GENERALIZED INFORMATION RETRIEVAL
LANGUAGE AND SYSTEM
(GIRLS)

USER REQUIREMENTS SPECIFICATION

DONALD B. NELSON

March 19, 1965

TABLE OF CONTENTS

1. INTRODUCTION

The generalized information retrieval language and system defined in this specification is designed for a communication network of many remote stations and a single computer. The design of the system is based on several basic assumptions and externally imposed conditions.

1. A keyboard input and a print output is assumed for each remote station.

2. The definition of "generalized" imposed on the system requires the design to be independent of specific equipment and to entail a minimum allotment in the computer's fast access memory.

3. The definition of "generalized" imposed on the system also requires the design to accommodate a maximum variety of information retrieval inputs from many different application areas.

4. The definition of "language" imposed on the system requires the accommodation of inputs stated in a natural language limited by a minimum of system rules and restrictions.

5. Such a "natural language" requirement is assumed literally, and not merely as the antonym of "machine language."

6. The imposed definition of "information retrieval system" specifies it as a sub-system of a network EXECUTIVE SYSTEM for all remote station inputs; and, therefore, all inputs to the "information retrieval system" arbitrarily are to be EXECUTIVE SYSTEM outputs, and EXECUTIVE SYSTEM inputs unrelated to information retrieval arbitrarily are excluded from the IR system.

7. The definition of "system" is assumed to specify a single integrated system design, and not a fragmentation of the problem into separate sub-system designs requiring interface solutions for subsequent assembly.

The apparent contradiction in the two definitions of "generalized" can be resolved by the creation of a "procedure word structure", a basic design technique in digital computer definition. The analysis of each procedure as information, and each process as a sequence of operational modes, then will permit the implementation of a maximum system capability with a minimum allotment in the computer's fast access memory. This dynamic information and the static information to be stored by the system will be assigned a common format and common rules for data storage and retrieval.

A general application module and rules for its use are defined in Section 2., with discussions of the IR language, the RSI format and four major processors. The description of this module with general suitability for use in an IR network of remote stations, and by many application areas, places particular emphasis on user requirements. The operational system requirements are defined in the System Operational Requirements Specification and include all the requirements for the IR preprocessor and any special requirements for each of the four processors.

Note:

The precise details for remote communication with the EXECUTIVE SYSTEM, especially from the point of view of administrative information (see section 2.2.2) will be published separately but included as an attachment to this report.

2. DEFINITION OF A GENERAL APPLICATION MODULE AND RULES FOR ITS USE

2.1 Definition of the Language for Remote Station Inputs (RSI)

2.1.1 Introduction

The primary intent of this system design for remote station inputs will be (1) to accommodate a maximum variety of information retrieval inputs from many different application areas, and (2) to accommodate inputs stated in a natural language limited by a minimum of system rules and restrictions. In combining natural language with computer processes, however, the necessity for some language rules and restrictions will be unavoidable, since there are many differences in the basic characteristics of natural language and those of numbers. Wherever possible, therefore, resolution of these interface differences will be defined as computer requirements rather than as language restrictions.

The specification of a truly natural language with ambiguities permitted in word meanings and grammatical context would require an assignment of processor capacity and operating time incompatible with the conceptual definitions listed for the system in Section 1. Therefore, the IR language to be specified for this system will be a simulated natural language.

2.1.2 Conceptual Development

All IR information in the computer will be stored as items of information within data lists, with all the items for a single type of information stored together as one data list, and with many data lists stored together as all the IR information. The storage location for each item within a data list will be recorded in a special data list index, and the item itself will be stored without its DATA LIST I.D. Each IR item in storage then will contain only its own ITEM I.D. followed by information data, but its DATA LIST I.D. will be implied by its location in storage. The retrieval then of any IR item in storage will require both its ITEM I.D. and its DATA LIST I.D.

The information data stored with an ITEM I.D. will contain one or more types of data, and each type of data will be stored together with its own identification code. To create a terminology for this information data, each type of data will be defined as an ATTRIBUTE, and each attribute will be defined as containing one ATTRIBUTE I.D. together with one or more ATTRIBUTE VALUES.

Any IR entry will be defined, therefore, as one DATA LIST I.D. and one ITEM I.D. followed by one or more ATTRIBUTES, with each ATTRIBUTE I.D. followed by one or more ATTRIBUTE VALUES. This definition is shown as a format for two IR entries in the following illustration.

DATA LIST I.D.	ITEM I.D.	ATTRIBUTE I.D.	ATTRIBUTE VALUE
ASSY	2A62483	Name	Amplifier
		Drawing Number	1230
			6780
		Release Date	Dec. 7, 1964
		Next Higher Assembly	2A62776
			3B82415
		Next Lower Assembly	1W99230
			2A78236
ASSY	2A64286	Name	TRANSISTOR 263
		Release Date	Aug. 8, 1963

Therefore, each IR datum will be a DATA LIST I.D., an ITEM I.D., an ATTRIBUTE I.D. or an ATTRIBUTE VALUE. Also when the first two I.D.'s are considered together as an ENTRY I.D., the implied subdivisions of an IR entry will relate IR data to the general definition for all information.

ENTRY I.D.		ATTRIBUTE	
DATA LIST I.D.	ITEM I.D.	I.D.	VALUE

Similarly, when each ATTRIBUTE VALUE is considered as requiring three I.D.'s, then the implied subdivisions will relate IR data to the storage and retrieval requirements.

IDENTIFICATION			ATTRIBUTE VALUE
DATA LIST	ITEM	ATTRIBUTE	

Further, each of the two subdivisions in the example relating IR data to the general definition for all information can be considered individually as representing an information I.D. and an attribute.

I.D.	ATTRIBUTE		I.D.	ATTRIBUTE
DATA LIST I.D.	ITEM I.D.		ATTRIBUTE I.D.	ATTRIBUTE VALUE

Consideration of the data in this form and as a sequence of two units of information is a concept of particular consequence, since it permits the automatic construction of extended and complex data formats by the correlation of data in two or more basic formats. For example, extended sequential orders of attributes can be accommodated in a directional sequence of basic formats by defining the attribute value of one data entry for automatic correlation with the item I.D. of another data entry.
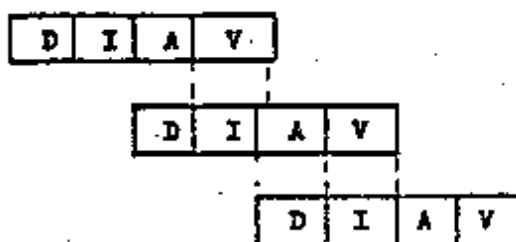
Similarly, the automatic cross-indexing of data entries and many other automatic correlations can be specified for data stored in the basic format defined for the system.

Statement of the IR data requirements in a natural language for remote station inputs will require the use of additional words and symbols to connect and interrelate the IR data words. All of these additional words, then, will be defined as CONNECTIVES, and the use of such connectives is illustrated in the following example of an RSI entry.

(LIST) EACH ASSEMBLY > 2A64398 WITH THE NAME AMPLIFIER

Connectives	EACH, > , WITH, THE
Data List I.D.	ASSEMBLY
Item I.D.	2A64398
Attribute I.D.	NAME
Attribute Value	AMPLIFIER

Considering (LIST) as an example of the program name to be associated with the data in each RSI, then each IR element in an RSI will be a PROGRAM I.D., a CONNECTIVE, a DATA LIST I.D., an ITEM I.D., an ATTRIBUTE I.D. and/or an ATTRIBUTE VALUE.

Therefore, the IR language and format must accommodate the foregoing data requirements in all RSI requests for the retrieval of data or the updating

of stored data. Also, the IR language and format must permit the IR preprocessor
to identify each element exactly and automatically, for translation into the
machine language and format defined for the computer processors.

Exact meanings, then, are required by the computer, and such exact
meanings in natural language often rely on both pattern recognition and context
for identification. Pattern recognition, however, is relatively difficult for
a computer, and such requirements for the IR preprocessor must be restricted;
but context identification requirements can be assigned to the preprocessor
with more freedom. For context identification of the different parts of an RSI,
then, by the EXECUTIVE SYSTEM, the RSI format will be defined as having two
basic sections; one for the administrative information, and the other for a
designated program name with its data.

For a program external to the IR section of the computer, an RSI must
include all the data to be processed by the "call up" program. Such data will
be subject to the format and scaling requirements of the particular program
requested and its "data audit" routines; and, therefore, the EXECUTIVE SYSTEM
will process only the administrative and program information in the RSI; and,
after character translation, transfer the given data intact. Similarly, if the
RSI program were to specify an IR program, the EXECUTIVE SYSTEM would transfer
the RSI data information section to the IR preprocessor for whatever procedures
may be defined by the system design.

Those inputs specifying information retrieval processes will require
access to one or more of the IR DATA LISTS; and the preprocessor, in such a
context, automatically will initiate the defined IR procedures. The definition
of rules for special programs, then, will be an initial requirement for
instituting such programs, and the rules for each program will be identified by
the preprocessor in a "program index" to be stored in the computer. Thus, any
RSI format and vocabulary requirements can be accommodated and the use of special
data lists also can be accommodated.

The data requirements to be accommodated by the IR language already have
been defined and exampled. In the RSI data section, then, the language must
accommodate names, numbers and alphanumeric data as well as the program name and
the connectives. The definition of language context rules, therefore, must
include consideration of the preprocessor capabilities and any contradiction of
preprocessor procedures which may be required by the several IR processes.
Another consideration in defining the IR language will be the prohibition of
words, numbers or symbols to many of the language elements by an exclusive
assignment to one element. The definition of context, then, will require
consideration of many variables, and the following examples illustrate some of
the many identification problems to be resolved by the IR language and preproces-
sor definitions.

    1)   IF THE RATING OF ASSY BX263 IS LESS THAN 14, LIST THE DRAWING NO.
        FOR EACH NEXT LOWER ASSY IN ASSY BX263 WITH RATING C14 AND EACH
        ASSY NO. FOR DRAWING NO. 2063.

