Peak	Process (X)	Rctd. (A,B or X)	Action (X)	Source Set Identification for Direct Recording	9300/00										Ref. Note (✓)
?		A	N	07					1	1			X	1512A											
0		A		13					1	1			X	1512A	X										
0			N	05	0				1	1			X	1512A	X										

OUTPUT DESCRIPTION

Process BILLING Set Name ORDER ERRORS Form Type STANDARD PRINT

Number Copies 2 Volume: Average 5 Peak 10 Special Form I. D. ______

Prepared by: HCD

Date 6/15/60

ELEMENT DESCRIPTION					Element Class: (X)			NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		SOURCE				IDENTIFYING ELEMENT CODES (X)		
			Char.							Set Type						
Suffix	Alpha	Numeric	Total	Decimal	Recog.	Ident.	Other	Average	Peak	Process (X)	Retd. (A, B or X)	Action (X)	Source Set Identification for Direct Recording	8368100		Ref. Note (✓)
	A	N	09			X		1	1			X	1512A			
	A	N	05				X	1	1	X				X		

OUTPUT DESCRIPTION

Process BILLING   Set Name ORDER ERRORS   Form Type STANDARD PRINT

Number Copies 2   Volume: Average 5   Peak 10   Special Form I. D. ______

Prepared by: HCD
Date 6/15/60

	ELEMENT DESCRIPTION				Element Class. (X)			NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		SOURCE						IDENTIFYING ELEMENT CODES (X)										
			Char.							Set Type			Source Set Identification for Direct Recording													
Suffix	Alpha	Numeric	Total	Decimal	Recog.	Ident.	Other	Average	Peak	Process (X)	Rcrd. (A,B or X)	Action (X)				8368100										Ref. Note (✓)
	A	N	09			X		1	1			X	1512A													
		N	05	0			X	1	1	X						X										
		N	04	0			X	1	1	X						X										

OUTPUT DESCRIPTION

Process BILLING Set Name ORDER ERRORS Form Type STANDARD PRINT

Number Copies 2 Volume: Average 20 Peak 40 Special Form I. D. ______

Prepared by: HCD

Date 6/15/60

	ELEMENT DESCRIPTION					Element Class. (X)			NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		SOURCE				IDENTIFYING ELEMENT CODES (X)	
				Char.							Set Type					
	Suffix	Alpha	Numeric	Total	Decimal	Recog.	Ident.	Other	Average	Peak	Process (X)	Retd. (A, B or X)	Action (X)	Source Set Identification for Direct Recording	83681000	Ref. Note (✓)
0		A	N	09			X		1	1			X	1512 A		
0		A	N	07				X	1	1	X				X	

OUTPUT DESCRIPTION

Process BILLING Set Name ORDER ERRORS Form Type STANDARD PRINT

Number Copies 2 Volume: Average 5 Peak 10 Special Form I. D. ____

Prepared by: HCD

Date 6/15/60

	ELEMENT DESCRIPTION					Element Class. (X)			NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		SOURCE: Set Type			SOURCE	IDENTIFYING ELEMENT CODES (X)										
	Suffix	Alpha	Numeric	Char. Total	Char. Decimal	Recog.	Ident.	Other	Average	Peak	Process (X)	Retd. (A, B or X)	Action (X)	Source Set Identification for Direct Recording	8368/00		9300/00								Ref. Note (✓)
)		A	N	09			X		1	1			X	1512 A											
		A	N	07			X		1	1			X	1512 A	X										
o			N	05	0			X	1	1	X				X		X								

OUTPUT DESCRIPTION

. .cess BILLING   Set Name ORDER ERRORS   Form Type STANDARD PRINT

Number Copies 2   Volume: Average 5   Peak 10   Special Form I. D. ______

Prepared by: HCD
Date 6/15/60

	ELEMENT DESCRIPTION					Element Class. (X)			NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		SOURCE: Set Type			SOURCE	IDENTIFYING ELEMENT CODES (X)											
	Suffix	Alpha	Numeric	Char. Total	Char. Decimal	Recog.	Ident.	Other	Average	Peak	Process (X)	Retd. (A, B or X)	Action (X)	Source Set Identification for Direct Recording	8368100		9300100								Ref. Note (✓)
0		A	N	09			X		1	1			X	1512 A											
0		A	N	07			X		1	1			X	1512 A											
30			N	05	0			X	1	1	X				X		X								

OUTPUT DESCRIPTION

Process BILLING Set Name ORDER ERRORS Form Type STANDARD PRINT

Number Copies 2 Volume: Average 2 Peak 5 Special Form I. D. ________

Prepared by: HCD

Date 6/15/60

	ELEMENT DESCRIPTION					Element Class. (X)			NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		SOURCE						IDENTIFYING ELEMENT CODES (X)										
				Char.							Set Type			Source Set Identification for Direct Recording													
	Suffix	Alpha	Numeric	Total	Decimal	Recog.	Ident.	Other	Average	Peak	Process (X)	Rcrd. (A, B or X)	Action (X)				8368/00		9300/00								Ref. Note (✓)
[illegible]		A	N	09			X		1	1			X	1512 A													
[illegible]		A	N	07			X		1	1			X	1512 A													
0			N	05	0			X	1	1	X						X		X								
[illegible]			N	05	0			X	1	1	X						X		X								

OUTPUT DESCRIPTION

Process BILLING Set Name ORDER ERRORS Form Type STANDARD PRINT

Number Copies 2 Volume: Average 5 Peak 10 Special Form I. D. ____

Prepared by: HCD

Date 6/15/60

ELEMENT DESCRIPTION: Suffix	Alpha	Numeric	Char.: Total	Char.: Decimal	Element Class. (X): Recog.	Ident.	Other	NO. TIMES AN ENTRY MAY APPEAR ON THIS SET: Average	Peak	SOURCE: Set Type: Process (X)	Retd. (A, B or X)	Action (X)	Source Set Identification for Direct Recording	IDENTIFYING ELEMENT CODES (X): 83681000	Ref. Note (✓)
	A	N	09			X		1	1			X	1512A		
		N	02	0			X	1	1	X				X	
		N	02	0			X	1	1	X				X	
		N	02	0			X	1	1	X				X	

OUTPUT DESCRIPTION

Process BILLING Set Name CUST CHANGE E.,ROR Form Type STANDARD PRINT

Number Copies 2 Volume: Average 1 Peak 10 Special Form I. D. ______

Prepared by: HCD

Date 6/15/60

	ELEMENT DESCRIPTION		Char.			Element Class. (X)			NO. TIMES AN ENTRY MAY APPEAR ON THIS SET		SOURCE: Set Type			Source Set Identification for Direct Recording			IDENTIFYING ELEMENT CODES (X)										
	Suffix	Alpha	Numeric	Total	Decimal	Recog.	Ident.	Other	Average	Peak	Process (X)	Retd. (A, B or X)	Action (X)				8981100		6813250								Ref. Note (✓)
10			N	05	0		X		1	1			X	1201													
-0			N	04	0		X		1	1			X	1201			X										
10			N	02	1			X	1.	1	X						X		X								
30		A		70				X	1	1	X						X		X								
10		A	N	70				X	1	1	X						X		X								
30			N	01	0			X	1	1	X						X		X								
0		A		09				X	1	1	X						X		X								

PROCESS DESCRIPTION

☒ Validation

Prepared by HCD

Date 6/15/60

SOURCE: ELEMENT — Prior Result Value	SUF.	Set Ident. or Rule No. for Prior Result	COND.: Greater Than	COND.: Less Than	COND.: Equal To	Oper (+ - / X ✕)	Set = To (V)	SOURCE/DISPOSITION: ELEMENT — Name, Result, Prior Result or Actual Value	SUF.	Set Ident. or Rule No. for Prior Result	Oper (+ - / X)	Note Ref. (V)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
PED		1512 A	✓		✓			(01)				1	Y	N																
				✓	✓			(12)					Y		N															
PED		1512 A	✓		✓			(01)					Y			N														
				✓	✓			(31)					Y				N													
PPED		1512 A		✓	✓			/CURRENT YR/					Y					N												
													S	/	/	/	/	/												
PPED		1512 A					✓	MO SHIPPED		040307				X	X	X	X	X												
PED		1512 A					✓	DA SHIPPED		040307				X	X	X	X	X												
PPED		1512 A					✓	YR SHIPPED		040307				X	X	X	X	X												
T MO /							✓	MO SHIPPED		1512 A				X	X	X	X	X												
T DA /							✓	DA SHIPPED		1512 A				X	X	X	X	X												
T YR /							✓	YR SHIPPED		1512 A				X	X	X	X	X												

