Design and implementation of an information base for decision makers*

by R. H. BONCZEK, C. W. HOLSAPPLE and A. B. WHINSTON
Purdue University
West Lafayette, Indiana

ABSTRACT
When considering the design and implementation of systems for decision support, a crucial point is the power and flexibility of available tools for representing data contexts. The value of such systems is constrained by the "richness" of patterning allowed by their data structure mechanisms. We introduce the notion of an information base as a natural step forward in the continuing evolution of data structures. The outstanding features of the information base are (1) its accommodation of the horizontal and vertical integration of information parcels into a single semantic mechanism, and (2) the integration of operators into this semantic structure.

INTRODUCTION
A topical and decidedly significant area of research involves the identification of those criteria which a computerized information system must satisfy if it is to be of value to non-programming decision makers. The ensuing discussion focuses upon such criteria and their implications for system design and implementation. In particular, we introduce the notion of an information base and demonstrate how it may be developed and implemented as an extension to the CODASYL DBTG* approach to data management. We commence with the characterization of an information base as a semantic network. It is then shown that this semantic network may be realized as an extension to the CODASYL DBTG approach to data management. We illustrate how the information base serves as the cornerstone for a generalized decision support system.

Within the scope of this paper a distinction is drawn between the terms information and data. Observe, first of all, that information is an abstraction; it is not something which can be pointed to or seen. However it may be conveyed by patterns of "matter-energy," i.e., by configurations of symbols, by data. Data and information invariably accompany one another. The words on this piece of paper are not information, but rather a pattern of matter-energy which as a consequence of certain activities (e.g., inputting, transmitting, decoding, associating, storing, deciding, etc.) conveys information. The important point is the patterning of data; the "richness" of a notation in terms of the kinds of data relationships which it can represent has obvious implications for its power in conveying information. With this in mind, we can note a pronounced trend in the history of information systems from the relatively impoverished linear data structure to the tree and network data structures, capable of a greater variety of data configuration; correspondingly the ease with which comparatively complex information can be conveyed has also grown. Summarizing, "... we can say that data is an objective notation which has no significance in itself, versus information as a subjective concept which relates a datum to a context." 5

In order to understand the varieties of contexts or configurations in which data must appear if there is to be a comprehensive conveyance of information, we examine the field of semantics. Of special interest is the notion of the semantic net. The results of this examination constitute a basis for the specification of information base features which permit the unambiguous representation of all types of information pertinent to decision support applications. This representation must configure data such that all significant relationships among parcels of information (e.g., among facts, procedures, empirical information, etc.) are accommodated. Furthermore, these objectives for information base features must be met in a manner that is amenable to processing for the purposes of inference and deduction.

Since semantics deals with the relationships between symbols and what they denote or mean, 6 what we call the information base may be viewed as a semantic mechanism capable of representing meanings in terms of data configurations. Its storage technique must be general enough to handle the basic kinds of information involved in decision making regardless of the specific decision application. These types of information are: directive information, conceptual information, empirical information, stimulatory information, information about expectations, information concerning valuations, and procedural information. In addition the information base must be flexible enough to represent the often intricate interrelationships among information parcels, relating them so as to capture their full meaning.
and impact with respect to other parcels of information. This latter point is particularly significant in that it furnishes a basis for the synthesis of separate parcels of information that are all related to the same object, concept, observations, etc.

Woods defines a semantic network to be an attempt to combine into a single mechanism both the ability to store factual knowledge and the ability to model associative connections which render certain parcels of information accessible from certain others. Moreover he indicates three criteria which must be satisfied by a notation used for semantic representation:

1. Logical adequacy. The notation must provide an exact, formal and unambiguous representation of any particular interpretation that may be given to a sentence.
2. There must be an algorithm for translating an initial sentence into this notation.
3. There must be algorithms capable of using the semantic representation in order to perform needed inferences and deductions.