2) IS TRANSISTOR OR THE NAME OF PART NO. 2X36; AND IF THE REPLY IS
   YES, THEN LIST EACH ASSY > ASSY BX962 WITH SMITH AND JONES AS
   VENDORS.

These examples illustrate not only the many interpretations possible for
one name or number, but also the very limited potential of the connectives as
a basis for context definition. Although a few connectives will have limited
or special definitions, restrictions in the natural use of connectives will be
considered undersirable. The previous examples illustrate also the LINEAL or
PROSE FORMAT to be accommodated by the IR language and preprocessor.

The prohibition of data to each element in the IR language will require
an enforcement; and, therefore, particular consideration of such prohibitions
will be imperative in context definitions. The data prohibitions for
ATTRIBUTE VALUES and for ITEM I.D.'s will require the greater consideration, but
any data prohibitions for any elements will be considered as an undesirable
restriction.

Since generality is a requirement in the definition of the IR language,
special context identification will not be based on the upper and lower case
alphabets available only on special keyboards. Such a limitation would contra-
dict the potential extension of the RSI system to accommodate external
customers, and such an extension will be considered as a requirement for the IR
language and preprocessor. Therefore, the DATA LIST I.D. will be defined as one
or more words; and any facility, project or customer identification will be
defined as the initial word or words in a DATA LIST I.D. For example,

            HOUSTON PURCHASE ORDER
            LEM DRAWING
            GDA LEM DRAWING

Each DATA LIST I.D., therefore, will require the preprocessor to identify not
only a particular data list, but also any access security codes associated with
it. Similarly, any format definitions for a data reliability audit of each
ITEM I.D. and ATTRIBUTE VALUE also will be associated with a DATA LIST I.D.
or an ATTRIBUTE I.D. and will require identification by the IR preprocessor.

The many concepts and parameters developed in this section, then, define
the IR language requirements; and the following list of six language elements
will require accommodation in any IR data section.

            PROGRAM I.D.
            CONNECTIVES
            DATA LIST I.D.
            ITEM I.D.
            ATTRIBUTE I.D.
            ATTRIBUTE VALUE

Since the preprocessor will be designated for identification of these elements,
the specification of the language and rules for its use will determine, in turn,
many of the requirements for the preprocessor.

2.1.3 Definition and Rules of the IR Language

This specification of the IR language is based on the concepts and parameters developed in Section 2.1.2. Particular consideration is given to generality, both in the language and in the system capability, and to a minimum of language rules and restrictions. Therefore, this language specification is interrelated with the many of the requirements for the IR preprocessor, as well as the RSI format.

The primary purpose of the language, however, is the retrieval of information from one or more data lists from remote stations; and both the stored information and its retrieval format will determine many of the words and rules for the IR language. Relative to natural language elements, these words can be considered as NOUNS; and this specification of the IR language, then, is interrelated particularly with the data list word and format definitions.

Therefore, assuming this interdependence between the RSI language, the preprocessor and the data list requirements, the following 5 rules are defined for an IR data section with one request.

RULE 1.   Only defined words are to be used as connectives and as program, data list and attribute I.D.'s.

RULE 2.   The program identification is to precede all other information.

RULE 3.   Any information not attributed to the next previous item is to be introduced by a data list I.D.

RULE 4.   Any attribute information not associated with the next previous attribute I.D. is to be introduced by another attribute I.D.

RULE 5.   Each attribute value is to be enclosed by quotation marks, and other quotation marks are not to be used.

These 5 language rules accommodate the present system requirements. However, to insure generality in this language specification, 2 additional rules will be defined for an IR data section extended to accommodate multiple requests.

RULE 6.   Each request subsequent to the first is to be preceded by a defined mark.

RULE 7.   Only defined words may be used to identify the output of one request as data in a subsequent request.

This future extension of system capabilities, then, can be accommodated by the present IR language.

Under RULE 1, only defined words are to be used as connectives and as program, data list and attribute I.D.'s. The words to be used as connectives are defined and discussed in the last part of this section, and the program I.D.'s will include several processor codes such as (GIRL), (GUPD), (COUZ) and (FILE). Also, program I.D.'s will include several words such as LIST, IS, COUNT, ADD and DELETE; and under the relevant processor section, all such words for use with a processor code are defined and illustrated. For example,

(GIRL) IS "SMITH" THE VENDOR FOR ASSY 2864

(GUPD) DELETE THE RATING IN PART XB12 AND XB26

IR data section rules for each of these 4 processor codes will be defined and discussed separately in Sections 2.3., 2.4., 2.5 and 2.6. The data list and attribute I.D. words will be defined by each application area; and they will include such data list names and mnemonic codes as ASSY, SUPPLIER, P/N and P.O., and such attribute names as PART NO., NAME, N.H. ASSY, RATING and STATUS.

Under RULES 6 and 7, the word IF as a program I.D., and the words THEN, IS and ARE as connectives, are reserved by definition for future use in multiple and interrelated requests.

The IR language RULE 1 is necessary to insure exact identification of each element by the preprocessor, and each word or words defined as an element will be stored in various dictionary listings for preprocessor inspection. The requirements for identifying new program or data list words are detailed under (FILE) in Section 2.6, but they can be summarized in two general requirements.

1) Words defined for connectives, programs and data lists are to be mutually exclusive.

2) An attribute I.D. cannot be identical to any data list I.D.

Each of the IR language elements, except connectives and attribute values, may require preprocessor identification of "n" words in a sequence. Although an attribute value also may contain several words, the quotation marks enclosing each attribute value will identify it as a single word. Each connective is defined as a single word, but any relevant combinations with other connectives, or with other IR elements, may be used. Any IR element, then, can be expressed as the following equation.

$$\text{Any IR element} = P + D + I + A + V + C$$

Also, identification of relevant dependencies can be expressed with subscripts; for example, N.H. ASSY as an attribute I.D. in the ASSY data list can be

expressed as $A_{D_{ASSY}}$ or as $(A1,A2,A3)_{D_{ASSY}}$. This relational terminology is used in the following equations to illustrate the IR language RULES 2, 3, 4 and 5.

Since any C is a unique word and any P must precede all other information (i.e., RULE 2), these elements can be identified exactly by the preprocessor. Thus, the unidentified elements in the former equation are reduced.

$$\text{a unidentified element} = D_a + I_{D_b} + A_{D_c} + V_{A_{D_d}}$$

RULE 3 insures the identification of any attributes or item as being interrelated in a data list I.D. context and RULE 4 defines a context for the assignment of each attribute value to its attribute I.D. The previous equation, then, can be restated.

$$\text{a unidentified element} = D_a + I_{D_a} + A_{D_a} \, V_{A_{D_a}}$$

RULE 5 permits preprocessor identification of each attribute value, and the equation can be reduced again.

$$\text{a unidentified element} = A_a + I_{D_a} + A_{D_a}$$

and finally, words of these three elements can be differentiated by sequence context and word definitions, subsequent to the identification of each attribute value under RULE 5. The following example illustrates several of the IR language rules, and particularly illustrates RULE 5.

(GIRL) LIST THE DRAWING REFERENCE 3B AND NAME FOR ASSY 28

INTENDED		P1	P2	C	$A_1 1_{D_1}$		$A_1 2_{D_1}$	$A_1 3_{D_1}$	C	A_{2D_1}	C	D_1	I
INCORRECT BUT POSSIBLE WITHOUT RULE 5.		P1	P2	C	D_1		A_{1D_1}	V_{1A_1}	C	A_{2D_1}	C	D_2	I

RULE 5, then, can be considered necessary for 3 reasons;
(1) attribute values often must include the literal use of other IR
elements; (2) differentiation between IR elements by the preprocessor requires
exact and unambiguous identifications; and (3) alternate rules include any
data prohibitions for attribute values, rather than the single prohibition
of internal quotation marks.

Additional examples of the IR language rules are illustrated with
the IR elements identified, and with the I.D.'s of 2 arbitrarily defined
data lists.

DATA LIST I.D.:	P/N	NCMR
ATTRIBUTE I.D.'s:	DATE	DATE
	QUANTITY	PART NO.
	STATUS	SUPPLIER NO.
	SUPPLIER NO.	MJO/SO NO.
	NCMR NO.	FR NO.
	DATE RECVD.	CAR NO.

(GIRL) LIST EACH NCMR FOR PART NO. "123" WITH THE DATE >"DEC.1,1964"

IR Elements: P1   P2   C D_1 C $A_1^1{}_{D_1} A_1^2{}_{D_1}$ V_{1A_1} C   C $A_2{}_{D_1}$ C V_{1A_2}

(GUPD) ADD SUPPLIER NO. "682" AND "721" IN P/N 1264X AND NCMR 6543

IR Elements: P1   P2 $A_1^1{}_{D_1}$ $A_1^2{}_{D_1}$ V_{1A_1} C V_{2A_1} C D_1 $I_1{}_{D_1}$ C D_2 $I_1{}_{D_2}$

(GIRL) LIST EACH NCMR WITH MJO/SO NO. "263" FOR P/N 1265X

IR Elements: P1   P2   C D_1 C $A_1^1{}_{D_1} A_1^2{}_{D_1}$ V_{1A_1} C D_2 $I_1{}_{D_2}$

(GIRL) IS THE STATUS OF P/N 1266X "H"

IR Elements: P1 P2 C $A_1{}_{D_1}$ C D_1 $I_1{}_{D_1}$ V_{1A_1}

(GUPD) DELETE "263" AS THE SUPPLIER NO. IN EACH P/N > 1262X AND > 126⁻

IR Elements: P1   P2 V_{1A_1} C   C $A_1^1{}_{D_1}$ $A_1^2{}_{D_1}$ C   C D_1 C $I_1{}_{D_1}$ C   C I_2

Each IR connective is defined as a single word, but these single words
may be used in any relevant combinations. The following list of defined IR
connectives include 5 symbols also to be used a single words. Therefore, each
connective symbol is to be isolated between blank spaces; and, except for the
2 bracket symbols, this use of blank spaces will be natural and will avoid
additional data prohibitions as well as special recognition procedures in the
RSI preprocessor. The 2 exact meanings for "and," "or" and "in" are defined in
demonostrations following this listing of 35 IR connectives.