MANAGEMENT RULE SUFFIX AND FREQUENCY

Cond: Y = Condition Is Satisfied; N = Condition Is Not Satisfied; Blank = Condition Not Applicable

Action: X = Action to Be Taken; Blank = Action Not to Be Taken

PROCESS DESCRIPTION

Prepared by HCD

Date 6/15/60

☒ Validation

MANAGEMENT RULE SUFFIX AND FREQUENCY

Cond: Y = Condition Is Satisfied; N = Condition Is Not Satisfied; Blank = Condition Not Applicable

Action: X = Action to Be Taken; Blank = Action Not to Be Taken

SOURCE: ELEMENT (Name, Prior Result or Actual Value)	SUF.	Set Ident. or Rule No. for Prior Result	COND.: Greater Than	COND.: Less Than	COND.: Equal To	Oper (+ - / X Σ)	Set = To (V)	SOURCE/DISPOSITION: ELEMENT (Name, Result, Prior Result or Actual Value)	SUF.	Set Ident. or Rule No. for Prior Result	Oper (+ - / X)	Note Ref. (V)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
'AX CODE		1512 A			√			SALES TAX CODE		04RD3		1	Y	N																
													9/5	5																
'X CODE		1512 A					√	SALES TAX CODE		040301				X																
-b)							√	SALES TAX CODE		1512 A				X																
AN NO		1512 A			√			SALESMAN NO		04RD1		2	Y	N																
CT NO		1512 A			√			CUST ACCT NO		04RD1			Y		N															
													9/5	2	3															
AN NO		1512 A					√	SALESMAN NO		040302				X	X															
CT NO		1512 A					√	SALESMAN NO		040302				X	X															
NO		1512 A			√			STOCK NO		04RD2		2	Y	N																
													9/5	5																
NO		1512 A					√	STOCK NO		040303				X																
HIPPED		1512 A			√			/NNNNN/				2	Y	N	Y	Y														
DERED		1512 A			√			/NNNNN/					Y	Y	N	Y														
DERED		1512 A	√		√			QTY SHIPPED		1512 A			Y			N														
													9/5	2	2	1														
IPPED		1512 A					√	QTY SHIPPED		040304				X																
DERED		1512 A					√	QTY ORDERED		040305					X															
PPED		1512 A					√	QTY SHIPPED		040306						X														
DERED		1512 A					√	QTY ORDERED		040306						X														

PROCESS DESCRIPTION

Prepared by HCD

Date 6/15/60

☐ Validation

MANAGEMENT RULE SUFFIX AND FREQUENCY

Cond: Y = Condition Is Satisfied
N = Condition Is Not Satisfied
Blank = Condition Not Applicable

Action: X = Action to Be Taken
Blank = Action Not to Be Taken

SOURCE: ...MENT (...rior Result / ...l Value)	SUF.	SOURCE: Set Ident. or Rule No. for Prior Result	COND.: Greater Than	COND.: Less Than	COND.: Equal To	Oper (+ - / X Σ)	Set = To (√)	SOURCE/DISPOSITION: ELEMENT (Name, Result, Prior Result or Actual Value)	SUF.	SOURCE/DISPOSITION: Set Ident. or Rule No. for Prior Result	Oper (+ - / X)	Note Ref. (√)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
SUP BON		04RD1				X		PRICE UN BASE		04RD2																				
							√	UNIT PRICE		1512 B																				
... QSA		1512 A			√			/BLANK/					Y	N																
													6 0	4 0																
... QSA		1512 A				X		AMT PRE TAX		1512 B																				
							√	AMT QTY ALLOW		1512 B				X																
							√	RESULT A						X																
.../							√	RESULT A					X																	
...RICE		1512B				X		QTY SHIPPED		1512 A																				
							√	AMT PRICE EXT		1512 B																				
							√	AMT PRICE EXT		0401																				
...ICE EXT						Σ	√	AMT PRE TAX		1512 B																				
...IPPED		1512 A				X		TAX FED EX UN		04RD2																				
							√	AMT FED EX TAX		1512B																				
...RE TAX		1512 B				+		AMT FED EX TAX		1512 B																				
							√	RESULT A																						

PROCESS DESCRIPTION

☐ Validation

Prepared by HCD

Date 6/15/60

SOURCE ELEMENT: Prior Result or Value	SUF.	Set Ident. or Rule No. for Prior Result	COND. Greater Than	COND. Less Than	COND. Equal To	Oper (+ - / X Σ)	Set = To (V)	SOURCE/DISPOSITION ELEMENT: Name, Result, Prior Result or Actual Value	SUF.	Set Ident. or Rule No. for Prior Result	Oper (+ - / X)	Note Ref. (V)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
AX CODE		1512 A			✓			(X b b b b)					Y	N																
													5	9/5																
AX PC		0ARD 3				X		AMT PRE TAX		1512 B																				
							✓	AMT SALES TAX		1512 B				X																
							✓	RESULT A						X																
/							✓	RESULT A					X																	
AX PC		0ARD3					✓	SALES TAX PC		1512 B				X																
- A		(010)				-		RESULT A		(007)	+																			
r A		(011)					✓	AMT CUST INV		1512 B																				

MANAGEMENT RULE SUFFIX AND FREQUENCY
Cond: Y = Condition Is Satisfied; N = Condition Is Not Satisfied; Blank = Condition Not Applicable
Action: X = Action to Be Taken; Blank = Action Not to Be Taken

☐ Validation

Prepared by HCD

Date 6/15/60

SOURCE: ELEMENT (…rior Result / … Value)	SUF.	Set Ident. or Rule No. for Prior Result	COND. Greater Than	Less Than	Equal To	Oper (+ - / X Σ)	Set = To (V)	SOURCE/DISPOSITION: ELEMENT (Name, Result, Prior Result or Actual Value)	SUF.	Set Ident. or Rule No. for Prior Result	Oper (+ - / X)	Note Ref. (V)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
AN NO		1201			√			SALESMAN NO		04RD1			Y	Y	Y	N		N		N										
CCT NO		1201			√			CUST ACCT NO		04RD1			Y	Y	Y		N		N		N									
3 IND		1201			√			(1)					Y			Y	Y													
					√			(2)						Y				Y	Y											
					√			(3)							Y					Y	Y									
													1/5	1	5/0	1	1	1/5	1/5	1	1									
VK/							√			04RD1			X																	
		1201					√			04RD1								X	X											
SUP BON		1201					√	DISC PC SUP BON		04RD1					X															
SUPBON		1201					√	DISC PC SUP BON		0404				X		X	X			X	X									
AME SOLD		1201					√	CUST NAME SOLD		0404				X		X	X													
OR SOLD		1201					√	CUST ADDR SOLD		0404				X		X	X													
VED)							√	ERROR REASON		0404				X																
HED)							√	ERROR REASON		0404						X	X			X	X									

MANAGEMENT RULE SUFFIX AND FREQUENCY

Cond: Y = Condition Is Satisfied; N = Condition Is Not Satisfied; Blank = Condition Not Applicable

Action: X = Action to Be Taken; Blank = Action Not to Be Taken

STANDARD REFERENCE NOTES FOR VALIDATIONS

Note Number	Explanation
1.	If the element is not valid, continue with the execution of the Management Rules for validations and processes indicated by the set that contains the invalid element.
2.	If the element is not valid, continue with the execution of the Management Rules for validations indicated by the set that contains the invalid element. Do not execute the Management Rules for processes indicated by the set that contains the invalid element.
3.	If the element is not valid, do not continue with the execution of the Management Rules for validations. Do not execute the Management Rules for processes.

SYSTEMS ENGINEERING SERVICES

CLEARINGHOUSE REPORT

AN INSURANCE COMPANY FILE MAINTENANCE PROBLEM

June 10, 1961
Ref. No. 1F3

Burton Grad

AN INSURANCE COMPANY FILE MAINTENANCE PROBLEM

Burton Grad
IBM

This report presents two methods (flow chart and decision table) for representing the decision logic of a complex problem; it thereby provides a means of comparing the relative merits of the two techniques. Some considerations in such a comparison are: clarity of understanding, ease of modifying, ability to detect logical errors and omissions, ability to see important relationships, etc. This example by no means represents a controlled test or evaluation of flow charts vs. decision tables; it has, however, provided some insight and firsthand experience with the two methods on an identical problem.