The information base detailed in the subsequent discussion will be shown to satisfy the definition of a semantic network. A query language will be described which, in conjunction with the information base, will be shown to satisfy the three requirements of notations for semantic representation.

FEATURES OF THE INFORMATION BASE

A specific design and implementation of the information base is described in a later section; this design and implementation is based in part upon the idea of a network data base advanced in the CODASYL DBTG Report of 1971 and subject to extensions and modifications outlined in Reference 6. The term information base, rather than data base, is used to emphasize its incorporation of two fundamental features which do not appear in the general data base management literature. Both features concern ways of patterning data that can convey information not commonly treated in the guise of data base management, but of value to decision makers; they furnish methods for introducing two novel kinds of context into data structures.

In observing the progression from linear structures to trees, to networks, we note increased facility for relating a datum with other data; there is an increased capacity for specifying the context of a datum in terms of data structures. Though there is little context inherent in linear data structures, the data content of groups of such structures may be used to represent trees. This becomes complex and cumbersome as the tree to be represented grows in size. Similarly, though it is possible to twist tree structures to the task of representing large or complex networks by using collections of tree-like structures, this cannot be accomplished in a facile, straightforward manner. This analogy may be continued with respect to the two features being introduced in this section. That is, they can, in some sense, be represented within network data structures, but such an approach leads to certain asymmetries (with respect to processing) and difficulties akin to those encountered when representing trees in linear structures. Since the two features are not inherent in the common notion of a network data base, we introduce the information base as a mechanism which encompasses both while allowing full network capabilities.

The first feature involves the introduction of the concept of resolution levels within the mechanism for information organization. A simple example of this is described by Winograd. Consider data about cars in which specific weights and colors are related (linked) to each car; on a higher level of resolution, we may want to somehow store information about what the properties of cars are. So on one level of resolution we are interested in specific attributes of specific cars and on another level we are concerned with properties of cars. Thus two distinctive characteristics of the information base are links which integrate individual information parcels on a given level of resolution into a single network structure and secondly, the integration of information of varying levels of resolution into a single structure. We term the former characteristic “horizontal integration” and the latter “vertical integration.” So horizontal refers to linkage of entities on the same level; whereas vertical denotes linkage among different levels via information parcels that participate in both levels (though the nature of participation is different on each level). A subsequent section of this paper describes both an implementation and the implications of this feature.

The second outstanding feature of the information base involves its ability to handle the integration of programs into its logical structure. Not only does this permit the linkage of a datum with a program that uses it; it allows the construction of networks (in both the horizontal and vertical sense) of programs. This capacity has two primary effects. First, it provides the basis for model formulation. Second, it furnishes a more comprehensive mechanism for semantic representation.

The aspect of model formulation involves the action of relating certain modules into a desired configuration. This necessitates a knowledge of which configurations are meaningful and which are not. Such knowledge is stored in the information base’s semantic network. This approach has much in common with the notion of structured programming. Programs devised according to the tenets of structured programming are readily amenable to storage within the information base; indeed there is also the ability to store alternative modules (e.g., alternative functional forms) for performing a particular role within the context of either other modules or a higher resolution level. The advantages of structured programming in terms of maintainability and extensibility are also apparent in the strategy of integrating program modules into the logical structure of an information base. That is, it is possible to add, replace or delete a module in the same manner that one would add, replace or delete an occurrence of data.

It is useful at this juncture to point out a distinction
between program modularity and program resolution. The idea of resolution level also goes under the name of level of abstraction. Dijkstra\(^4\) indicates that each level of a system's software hierarchy constitutes an abstract resource which participates in the next higher level and which has available to it the resources of lower levels. So "... at one level the programming amounts to manipulation of the abstraction resources supported by the next lower level of the hierarchy.

