GIM-1, a generalized information management language and computer system

by DONALD B. NELSON, RICHARD A. PICK AND KENTON B. ANDREWS

TRW Systems
Redondo Beach, California

The current demand for more comprehensive information systems had created an increasing need for generalized information management. Generalized information management denotes a complex of interrelated information capabilities encompassing both human functions and the operation of automatic devices which supplement and extend human capabilities. The methodology and techniques for such generalized information control have been defined and are exemplified in GIM-1, a generalized information management language and computer system now being implemented at TRW Systems. With the use of GIM-1, the problems inherent in defining more comprehensive information systems can be completely and economically resolved.

The definition of a system for such information management requires a communication network of many remote stations and a central complex of one or more computers. It also must accommodate natural language queries and multiple technical vocabularies. Additionally, such a system must provide automatic correlation of information both within and between data files, complete as well as selectively limited information security, and automatic controls for data reliability, data conversions and the synthesis of data as a function of other data. All of these requirements for comprehensive information transfer can be satisfied by the use of GIM-1.

GIM-1 permits natural language access to data list information stored in a random bulk storage device which also is available to other computer programs, and the remote user may select any available equipment in the entire network for an information output. Although GIM-1 is general to information management needs, its design accommodates the particular information requirements of any one application area with special purpose efficiency in computer operating times. GIM-1 is essentially machine independent, and the calculated addressing scheme permits retrieval of any required data from random storage in a single access.

The GIM-1 language is limited natural English, and the formats and rules for remote inputs are both simple and very general. The language processors, together with the use of dictionaries, permit inputs to be stated directly in the technical terminology natural to each application area, and also provides the user with plain language outputs. The GIM-1 language uses the lineal format natural to prose text, accepts any number of variable length words, and permits a limited freedom of word order. Minimal vocabulary prohibitions to ensure uniqueness apply only to the data list identifiers and are guaranteed by automatic language audits, but no vocabulary prohibitions are imposed on the information values. Additionally, the GIM-1 language permits the definition of any number of identifer synonyms, and provides a large selection of input conditionals which may be used to limit outputs to only relevant information.

In GIM-1, the "master" dictionary contains the data list identifier nouns, and the "user" dictionary associated with each data list contains the attribute identifier nouns. The master dictionary also contains the process identifier verbs and the language connectives; and, in each of these dictionaries, any number of synonyms may be defined for each identifier. The connective words include all conjunctions, prepositions, articles and special symbols specified for possible use in language inputs. The definition of these connectives, then, includes the designation of relational operators and other conditional words for limiting information searches to only relevant data. Therefore, the specification of connectives will dictate operational procedures not only for the language preprocessor but also for the automatic correlation of data and the numerous data processors within GIM-1. Additionally, a limited natural syntax has been defined for GIM-1, since any completely natural syntax would require a probabilistic processor contradictory both to practical efficiency and to the absolute identifications required for updating data lists with complete accuracy.

The GIM-1 data base consists of many different data lists, and all GIM-1 information is stored in data lists. Each data list is identified by a DATA LIST I. D. and consists of many different items of information; and each item consists of an ITEM I. D. followed or not by relevant information values. All information values, then, are stored in items within data lists, and each VALUE is identified by one of the many attributes defined for each data list. Each attribute is identified by an ATTRIBUTE I. D. and any number of attributes can be defined for the identification of all possible data list values. However, since items contain only relevant values or none at all, the usual document has values for only a few of the many attributes defined for a data list.

The standard format for organizing GIM-1 information, then, has a column heading for each of the following four information elements:

Information element	Mnemonic Code
DATA LIST I.D.	D
ITEM I.D.	I
ATTRIBUTE I.D.	A
VALUE	V

An example of information organized in accordance with such a format is illustrated in Figure 1.