The particular problem is concerned with master file maintenance and controlling key operating procedures of a large insurance company. The operations are presented at the systems level and while not precise enough for direct coding, should be accurate and structurally sound. Some of the logical inaccuracies that exist in the flow chart were corrected in the decision tables.

With the problem solution initially represented in flow chart form, it was then decided to explore the capability of decision tables for describing such a complex decision procedure. It took approximately 25 man-hours to study the flow chart, understand and structure the problem, and prepare the decision tables. This short time did not allow thorough review and debugging of the decision tables. The most difficult task was to understand the problem from the information available; considerable time and effort were required with the flow chart originator toward this end. However, once the flow chart was grasped, the problem could be subdivided into several major portions. It seemed at the time that this might be one main advantage of tables, i.e., they seem to force logical structure.

The Basic Problem Solution

Another complication of the job is that a single customer may have a multiple account, i. e., more than one policy; if a customer has a multiple account, he may go on a monthly pay plan instead of the normal three-payment or one-payment method. Tables 002, 003, and 012 detect and handle multiple account and monthly pay cases.

There are two major types of activities to the file. The first is that which is scheduled because of the date, such as renewals, terminations, etc. This is handled by Tables 005 and 009, which are concerned with scheduled activities. The other type of work involves handling transactions, where a change in policy status or introduction of a new policy takes place. Tables 008 and 011 take care of transaction activities.

The remaining table, Table 010, is a closed procedure table which is used in a variety of cases to write out a previous policy and obtain the policy to be examined.

Attached you will find the flow chart used to describe the insurance company job, followed by the decision tables which cover the same ground.

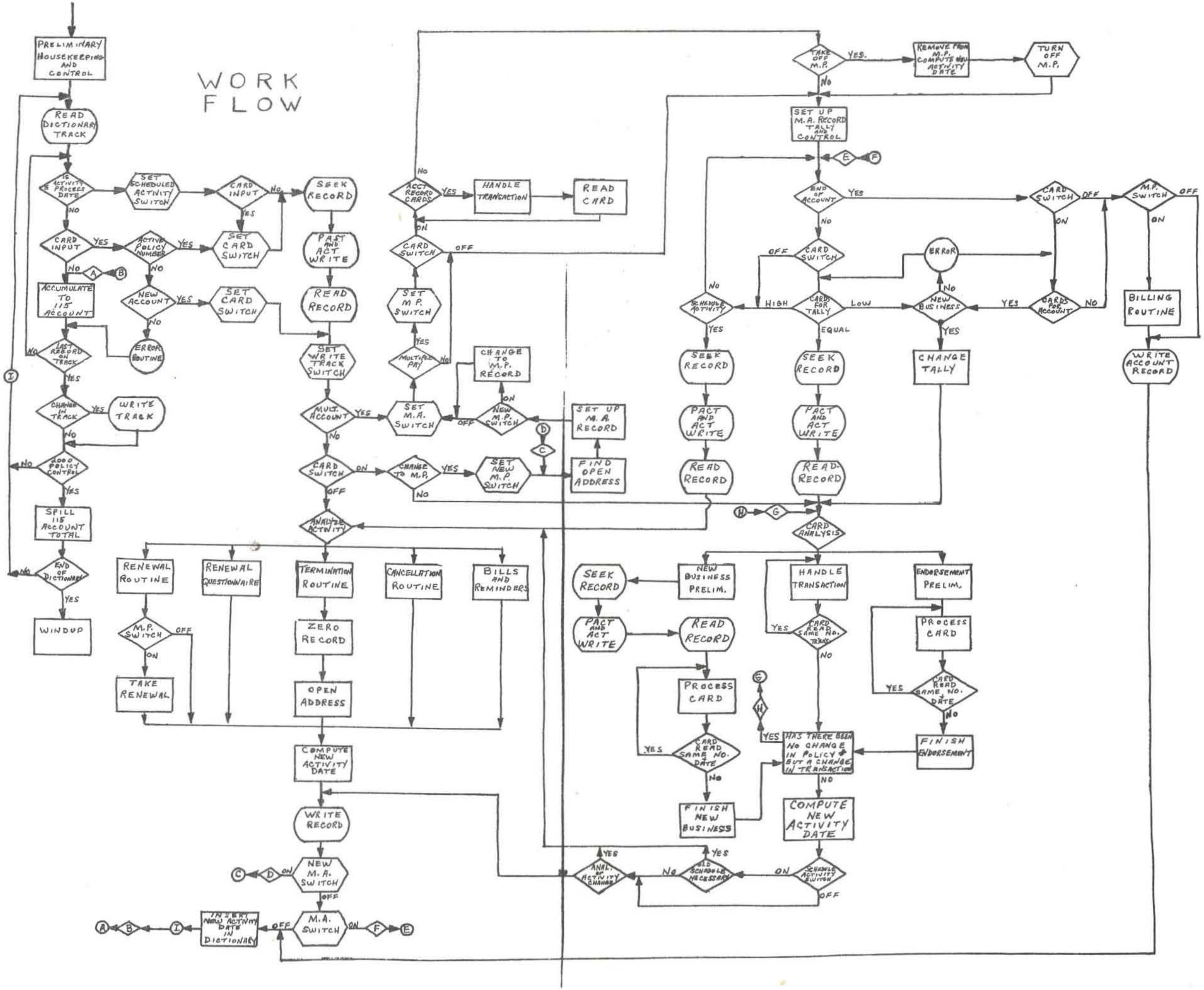

WORK FLOW
PRELIMINARY HOUSEKEEPING AND CONTROL
READ DICTIONARY TRACK
SET SCHEDULED ACTIVITY SWITCH
CARD INPUT
SEEK RECORD
ACTIVE POLICY NUMBER
SET CARD SWITCH
PACT AND ACT WRITE
ACCUMULATE TO 115 ACCOUNT
NEW ACCOUNT
READ RECORD
ERROR ROUTINE
LAST RECORD ON TRACK
SET WRITE TRACK SWITCH
CHANGE IN TRACK
WRITE TRACK
MULT. ACCOUNT
SET M.A. SWITCH
2000 POLICY CONTROL
CHANGE TO M.P.
SET NEW M.P. SWITCH
SPILL 115 ACCOUNT TOTAL
ANALYZE ACTIVITY
END OF DICTIONARY
WINDUP
RENEWAL ROUTINE
RENEWAL QUESTIONNAIRE
TERMINATION ROUTINE
CANCELLATION ROUTINE
BILLS AND REMINDERS
M.P. SWITCH
ZERO RECORD
TAKE RENEWAL
OPEN ADDRESS
COMPUTE NEW ACTIVITY DATE
WRITE RECORD
NEW M.A. SWITCH
M.A. SWITCH
INSERT NEW ACTIVITY DATE IN DICTIONARY
ACCT RECORD CARDS
HANDLE TRANSACTION
READ CARD
CARD SWITCH
SET M.P. SWITCH
MULTIPLE PAY
CHANGE TO M.P. RECORD
NEW M.P. SWITCH
SET UP M.A. RECORD
FIND OPEN ADDRESS
TAKE OFF M.P.
REMOVE FROM M.P. COMPUTE NEW ACTIVITY DATE
TURN OFF M.P.
SET UP M.A. RECORD TALLY AND CONTROL
END OF ACCOUNT
CARD SWITCH
M.P. SWITCH
ERROR
SCHEDULE ACTIVITY
CARDS FOR TALLY
NEW BUSINESS
CARDS FOR ACCOUNT
BILLING ROUTINE
SEEK RECORD
CHANGE TALLY
WRITE ACCOUNT RECORD
PACT AND ACT WRITE
READ RECORD
CARD ANALYSIS
SEEK RECORD
NEW BUSINESS PRELIM.
HANDLE TRANSACTION
ENDORSEMENT PRELIM.
PACT AND ACT WRITE
READ RECORD
CARD READ SAME NO. TRANS
PROCESS CARD
PROCESS CARD
CARD READ SAME NO. DATE
HAS THERE BEEN NO CHANGE IN POLICY BUT A CHANGE IN TRANSACTION
FINISH ENDORSEMENT
CARD READ SAME NO. DATE
FINISH NEW BUSINESS
COMPUTE NEW ACTIVITY DATE
ANAL. OF ACTIVITY CHANGE
OLD SCHEDULE NECESSARY
SCHEDULE ACTIVITY SWITCH
YES
NO
ON
OFF
HIGH
LOW
EQUAL
A
B
C
D
E
F
G
H
I