The programs at that level manipulate abstractions—the abstractions of the resource, whatever it may be—and at the same time participate in generating a higher level of abstraction for the next level of the hierarchy to manipulate."\(^10\) Furthermore, Miller and Lindamood suggest that a "... highly modular implementation is one in which specific functions are performed by specific modules (and nowhere else); on the other hand, a system which preserves a hierarchy of abstract resources would appear to require modularity as a minimum, and perhaps a great deal more 'structure'."\(^10\) Such a structure is effectively treated by the information base feature of resolution levels which allows the arrangement of program resources into levels of abstraction.

The second effect of allowing the integration of programs into the structure of the information base is the more comprehensive semantic representation that is permitted. Much literature about semantic networks is concerned with the network representation of English sentences (e.g., see References 5 and 11). These sentences consist of patterns of verbs and arguments. The typical decision maker who queries the information base requests the execution of some model (i.e., operators, verbs) using certain data (i.e., operands, arguments) as inputs. The usual data base structures handle information about arguments only; the meaningful operator contexts in which such arguments may appear is not represented in standard types of data base structures. A more detailed discussion and practical application of this feature of representing programs in an information base is presented in Reference 12. The remainder of this paper focuses on details and examples of the resolution level feature and on the utility of the information base as a device for semantic representation.

REQUIREMENTS FOR A DECISION SUPPORT SYSTEM

Recall that, in this paper, we are principally concerned with the information base from the standpoint of its contribution to the realization of a general decision support system. Although there are several facets involved in reaching decisions, we investigate three in particular: information access, model formulation, and analysis. The efficacy of a decision support system may be evaluated in terms of its flexibility, facility, scope, timeliness and cost in supporting these three facets.

With respect to information access there must be a mechanism for the systematic, integrated storage of all pertinent information. The information base outlined above provides just such a mechanism, through both horizontal and vertical integration and through its capacity to relate operators with each other and with arguments. Given such a storage mechanism there must be a technique for interrogating (and modifying) it that can be used by decision makers who are not computer experts or programmers. The query language for accomplishing this is presented later.

The second facet which must be supported is the activity of model formulation. This facet refers both to models that are subsequently used for purposes of analysis and to models in the sense of plans to be implemented. This is a crucial aspect for resolving unstructured problems and for supporting the exploratory aspects of decision making. In short, the decision support system must have a component for the generation and evaluation of alternatives for achieving a stated goal. As already indicated, the information base contributes to such an end.

The decision support system must also provide for the activity of analysis; i.e., the fitting of data with models and models with data, thereby resulting in some expectation, beliefs or knowledge. Implicit in the very nature of the planning activity is the dynamic quality of the interface between model and data; for even though a collection of data may be comparatively stable over some time period, both the problems and the models used for problem solving may be subject to frequent alteration. Notice that a model operates on a particular subset of the entire collection of operands available, and it requires a certain configuration of this data as input. We contend that the tedious, cumbersome task of interfacing data and models for purposes of analysis should be automatically handled by the decision support system in response to the commands of a non-programming user. The method for accomplishing this is discussed in subsequent sections.

FORMALIZATION OF THE INFORMATION BASE

We now present a formal description of what is meant by the term "information base." We define a record occurrence to be a uniquely labeled aggregate of data (i.e., string of symbols). Where \(I^+\) is the set of positive integers, let \(X_k\) be the set of labels associated with a finite set of record occurrences, such that \(X_k \subseteq I^+\). A record type, uniquely denoted by the label \(p_i\), may be described by a function \(r_i\) as follows. Define \(R_k\) as the set of all \(r_i: X_k \rightarrow \{0,1\}\) such that:

1. \(\forall x \in X_k, \ r_i(x) = \begin{cases} 1 & \text{if } x \text{ is of the type labeled } p_i \\ 0 & \text{otherwise} \end{cases}\)
2. \(\forall x \in X_k, \ \sum_i r_i(x) \leq 1\)
3. \(\forall r_i \in R_k, \ \sum_j r_i(x_j) > 0 \quad \text{where } X_k = \{x_j\}\)

Property (1) states that \(r_i\) defines the collection of \(x \in X_k\) of the type labeled \(p_i\). Property (2) indicates that each \(x \in X_k\) can belong to at most one \(p_i\). Property (3) states that each \(r_i\) is non-trivial.

