IRIS—AN INFORMATION RETRIEVAL SYSTEM FOR BRIDGE INVENTORY MAINTENANCE

Charles J. Conley, Colorado Department of Highways;
Jerome C. Kaltenhauser, University of Colorado; and
E. W. Klaphake, Sperry-Univac

The Colorado Department of Highways bridge structure inventory has been automated by using the IRIS information storage and retrieval program. IRIS allows rapid and convenient bridge inventory updates and maintenance, selective retrieval of information for report generation, and on-line queries of the data bank. IRIS maintains its principal data bank as a pseudo-inverted list structure on mass-storage devices. Selection of records for retrieval is made by calculation rather than by file search and comparison, and the selected records are reconstructed rather than retrieved as entities. Each category of information (descriptor) is a candidate for use as a key for retrieval, so maximum flexibility and specificity are possible in describing records for retrieval. The user describes a set of bridge records for retrieval by involving 1, some, or all of the descriptors as retrieval keys in Boolean expressions by using a simple and convenient language. Records that belong to the described set are retrieved. The generality of the IRIS design allows it to be used in many applications other than bridge inventory. IRIS is written in FORTRAN and is operational in both batch and interactive modes on the CDC 6400 at the University of Colorado.

•EACH state highway department is required by Section 26 of the Federal-Aid Highway Act of 1968 to develop an inventory of all bridges in the state carrying or spanning a federal-aid highway. At intervals of a year or less, several different reports on the inventoried bridges must be submitted to the Federal Highway Administration. Bridge inventory maintenance and report-generation requirements are being met by the Colorado Department of Highways by the computer program IRIS and associated report-generation programs.

IRIS maintains the state bridge inventory for fast access in a data bank on mass-storage devices (disks) rather than magnetic tapes. Correction facilities are provided to allow the inventory to be rapidly updated as errors are found and new information on bridges is acquired. Retrieval of bridge data is by way of a simple Boolean language that enables the user to specify the kind of bridges he or she wants retrieved in terms of the information stored about each bridge. The user can request that the retrieved bridges simply be counted, that some or all of the information on each retrieved bridge be printed out, or that some or all of the information be placed in a machine-readable file for processing by auxiliary programs such as report generators.

IRIS can operate in an interactive mode as well as a batch mode. The interactive mode enables the user to explore the bridge inventory through successive queries, and answers return in a matter of seconds. A dialogue with the inventory in which the user asks the questions is allowed by the simplicity of the query language and the rapid response to queries in the interactive mode.

Publication of this paper sponsored by Committee on General Structures.

IRIS was acquired and developed by the Colorado Department of Highways for bridge inventory and reporting requirements, but it has potential for being implemented by a much wider range of users. The data structure and format are quite general—the parent program, TAXIR, from which IRIS descended, was developed for handling taxonomic information in botany (1)—and program capabilities can be extended in several directions with minor modifications.

This paper addresses 2 audiences: bridge engineers responsible for maintaining bridge information and data processing personnel responsible for facilitating the maintenance of the information.

BRIDGE INVENTORY APPLICATIONS

Data Description

The Colorado Department of Highways bridge inventory is coded in accordance with the Recording and Coding Guide for the Structure Inventory and Appraisal of the Nation's Bridges (2). Each structure in the inventory is represented by a record of 84 items or descriptors that convey various types of information. The following example descriptors are taken from the Recording and Coding Guide (2):

1. Identification, which includes information on inventory route, structure number or designation, features intersected by the structure, local topography (narrative description), and latitude and longitude of the structure;
2. Structural data, which include information on year built, design load, and average daily and yearly traffic;
3. Type of structure, which includes information on materials, bridge structural type, and number of spans;
4. Condition, which includes information on operation rating, condition of deck, and condition of superstructure; and
5. Proposed improvements, which include information on estimate of year needed, proposed design of improvements, and estimated cost of improvement.