TABLE 001 Overall Control

Rule No.	1	2	3	4	5	6	7	8	9	10	11	12	13
START	Y	N	N	N	N	N	N	N	N	N	N	N	N
Activity Date ⩽ Process Date		N	N	N	N	N	N	N	N	N	Y	Y	N
Card Input for this account		N	N	N	N	N	N	N	Y	Y	N	Y	Y
Last Record on Track		N	Y	Y	Y	Y	Y	Y					
Change Track Switch ON			N	N	N	Y	Y	Y					
Policy Control No. = 2000			N	Y	Y	N	Y	Y					
End of Dictionary				N	Y		N	Y					
Active Policy No.									N	N			Y
New Account									N	Y			
Preliminary Housekeeping & Control	X												
Error... Not Active Policy									X				
Set Status Code =	0				2			2					
Set Track Change Switch ON										X	X	X	X
Accumulate to 115 Account		X	X	X	X	X	X	X					
Write Dictionary Track						X	X	X					
Set Track Change Switch OFF						X	X						
Spill 115 Account Total				X	X		X	X					
Read Next Dictionary Track	X		X	X		X	X						
Setup Next Dictionary Item	X	X	X	X		X	X						
Set Policy Control No. =	100		+100	100		+100	100						
Windup (incl. Table 010)					X			X					
Read Input Card	X								X				
Set Card Switch ON										X		X	X
Set Schedule Activity Switch ON											X	X	
GO TO TABLE	001	001	001	001	Stop	001	001	Stop	001	002	002	002	002

TABLE 002 Special Processing

Rule No.	1	2	3	4	5	6	7	8	9	10	11	12	13
Multiple Account	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Monthly Pay	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
Card Switch ON	Y	N	Y	N	Y	Y	N	N	Y	Y	Y	Y	Y
Type of Transaction = Status Change	N		Y		N	Y			N	N	Y		
= Cash									N	N		Y	Y
Transaction Switch = C							N	Y	N	Y		N	Y
Set Multiple Account Switch ON				X	X	X	X		X		X	X	
Set Monthly Pay Switch ON							X		X		X	X	
Read Multiple Account Record				X	X		X		X			X	
Setup Multiple Account Tally & Control				X	X		X	X	X	X			
Do Policy Record Setup (Table 010)	X	X											
Set Transaction Switch =								b		b		"C"	
Handle Monthly Pay Cash Transaction												X	X
Read Input Card												X	X
GO TO TABLE	008	005	012	003	003	012	003	003	003	003	012	011	011

TABLE 003 Multiple Account Control

Rule No.	1	2	3	4	5	6	7	8	9	10	11
End of Account	Y	Y	Y	Y	N	N	N	N	N	N	N
Card Switch ON	N	N	Y	Y	Y	Y	Y	Y	Y	N	N
Cards for Tally					High	High	Low	Low	EQ		
New Business			Y	N			Y	N			
Monthly Pay Switch ON	N	Y									
Schedule Activity					N	Y				N	Y
Error...Card out of line				X				X			
Set Transaction Switch =				"L"				"L"			
Change Tally			X				X				
Set Schedule Activity Switch OFF	X	X									
Billing Routine		X									
Write Account Record	X	X									
Tally					X					X	
Read Input Card				X				X			
Do Policy Record Setup (Table 010)			X			X	X		X		X
Set Multiple Account Switch OFF	X	X									
GO TO TABLE	005	005	008	011	003	005	008	011	008	003	005

TABLE 005 Scheduled Activity

Rule No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Schedule Activity Switch ON	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
Monthly Pay Switch ON	Y	N												
Multiple Account Switch ON	Y	Y	N	Y	N	Y	N	Y	N	Y	N		Y	N
Type of Activity = Renewal	Y	Y	Y											
= Renewal Questionnaire				Y	Y									
= Termination						Y	Y							
= Cancellation								Y	Y					
= Bills & Reminders										Y	Y			
Transaction Switch = "T"												N	Y	Y
Renewal Routine	X	X	X											
Renewal Questionnaire Routine				X	X									
Termination Routine						X	X							
Cancellation Routine								X	X					
Bills & Reminders Routine										X	X			
Zero Record & Open Address						X	X							
Take Renewal	X													
Compute New Act. Date	X	X	X	X	X	X	X	X	X	X	X			
Write Policy Record	X	X	X	X	X	X	X	X	X	X	X		X	X
Insert New Act. Date in Dictionary			X		X		X		X		X	X		X
Tally	X	X		X		X		X		X			X	
Change Multiple Account Record	X	X		X		X		X		X			X	
Set Schedule Activity Switch OFF			X		X		X		X		X			
GO TO TABLE	003	003	001	003	001	003	001	003	001	003	001	001	003	001

TABLE 008 Transaction Activity

Rule No.	1	2	3	4	5	6	7	8	9	10	11	12
Card Switch ON	Y	Y	Y	N	N	N	N	N	N	N	N	N
Transaction Type = New Business	Y		N	Y		N				N		
= Endorsement		Y	N		Y	N					N	Y
Control Code =				"NB"	"EN"	"RT"	"NB"	"EN"	"RT"	"NB"	"EN"	"RT"
Policy Nos. match				Y	Y	Y	N	N	N	Y	Y	Y
New Business Preliminary	X											
Endorsement Preliminary		X										
Handle Transaction			X			X						
Process Card	X	X		X	X							
Read Input Card	X	X	X	X	X	X						
Set Card Switch OFF	X	X	X									
Set Card Switch ON										X	X	X
Set Control Code =	"NB"	"EN"	"RT"									
Finish Endorsement								X			X	
Finish New Business							X			X		
Compute New Activity Date							X	X	X			
GO TO TABLE	008	008	008	008	008	008	009	009	009	008	008	008

TABLE 009 Scheduled Activity Check

Rule No.	1	2	3	4	5
Schedule Activity Switch ON	Y	Y	Y	N	N
Old Schedule Necessary	Y	N	N		
Analysis of Activity Change		Y	N	Y	N
Set Schedule Activity Switch ON				X	
Set Schedule Activity Switch OFF			X		
Set Transaction Switch = "T"			X		X
GO TO TABLE	005	005	005	005	005

TABLE 010 (DO) Policy Record Setup

Rule No.	1	2	3
Status Code	1	0	2
Seek Policy Record	X	X	
Pre Activity Write	X		X
Activity Write	X		X
Read Policy Record	X	X	
Set Status Code =		1	

TABLE 011 Card Input Control

Rule No.	1	2	3	4	5
Card Input for this Policy	Y	Y	N	N	N
Transaction Switch =	"C"	"L"	"C"	"C"	"L"
Schedule Activity Switch ON			Y	N	
Set Card Switch OFF			X	X	X
GO TO TABLE	002	003	002	003	003

TABLE 012 Status Change Control

Rule No.	1	2	3	4	5
Multiple Account	Y	Y	Y	N	N
Monthly Pay		Y	N	N	N
Type of Status Change = On Multiple Account				Y	
= On Monthly Pay			Y		
= On Multiple Account & Monthly Pay					Y
= Off Multiple Account	Y				
= Off Monthly Pay		Y			
Read Multiple Account Record	X	X	X		
Setup & Write Multiple Account Record				X	X
Change Dictionary Track this item	X	X	X	X	X
Change Multiple Account Record	X	X	X		
Set Transaction Switch =		"C"	"C"		
Read Input Card		X	X	X	X
Set Type of Transaction = blank	X				
GO TO TABLE	002	011	011	002	002

SYSTEMS
ENGINEERING
SERVICES

CLEARINGHOUSE REPORT

TABLES SIGNAL BETTER COMMUNICATION

June 1, 1961
Ref. No. 1F1

Burton Grad

INTERNATIONAL BUSINESS MACHINES CORPORATION
White Plains, New York

Tables Signal Better Communication

Talk Given by Burton Grad, Manager IBM Systems Engineering Development

The pilot is preparing to land his single engine plane at the airport; it is late at night and his fuel supply is low. He calls to the radio tower and asks for landing instructions. All he hears in return is a babble in a foreign language which he can't understand.

The executive has spent the last hour of his day dictating an important speech; the next morning he comes in and wants to review the material. His secretary is out ill. The other girls in the office all read Gregg, not Pitman.

A design engineer has carefully prepared a number of complex Boolean equations to explain the operation of a new computer circuit. He shows these to the manufacturing engineer to give an indication of what needs to be constructed. The manufacturing engineer says, "I don't understand Boolean algebra."

We could go on and on citing examples like these of events and occurences where lack of a common language for communication causes difficulties ranging all the way from the most trivial to the deadly. Systems Engineering faces communication barriers as serious as those of any profession. The systems engineer today does not have a language to communicate with management; he does not have a language to communicate with computer programmers; he does not have a language to communicate with functional specialists; he does not even have a language to communicate with other systems engineers.