Before defining \(X_k\) for \(k > 0\), we note that \(P_k = \{p_i\}\) is the set of all labels associated with the elements of \(R_k\). Since \(X_k\)
is finite, we can define these labels such that \( P_k \subseteq \Gamma^+ \), \( P_k \cap X_k = \emptyset \); furthermore we can define each of these sets of labels such that it has no elements in common with any other \( P_k \). Define:

\[
X_1 = \{ p \in P_1 \mid \exists x \in X_1 ; r_1(x) \neq 0 \} \cup X_1
\]

\[
X_2 = \{ p \in P_1 \mid \exists x \in X_1 ; r_2(x) \neq 0 \} \cup X_1
\]

\[
X_3 = \{ p \in P_1 \mid \exists x \in X_1 ; r_3(x) \neq 0 \} \cup X_1
\]

\[
X_n = \{ p \in P_{X_n-1} \mid \exists x \in X_{n-1} ; r_i(x) \neq 0 \} \cup X_{n-1}
\]

It follows from the definition of \( X_0 \) and \( R \) that there must exist a \( K \) such that \( X_k = X_{k+1} \ldots \) then let \( X = X_k \). Observe that \( X \) is the set of labels of all record occurrences within an information base; these labels are unique identifiers, thereby serving as information base keys. All occurrences of a record type denoted by the label \( p \) can be determined by successive applications of the function \( r \) to the set \( X \). The magnitude of \( K \) indicates the levels of resolution inherent in the information base. The reader will notice that \( P \) is always a subset of \( X \); if it were not desired to treat all record types as record occurrences, one could define \( X = X_k \). There are advantages to defining \( X = X_k \), especially for purposes of altering the logical structure of an information base after it has been loaded. This will be elaborated in a subsequent section.

Continuing, we now formally define the information-set (in-set). This construct, as implemented in the information base, is drawn in part from the "set" idea of the CODASYL DBTG Report, hence the term "in-set." It is important to differentiate this from the familiar notion of a mathematical set. Let \( Q_i = \{ x \in X \mid r_i(x) \neq 0 \} \). If a function associates each element of its domain with no more than one element of its range it is said to be a functional relation. Then each functional relation \( f_i ; Q_i \rightarrow Q_i \) uniquely defines an in-set of which the record type \( r_i \) is said to be the owner and the record type \( r_j \) is called the member. It is important to make several observations about the in-sets of an information base. It is permissible, and sometimes useful, to allow \( i = j \). Second, an in-set may be used to associate record types of different levels of resolution. Third, the set \( F \) of in-sets of an information base must be carefully defined so that its elements are consistent; e.g., one should exercise caution in defining both \( f_1 : Q_1 \rightarrow Q_2 \) and \( f_2 : Q_2 \rightarrow Q_3 \) as elements of \( F \). Finally if \( f_1 ; Q_1 \rightarrow Q_2 \) and \( f_2 ; Q_2 \rightarrow Q_3 \), then we can form the composite in-set \( f_2 ; f_1 ; Q_1 \rightarrow Q_3 \) defined by

\[
(f_2 f_1)(x) = f_1(f_2(x)) \forall x \in Q_1.
\]

This is sometimes desirable from the standpoint of access efficiency; it also allows us to attach special significance or meaning to certain groups of sets.