>	FOR	AND (sequential)
<	OF	ANDD (logical)
=	WITH	OR (inclusive)
[	EACH	ONLY (one and only)
]	EVERY	FROM (not <)
NOT	ANY	TO (=)
EQUAL	AS	ON (=)
GREATER THAN	THE	AFTER (>)
LESS THAN	A	SINCE (>)
IN	AN	BEFORE (<)
INN (for vertical searches and can precede only a data list I.D.)	HAVING (with)	

Also, the following words are defined as IR connectives and reserved for future use in multiple requests.

THEN

IS (=)

ARE (=)

Special connectives can be defined for any special programs, such as CALCULATE and LOGICALLY REDUCE, as well as special procedures; and, under present system requirements, each such special program is to be identified by a special processor code, such as (CALC).

The IR connective ONLY is to be used with its natural meaning, and when identified by the preprocessor, will be automatically interpreted as "one and only." Therefore, the preprocessor will replace ONLY by EQUALS and insert a second entry below the first with the connective ANDD followed by a zero or blank in the data element.

As connectives in the IR language, "and," "or" and "in" each have two exact meanings, and each is defined in 2 forms, i.e., AND, ANDD, OR, ORR, IN and INN. All of these except ORR are defined as IR connectives, and the definitions of these five connectives are illustrated in the following two demonstrations.

Assuming a single quality or quantity "A," with everything else being "NOT A," the definition of "everything" can be stated as a diagram.
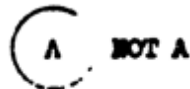
By also assuming the 2 symbols "1" and "0" as another 2 value system for expressing "everything," the above diagram can be restated as a TRUTH TABLE (a table of all possible values).

A	
0	(NOT A)
1	(A)

Similarly, assuming the TWO qualities or quantities "A" and "B," the definition of "everything" can be stated both as a diagram and as a truth table.
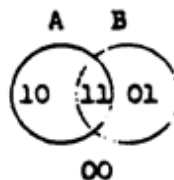
A	B
0	0
0	1
1	0
1	1

If "A and B" are to be considered one at a time and in sequence, the COLUMNS of the truth table will be relevant; and the IR connective with this SEQUENTIAL definition will be "AND."

However, if "A and B" are to be considered together and at the same time, the ROWS of the truth table will be relevant; and the IR connective with this COINCIDENT (or LOGICAL) definition will be "ANDD."

If "A or B" is to be considered as "either A or B or both together," the IR connective with this INCLUSIVE definition will be "OR"; but if "A or B" is to be considered as "either A or B but NOT both together," the IR connective with this EXCLUSIVE definition will be "ORR."

2-11

The 2 connectives with natural spelling are commonly used in natural language, and the 2 with an extra letter are not.

The definitions for these 4 IR connectives are summarized in the following illustration.

A	B
0	0
0	1
1	0
1	1

ORR
ANDD
OR
AND

DEMONSTRATION NO. 2

In an IR data section, there also will be 2 definitions and 2 spellings (IN and INN) for the word "in." However, use of the connective "INN" will be restricted by definition to "vertical" (or "Christmas Tree") searches of data entries within a DATA LIST; and, therefore, the connective INN can precede only a data list I.D. A vertical search is possible only in a data list which permits the ITEM I.D. of one entry to be also the ATTRIBUTE VALUE of another entry. For example, an entry in a data list with ASSEMBLY as the data list I.D. may include NEXT HIGHER ASSEMBLIES or NEXT LOWER ASSEMBLIES as attribute I.D.'s; and the attribute value for either I.D. would be an ASSEMBLY number and, therefore, also included as an item I.D. elsewhere in the same data list. This vertical interrelationship of information in a data list is diagrammed in the following illustration.
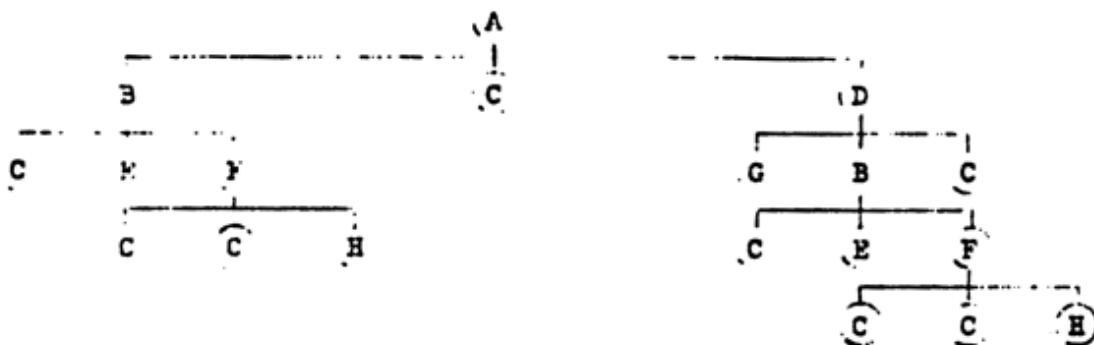
If the entries for such a data list represented QULITIES, C added to C would = C and the entry for F would be:

F	NEXT HIGHER	B.
	NEXT LOWER	C, H

However, if the entries for such a data list represented QUANTITIES, C added to C would = 2C and the entry for F would be:

F	NEXT HIGHER	B
	NH/QUANTITY	2
	NEXT LOWER	C, H
	NL/QUANTITY	2, 1

With both the "next higher" and "next lower" quantities, vertical search procedures could be defined for either the "next higher" attributes (an ASCENDING vertical search) or the "next lower" attributes (a DESCENDING vertical search).

However, in an ASSEMBLY data list, only the "next lower" quantities are defined. Therefore, assuming the example diagram as representing assemblies, the data list entries would be:

ASSY A	N.L.ASSY	B, C, D
	QUANTITY	1, 1, 1
ASSY B	N.H.ASSY	A, D
	N.L.ASSY	C, E, F
	QUANTITY	1, 1, 1
ASSY C	N.H.ASSY	A, B, D, F
ASSY D	N.H.ASSY	A
	N.L.ASSY	G, B, C
	QUANTITY	1, 1, 1
ASSY E	N.H.ASSY	B
ASSY F	N.H.ASSY	B
	N.L.ASSY	C, H
	QUANTITY	2, 1
ASSY G	N.H.ASSY	D
ASSY H	N.H.ASSY	F

Considering only this list of data entries, such a question as "How many C's in A?" would be answered naturally as 1. However, with a vertical search down from the entry A, the answer would be 8.

Therefore, since both meanings must be accommodated in an RSI, the RI language connective "IN" will be defined for natural use; and "INN" will specify any type of vertical search previously defined for a data list and stored in a special index.

Restating the example question in the IR language and data section format, then, the RSI data section for "IRN" would be:

(GIRL) COUNT THE QUANTITY OF N.L.ASSY "C" IN ASSY A

and the RSI data section for "IN" would be:

(GIRL) LIST THE QUANTITY OF N.L.ASSY "C" IN ASSY A

2.1.4   Conclusion

The IR language is restricted and does have rules, since the inherent differences between natural language and the computer data list language make some restrictions and rules necessary. The data list retrieval requirements are not compatible in practice with heuristic identifications of ambiguous words and grammatical constructions; and, therefore, only a simulated natural language is possible in this system design without an impractically large language processor. The integrated definition of the IR language, format and preprocessor, however, has permitted many of the interface requirements to be defined for the preprocessor, with a minimum of rules and restrictions defined for the language. Also, the 35 IR connectives provide considerable flexibility and naturalness to the language; and particular freedom of word sequence in the initial part of each IR data section simulates natural language practices.

The generality of the language and format is evidenced by the accommodation of either statements or questions; and, for future extension of the system, any of the RSI programs can be sequenced and interrelated in one RSI data section. Generality also is evidenced by the accommodation of any external customers, and the potential of a remote network of such customers is considered desireable. The conceptual definitions and parameters developed in Section 2.1.2., then, all are accommodated in this IR language specification.

2.2 Definition of the RSI Format and Rules for Its Use

2.2.1 Introduction

The RSI format and rules for its use will permit the automatic identification of the user and the various operations and processes required for one or more groups of data. Two basic sections are defined for the RSI format: the first is the ADMINISTRATIVE SECTION to include information and codes for general identification of the RSI; and the second is the DATA SECTION to include the identification of any program and processor together with the relevant data information. Each RSI includes one administrative section followed by one data section, and each data section potentially can include one or more requests. The following specification of the RSI format, then, includes the definition for each of these two sections.

2.2.2 The Administration Section

The administrative section will be defined to include the identification necessary for remote outputs, processing priority, accounting charges and information security. Each of these identifications will be defined as a name, number or code in an exact form. The following list includes the definition of each of these numbers or codes assigned to the administrative section.

1) REMOTE STATION EXTENSION. 5 digits defined by the telephone extension number of the input station, and required for call up of the input station.

2) NAME. Alphabetic characters defined by the name of the person initiating the RSI.

3) ORGANIZATION CODE. 6 digit code required for report records and accounting charges.

4) CHARGE NUMBER. 6 character code required for accounting charges.

5) REQUEST PRIORITY. 1 character code.

6) BUILDING. 2 characters defined by the sender's office address, and required for delivery of the output by either the remote station or CDRC.

7) ROOM. 4 digits defined by the sender's office address (see 6.).

8) EXTENSION. 5 digits defined by the sender's own telephone extension number (see 6.).

9) OUTPUT EQUIPMENT CODE. 3 character code to be defined as the remote or CDRC equipment selected by the sender, and required in the output address.

10) REMOTE STATION EXTENSION. 5 digits defined by the telephone
extension number of the output station selected by the sender and
required in the output address. For programs external to the IR
section of the computer, this five digit number will be stored as
the "Alternate Extension" entry in the Job Information Block.