Since a data list may have virtually unlimited attributes which may be used or not in any given item, a user is neither restricted to a small set of fixed identifiers nor required to assign a value to each identifier for every item. For example, assume all references for transporation systems are itemized by library code, with each reference containing such standard information as TITLE, AUTHOR, ABSTRACT, etc. In addition to these standard attributes, however, much greater indexing depth can be achieved by also defining many special attributes such as GROUND EFFECT MACHINES, HELICOPTERS, MONORAILS, TOTAL COST, and PRINCIPAL CITY. These extra attributes, then, define special information which may or may not be relevant to any one transportation system reference. For instance, the majority of references may include absolutely no information for these extra attributes and, therefore, no extra values would be stored. However, for any item which may contain such special information, these extra attributes would enable an indexer to describe the reference item in much greater depth. Provided such attributes were defined, then, a GIM-1 user might initiate the following input:

> LIST THE TITLE AUTHOR AND ABSTRACT OF EVERY TRANSPORTATION SYSTEM REFERENCE WITH THE PRINCIPAL CITY "LOS ANGELES" AND THE GROUND EFFECT MACHINES "AVC-1" OR "HOVERCRAFT"

Furthermore, the organization of GIM-1 information permits the remote user to add new attributes for a data list whenever he may wish, and without any effect on the existing data list information. For instance, an indexer might encounter a transportation system reference which included a description of the net change in smog levels during trial tests of some proposed system. Assuming all of the existing attributes to be unsuitable for a description of this information, the indexer might wish to add the new attribute HYDROCARBONS. This new attribute then would be available for possible use in any old or new transportation system reference.

The four elements within the standard format for GIM-1 information can be considered both as an attribute value DATUM having three IDENTIFIERS (Figure 2), and as a set of two dyads, with each dyad having one DATUM and one IDENTIFIER (Figure 3).

Consideration of the standard format as a sequence of two dyads of information is a concept of particular importance. Together with the automatic correlation of GIM-1 data, both within and between data lists, this concept permits the definition of extensive and very complex information formats. The construction of an extended data format by a unidirectional chaining of dyad units is illustrated in Figure 4.

As an elementary example of data format extension, assume the unidirectional construction in Figure 5 as defining both the retrieval and update-add correlation between the attribute AUTHOR in the ANEW DOCUMENT data list (defined in Figure 1) and another data list identified by TECH/AUTHOR.

In Figure 6, therefore, any value for AUTHOR automatically will be also an item I. D. for TECH/

DATA LIST I.D.	ITEM I.D.	ATTRIBUTE I.D.	ATTRIBUTE VALUES
ANEW DOCUMENT	NADC-AX-6123	TITLE	MOD 3 SIMULATION DESIGN
		AUTHOR	JOHN DOE
			WALTER SMITH
		SOURCE	DDC 457 321
		DESCRIPTORS	P-23 SIMULATION
			ADX AIRCRAFT SIMULATION
		DATE	12 NOV 65
		ABSTRACT	DESIGN FOR SIMULATION OF JULIE AND JEZEBEL EXERCISES WITH P-23 AIRCRAFT
	NEL-251367	TITLE	S012 REPORT
		SOURCE	NEL
		REMARKS	ORDERED ON MAY 7, 1966
	NOL-25682	TITLE	BALLISTIC PERFORMANCE
		AUTHOR	PETER BROWN
		SOURCE	DDC 123 896
		DESCRIPTORS	X503 TRAJECTORY
			MISS DISTANCE