The foregoing is a formal description of the major features of the information base. It accounts for both the horizontal integration (via in-sets) and vertical integration (via resolution levels) of information into a single mechanism. In order to illustrate the use of resolution levels, we apply the above formalisms to the problem (see Winograd) of representing information about cars. In this problem cars are to be described in terms of color and weight; in addition we would like to denote that color and weight are properties. Suppose we have record occurrences as shown in Figure 1a; these are identified by the respective labels in \( X_0 \). The set \( R_0 \) is also shown; by inspection we see that \( R_0 \) satisfies the needed conditions as given at the beginning of this section. The function \( r_1 \) determines whether or not an element of \( X_0 \) is of the type color. Similarly \( r_2 \) is associated with the type weight and \( r_3 \) is associated with the type car. In our implementation each \( r_i \) defines (and is defined by) a linked list of occurrences of its type. Given \( X_0 \), \( R_0 \), and \( P_0 \) we apply the rule for defining \( X_1 \) to obtain the result shown in Figure 1b. \( R_1 \) is also given and clearly satisfies the necessary conditions for its definition. Application of \( r_4 \) to elements of \( X_1 \) can be used to determine which elements are vehicle properties. Figure 1c gives the \( X_2 \) that follows from the definition. If we take \( R_2 = \emptyset \), then \( X = X_2 \).

The occurrences and their "vertical" relations with each other are diagrammed in Figure 2. Also depicted are two in-sets: \( f_1 \) and \( f_2 \). Using the definitions of \( Q_1 \), \( Q_2 \), and \( Q_3 \) given in Figure 2, \( f_1 ; Q_2 \rightarrow Q_1 \) and \( f_2 ; Q_3 \rightarrow Q_2 \). The arrows in the diagram point from the owner of the in-set to the member; i.e., each arrow points in the direction opposite to that in the notation of its corresponding functional relation. Using the formalisms introduced here it is a simple matter to represent an extended problem including other kinds of vehicles, more properties, subclassifications of properties (e.g., structural, functional, etc.) and even properties of properties.

An information base for water quality management

More detailed discussions of the water quality management problem may be found in References 14 and 15. The objective of the example presented in this section is to demonstrate the applicability of the information base as a

\[
X_0 \rightarrow \{ 1 \} \cup \{ 2,4,5,15,16,8,9,11 \}
\]

\[
R_0 = \{ (r_1, r_2, r_3) \} \text{ with labels } R_0 = \{ (0,9,11) \}
\]

where

\[
r_1(x) = \begin{cases} 1 & x < 6 \\ 0 & \text{otherwise} \end{cases}
\]

\[
r_2(x) = \begin{cases} 1 & x \in \{ 1,2 \} \\ 0 & \text{otherwise} \end{cases}
\]

\[
r_3(x) = \begin{cases} 1 & x \in \{ 1,7 \} \\ 0 & \text{otherwise} \end{cases}
\]

\[
X_1 \rightarrow \{ 1,2,4,15,16,8,9,11 \}
\]

\[
R_1 = \{ (r_2) \} \text{ with label } R_1 = \{ (8) \}
\]

where

\[
r_2(x) = \begin{cases} 1 & 8 < x < 10 \\ 0 & \text{otherwise} \end{cases}
\]

\[
X_2 = \{ 1,2,4,5,15,16,8,9,12,13 \}
\]

Figure 1—Resolution levels for representing information about cars
Let:

\[ s_1(x) \cup \{ x \in \mathcal{E} \cup r(x) \cup 0 \} \]
\[ s_2(x) \cup \{ x \in \mathcal{E} \cup r(x) \cup 0 \} \]
\[ s_3(x) \cup \{ x \in \mathcal{E} \cup r(x) \cup 0 \} \]

Figure 2—Occurrences in car information base

device for capturing the semantics used to support practical decision problems. At this point, we presume that the reader has a sufficient concept of what an information base entails to obviate the need for complete formalistic description. So for the sake of economy, the following example is presented in a less formal manner than the previous one. It will be used to depict certain implementational details (e.g., languages in which the information base is specified and with which it is utilized).