11) ATTEND. 2 characters to be defined for the sender to request prior
notification of any remote or CDRC output.

12) SECURITY CODES. 5 characters to be defined for each person
authorized to initiate an RSI, and for any restricted program,
IR data list or IR data list ATTRIBUTE, and required for access.

Among these twelve administrative identifications, six will be mandatory
in IR requests; i.e., 1 through 4 for input identification, and 9 and 10 for
the output address. Each of the other six will be defined for either omission
or inclusion; and the format for these identifications will be defined as
lineal with the sequence of identifications as in the above list, and with a
blank space following each identification. Except in the NAME, no blank spaces
will occur within an identification; and, therefore, the RSI preprocessor will
recognize any one identification by its sequence and the type or number of its
characters. Any number of consecutive security codes will be recognized by the
preprocessor, and the preprocessor will compare any security requirements
encountered in the retrieval of data with these security codes.

2.2.3 The Program and Data Section

When the Administrative Section is completed, a new line will be started
for the Program and Data Section; and the mechanical operation codes in the
equipment to be used will be assumed as transmitting a "NEW LINE" code, or an
"END OF BLOCK" code at the end of the old line to permit efficiency in the
"LONGITUDINAL REDUNDANCY CHECK." Any arbitrary mark, however, could be defined
to mark this new line.

The IR data section then is to be typed in a lineal (prose) format, line
after line, until the end of the data is reached. Then the "END OF TRANSMISSION"
or "END OF DATA" code will be required to identify the data being at an end.

Under the present system requirements, each IR data section will be
introduced by a processor code, such as (GIRL), (GUPD), (GOUT) and (FILE).
The formats for each of these processors are defined and discussed separately
in Sections 2.3, 2.4, 2.5 and 2.6; and several examples are included for each
processor.

2.2.4

Since the RSI format and language both are interrelated closely in
definition, and since particular consideration was given the simplicity and
naturalness of the IR language, the RSI format and the rules for its use are
extremely simple.

The definition of the lineal format natural to prose text, and each data section starting with a processor code, left no unique requirement for an "END OF LINE" function code, other than to occur between the Administrative section and the Data section. Therefore, minimum rules and restrictions have been defined for the use of the RSI format Data section.

The Administrative section has many rules and regulations, but this data is considered to be essential to permit identification and the reference information for establishing JOB TABLE entries either in the RSI program section or in the external part of the computer.

The IR preprocessor requirements for each processor are discussed in the next 4 sections (2.3, 2.4, 2.5 and 2.6), and many of the solutions for the interface between natural language and computer retrieval of stored data have been assigned to the processors as well as the preprocessors.

2.3 (GIRL)

(GIRL) identifies the processor to be used for information retrieval. Other processors are defined for updating stored information, for special outputs and for initiating new data lists; and these other processors are defined and discussed in Sections 2.4, 2.5 and 2.6. The following words are defined as program I.D.'s for use in (GIRL) remote station inputs:

COUNT

LIST

IS

ARE

DICT. LIST

Except for (GIRL) DICT. LIST requests, the language and format rules for a (GIRL) RSI are the same as those defined for the IR language in Section 2.1.3. and for the format in Section 2.2. Each RSI data section is to be introduced by the processor code (GIRL) followed by one of the program I.D. word definitions in the above list, and the separate context use of IS and ARE as connectives is not contradicted by this definition of the 2 words as program I.D.'s. An RSI to the (GIRL) processor, then, can be either imperative using COUNT and LIST, or a question using IS and ARE. For example:

(GIRL) LIST EACH NCMR AND DATE FROM "NOV. 1, 1964" TO "DEC. 1, 1964" FOR PART NO. "12345" AND SUPPLIER NO. "75439"

(GIRL) IS "JAN. 10, 1965" THE DATE RECVD. FOR P/N X6238

(GIRL) IS "SMITH" THE ONLY VENDOR FOR ASSY X2836

(GIRL) COUNT THE TRANSISTORS IXN ASSY X2836

(GIRL) LIST EACH NCMR AND DATE FOR P/N 12345 ANDD SUPPLIER 75439

Any connectives which are irrelevant will be identified as such in the dictionary, and special connectives may require special procedures either in the preprocessor or in the processor.

The output format for the (GIRL) processor will be defined by the given elements in the processor format, and data retrieval "bridges" will not be included in the output column headings. In the previous example, then, the output format would be:

SUPPLIER	P/N	NCMR	DATE
75439	12345	6282	Dec. 6, 1964
		6288	Jan. 4, 1965

If the program I.D. words are (GIRL) IS or (GIRL) ARE, however, the output will be based on a comparison of the retrieval data and the given data. If the data agrees, the output format will be preceded by YES; but if the data does not agree, the output format will be preceded by NO.

The language and format rules for a (GIRL) DICT. LIST entry are the same as for other (GIRL) requests, except the noun vocabulary for data list, item and attribute I.D.'s is defined by the preprocessor dictionary data lists. This vocabulary is defined in the following table.

Program I.D.	Data List I.D.	Item I.D.	Attribute I.D.
(GIRL) DICT. LIST	IR/DICT. ----ATTR. GOUT/DICT.	(IR data list I.D.'s) (IR attribute I.D.'s)	CONVERSION CORRELATIVE SIZE/DL SIZE/ITEM C/TYPE C/MIN. C/MAX. C/PATTERN IR/SC UPD/SC

Except for the different vocabulary for data list, item and attribute I.D.'s, then, the language and format rules for a (GIRL) DICT. LIST request are the same as those defined for the other (GIRL) inputs. For example,

        (GIRL) DICT. LIST THE CORRELATIVE OF NCMR NO. IN THE IR/DICT.
        FOR P/N ATTR.

In the IR data list attributes above, the security codes are protected from retrieval and unauthorized updating, since they are to be stored with "no-print" procedure codes. Each of the several attribute I.D.'s listed above are discussed and defined in Section 2.6.

The (GIRL) processor requirements, then, will include vertical search procedures, counting, cumulative counting, list making, comparison of lists and individual values as well as data retrieval. These several requirements, however, in addition to the many preprocessor requirements, permit a minimum of rules and restrictions in the (GIRL) language and format.

2.4 (GUPD)

     (GUPD) identifies the processor to be used for updating information
stored in any data list, and the following words are defined as program
I.D.'s for use with the processor code (GUPD).

          ADD
          DELETE
          CHANGE (associated with the connective TO)

          DICT. { ADD
                   DELETE
                   CHANGE

Except for (GUPD) DICT. requests, all language rules for a (GUPD) request
are the same as those defined for the IR language in Section 2.1.3., and
the (GUPD) format rules are the same as those defined in Section 2.2.
Therefore, except for the defined program I.D. words, (GIRL) and (GUPD)
have identical language and format requirements. The (GUPD) RSI data section,
then, is illustrated in the following examples:

          (GUPD) ADD NCMR 6543 WITH DATA "JAN. 8, 1965"
                 PART NO. "1265X" SUPPLIER NO. "682"
                 AND MJO/SO NO. "263"

          (GUPD) DELETE "H" AS STATUS IN P/N 1268X

          (GUPD) CHANGE STATUS OF P/N 1268X TO "H"

          (GUPD) CHANGE EACH DASH NO. GREATER THAN "12"
                 TO "10" IN P/ASSY 12468 12469 AND 13016

          (GUPD) DELETE P/N 1268X

     The preprocessor dictionaries, however, are to be updated by (GUPD)
DICT. requests; and this dictionary information requires a different vocabulary
for the data list, item and attribute I.D.'s. This vocabulary is the same
as that defined in the previous section for a (GIRL) DICT. LIST input.
Except for a different noun vocabulary, however, the language and format
rules for the (GUPD) DICT. requests are the same as those defined for the
other (GUPD) inputs, for example:

     (GUPD) DICT. CHANGE THE IR/SC OF SALARY TO "16908" IN EMPL/NO.
            ATTR. IN THE IR/DICT.

2.5 (GOUT)

(GOUT) identifies the RSI processor to be used for requesting an output with a special format, and the following words are defined as program I.D.'s for use in (GOUT) requests:

LIST
FORMAT
PRINT

(GOUT) PRINT may not be used under the present system requirements, but it is defined as an extended capability for possible future accommodation of remote requests for special reports to be created from GOUT procedures stored as information data lists.

Since the initial system implementation is not to include the storage of procedural information for special outputs, each (GOUT) RSI for the present is required to furnish all relevant information both for data retrieval and for format procedures. Each (GOUT) RSI data section, then, is required to include 2 parts; the first part, identified by (GOUT) LIST, for the data retrieval input: and the second part, identified by (GOUT) FORMAT, for the output format procedures.

The language and format requirements for the (GOUT) LIST part are identical to those for a (GIRL) LIST entry, and this first part of a (GOUT) RSI data section is to identify all information to be retrieved from data storage for use by the (GOUT) FORMAT procedures. Therefore, the (GOUT) LIST noun vocabulary for data list and attribute I.D.'s is identical to the IR language vocabulary, and is defined by the terms listed in the preprocessor dictionary of IR data lists. For example, assume the following IR data lists and the special output headings.

Data List I.D.'s :   PURCHASE ORDER or P/O   COMMODITY CODE or C/C

Attribute I.D.'s :   DATE   NAME
                       CUSTOMER   STD. HRS./$100
                       COMM. CODE
                       QUANTITY
                       VALUE
                       SCHED. DEL.

Title: WORKLOAD FORECAST BY WEEK

Column Headings: SCHEDULED DEL. DATE
                       COMMODITY CODE
                       QUANTITY SCHEDULED
                       MANPOWER REQUIREMENTS

In this example, then, the (GOUT) LIST part of the input data section might be:

(GOUT) LIST THE QUANTITY AND VALUE FOR EACH P/O WITH SCHED. DEL. FROM "NOV. 30, 1964" AND BEFORE "DEC. 25, 1964" ANDD WITH COMM. CODE > "1199" AND < "1400" AND THE C/C STD. HRS./$100

A data list "bridge" between the two data lists is assumed as being the correlative value "B, C/C" listed with the attribute I.D. COMM. CODE in the IR dictionary. Such a data list correlative is defined and exampled under (FILE) in Section 2.6.