Programmers who have learned one computer at the machine language level can't understand the programming of another machine at the machine language level without spending the time necessary to learn the second machine's special codes and instructions. For this reason (among others) there has been intensive effort to develop common languages like FORTRAN, Commercial Translator and COBOL which will be applicable to a number of machines. But the communication between programmer and machine is merely a small part of the total problem.

For Systems Engineering it is vital to develop tools and techniques to permit a manager to state his decision criteria and decision rules. We must find a common language so systems engineers can communicate with product engineers, accountants, and manufacturing planners, to find out their decision rules and decision logic; that is critical to determine the characteristics of the system that is going to be modelled or controlled. A method must be found for two-way communication with computer programmers to be sure that the intended decision rules are in fact being executed. A technique is needed to aid the systems engineer in establishing complete decision rules and in predetermining that these rules will accomplish the intended goals.

In the past, this problem has not been as severe. Because of the limited size of business systems problems, we could depend on the programmer to understand the particular problems well enough to be sure the logic was correct and to check the problem out thoroughly. However, as the systems we are trying to solve become larger and more complex, this expedient is no longer satisfactory. Systems engineers must take on the responsibility for designing the decision logic and for insuring that it is being executed properly. To do this systems engineers must have a professional language which will serve for effective intercommunication.

What has caused the communication void? What has caused this communication moat surrounding the systems engineer? There are at least three major factors involved:

1- The inability to clearly and concisely express decision logic and decision rules for describing business systems.

2- The inability to show cause-effect relationship between conditions and actions.

3- The inability to guarantee or even aid in achieving logical completeness in establishing decision rules.

Today, we have available a number of techniques which have been applied to solving the communication problem: we've tried to use narrative, flow charts and even logical equations. But none of these has filled the bill. Each has major drawbacks; the failure of these known techniques has led to consideration of another alternative: decision tables.

Decision Tables

Decision tables are a formal method for describing decision logic in a two-dimensional display. The layout clearly shows the cause and effect relationship between conditions and actions; it explicitly relates decision alternatives.

Decision tables use a format which is familiar to us from analytical, financial, and statistical tables. Since the days of the Babylonians, people have used tables as a means of organizing information where the relationships were complex or the amount of data great. These data tables appear to be superior to many other forms of information organization because:

1- They provide clarity and conciseness through data classification.

2- They clearly show relationship of dependent to independent variables.

3- They explicitly indicate omissions.

Decision tables use tabular format to represent dynamic situations. Where we have used flow charts, narrative, or logical equations to describe decision logic, or an operating procedure, we now find it possible to use decision tables for these same jobs. The argument in favor of tables is their relative convenience and effectiveness, not that they can describe systems that cannot also be described in other ways.

Tabular form has been used by programmers since the earliest days of computers. The most common use of tables has been to relate some function to an argument. Given the value of one factor, the table provides the value of another dependent factor. For example, a table might relate capitals to states (Figure 1). Given the state name, determine the name of the capital.

STATE	Alabama	Alaska			Wyoming
CAPITAL	Montgomery	Juneau			Cheyenne

In this example State appears above the double line and Capital below; each different state name is in a column and physically below it, the name of the corresponding capital. If the State is Alabama, then the Capital is Montgomery; if the State is Alaska, then the Capital is Juneau.

An extension of this concept is seen in Figure 2 in the use of a matrix to display the value of a particular factor as a function of multiple variables.

HEALTH / AGE	EXCELLENT	GOOD	FAIR	POOR
>25 <35	1.27	1.62	1.98	2.73
>35 <45	1.83	2.12	2.53	3.42
>45 <55	2.51	2.93	3.47	5.27
>55 <65	3.29	3.91	4.85	8.73
>65	5.21	6.45	7.61	10.97

Insurance premium rates are shown as a function of health and age. In the example, if health is excellent and age is between 25 and 35, then the rate is \$1.27. However, if health is poor and age between 55 and 65, then the rate is \$8.73. Unfortunately, the visual effectiveness of a matrix is reduced when the number of independent variables exceeds two or the number of dependent variables is greater than one.

Because of the natural benefits from using tables, it seems that there should be some way to generalize tabular form so that any number of independent and dependent variables might be shown with clear visual correspondence. Figure 3 (on the next page) shows a table with four indenendent and three

Health	Excellent	Excellent		Poor
Age	25, 35	25, 35		65
Section of Country	East	East		West
Sex	Male	Female		Female
Premium Rate	1.27	1.18		9.82
Policy Limit	200,000	100,000		10,000
Type of Policy	A, B, or C	A, B, or C		R

dependent factors where clarity, interrelationship and comprehensiveness have been maintained.

In this example, the decision table indicates insurance premium rate, policy limit, and type of policy as a function of health, age, section of country, and sex. If the applicant is in excellent health, between 25 and 35 years of age, from the East, and is a male, his rate is $1.27, the insurance limit is $200,000, and he may be issued policy type A, B, or C. All of the alternatives are clearly set forth, one by one, across the table.

To obtain a better understanding of a decision table, let's look at its fundamental elements as shown in Figure 4.

Decision Rule

TABLE HEADER	RULE HEADER
Condition Stub	Condition Entry
Action Stub	Action Entry

The double lines serve as demarcation: CONDITIONS are shown above the horizontal double lines, ACTIONS below. The STUB is to the left of the vertical double line, ENTRIES to the right. A condition states a relationship. An action states a command.

If all the conditions in a column are satisfied then the actions in that column are executed. Each such vertical combination of conditions and actions is called a RULE. In the same column with the entries for each rule, there may be specialized data relating to that rule; this is called the RULE HEADER. Similarly, each table may have certain specialized information which is called the TABLE HEADER.

Consider another sample table which contains all the same elements, but has some different properties. This table is Figure 5.

TABLE:CREDIT	Rule 1	Rule 2	Rule 3	Rule 4
Credit limit is o.k.	Y	N	N	N
Pay experience is favorable		Y	N	N
Special clearance is obtained			Y	N
Approve order	X	X	X	
Return order to Sales				X

The first rule would be read: If credit limit is OK, then approve order. The second rule would be read: If credit limit is not OK and pay experience is favorable, then approve order. In this LIMITED ENTRY table, the entire condition or action must be written in the stub. The condition entry is limited to indicating whether the corresponding condition should be asserted, negated or ignored; the action entry indicates if the action stub should be executed or ignored.

This is in contrast, as you may note, to the table shown in Figure 3, which is called an EXTENDED ENTRY table. In this case the individual condition or action information extends from the stub into the corresponding entries. In any given table, we can, of course, mix extended and limited entry form, whichever is more convenient for a particular condition or action.

The Use of Decision Tables

To this point sample decision tables and their elements have been discussed to describe concept and structure. Now the application and use of decision tables will be presented. A number of experiments conducted over the past four years have used decision tables on a variety of problems; these will be reviewed briefly.

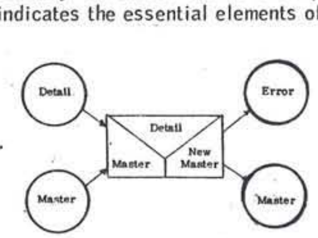
While I was project leader for General Electric's Integrated Systems Project, the potential application of tables to a wide variety of problems was explored including its use for product design, operation planning, cost determination, factory scheduling, etc. The results certainly revealed the opportunity of using decision tables as a major new tool to clarify communication among different technical specialists as well as between these specialists and computer programmers. It was stimulating to watch a manufacturing engineer suddenly grasp product design decision logic and then point out where restraints had been introduced by the product engineer that were of little value to anybody. Through this kind of examination, significant improvements might be made in the total product.

At Sutherland Company, a consulting firm in Peoria, Illinois, management decision rules have been studied with various customers and expressed in tabular form. These decision tables have been applied to Air Force logistics and various commercial situations such as accounts receivable, accounts payable, etc. From all reports, this work has permitted a more effective and comprehensive statement of the current decision logic and provided more meaningful and understandable communication between systems men and programmers.

An area of experimentation already familiar to many of you is the work done at Hunt Foods and Industries by Mr. O. Y. Evans, who is now with IBM. Mr. Evan's work was directed toward communication among different systems men, and from systems men to programmers, concerning the complex decision rules involved in stock control, sales analysis, etc. The results demonstrate that this approach was an effective formal way to state very complex logic without requiring knowledge of Boolean algebra or any other precise mathematical technique.