Consider the record type POLLUTER, displayed in Figure 3a. This aggregate of data item types represents measures of types of polluter activity for a given date. So occurrences of this record type correspond to measurements taken on various dates. In order to build a semantic network, we must indicate how this concept of POLLUTER fits into the pattern of knowledge concerning water quality management. A polluter is properly characterized as being a property of a river reach. Other properties of a reach include reach parameters, headwater, incremental flow, and treatment plan. So a reach is characterized in terms of these properties as follows: a reach is a portion of a river in which certain water quality parameters are relatively invariant; which has no more than one (point-source) polluter, one incremental flow or one headwater; and which must possess treatment plans. This could be represented in the information base by occurrences of the REACH PROPERTY record type displayed in Figure 3b. However, observe that each occurrence of the data item NAME (e.g., "POLLUTER," "HEADWATER," "PARAMETER," etc.) is also the label of a record type which is itself an aggregate of item types and which may have numerous occurrences. So, for instance, "POLLUTER" denotes an occurrence of REACH PROPERTY; but it also denotes a record type (shown in Figure 3a). The same circumstance holds for the other reach properties, though their record types are not depicted here. The resultant logical structure is illustrated in Figure 3c; a record type enclosed by another record type indicates that the enclosed record type is also an occurrence of the enclosing record type.

We continue the example by examining general water quality modeling characteristics. In order to simulate water quality we need information about the following: the rivers involved, the reaches which are in each river, each reach's properties, junctions, piping plans, and model parameters. This is shown in the structure of Figure 4. Note that the record type GMC has two item types: CHAR (characteristic) and IMPT (a measure of the relative importance of each characteristic). Five occurrences of GMC are shown: RIVER, REACH, JUNCTION, MODEL, and PIPE PLAN. General Modeling Characteristic is not the only property of
a segment that needs to be represented; Local Modeling Characteristics (LMC) are also needed. (The term segment is used to indicate a particular area of a river basin.) The details of the high level record type LMC are not shown here, but they describe information about non-point sources of pollution, permits for point-source pollution, treatment plant construction status, permit violation data, etc. As shown in Figure 5, GMC and LMC are occurrences of SEGMENT PROPERTY which is itself an occurrence of the record type WQMA; BASIN and SEGMENT are also occurrences of this record type. The information base could be further extended to incorporate aspects of land use planning since they influence and are influenced by water quality management.

The foregoing logical structures are initially defined in terms of an Information Description Language (IDL). Use of the IDL to define the logical structure of Figure 4 is presented in Figure 6. The specification shown is largely self-explanatory. Each record type is followed by the item types which compose it. If the record type is of a high level, then its item types are followed by a specification of those record types which are its occurrences. Definition of an in-set must be preceded by specifications of its owner and member record types. For simplicity, details of the type and size of items are not shown; also the ordering criterion of each in-set is not shown.

A LANGUAGE FOR DECISION SUPPORT

The reader will observe from the preceding discussion that the decision support system has two basic components: an information base and a query language. Clearly the usability of a semantic network depends upon implementation of a language with which one can extract (insert) meanings that are held in the semantic net. Not only are semantics conveyed by a particular language, they are limited by it as well. The language is used to express meanings, but it also delineates the kinds of meanings which are expressed. We can devise arbitrarily complex semantic networks, but their usability is (from the practical standpoint) constrained by the languages (and language processors) which can be interfaced with them. Observe then that there is a fundamental duality of (1) the language in which ideas are expressed, and (2) the structural representation of ideas in an information base. On the other hand the semantic mechanism must be capable of taking full advantage of the language's power. In the case of the
Implementation described in this paper, the query language is the constraining factor since it is intended primarily for the practical support of decision activities of managers in both the public and private sectors.

Implementation of a natural language (e.g., English) processor is certainly a noble objective. It is our experience that the typical decision maker neither uses, nor needs, a complete facility for conversing in a natural language. It often happens that phrases or clauses are sufficient to convey an idea; there are grammatical constructs (e.g., reflexive, passive) which are not particularly germane to the decision activities of information access, model formulation, and analysis. In addition the decision maker is more prone to desire information conveyed in a tabular or graphical fashion than in a narrative mode. It has also been found that the user sitting at a computer terminal has a tendency to use abbreviations and concise mathematical notation.