The language and format requirements for the (GOUT) FORMAT part, however, differ from those for the IR language, since connectives are not to be used and the noun vocabulary for (GOUT) FORMAT is defined differently. The definition of this special vocabulary for (GOUT) FORMAT is based on the specifications of stored GOUT data lists for future retrieval by (GOUT) PRINT requests. These future data lists will be indexed in a preprocessor GOUT dictionary by the number or mnemonic code of each report, and each GOUT data list will use item and attribute I.D.'s from defined common lists. For example, the vocabulary nouns relevant to (GOUT) FORMAT are defined in the following lists:

Data List I.D.:   XX

Item I.D.'s: TITLE
                    COL/1. . . n
                    TP/1 . . . n

Attribute I.D.'s:

                    CORRELATIVE
                    HEADING
                    SORT
                    GROUP 1 START
                    GROUP 2 START

The item I.D. TP/1. . . n is the mnemonic code for TERMINAL PROCEDURE/1. . . n, and it identifies calculation procedures which are to be performed using the completed GOUT columns of data or the column totals. For example,

TP/1	CORRELATIVE	"F1, T8" "F2,T9" "F = F1/F2"
	HEADING	"OVERALL DELIVERY MEAN"
TP/2	CORRELATIVE	"F1, C7" "F = MEDIAN F1"
	HEADING	"% LATE MEDIAN"

The CORRELATIVE attribute value is to include any relevant values for three types of interrelational procedures. The first type of GOUT correlative defines the IR data to be copied as column data, and this type of correlative is identified by the code letter R followed by the relevant attribute and data list I.D.'s. For example,

| COL/2 | CORRELATIVE | "R, COMM. CODE, P/O" |
| | HEADING | "COMMODITY CODE" |

The second type of COUT correlative specifies totals for a column, and this type is defined by the code letter T. For example,

    COL/3   CORRELATIVE   "B, QUANTITY, P/O" "T"
             HEADING   "QUANTITY SCHEDULED"

The third type of COUT correlative defines the function for calculating the data of a column, and is identified by the code letter F used to specify both the function and the IR data variables. For example,

    COL/4   CORRELATIVE   "F1, VALUE, P/O" "F2, STD.
                                  HRS./$100, C/C" "F = F1/F2" "T"

             HEADING   "MANPOWER REQUIREMENTS"

      The SORT attribute value defines the data format for the ascending sort procedures, and is identified by the code letters, D, An or Bn. D defines a numerical sorting with the decimal point on the right of the least significant sorting digit. An defines an alphabetic sorting with the nth character counted from the left being the most significant letter. Similarly, Bn defines a numerical sorting with the nth character counted from the right being the least significant sorting digit. Alphanumeric values, then, can be defined with either numeric or alphabetic sorting procedures. No accommodation for descending sort procedures is defined for COUT column data.

      The GROUP 1 START and GROUP 2 START attribute values define the data value grouping within a column of data. The difference between the 2 values will be calculated by the COUT processor and used as a constant increment for grouping all relevant data. For example,

    COL/1   HEADING   "SCHEDULED DEL. DATE"
             GROUP 1 START   "NOV. 30, 1964"
             GROUP 2 START   "DEC. 7, 1964"

Assuming data for this example, the grouping of the column data then might be:

                    SCHEDULED
                    DEL. DATE

                    DEC. 3, 1964
                    DEC. 2, 1964
                    DEC. 5, 1964
                    DEC. 3, 1964
                    DEC. 4, 1965

                    DEC. 12, 1964
                    DEC. 9, 1964
                    DEC. 10, 1964
                    DEC. 8, 1964

                    DEC. 15, 1964
                    DEC. 20, 1964

The COUT processor automatically will total the data for each column with
a correlative value T after the column procedures are completed, and before the
TP/1. . . n procedures are started. Also, all columns with a correlative value
T automatically will be sub-totalled at the end of any column grouping. The
previous example, then, would have the following control breaks.

```
                              SCHEDULED
                              DEL. DATE

                              DEC. 3, 1964
                              DEC. 2, 1964
                              DEC. 5, 1964
                              DEC. 3, 1964
                              DEC. 4, 1964

              TOTAL             .

                              DEC. 12, 1964
                              DEC.  9, 1964
                              DEC. 10, 1964
                              DEC.  8, 1964

              TOTAL

                              DEC. 15, 1964
                              DEC. 20, 1964

              TOTAL
         GRAND  TOTAL
```

The COUT processor automatically is to specify the horizontal tab and
line feed for each (COUT) RSI, except in the future processing of (COUT)
PRINT requests using stored COUT data lists which include these specifications
as attribute values. Also, the COUT processor automatically is to determine
the type of output equipment specified in the RSI Administrative Section;
and, for equipment with page control, the COUT processor is to include on each
page of the output a PAGE NO., the REPORT NO. and COLUMN HEADINGS and, on the
last line of each page except the last page, the entry "CONTINUED ON NEXT PAGE".

Therefore, considering the example assumed earlier in this section, the
(COUT) FORMAT part of the RSI might be:

```
(COUT) FORMAT XX TITLE HEADING "WORKLOAD FORECAST BY WEEK" COL/1
       CORRELATIVE "R, SCHED. DEL., P/O" HEADING "SCHEDULED DEL. DATE"
       GROUP 1 START "NOV. 30, 1964" GROUP 2 START "DEC. 7, 1964"
       COL/2 CORRELATIVE "R, COMM. CODE, P/O" HEADING "COMMODITY CODE"
       SORT "N1" GROUP 1 START "1200" GROUP 2 START "1300" COL/3
       CORRELATIVE "R, QUANTITY, P/O" "T" HEADING "QUANTITY SCHEDULED"
       COL/4 CORRELATIVE "F1, VALUE, P/O" "F2, STD. HRS./$100, C/C"
       "F = F1/F2" "T" HEADING "MANPOWER REQUIREMENTS"
```

The output format, then, for this (GOUT) LIST and (GOUT) FORMAT example would be:

SPECIAL REPORT
WORKLOAD FORECAST BY WEEK

	SCHEDULED DEL. DATE	COMMODITY CODE	QUANTITY SCHEDULED	MANPOWER REQUIREMENTS
	Dec. 5, 1964	1234	300	12.0
	Dec. 3, 1964	1265	700	42.0
	Dec. 4, 1964	1291	100	3.2
SUB-TOTAL			1100	57.2
	Dec. 3, 1964	1306	10000	7.0
	Dec. 2, 1964	1309	2000	4.0
SUB-TOTAL			12000	11.0
TOTAL			13100	68.2
	Dec. 9, 1964	1234	200	5.0
	Dec.10, 1964	1239	1200	7.8
SUB-TOTAL			1400	12.8
	Dec.12, 1964	1310	5000	37.2
	Dec. 8, 1964	1362	10000	20.2
SUB-TOTAL			15000	57.4
TOTAL			16400	70.2
	Dec.20, 1964	1315	1000	6.8
	Dec.15, 1964	1339	4000	2.5
SUB-TOTAL			5000	9.3
TOTAL			5000	9.3
GRAND TOTAL			34500	147.7

The initial system implementation, however, will not permit the grouping of column data, terminal procedures or the use of column data and column totals in correlative functions; and these capabilities are defined only to permit possible future extension of the system. Under the present system, then, the (GOUT) FORMAT vocabulary is limited to the following list.

Data List I.D.:   XX

Item I.D.'s: TITLE
                    COL/1 . . . n

Attribute I.D.'s: CORRELATIVE
                    HEADING
                    SORT

Therefore, in the previous illustration, the present system capabilities would limit the (GOUT) RSI data section to being, for example,

(GOUT) LIST THE QUANTITY AND VALUE FOR EACH P/O WITH SCHED. DEL. FROM "NOV. 30, 1964" AND BEFORE "DEC. 7, 1964" ANDD WITH COMM CODE > "1199" AND < "1400" AND THE C/C STD. HRS./$100 (GOUT) FORMAT XX TITLE HEADING "WORKLOAD FORECAST FOR WEEK ENDING DEC. 7, 1964" COL/1 CORRELATIVE "R,COMM. CODE, P/O" HEADING "COMMODITY CODE" SORT "R1" COL/2 CORRELATIVE "R,QUANTITY, P/O" "T" HEADING "QUANTITY SCHEDULED" COL/3 CORRELATIVE "F1,VALUE, P/O" "F2,STD. HRS./$100, C/C" "T = F1/F2" "T" HEADING "MANPOWER REQUIREMENTS"

The corresponding output format then would be:

SPECIAL REPORT

WORKLOAD FORECAST FOR WEEK ENDING DEC. 7, 1964

COMMODITY CODE	QUANTITY SCHEDULED	MANPOWER REQUIREMENTS
1234	300	12.0
1265	700	42.0
1291	100	3.2
1306	10000	7.0
1309	2000	4.0
GRAND TOTAL	13100	68.2

The availability of current data with use of (GOUT) remote station inputs is intended to eliminate the need for complex reports and periodic bulk report printings. However, the language and format requirements for the (GOUT) processor have been defined to include these capabilities for possible extension of the present system to accommodate any transitional requirements. The use of stored GOUT data lists, however, may be desirable for frequently used GOUT requests; and extension of the present system specifications to include (GOUT) PRINT inputs would permit retrieval and updating of both the present (GOUT) LIST information and the (GOUT) FORMAT procedures. For example, assuming the last previous example above were store in GOUT data list form under the data list I.D. FW-3, the entire (GOUT) RSI data section then could be:

(GOUT) PRINT F4-3 WITH P/O SCHED. DEL. FROM "NOV. 30, 1964" AND
        BEFORE "DEC. 7, 1964" COMM. CODE > "1199" AND < "1400" AND
        TITLE HEADING "WORKLOAD FORECAST FOR WEEK ENDING DEC. 7, 1964"

The (GOUT) processor then would interpret the (GOUT) PRINT input as updating
corrections to the information stored in the data list FW-3, and the processor
would correct the retrieval and output formats to correspond with the input
corrections. Therefore, the language and format rules defined for the (GOUT)
processor will accommodate system extensions either for the future convenience
of the user or for transitional report requirements.