IBM has been working with several of its customers investigating potential applications of decision tables to a wide variety of problems. From these experiments, it seems clear that decision tables are frequently easier to prepare than comparable programming methods, and that they are an effective aid to systems analysis. In these experiments, communication between systems engineer and programmer has been substantially improved; communication between systems engineer and management has also benefitted from the common description of decision rules.

To convey how tables can be developed, let's follow the process through the significant problem of file maintenance . The block diagram in Figure 6 indicates the essential elements of the problem solution.



A detail file and a master file are the two inputs. The updated master file and an error file are the principal outputs. Within the computer, three basic areas are assigned: master, detail, and new master. The purpose of the update logic is to modify the incoming master file by the detail information to produce an updated master file containing any additions and changes and from which deleted records have been eliminated.

Figure 7 (on the following page) is one of two tables prepared to perform this job.

Rule 1 states the starting condition. At the start, one master record and one detail record are read into the corresponding memory areas. At this point, sequence control returns to the beginning of the table.

Rule 2 and all the following ones are now pertinent. Rule 2 specifically handles the end of job conditions, i.e., end of detail and end of master. In this case, control is transferred to End, a closing routine to provide for sentinels, tape marks, etc.

TABLE: Update Rule #	01	02	03	04	05	06	07	08
Start	Y	N	N	N	N	N	N	Else
End of detail		Y	Y	N	N	N	N	
End of Master		Y	N	Y	N	N	N	
Detail					<Master	>Master	=Master	
Detail an "addition"				Y	Y			
Do Error Routine								x
Move Master to New Master							x	
Move Detail to New Master				x	x			
Set Addition Switch				On	On		Off	
Write Master			x			x		
Read Master	x		x			x		
Read Detail	x			x	x			x
Go to Table	Update	End	Update	Change	Change	Update	Change	Update

Rule 3 describes the situation when the end of detail has been reached, but not the end of master. Since there can be no further changes, additions, or deletions to the original master, the actions are to write the updated master from the master area, read another master, and then return to the beginning of the table.

In Rule 4, the end of master has been found, but not the end of detail; the remaining details should only be additions. Therefore, the information in the detail area is moved to the new master area, the addition switch is set on, a new detail record is read, and control transferred to the Change Table.

Rules 5, 6, and 7 are concerned with cases where neither the detail nor the master file has ended. The identification number in the detail area is compared to the identification number in the master area. Rule 5 considers the event when the detail is less than the master; in this case the detail should be an addition in order to follow the same logic of Rule 4. In Rule 6 the detail is greater than the master; consequently the same logic as Rule 3 applies. Rule 7 covers the case where master and detail are equal. The information in the master area is moved to the new master area, and control is transferred to the Change Table.

The final rule, Rule 8, is the ELSE Situation. When this occurs something has gone wrong, since all legitimate possibilities have already been examined. An error routine is carried out; then another detail record is read. Rule 8 will take care of cases involving sequence errors in the master file and certain types of sequence errors in the detail file (if the out-of-sequence detail is not an addition). It will also take care of any non-matching detail which is not an addition.

The table can be rearranged to aid programming efficiency: columns with higher frequency of success should be moved to the left and those with lower frequency to the right. Rules 1 and 2 would be way over to the right since they occur only once in each program. Depending upon the particular data, Rule 6 (the column where the detail is greater than the master) will probably be the most frequent case and should be the first one considered. One recommended order is: 6, 7, 5, 3, 4, 1, 2, 8.

Another concept for improving program efficiency is to rearrange the conditions to present the most discriminating condition at the top and the least discriminating at the bottom. For example, the start condition, which is shown first, probably should be last since this only distinguishes one case out of all the thousands that will occur. A similar statement can be made about end of detail and end of master. It seems evident that the comparison of detail to master would be the most discriminating criteria and therefore placed first in the table.

The Case for Tabular Form

Look once more at Figure 7 and compare its statement of the update decision logic with that in the following narrative. Which is clearer and more concise, which shows cause-effect relationships better, which aids more in determining logical completeness.

Mr. T. F. Kavanagh speaking at the 1960 Eastern Joint Computer Conference had this to say: "the decision... table is a fundamental language concept... broadly applicable to many classes of information processing and decision making problems... tables force a step-by-step analysis of the decision... are easily understood by humans regardless of their functional background (they are) simple and straightforward (enough) that... specialists can write tables... with very little training... tables are easy to maintain (and) errors are reported at the source language level."

Mr. O. Y. Evans states of his work on tabular techniques: "The tabular approach... aids... in visualizing the numerous relationships and alternatives... (and) permits data rules to be readily reviewed for omissions and inconsistencies... (in addition it) provides flexibility in changing any portion of the analysis."

The CODASYL Systems Group, part of The Development Committee of the Conference on Data System Languages, has been looking into the use of decision tables. In a recent release the following statement was made: "Investigation... indicates that the systems analysis method discussed above (decision tables) will provide a precise and orderly method of documenting the analysis independent of the processing method. It will offer the analyst an aid in visualizing the relationships and alternatives of the problem, will provide flexibility in changing any portion of the analysis, and will establish a framework for the complete definition of the systems problem. The CODASYL Systems Group will continue to develop and experiment with these concepts."

To further indicate the potential results from use of tabular form, the following statements paraphrase various user opinions: Clarity and conciseness -- Decision tables are easy to prepare, read, and teach to others; experience shows that non-programmers can learn to prepare satisfactory tables in less than a day. The amount of writing, or number of words, lines and symbols used in describing complex decisions, is reduced by 25-50% as compared to flow charting. For certain specific cases, problem statement and programming time combined have been reduced significantly.

Meaningful Relationships -- Table structure serves to improve systems logic by aligning alternatives side by side. It also sharpens cause and effect understanding, so relationships which are accidental or incidental become clearer. Furthermore, actions based on similar or related conditions are apt to be drawn into the same table, making it easier to appreciate and consider interdependent factors.

Completeness -- Tabular form allows effective visual or deck debugging both by the analyst and the reviewer. There are fewer errors to start with since the analyst tends to catch his own mistakes; moreoever, the reviewer will typically detect a high percentage of the remaining errors by visual examination. Finally, experience shows that with this foundation and suitable test problem construction, it is easy to rapidly detect the balance of the errors during machine debugging.

The evidence quoted on the advantages of decision tables for systems analysis and computer programming is based on actual study projects. Some of these studies even tested decision tables on various data processing machines. There are many current studies which are experimenting with a variety of tabular forms.

A Plan for Action

With all its potential advantages, it is apparent that tabular form has not yet achieved full growth and stature; there are major technical and application areas still unprobed, awaiting only the touch of creativity to make practical breakthroughs. While current table methodology does not yet provide a drawbridge to cross the communications moat surrounding systems engineers, it appears to offer the greatest chance for a significant advance.

To bring these possibilities to fruition requires experimental development. Tabular form will have to be tried and used on a wide variety of applications to provide practical evaluation and determine desirable characteristics. Along with this field pre-testing, there will be a need for effective technical developments to explore new table concepts and structures.

There are many areas which need experimental and technical development:

1. Table structure
   -- multiple successes per table
   -- interspersing conditions and actions
   -- explicit control of sequence of actions
2. Relations among tables
   -- prior rule concepts
   -- use of library functions
   -- use of open and closed subroutines
3. Language considerations
   -- statement construction
   -- macro or jargon operators
   -- machine independence
4. Associated data description
   -- defining factors and expressions for man-to-man and man-to-machine use
   -- conditioned definitions
   -- input/output format
   -- preassigned values and constants
5. Implementation considerations
   -- compiling vs. interpreting
   -- sequential vs. random access to tables
   -- possibility of made-to-order processors
   -- ability to introduce specialized operators and table structures

The explosive innovations in computer hardware have not been matched by corresponding developments in systems communication. But we are on the threshold of a major breakthrough, we are on the verge of a significant advance. It's up to you and it's up to us to show equal creativity in software to that shown in hardware: To use tabular form to develop a clear, concise, meaningful, comprehensive Systems Engineering language.

Tables Signal Better Communication

Talk Given by Burton Grad, Manager IBM Systems Engineering Development

The pilot is preparing to land his single engine plane at the airport; it is late at night and his fuel supply is low. He calls to the radio tower and asks for landing instructions. All he hears in return is a babble in a foreign language which he can't understand.

The executive has spent the last hour of his day dictating an important speech; the next morning he comes in and wants to review the material. His secretary is out ill. The other girls in the office all read Gregg, not Pitman.