With these factors in mind, the query language to be outlined here has been designed to meet the needs of decision makers for flexibility and brevity of expression, while at the same time being easy to learn and utilize. The query language is effectively a subset of English that has been extended to include standard mathematical operators (i.e., relational, arithmetic, and univariate and multivariate functions). The focus here is upon use of this language for interrogation, though it may be used for data creation and modification as well.

\[(\text{COMMAND})(\text{FIND clause})(\text{CONDITIONAL clause})\]

or alternatively,

\[(\text{CONDITIONAL CLAUSE})(\text{COMMAND})(\text{FIND CLAUSE}).\]

So some sample queries are:

**LIST REACH.NAME,REAERATION.PARAMETER AND REAERATION.EXPONENT. FOR DATE=110175 AND REACH.LENGTH<.9**

**WHEN DATE=110175, PLOT REACH.NUMBER VERSUS AMMONIA.CONCENTRATION AND DO.CONCENTRATION/LOG(TEMPERATURE)**

**LIST GENERAL.MODELING.CHARACTERISTIC IF IMPORTANCE>3**

The language allows any meaningful configuration of arguments and mathematical operators to appear in the FIND and CONDITIONAL clauses.

**PROCESSING, HIGHER LEVEL RECORD TYPES**

Upon receipt of a query, the query processor generates appropriate commands for traversal of a multi-level network. These commands are operators in an Information Manipulation Language (IML). We use the term IML to distinguish from the Data Manipulation Language (DML) proposed in the CODASYL DBTG Report. The DML is intended to permit access, modification and retrieval for a single level network database. The IML has the more extensive function of furnishing tools for manipulation of the information base. Thus the IML contains operators for handling traditional DML functions and operators for processing higher level record types. The latter are discussed here.
In the DML, routines exist for creating a record occurrence at a unique location denoted by its key. The IML includes analogous routines for specifying that an existing record occurrence be treated as a record type as well. There are four such commands:

- **CRTK**—Create Record Type based on a given Key
- **CRTR**—Create Record Type based on the current occurrence of another Record type
- **CRTO**—Create Record Type based on the current Owner of a given set
- **CRTM**—Create Record Type based on the current Member of a given set

Another traditional DML operator (AMS) adds a specified record occurrence as a Member of a given Set. A similar IML operator is used for adding an existing occurrence of one record type as an occurrence of another record type. Note that utilization of this operator must be preceded by a generalization of the definition of a record type which was introduced above (i.e., the definition is generalized by removing the restriction that \( \sum_{i} r_i(x) = 1 \), \( \forall x \in X_k \), where \( r_i \in R_k \)). This operator is **AORT**, Add Occurrence to Record Type, and it uses the key of the occurrence to be added. In conjunction with commands for the logical restructuring of a network data base, \(^6\) AORT provides the ability to add and delete higher level record types and add existing occurrences to higher level record types; and this is accomplished without dumping and reloading data.

Finally, operators are needed for determining the key of a record type, given an occurrence of the record type. These commands are:

- **GKRR**—Get Key of the Record type for the current Record occurrence of that type
- **GKRO**—Get Key of the Record type whose occurrence is the current Owner of some set
- **GKRM**—Get Key of the Record type whose occurrence is the current Member of some set

These operators provide the capacity to proceed from a lower level occurrence to a higher level occurrence, when used in conjunction with traditional DML operators. It must be emphasized that the typical user of the query system needs to have no knowledge of the IML operators, for they are automatically set up and executed by the query processor in response to a user query.

ADVANTAGES OF THE RESOLUTION LEVEL FACILITY

We contend that the concept of resolution levels effectively adds a new dimension to the field of information storage. The preceding discussion has suggested a means for operationalizing this concept as an extension to the traditional single-level network approach. One advantage is that multi-level semantic networks may be stored without introducing asymmetry in the interpretation and processing of in-sets and record types. Since a record type may also be defined to be an occurrence of a higher level record type, the addition of a record type is treated by creating a new record occurrence at the next higher level. That is, we remove the distinction between data values and the structural pattern according to which data is organized. In other words, the terms "attribute" and "value" are recognized as being relative, so that what is a value on one level is an attribute on another and vice versa.