2-6   (FILE)

      (FILE) identifies the RSI processor to be used for initiating new data
lists; and the following words are defined as program I.D.'s for use in (FILE)
requests.

DICT.
DATA

      For the initiation of a new data list, each (FILE) data section is to
have 2 parts: the first part is to be (FILE) DICT., and the second part is
to be (FILE) DATA. The first part is required to update the vocabulary in
the preprocessor dictionary to include the new data list I.D. and the attribute
I.D.'s, and to include for each I.D. any special security codes, any data format
audit codes and any data correlation codes. Also, with the IR data list I.D.,
the estimated size of the new data list is required to permit the efficient
assignment of disk storage location for the new data list by the (FILE) proc-
essor. The second part is to provide the data for storage in the data list
form. Although a (FILE) RSI data section may include either (FILE) DICT. or
(FILE) DATA as a separate entry, the (FILE) DATA input cannot be processed
unless it has been preceded by the associated (FILE) DICT. input.

      The (FILE) DATA information is to be stored sequentially before input,
and the (FILE) processor is to store the information directly and without
sorting procedures. Also, the (FILE) processor does not create cross-index
data lists interrelated with existing data lists, and the user is responsible
for the reliability of the (FILE) DATA information. However, any information
of uncertain reliability can be entered separately into the new data list as a
subsequent (GUPD) ADD input, since the (GUPD) processor does create data entrie
required by defined interrelationships between existing data lists. These data
list interrelationships are defined as correlatives and are discussed later in
this section.

      The language and format requirements for a (FILE) input differ from
those for a (GIRL) or (GUPD) request, and (FILE) DICT. and (FILE) DATA each
have strictly defined forms and uses. The following vocabulary is defined
for each of these 2 parts of a (FILE) input.

Program I.D.	Data list I.D.	Item I.D.	Attribute I.D.
(FILE) DICT.	IR/DICT. ---ATTR. GOUT/DICT.	(IR data list I.D.'s) (IR attribute I.D.'s)	CONVERSION CORRELATIVE SIZE/DL SIZE/ITEM C/TYPE C/MIN. C/MAX. C/PATTERN IR/SC UPD/SC
(FILE) DATA	(IR data list I.D.'s)	(IR item I.D.'s)	(IR attribute I.D.'s)

As discussed in Section 2.5., the GOUT/DICT. is not included in the
present system requirements; and, therefore, definition of its associated
vocabulary and further discussion of this type of (FILE) input is not included
in this section.

The data list I.D.'s and attribute I.D.'s for a new IR data list are to
be defined under the following rules:

1. All words defined for IR connectives, and all program, data
    list and item I.D.'s are to be mutually exclusive.

2. All words defined for IR connectives, and all program, item
    and attribute I.D.'s are to be mutually exclusive.

3. An attribute I.D. cannot be identical to any data list I.D.
    or to any data list I.D. in a sequence with one of its
    attribute I.D.'s.

These rules can be summarized and expressed as equations.

1. $D \neq I \neq P \neq C$

2. $A \neq I \neq P \neq C$

3. $A_{D_x} \neq D + D_y A_{D_y} + A_{D_y} D_y$

Although the format and language defined for (FILE) DATA is very
similar to those for (GIRL) and (GUPD), the (FILE) DATA language does not
include connectives and does have strict rules of sequence. All item I.D.'s
for the new data list are to be in an ordered sequence, each item I.D. is to
precede its associated information, and each attribute I.D. is to precede any
relevant attribute values. These same rules of sequence also are defined for
(FILE) DICT.

The SIZE/DL and SIZE/ITEM values are relevant only to the new data list I.D.'s, with the SIZE/DL being the estimated number of items in a new data list and the SIZE/ITEM being the estimated number of characters in an average item. For example,

IR/DICT.	P/N	SIZE/DL	"1500"
		SIZE/ITEM	"25"
IR/DICT.	NOMR	SIZE/DL	"1000"
		SIZE/ITEM	"30"

Separate security codes may be assigned for information retrieval requests, i.e., (GIRL) and (GOUT), and for information updating, i.e., (GUPD), to any data list I.D. or any attribute I.D. Each such security code is to be included as a value for either IR/SC or UPD/SC. For example,

TR/DICT.	EMPL/NO.	UPD/SC	"30649"
EMPL/NO. /TTR.	SALARY	IR/SC	"4026B"
		UPD/SC	"30682"

Any security code assigned to a data list I.D. is to be effective also for any attribute I.D. within the data list, but individual security codes also may be assigned additionally to any attribute I.D.

Any format audit required by data reliability of an item I.D. or an attribute value is to be listed in the IR/DICT. under the associated data list I.D. or attribute I.D. Any such format audit is limited to one or more of the character specifications defined by C/TYPE, C/MIN., C/MAX., and C/PATTERN. The C/TYPE value is to be the letter code A for alphabetic, N for numeric or AN for alphanumeric. The C/MIN. value is to be the minimum number of characters defined for a format audit; and, similarly, the C/MAX. value is to be any defined maximum number of characters. The C/PATTERN value is to be the pattern sequence of alphabetic, numeric and symbol characters defined for a format audit; and the pattern automatically will be justified to the right except with an associated C/TYPE "A". For example, assume each item I.D. for a new data list I.D. ACSY is to be alphanumeric, is to have exactly 8 characters, and is to have an exact pattern of 5 digits followed by a hyphen and 2 letters. This format audit, then would be defined by the following entries.

IR/DICT.	ACSY	C/TYPE	"AN"
		C/MIN.	"8"
		C/MAX.	"8"
		C/PATTERN	"NNNNN-AA"

The CONVERSION value is relevant only to attribute I.D.'s with input values specified by the user for conversion to numeric form to permit arithmetic comparison procedures. For example, if an attribute I.D. value which is a calendar date is to be compared as being less than or greater than other calendar dates, it must be converted from DEC. 3 1964 to 641203 to permit either comparison or sequencing with other calendar dates. The specification of any such conversion is to be defined by the user in the IR/DICT. and the CONVERSION value is to be the code letter defined for each type of data format change, and the letter D is defined for the conversion of calendar dates to number form. For example,

P/N ATTR.   DATE   CONVERSION "D"

The CORRELATIVE values are to include any interrelationships specified
by the user for the automatic updating of cross-indexed data list and attribute
I.D.'s, for the creation of retrieval "bridges" between data lists by the
preprocessor, for calculating an attribute value as a function of other
stored data, for coupling the multiple values of 2 attribute I.D.'s, for
vertical cross-indexing within a data list and for the elimination of redundant
data storage requirements. Any of nine types of data correlation codes may
be defined for a data list I.D. or an attribute I.D., and these nine types
are defined in the following table.

DEFINITION	AUTOMATIC CORRELATION		CODE LETTER	TYPE
	IR	UPD		
An I.D. with data which also is stored under another I.D.	no	no	B	Data list "bridge"
	no	yes	X	Cross-index
	yes	yes	Y	Directional data chain
An I.D. with data which is stored ONLY under another I.D.	yes	no	R	Reference address
	yes	yes	S	Reference address
An I.D. with each consecutive datum correlative with a datum under another I.D.	no	yes	C	Coupled values of different I.D.'s
	yes	yes	D	
Vertically interrelated I.D.'s within a data list	only with INH	yes	V	Vertical cross-index
An I.D. with data which is to be calculated as a defined function of data stored under one or more other I.D.'s	yes	no	F	Function for data synthesis

Each CORRELATIVE value is to include a code letter; and, except for
V and F, each code letter is to be followed either by the data list I.D.
or by the attribute I.D. and the data list I.D. The code letter V is to
be used alone, and the code letter F is to specify both the function and
the IR data variables.

EXAMPLE 1)   Assume a new data list P/ASSY which is to have vertical
interrelationships, and which also is to have coupled
values for N.L. ASSY and NL/QUANTITY. Retrieval of N.L.
ASSY data is independent of the NL/QUANTITY; and,
therefore, the N.L. ASSY is assigned the code letter "C."
The retrieval of NL/QUANTITY data, however, is dependent

for meaning on the retrieval of the N. L. ASSY data; and, therefore, the NL/QUANTITY is assigned the code letter "D".

```
P/ASSY          (V)
    NAME
    DWG. NO.
    N. H. ASSY (V)
    N. L. ASSY (V)(C,NL/QUANTITY,P/ASSY)
    NL/QUANTITY (D,N. L. ASSY,P/ASSY)
```

Therefore, the relevant (FILE) DICT. entries can be expressed figuratively as:

```
IR/DICT.        P/ASSY        CORRELATIVE    "V"
P/ASSY ATTR.    NAME
                DWG. NO.
                N.H.ASSY      CORRELATIVE    "V"
                N.L.ASSY      CORRELATIVE    "V" "C,NL/QUANTITY, P/ASSY"
                NL/QUANTITY   CORRELATIVE    "D,N.L.ASSY,P/ASSY"
```

EXAMPLE 2) Assume a new data list is to have two names, PART and P/N. Therefore, the relevant (FILE) DICT. entries can be expressed figuratively as:

```
IR/DICT.        PART          CORRELATIVE    "B,P/N"
                P/N
```

In this example, all the data then is stored in the P/N data list.