A design engineer has carefully prepared a number of complex Boolean equations to explain the operation of a new computer circuit. He shows these to the manufacturing engineer to give an indication of what needs to be constructed. The manufacturing engineer says, "I don't understand Boolean algebra."

We could go on and on citing examples like these of events and occurences where lack of a common language for communication causes difficulties ranging all the way from the most trivial to the deadly. Systems Engineering faces communication barriers as serious as those of any profession. The systems engineer today does not have a language to communicate with management; he does not have a language to communicate with computer programmers; he does not have a language to communicate with functional specialists; he does not even have a language to communicate with other systems engineers.

Programmers who have learned one computer at the machine language level can't understand the programming of another machine at the machine language level without spending the time necessary to learn the second machine's special codes and instructions. For this reason (among others) there has been intensive effort to develop common languages like FORTRAN, Commercial Translator and COBOL which will be applicable to a number of machines. But the communication between programmer and machine is merely a small part of the total problem.

For Systems Engineering it is vital to develop tools and techniques to permit a manager to state his decision criteria and decision rules. We must find a common language so systems engineers can communicate with product engineers, accountants, and manufacturing planners, to find out their decision rules and decision logic; that is critical to determine the characteristics of the system that is going to be modelled or controlled. A method must be found for two-way communication with computer programmers to be sure that the intended decision rules are in fact being executed. A technique is needed to aid the systems engineer in establishing complete decision rules and in predetermining that these rules will accomplish the intended goals.

In the past, this problem has not been as severe. Because of the limited size of business systems problems, we could depend on the programmer to understand the particular problems well enough to be sure the logic was correct and to check the problem out thoroughly. However, as the systems we are trying to solve become larger and more complex, this expedient is no longer satisfactory. Systems engineers must take on the responsibility for designing the decision logic and for insuring that it is being executed properly. To do this systems engineers must have a professional language which will serve for effective intercommunication.

What has caused the communication void? What has caused this communication moat surrounding the systems engineer? There are at least three major factors involved:

1- The inability to clearly and concisely express decision logic and decision rules for describing business systems.

2- The inability to show cause-effect relationship between conditions and actions.

3- The inability to guarantee or even aid in achieving logical completeness in establishing decision rules.

Today, we have available a number of techniques which have been applied to solving the communication problem: we've tried to use narrative, flow charts and even logical equations. But none of these has filled the bill. Each has major drawbacks; the failure of these known techniques has led to consideration of another alternative: decision tables.

Decision Tables

Decision tables are a formal method for describing decision logic in a two-dimensional display. The layout clearly shows the cause and effect relationship between conditions and actions; it explicitly relates decision alternatives.

Decision tables use a format which is familiar to us from analytical, financial, and statistical tables. Since the days of the Babylonians, people have used tables as a means of organizing information where the relationships were complex or the amount of data great. These data tables appear to be superior to many other forms of information organization because:

1- They provide clarity and conciseness through data classification.

2- They clearly show relationship of dependent to independent variables.

3- They explicitly indicate omissions.

Decision tables use tabular format to represent dynamic situations. Where we have used flow charts, narrative, or logical equations to describe decision logic, or an operating procedure, we now find it possible to use decision tables for these same jobs. The argument in favor of tables is their relative convenience and effectiveness, not that they can describe systems that cannot also be described in other ways.

Tabular form has been used by programmers since the earliest days of computers. The most common use of tables has been to relate some function to an argument. Given the value of one factor, the table provides the value of another dependent factor. For example, a table might relate capitals to states (Figure 1). Given the state name, determine the name of the capital.

STATE	Alabama	Alaska		...	Wyoming
CAPITAL	Montgomery	Juneau		...	Cheyenne

In this example State appears above the double line and Capital below; each different state name is in a column and physically below it, the name of the corresponding capital. If the State is Alabama, then the Capital is Montgomery; if the State is Alaska, then the Capital is Juneau.

An extension of this concept is seen in Figure 2 in the use of a matrix to display the value of a particular factor as a function of multiple variables.

AGE \ HEALTH	EXCELLENT	GOOD	FAIR	POOR
>25 <35	1.27	1.62	1.08	2.73
>35 <45	1.83	2.12	2.53	3.42
>45 <55	2.51	2.93	3.47	5.27
>55 <65	3.29	3.91	4.85	8.73
>65	5.21	6.45	7.61	10.97

Insurance premium rates are shown as a function of health and age. In the example, if health is excellent and age is between 25 and 35, then the rate is \$1.27. However, if health is poor and age between 55 and 65, then the rate is \$8.73. Unfortunately, the visual effectiveness of a matrix is reduced when the number of independent variables exceeds two or the number of dependent variables is greater than one.

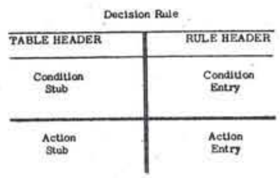
Because of the natural benefits from using tables, it seems that there should be some way to generalize tabular form so that any number of independent and dependent variables might be shown with clear visual correspondence. Figure 3 (on the next page) shows a table with four independent and three

Health	Excellent	Excellent	Poor
Age	25, 35	25, 35	65
Section of Country	East	East	West
Sex	Male	Female	Female
Premium Rate	1.27	1.18	9.82
Policy Limit	200,000	100,000	10,000
Type of Policy	A, B, or C	A, B, or C	R

dependent factors where clarity, interrelationship and comprehensiveness have been maintained.

In this example, the decision table indicates insurance premium rate, policy limit, and type of policy as a function of health, age, section of country, and sex. If the applicant is in excellent health, between 25 and 35 years of age, from the East, and is a male, his rate is $1.27, the insurance limit is $200,000, and he may be issued policy type A, B, or C. All of the alternatives are clearly set forth, one by one, across the table.

To obtain a better understanding of a decision table, let's look at its fundamental elements as shown in Figure 4.



The double lines serve as demarcation: CONDITIONS are shown above the horizontal double lines, ACTIONS below. The STUB is to the left of the vertical double line, ENTRIES to the right. A condition states a relationship. An action states a command.

If all the conditions in a column are satisfied then the actions in that column are executed. Each such vertical combination of conditions and actions is called a RULE. In the same column with the entries for each rule, there may be specialized data relating to that rule; this is called the RULE HEADER. Similarly, each table may have certain specialized information which is called the TABLE HEADER.

Consider another sample table which contains all the same elements, but has some different properties. This table is Figure 5.

TABLE:CREDIT	Rule 1	Rule 2	Rule 3	Rule 4
Credit limit is o.k.	Y	N	N	N
Pay experience is favorable		Y	N	N
Special clearance is obtained			Y	N
Approve order	X	X	X	
Return order to Sales				X

The first rule would be read: <u>If</u> credit limit is OK, <u>then</u> approve order. The second rule would be read: <u>If</u> credit limit is not OK <u>and</u> pay experience is favorable, <u>then</u> approve order. In this LIMITED ENTRY table, the entire condition or action must be written in the stub. The condition entry is limited to indicating whether the corresponding condition should be asserted, negated or ignored; the action entry indicates if the action stub should be executed or ignored.

This is in contrast, as you may note, to the table shown in Figure 3, which is called an EXTENDED ENTRY table. In this case the individual condition or action information extends from the stub into the corresponding entries. In any given table, we can, of course, mix extended and limited entry form, whichever is more convenient for a particular condition or action.

The Use of Decision Tables

To this point sample decision tables and their elements have been discussed to describe concept and structure. Now the application and use of decision tables will be presented. A number of experiments conducted over the past four years have used decision tables on a variety of problems; these will be reviewed briefly.

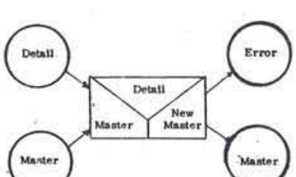
While I was project leader for General Electric's Integrated Systems Project, the potential application of tables to a wide variety of problems was explored including its use for product design, operation planning, cost determination, factory scheduling, etc. The results certainly revealed the opportunity of using decision tables as a major new tool to clarify communication among different technical specialists as well as between these specialists and computer programmers. It was stimulating to watch a manufacturing engineer suddenly grasp product design decision logic and then point out where restraints had been introduced by the product engineer that were of little value to anybody. Through this kind of examination, significant improvements might be made in the total product.

At Sutherland Company, a consulting firm in Peoria, Illinois, management decision rules have been studied with various customers and expressed in tabular form. These decision tables have been applied to Air Force logistics and various commercial situations such as accounts receivable, accounts payable, etc. From all reports, this work has permitted a more effective and comprehensive statement of the current decision logic and provided more meaningful and understandable communication between systems men and programmers.