Figure 1 – Example of GIM-1 data organization

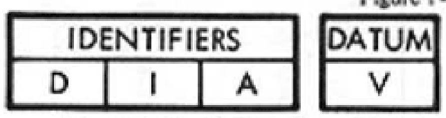

Figure 2 – Standard data format A

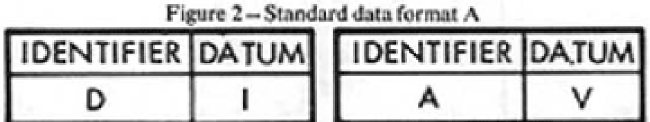

Figure 3 – Standard data format B

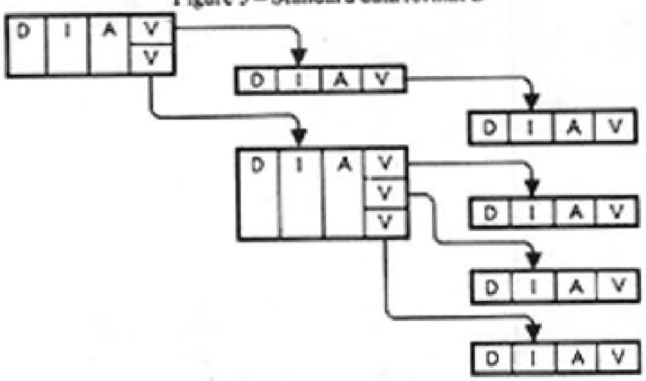

Figure 4 – Extended data format

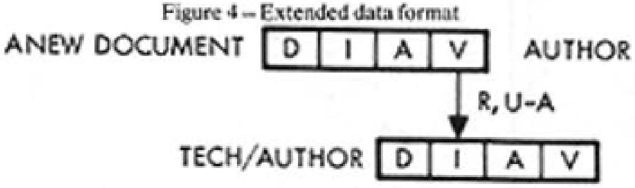

Figure 5 – Example of extended data format

AUTHOR, although each item I.D. for TECH/AUTHOR is not necessarily also a value for AUTHOR automatically also will include, as associated data, all the information stored under the relevant item I. D. for TECH/AUTHOR. However, any retrieval of an item I. D. for TECH/AUTHOR will not include any associated data.

The elementary correlations in Figures 4, 5, and 6 are constructed by unidirectional chaining of the values of an attribute in one DIAV unit with the item I. D. in another DIAV unit. However, many other types of automatic correlation also are available to the GIM-1 user. For instance, "Christmas Tree" interrelationships between different items within the same data list require bidirectional chaining of the values of an attribute with the item I. D. in a single DIAV unit. Additionally, some commonly required correlations are considerably more complex. For instance many interrelationships require correlations between the values of two attributes in different data lists, together with automatic item identifications. This type of correlation is illustrated in Figure 7 as an elementary construction with only two unidirectional correlatives and with only one correlative defined for the values of an attribute.

In Figure 7 then, each update of a value in $D_1 A_3$ requires the automatic update of an item I.D. in D_2. Similarly, each update of a value in D_1A_1 requires the automatic update of a value in D_2A_4. However, the automatic update of D_2A_4 further requires the automatic identification of each item I.D. in D_2 which also is defined as a value in $D_1 A_3$. The automatic creation of such "bridge" values