EXAMPLE 3) Assume a new data list is to be interrelated with another data list in IR requests and output formats, but the interrelationship between the data lists is to be automatic only when required as a "bridge" between apparently unrelated I.D.'s.

```
P/N                         NCMR
    NAME                        DATE
    NCMR NO.(B,NCMR)            SUPPLIER NO.
```

In this example, the attribute I.D. NCMR NO. also may be considered as the data list I.D. NCMR to permit automatic correlation of apparently unrelated I.D.'s in an IR entry such as:

(GIRL) LIST THE SUPPLIER NO. FOR P/N 2366

The relevant (FILE) DICT. entry can be expressed figuratively as:

P/N ATTR. NCMR NO. CORRELATIVE "B,NCMR"

EXAMPLE 4.) Assume a new data list is to include a reference to a value
stored in another data list.

```
P/ASSY                              DWG.
    NAME                                TITLE
    DWG. NO. (B, DWG.)                  2A-REF.
    REF. (R, 3B-REF., DWG.)             3B-REF.
```

Therefore, the relevant (FILE) DICT. entries can be expressed
figuratively as:

| P/ASSY ATTR. | DWG. NO. | CORRELATIVE | "B, DWG." |
| | REF. | CORRELATIVE | "R, 3B-REF., DWG." |

EXAMPLE 5) Assume a new data list is to be cross-indexed for IR requests
and output formats by either name or number.

```
EMPL./NO. (X,NUMBER,EMPL/NAME)    EMPL/NAME (X,NAME,EMPL/NO.)
    DEPT.                             NUMBER (X,EMPL/NO.)
    NAME (X,EMPL/NAME)
```

In this example, the cross-index also describes a double
bi-directional chaining of information I.D.'s. The relevant
(FILE) DICT. entries can be expressed figuratively as:

IR/DICT.	EMPL/NO.	CORRELATIVE	"X,NUMBER,EMPL/NAME"
	NAME	CORRELATIVE	"X,NAME,EMPL/NO."
EMPL/NO. ATTR.	NAME	CORRELATIVE	"X,EMPL/NAME"
EMPL/NAME ATTR.	NUMBER	CORRELATIVE	"X,EMPL/NO."

EXAMPLE 6) Assume a new data list is to be interrelated and interdependent
with another data list in a directional chaining of information.

```
P/N                               SUPPLIER (Y,SUPPLIER NO.,P/N)
    NAME                              NAME
    COST                             ADDRESS
    SUPPLIER NO.(Y,SUPPLIER)          RATING
```

In this example, the single bi-directional chaining of I.D.'s
can be considered as linear-indexing in contrast to cross-indexing.
The relevant (FILE) DICT. entries can be expressed figuratively
as:

| P/N ATTR. | SUPPLIER NO. | CORRELATIVE | "Y,SUPPLIER" |
| IR/DICT. | SUPPLIER | CORRELATIVE | "Y,SUPPLIER NO.,P/N" |

EXAMPLE 7) Assume a new data list is to include an attribute I.D. with
a retrieval value to be calculated from other stored data.

```
P/N NAME
    COST
    WEIGHT
    $/LB. (F1,COST,P/N)(F2,WEIGHT,P/N)(F = F1/F2)
```

Therefore, the relevant (FILE) DICT. entry can be expressed figuratively as:

P/N ATTR. $/LB. CORRELATIVE "F1,COST,P/N"
                               "F2,WEIGHT,P/N"
                               "7 = F1/F2"

Therefore, by combining these individual definitions, an example can be constructed for a (FILE) input initiating a new data list. Assume the following new data list, attributes and correlatives with existing data lists.

PART or P/N

    DATE
    QUANTITY
    NCMR NO. (B, NCMR)
    STATUS

Also, assume the item I.D. format is to be at least 7 alphanumeric characters with the first 4 characters from the right being numeric, the estimated number of items as being 2000 with an average item size of 25 characters, the attribute value for DATE is to be converted to number form for comparison processes, there are to be no security codes for IR and any (GUFD) input for the data list is to have the security code 13609. The (FILE) input, then, would be a standard Administrative Section followed by the (FILE) data section which can be expressed figuratively as:

(FILE) DICT. IR/DICT. PART			CORRELATIVE	"S,P/N"
			SIZE/DL	"O"
	P/N		C/TYPE	"AN"
			C/MIN	"7"
			C/PATTERN	"NNNN"
			SIZE/DL	"2000"
			SIZE/ITEM	"25"
			UFD/SC	"13609"
	P/N ATTR.	DATE	CONVERSION	"D"
		QUANTITY		
		NCMR NO.	CORRELATIVE	"B,NCMR"
		STATUS		
(FILE) DATA P/N		12345	DATE	"JAN. 7, 1965"
			QUANTITY	"20"
			NCMR NO.	"3604"
			STATUS	"H"
		12346	DATE	"JAN. 8, 1965"
			QUANTITY	"18"
			STATUS	"A"
		12348		

(etc. to the end of the last item I.D.)

This figurative arrangement of the (FILE) data section then can be

expressed directly in the lineal format defined for the (FILE) RSI data section.

> (FILE) DICT. IR/DICT. PART CORRELATIVE "S,P/N" SIZE/DL
> "0" P/N C/TYPE "AN" C/MIN. "7" C/PATTERN "NNNN"
> SIZE/DL "2000" SIZE/ITEM "25" UFD/SC "13609" P/N
> ATTR. DATE CONVERSION "D" QUANTITY NCMR NO.
> CORRELATIVE "B,NCMR" STATUS (FILE) DATA P/N
> 12345 DATE "JAN. 7, 1965" QUANTITY "20" NCMR NO.
> "3604" STATUS "R" 12346 DATE "JAN. 8, 1965" QUANTITY
> "18" STATUS "A" 12348 ...

This lineal format defined for the (FILE) RSI data section is suitable for a remote or a computer input using a sequence of punched cards, a punched tape or a magnetic tape; and, of course, the lineal format also permits the use of a remote typewritten input for initiating new data lists with very few items. Also, for convenience in large volume inputs certain additional compressions are possible to minimize the (FILE) input language.

As in the other parts of this system design, emphasis is given to a minimum of rules and restrictions on both the language and the format for initiating a new data list; and, wherever possible, interface requirements have been assigned to the computer rather than to the user. The complex security safeguards, for example, are to be enforced by the computer, as well as the requirements for word recognition and a complex audit of the vocabulary in a new data list and the existing dictionary words. The correlatives, however, cannot be defined except by the user; and the 9 defined correlative types permit great flexibility in cross-referencing data for retrieval, for updating or for output formats. With relatively simple language and format requirements, then, a new data list can be initiated with very complex interrelationships to existing data and with automatic data security.

3.   DEFINITION OF USER REQUIREMENTS FOR INITIAL IMPLEMENTATION
       OF THE SYSTEM

3.1. Introduction

       This section defines the user requirements and any changes in
system capabilities for the implementation of a limited version of
the IR system. Limitation of the system as defined in Section 2. is
required both by equipment constraints and to expedite an early
implementation schedule. This necessary limitation, however, has
been confined to a few deletions of language and processor capabilities
and these deletions are (1) the freedom of word order in the initial
part of an input and, therefore, (2) the interrogative form for
retrieval requests, (3) many of the words defined as language
connectives, (4) the storage of special output requests and (5) some
of the possible definitions for the automatic conversion and calculatio.
of data. Any of these deleted capabilities, however, can be restored
to the system in the future as supplemental procedures and specificatic

3.2. Data Format Rules

       The data requirements for the initial implementation system are
identical to those defined in Section 2.1.2. All data, then, is
required to be alphanumeric character information in data list form;
and the data list form is defined as one DATA LIST I.D. followed by
one or more ITEM I.D.'s, with each ITEM I.D. followed by one or more
ATTRIBUTE I.D.'s, and with each ATTRIBUTE I.D. followed by one or more
ATTRIBUTE VALUES. This definition of the data list form can be
exampled figuratively as:

```
        D    I    A    V
                       V
                  A    V
                  .    V
                       V
                  A    V
             I    A    V
             I    A    V
                       V
                  A    V
                       V
             I    A    V
                       V
            (et cetera)
```

All information in the IR system is to be stored in data list form;
and, therefore, the associated DICTIONARY of data list I.D.'s and
attribute I.D.'s, which is to be stored in the computer, also will
be in data list form.

This data list form, together with the use of the computer dictionary, permits inputs to be stated directly in the technical terminology natural to each application area; and, therefore, the user is not required to translate his own terminology into an artificial vocabulary common to all system inputs. This accommodation of the user, however, requires the initiation of each new data list to include dictionary inputs as well as the new data; and these inputs are defined and exampled under (FILE) in Section 2.6.

The defined data list form, together with the use of the computer dictionary, also permits the automatic correlation of interrelated data. By including one or more CORRELATIVE values in the dictionary inputs, the user can specify very complex data interrelationships for automatic execution. These CORRELATIVE values also are defined and exampled under (FILE) in Section 2.6.

3.3. Input and Language Format Rules

Each IR system input is defined with 2 basic sections; an ADMINISTRATIVE SECTION which is defined elsewhere under the EXECUTIVE system, and a DATA SECTION which is defined in this section for the initial implementation system.

The LINEAL or PROSE FORMAT used in natural language is defined for all IR system requests. The computer recognition procedures will identify any blank space as the interval between 2 words, and two or more consecutive blank spaces will be identified as only one interval. The LINEAL format, therefore, can accommodate both punched card inputs and tabulated typing inputs.

Four of the six IR language elements are defined by the data format and these four elements are the DATA LIST I.D., ITEM I.D., ATTRIBUTE I.D and ATTRIBUTE VALUE. The input format, however, requires two additional language elements. The PROGRAM I.D. is defined as the identification of any computer process; and CONNECTIVES are defined either as relations operators, or as extra words included for language naturalness but without significance to the computer. The six elements defined for the IR language, then, are:

PROGRAM I.D.
CONNECTIVES
DATA LIST I.D.
ITEM I.D.
ATTRIBUTE I.D.
ATTRIBUTE VALUE

The following language format rules are defined for these 6 languag elements and for the data section of any IR system input.

RULE 1. Only defined words are to be used as connectives and as program, data list and attribute I.D.'s.

3-2

RULE 2. The program I.D. is to precede all other information.