From one viewpoint this abolishes the special status of an IDL specification by permitting record type definition to be a dynamic process. That is, the creation of a new record type is synonymous with the creation of a new record occurrence of a higher level record type. Thus the IDL specification of the highest level of resolution is effectively reduced to the definition of three record types (one describing information about record types, another relating to information about sets, and one with various system information\(^6\)) and some in-sets between them. This definition is always the same regardless of the content and structure of lower resolution levels.

A second advantage, already mentioned in connection with integration of programs into the information base, concerns a mechanism for handling levels of abstraction in software. A third advantage is that higher level record types may be used to characterize areas of an information base by assigning record types of a particular area to be occurrences of a higher level record type; these areas may be defined for a variety of reasons (e.g., for information security, to denote scenarios, to delimit functional areas— which may overlap, etc.). As the information base becomes large and varied in content, this technique may also be used to realize efficiencies in path determination processing by limiting the scope of network traversal to a particular information base area.

THE INFORMATION BASE AS A DEVICE FOR SEMANTIC REPRESENTATION

With the foregoing background, we can now address the three criteria proposed by Woods,\(^5\) which must be satisfied by a notation used for semantic representation. First observe that the information base is a tool for the representation of a semantic network (i.e., a single mechanism with both the ability to store factual knowledge and the ability to model associative connections which render certain parcels of information accessible from certain others).

The first criterion of a notation for semantic representation is logical adequacy. The notation must provide an exact, formal and unambiguous representation of any particular interpretation that may be given to a sentence. Recall that the sentences with which we are concerned are those allowed in the query language for decision makers.
The information base allows a given query to have a multitude of interpretations. The query specifies a group of data items which may be related to each other in many ways via vertical and horizontal linkages in the information base. Each path of linkages on which these items lie corresponds to a particular interpretation of the query. Upon receiving a query which is subject to multiple interpretations the query processor prompts the system’s user in order to ascertain which interpretation (i.e., path) is intended. Details of the manner in which this has been implemented may be found in References 16 and 18.

The second criterion is that there must be an algorithm for translating an initial query into the notation of the information base. This is the central function of the query processor whose operation has already been described; implementational details appear in Reference 13. The third criterion, concerning algorithms capable of using the semantic representation, has also been addressed in the discussion of the query language. Observe that the IML provides the means for interfacing algorithms with the semantic representation. Algorithms which have been used range from relatively commonplace report generators to large scale water quality simulation models. 15

CONCLUSION

In the beginning we observed that it is the patterning of symbols which can convey information; a datum’s meaning derives from its context, from its relationships with other data. Thus, when considering the design and implementation of systems for decision support, a crucial point is the power of available tools for representing contexts. The value of such systems is constrained by the “richness” of patterning allowed by their data structure mechanisms. Observing the progression from relatively impoverished linear structures to trees and networks, we note that each stage has provided a more powerful and flexible tool for semantic representation. In this paper we have introduced the notion of an information base as a natural step forward in the continuing evolution of data structures. An outstanding feature of the information base is its accommodation of both the horizontal and vertical integration of information parcels into a single mechanism. An information base implementation which builds upon network concepts was discussed. A topic for future research is the investigation of an information base implementation which builds upon the relational data base notions. 16 A second distinctive feature of the information base, namely the integration of operators into its structure, was briefly described. The information base is utilized by a non-procedural, English-like query language, that has been designed for decision support applications. This language, in conjunction with the information base, satisfies the requirements for a notation for semantic representation.

ACKNOWLEDGMENTS

The authors are indebted to Professor Walter Reitman of the University of Michigan, Professor Tibor Vamos of the Hungarian Academy of Sciences and Dr. Bertram Raphael of SRI for useful discussion on semantic network and data base management.

REFERENCES