An area of experimentation already familiar to many of you is the work done at Hunt Foods and Industries by Mr. O. Y. Evans, who is now with IBM. Mr. Evan's work was directed toward communication among different systems men, and from systems men to programmers, concerning the complex decision rules involved in stock control, sales analysis, etc. The results demonstrate that this approach was an effective formal way to state very complex logic without requiring knowledge of Boolean algebra or any other precise mathematical technique.

IBM has been working with several of its customers investigating potential applications of decision tables to a wide variety of problems. From these experiments, it seems clear that decision tables are frequently easier to prepare than comparable programming methods, and that they are an effective aid to systems analysis. In these experiments, communication between systems engineer and programmer has been substantially improved; communication between systems engineer and management has also benefitted from the common description of decision rules.

To convey how tables can be developed, let's follow the process through the significant problem of file maintenance. The block diagram in Figure 6 indicates the essential elements of the problem solution.



A detail file and a master file are the two inputs. The updated master file and an error file are the principal outputs. Within the computer, three basic areas are assigned: master, detail, and new master. The purpose of the update logic is to modify the incoming master file by the detail information to produce an updated master file containing any additions and changes and from which deleted records have been eliminated.

Figure 7 (on the following page) is one of two tables prepared to perform this job.

Rule 1 states the starting condition. At the start, one master record and one detail record are read into the corresponding memory areas. At this point, sequence control returns to the beginning of the table.

Rule 2 and all the following ones are now pertinent. Rule 2 specifically handles the end of job conditions, i.e., end of detail and end of master. In this case, control is transferred to End, a closing routine to provide for sentinels, tape marks, etc.

TABLE: Update Rule #	01	02	03	04	05	06	07	08
Start	Y	N	N	N	N	N	N	Else
End of detail		Y	Y	N	N	N	N	
End of Master		Y	N	Y	N	N	N	
Detail					<Master	>Master	=Master	
Detail an "addition"				Y	Y			
Do Error Routine								x
Move Master to New Master							x	
Move Detail to New Master				x	x			
Set Addition Switch				On	On		Off	
Write Master			x			x		
Read Master	x		x			x		
Read Detail	x			x	x			x
Go to Table	Update	End	Update	Change	Change	Update	Change	Update

Rule 3 describes the situation when the end of detail has been reached, but not the end of master. Since there can be no further changes, additions, or deletions to the original master, the actions are to write the updated master from the master area, read another master, and then return to the beginning of the table.

In Rule 4, the end of master has been found, but not the end of detail; the remaining details should only be additions. Therefore, the information in the detail area is moved to the new master area, the addition switch is set on, a new detail record is read, and control transferred to the Change Table.

Rules 5, 6, and 7 are concerned with cases where neither the detail nor the master file has ended. The identification number in the detail area is compared to the identification number in the master area. Rule 5 considers the event when the detail is less than the master; in this case the detail should be an addition in order to follow the same logic of Rule 4. In Rule 6 the detail is greater than the master; consequently the same logic as Rule 3 applies. Rule 7 covers the case where master and detail are equal. The information in the master area is moved to the new master area, and control is transferred to the Change Table.

The final rule, Rule 8, is the ELSE Situation. When this occurs something has gone wrong, since all legitimate possibilities have already been examined. An error routine is carried out; then another detail record is read. Rule 8 will take care of cases involving sequence errors in the master file and certain types of sequence errors in the detail file (if the out-of-sequence detail is not an addition). It will also take care of any non-matching detail which is not an addition.

The table can be rearranged to aid programming efficiency: columns with higher frequency of success should be moved to the left and those with lower frequency to the right. Rules 1 and 2 would be way over to the right since they occur only once in each program. Depending upon the particular data, Rule 6 (the column where the detail is greater than the master) will probably be the most frequent case and should be the first one considered. One recommended order is: 6, 7, 5, 3, 4, 1, 2, 8.

Another concept for improving program efficiency is to rearrange the conditions to present the most discriminating condition at the top and the least discriminating at the bottom. For example, the start condition, which is shown first, probably should be last since this only distinguishes one case out of all the thousands that will occur. A similar statement can be made about end of detail and end of master. It seems evident that the comparison of detail to master would be the most discriminating criteria and therefore placed first in the table.

The Case for Tabular Form

Look once more at Figure 7 and compare its statement of the update decision logic with that in the following narrative. Which is clearer and more concise, which shows cause-effect relationships better, which aids more in determining logical completeness.

Mr. T. F. Kavanagh speaking at the 1960 Eastern Joint Computer Conference had this to say: "the decision... table is a fundamental language concept... broadly applicable to many classes of information processing and decision making problems... tables force a step-by-step analysis of the decision... are easily understood by humans regardless of their functional background (they are) simple and straightforward (enough) that... specialists can write tables... with very little training... tables are easy to maintain (and) errors are reported at the source language level."

Mr. O. Y. Evans states of his work on tabular techniques: "The tabular approach... aids... in visualizing the numerous relationships and alternatives... (and) permits data rules to be readily reviewed for omissions and inconsistencies... (in addition it) provides flexibility in changing any portion of the analysis."

The CODASYL Systems Group, part of The Development Committee of the Conference on Data System Languages, has been looking into the use of decision tables. In a recent release the following statement was made: "Investigation... indicates that the systems analysis method discussed above (decision tables) will provide a precise and orderly method of documenting the analysis independent of the processing method. It will offer the analyst an aid in visualizing the relationships and alternatives of the problem, will provide flexibility in changing any portion of the analysis, and will establish a framework for the complete definition of the systems problem. The CODASYL Systems Group will continue to develop and experiment with these concepts."

To further indicate the potential results from use of tabular form, the following statements paraphrase various user opinions: Clarity and conciseness -- Decision tables are easy to prepare, read, and teach to others; experience shows that non-programmers can learn to prepare satisfactory tables in less than a day. The amount of writing, or number of words, lines and symbols used in describing complex decisions, is reduced by 25-50% as compared to flow charting. For certain specific cases, problem statement and programming time combined have been reduced significantly.

Meaningful Relationships -- Table structure serves to improve systems logic by aligning alternatives side by side. It also sharpens cause and effect understanding, so relationships which are accidental or incidental become clearer. Furthermore, actions based on similar or related conditions are apt to be drawn into the same table, making it easier to appreciate and consider interdependent factors.

Completeness -- Tabular form allows effective visual or deck debugging both by the analyst and the reviewer. There are fewer errors to start with since the analyst tends to catch his own mistakes; moreoever, the reviewer will typically detect a high percentage of the remaining errors by visual examination. Finally, experience shows that with this foundation and suitable test problem construction, it is easy to rapidly detect the balance of the errors during machine debugging.

The evidence quoted on the advantages of decision tables for systems analysis and computer programming is based on actual study projects. Some of these studies even tested decision tables on various data processing machines. There are many current studies which are experimenting with a variety of tabular forms.

A Plan for Action

With all its potential advantages, it is apparent that tabular form has not yet achieved full growth and stature; there are major technical and application areas still unprobed, awaiting only the touch of creativity to make practical breakthroughs. While current table methodology does not yet provide a drawbridge to cross the communications moat surrounding systems engineers, it appears to offer the greatest chance for a significant advance.

To bring these possibilities to fruition requires experimental development. Tabular form will have to be tried and used on a wide variety of applications to provide practical evaluation and determine desirable characteristics. Along with this field pre-testing, there will be a need for effective technical developments to explore new table concepts and structures.

There are many areas which need experimental and technical development:

1. Table structure
   -- multiple successes per table
   -- interspersing conditions and actions
   -- explicit control of sequence of actions
2. Relations among tables
   -- prior rule concepts
   -- use of library functions
   -- use of open and closed subroutines
3. Language considerations
   -- statement construction
   -- macro or jargon operators
   -- machine independence
4. Associated data description
   -- defining factors and expressions for man-to-man and man-to-machine use
   -- conditioned definitions
   -- input/output format
   -- preassigned values and constants
5. Implementation considerations
   -- compiling vs. interpreting
   -- sequential vs. random access to tables
   -- possibility of made-to-order processors
   -- ability to introduce specialized operators and table structures

The explosive innovations in computer hardware have not been matched by corresponding developments in systems communication. But we are on the threshold of a major breakthrough, we are on the verge of a significant advance. It's up to you and it's up to us to show equal creativity in software to that shown in hardware: To use tabular form to develop a clear, concise, meaningful, comprehensive Systems Engineering language.