RULE 3. A data list I.D. is to precede each item I.D. and any other associated information.

RULE 4. An attribute I.D. is to precede any associated attribute values.

RULE 5. Each attribute value is to be enclosed by quotation marks, and other quotation marks are not to be used.

Any IR language element, except a connective, may be defined as one or more words; and more than one identification may be defined for any data list or attribute I.D. Each connective is defined as a single word, but these single words may be used in any relevant combinations; and a connective may precede any IR language element except a program I.D.

RULES 1, 2 and 5 are identical to those defined in Section 2.1.3., but RULES 3 and 4 are redefined in this section to delete the freedom of word order in the initial part of an input. This deletion is required by equipment constraints and to expedite an early implementation schedul Initial implementation requirements also restrict each input data sectio to a single system request; and any additional requests with the same program I.D. are to be divided, and each is to be considered as a separate input.

3.4. Language Vocabulary Table and Rules

In this section, six rules are defined for the IR language vocabulary of the initial implementation system; and these 6 vocabulary rules are summarized in the following VOCABULARY TABLE.

3-3

VOCABULARY TABLE

Program I.D.	Data List I.D.	Item I.D.	Attribute I.D.	Attribute Value
(GIRL) DICT. LIST (GIRL) DICT. COUNT (GUPD) DICT. ADD (GUPD) DICT. DELETE (GUPD) DICT. CHANGE (FILE) DICT.	IR/DICT. --- ATTR.	(IR data list I.D.'s) (IR attribute I.D.'s)	CONVERSION CORRELATIVE SIZE/DL SIZE/ITEM C/TYPE C/MIN. C/MAX. C/PATTERN IR/SC UPD/SC	D (B,X,Y,R,S,C,D, V and F codes) (numeric value) (numeric value) M, A, N, AN (numeric value) (numeric value) (code) (code) (code)
(GOUT) FORMAT	SPECIAL	TITLE COL/1...n	CORRELATIVE HEADING SORT	T and (R and F codes) (text) D, An, Nn
(GIRL) LIST (GIRL) COUNT (GUPD) ADD (GUPD) DELETE (GUPD) CHANGE (GOUT) LIST (FILE) DATA	(IR data list I.D.'s)	(IR item I.D.'s)	(IR attribute I.D.'s)	(IR attribute values)

CONNECTIVES	
Relational	Extra Words
> < = NOT ANDD TO EACH IRN	AND OR WITH IN FOR OF THE

RULE 1. For each data list initiated by the user, the
DATA LIST I.D. and all ITEM I.D.'s, ATTRIBUTE I.D.'s
and ATTRIBUTE VALUES are to be defined by the user.

RULE 2. For each data list initiated by the user, the
DATA LIST I.D. and all ATTRIBUTE I.D.'s defined by
the user are to be entered in the computer DICTIONARY
as (FILE) DICT. inputs.

New data list I.D.'s and attribute I.D.'s, therefore, are to be
defined within whatever limits may be imposed by the existing
dictionary definitions; and these interrelationships are defined
both as rules and as equations in Section 2.6.

RULE 3. The PROGRAM I.D. vocabulary is defined to be:

(GIRL) LIST
(GIRL) COUNT
(GIRL) DICT. LIST
(GIRL) DICT. COUNT

(GUPD) ADD
(GUPD) DELETE
(GUPD) CHANGE (to be associated with the connective TO)
(GUPD) DICT. ADD
(GUPD) DICT. DELETE
(GUPD) DICT. CHANGE (to be associated with the connective T

(GOUT) LIST
(GOUT) FORMAT

(FILE) DICT.
(FILE) DATA

The program I.D.'s which include the word DICT. are to be used only for
inputs associated with the computer dictionary information. Each
program I.D. in the vocabulary identifies a particular computer process;
and, except for (GIRL) DICT. COUNT, each is defined and exampled under
(GIRL) in Section 2.3., under (GUPD) in Section 2.4., under (GOUT) in
Section 2.5. or under (FILE) in Section 2.6. The program I.D.
(GIRL) DICT. COUNT is added to the vocabulary for the initial implementat
but the program I.D.'s (GIRL) IS, (GIRL) ASK and (GOUT) PRINT which also
are listed in Sections 2.3. and 2.5. are not to be used for the initial
implementation.

RULE 4. The CONNECTIVE vocabulary is defined to be:

Relational		Extra Words	
>	ANDD	AND	WITH
<	TO (=)	OR	OF
=	INN	IS	THE
NOT	EACH	FOR	

3-5

The connective INN is to be used only for defining a vertical search, and can precede only a data list I.D. Each of the connectives ANDD, AND, OR, IN and INN are defined and exampled in Section 2.1.3. Every connective is defined as a single word; and, therefore, each connective, whether word or symbol, is to be isolated between blank spaces. Connectives may be used in any IR system input except the (COUT) FORMAT and (FILE) DATA inputs, and may be used singly or in any relevant combinations to precede any IR language element except the PROGRAM I.D.

RULE 5. For each input associated with the computer dictionary information, the following vocabulary is defined.

Data List I.D.	Item I.D.	Attribute I.D.	Attribute Value
IR/DICT. --- ATTR.	(IR data list I.D.'s) (IR attribute I.D.'s)	CONVERSION CORRELATIVE	D (B,X,Y,Z,S,C,D, V and F codes)
		SIZE/DL	(numeric value)
		SIZE/ITEM	(numeric value)
		C/TYPE	M, A, N, AN
		C/MIN.	(numeric value)
		C/MAX.	(numeric value)
		C/PATTERN	(code)
		IR/SC	(code)
		UPD/SC	(code)

Except for the deletion of COUT/DICT. as a data list I.D., and for special definitions of 3 attribute values, this vocabulary is identical to that defined and exampled under (FILE) DICT. in Section 2.6. The three special definitions of attribute values for the initial implementatic system are (1) the "M" value for C/TYPE, (2) the "D" value for CONVERSION and (3) the "F codes" for CORRELATIVE.

The letter code "M" is defined as an attribute value for C/TYPE to permit the user to specify attribute I.D.'s for which an attribute value is mandatory in the input. Therefore, this extended capability of the data reliability audit is limited by definition to (FILE) DATA and (GUPD) ADD inputs. The attribute I.D. C/TYPE, then, may have 2 attribute values. For example,

```
P/N ATTR.   DATE   CONVERSION   "D"
                   C/TYPE       "M" "N"
```

For the initial implementation system, CONVERSION is defined with a single attribute value "D", and it is to be specified only for a data list I.D. or attribute I.D. with calendar date values requiring arithmetic comparison procedures. Also, specification of CONVERSION "D" requires each calendar date input to be in the following numeric form.

```
        XXXXX
        ‾‾‾‾
        | | |___Last digit
Month___| |_____of the Yr.
Day_____|
```

The CONVERSION "D" procedures then automatically will convert the input
format to the following numeric format defined for arithmetic comparison
procedures within the computer.

```
               XXXX
               ‾‾
Last digit___| |___Day of
of the Yr.         the Yr.
```
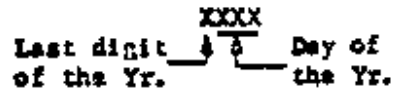
For outputs, each such value will be reconverted automatically to the
numeric input form, but with hyphens inserted between the three numbers.

```
        XX-XX-X
        ‾‾ ‾‾ ‾
        |  |  |___Last digit
Month___|  |_____of the Yr.
Day_____|
```
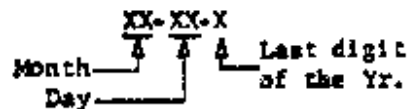
The "F code" as an attribute value for CORRELATIVE is defined
explicitly for this implementation. The "F codes" are to be used
to define the function for calculating a data value, and the code letter
"F" is used to specify both the function itself and one or both of the
IR data variables. For instance,

        EXAMPLE 1.   "F = F1/F2" "F1, VALUE, P/O" "F2, STD. HRS.,

        EXAMPLE 2.   "F = F1-14" "F1, DATE, P/O"

The definition of these functions is limited in this implementation,
and each function is to be defined either with 2 "F" variables or with
1 "F" variable and 1 constant, with every constant being an integer.
In defining the calculation, any one of the following four symbols
may be used.

                +   Addition
                -   Subtraction
                *   Multiplication
                /   Division

Any value defined as the product or quotient of decimal values will be
assigned the same decimal accuracy as the more accurate of the two values.

RULE 6. For each (GOUT) FORMAT input, the following special
vocabulary is defined.

Data List I.D.	Item I.D.	Attribute I.D.	Attribute Value
SPECIAL	TITLE COL/1...,n	CORRELATIVE HEADING SORT	T and (R and F codes) (text) D, An, Xn

Except for the data list I.D. being defined as SPECIAL rather than XX,
this vocabulary is identical to that defined and exampled for present
system implementation under (GOUT) in Section 2.5. The attribute I.D.'s
HEADING and SORT, and the attribute values for each, also are the same
as those defined and exampled under (GOUT) in Section 2.5. The "F code"
as an attribute value for CORRELATIVE, however, is defined explicitly
for this implementation, and is identical to the limited definition given
under RULE 5. Therefore, the T and C variables described in Section 2.5.
are not to be used.

3.5. Conclusion

This system defined for initial implementation is nearly the same
as the generalized system defined in Section 2., and the few differences
between them are limitations rather than deletions. Most of the
differences limit the IR language rather than the system operational
capabilities, but these limitations only restrict the full use of a
few features of the language rather than eliminating basic characteristics.
Therefore, any or all of these differences can be added to the initial
implementation system, as well as to the user requirements, as supplementar
extensions rather than as changes.

The IR language defined for initial implementation , then, accommodate
the technical terminology natural to any application area, and the user
is not required to state inputs in an artificial language or an artificial
and unnatural format. Storage of correlatives in the computer dictionary
permits the user to define very complex interrelationships for automatic
data correlations; and this version of the IR language also accommodates
the user by including a generalized capability for automatic data reliabili
audits and full information security.