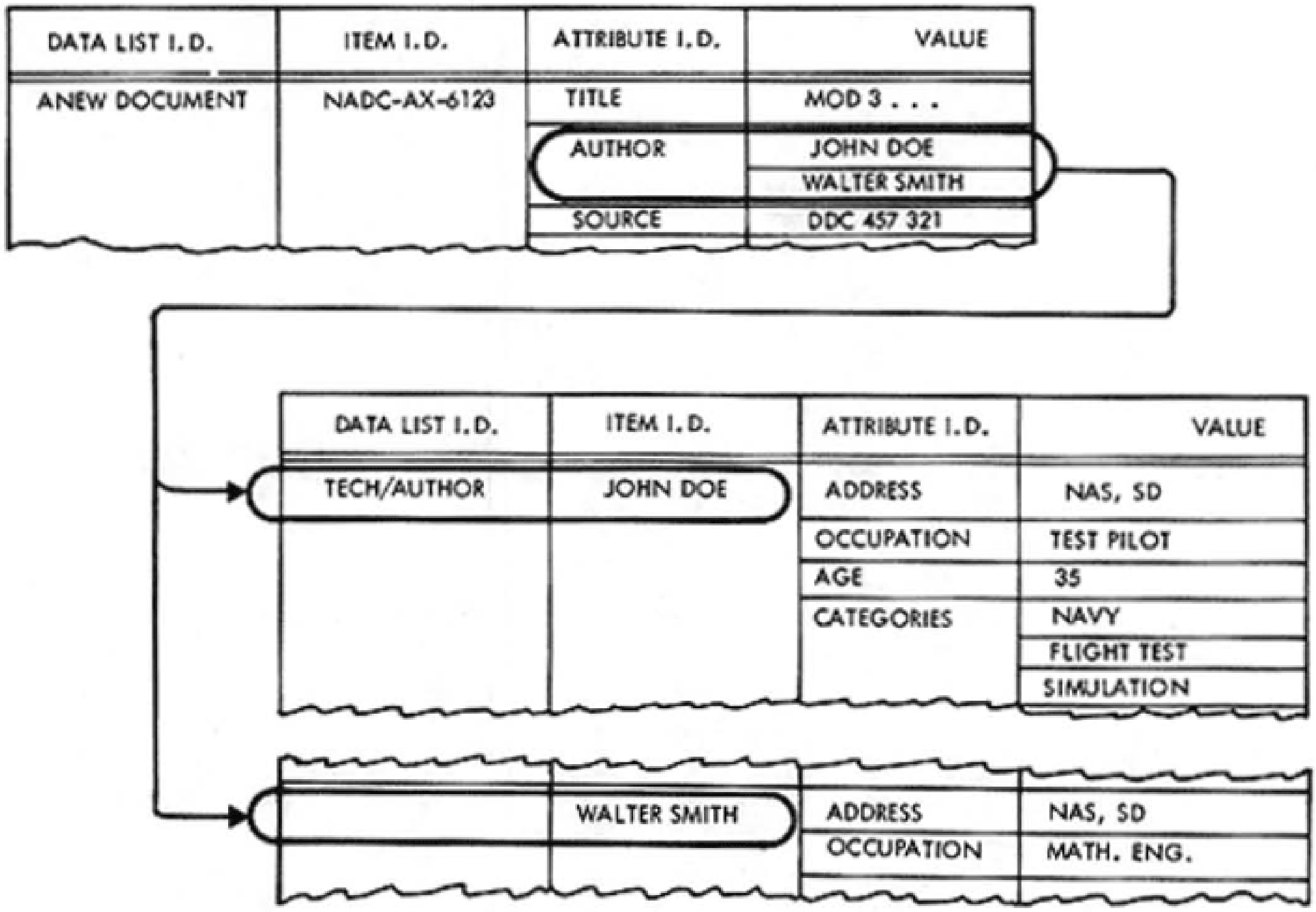

Figure 6 – Example of extended data format

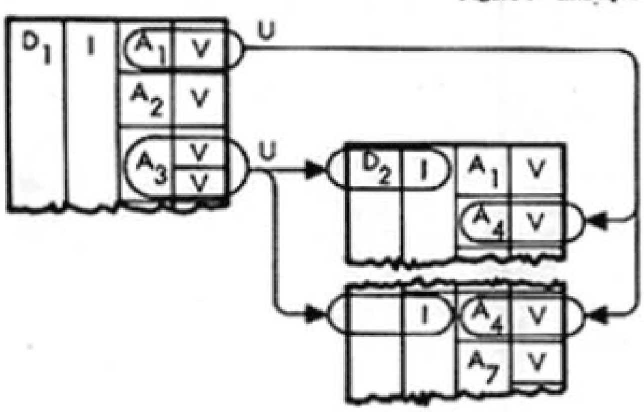

Figure 7 – Example of data list correlation

between data lists, then, is required for any correlation between the values of two attributes in different data lists. Although these few examples of automatic correlation are elementary, they are sufficient to indicate the flexibility provided the GIM-1 user in defining extensions of the standard format for organizing information.

GIM-1, then, permits the user to specify very complex data list interrelationships for automatic correlation. Only a simple dictionary code enforces any defined correlation for all subsequent inputs, and GIM-1 processes all interrelated information with total relevance and with complete accuracy. The several correlations available for specification by the GIM-1 user include:

- The automatic retrieval of associated information stored in other data lists.
- The automatic updating of correlated information stored in other data lists.
- The automatic updating of cross-indexed data lists.
- The automatic calculation of data values as a function of other stored values.
- The automatic coupling of unit values in associated sequences of multiple values.
- The automatic creation of "bridge" values for correlation between data lists and items.
- The automatic updating of all relevant information in "Christmas tree" data lists.
- Full "Christmas tree" searches for retrieval and counting requests.
- The automatic retrieval and updating of nonredundant information stored only in another data list.