The information carried by the items in the Recording and Coding Guide (2) is in the form of either character strings or integers, but 3 kinds of descriptors will be distinguished to clarify some characteristics of the IRIS program. A character string can be either a name or text, but in IRIS usage a name is a character string that the user expects to occur in more than 1 record. For instance, because there are many bridge structures in Denver, Colorado Springs, Pueblo, and other cities, the names of these cities would recur in many bridge records. Text is also a character string, but no such recurrence is expected. Certain items in the data bank call for descriptions of features near a bridge, and, because the user is free to improvise, any recurrence of a character string would be by chance. This is an example in which a character string would be regarded as text. IRIS allows name character strings of up to 90 characters and text character strings of up to 40 characters for any item. (These limits can be varied at will by simple program changes.) Integer numbers have the usual properties of integers and can express virtually any magnitude. The restriction to integer values of numbers is not serious because a simple program modification would allow decimal input. Decimal output is currently an option of the program.

Query Description

Until now the terms item and descriptor have been used interchangeably to designate a category of information; the term item is from the Recording and Coding Guide (2). In this paper, it will be convenient to use the term descriptor exclusively. The information carried by a descriptor is designated a descriptor state and can be thought of as

an entry under the descriptor. Each descriptor has a descriptor state or entry in each bridge structure record in the bridge inventory data bank. For example, for the descriptor, average daily traffic, 1,231 might be the descriptor state in a particular bridge record. Figure 1 shows the relation of descriptors, descriptor states (S_{ij}'s), and bridge records. This figure shows a bridge inventory in abstract form as a large table that has all the information on a single bridge written on a single line.

Queries on bridge inventory require that the user specify the kind of bridge in which he or she is interested. Bridge records meeting the specification are identified for further processing. IRIS query specifications are written by involving 1 or more descriptor states in a Boolean expression by using the Boolean operators, OR, AND, and NOT. The simplest possible Boolean expression is the name of the state in which the user is interested. Because the program must be informed of the descriptor to which the state belongs, the descriptor also must be named. This forms a descriptor-descriptor state pair. A query containing the information average daily traffic, 1,231 would identify all bridges with an average daily traffic of 1,231 vehicles. If the user is more interested in a range of traffic conditions, he or she might write average daily traffic, from 1,000 to 1,500, and all structures that have average daily traffic from 1,000 to 1,500 vehicles would be identified.

Query specifications of greater complexity can easily be written by using the Boolean operators in the conventional way. The user might be concerned with heavily used bridges. He or she would write a Boolean expression that would be set up as follows: (average daily traffic, from 2,000 to 5,000) AND [(number of lanes, 1) OR (number of lanes, 2)]. Bridges with 1 or 2 lanes carrying 2,000 to 5,000 vehicles per day would be identified. If the user is interested in only a particular highway or section of a highway, the highway and section numbers would be incorporated into the expression by using AND operators. The expression can incorporate as many terms as the user needs to single out the bridge or bridges in which he or she is interested. There is no restriction on which descriptors and states are entered into the retrieval expressions.

Retrieved queries return either a count of the records that meet the user's specification or part or all of each such record. A sequence of queries to determine whether any structures on US-6 have a rating of 30 tons (27 metric tons) or less for 3-axle rigs was performed. Because some of the structures do have such a rating, additional information identifying the particular bridges, their structural conditions, and length of detour necessary was requested. Some terms used in the actual queries have been changed for clarity but the essentials remain. In the sequence of queries shown in Figure 2, a set of records (bridges in this case) is selected at each stage, and each successive set is a subset of the set from the previous query. In IRIS the word RESULT is reserved to designate the set of records from the previous query. Therefore, when RESULT appears in the Boolean expression, it means that subsequent queries will be made only on the set of records identified in the previous query. Figure 2 shows this series of queries. The limiting structures and length of detours are now identified. Subsequent queries on the detour routes would reveal whether they will accept the vehicle of interest. In the illustrated queries the user added some comments to the query language to aid later interpretation.

The principal use of IRIS by the Colorado Department of Highways is for storing and reporting on the required state bridge structure inventory, and for this purpose the retrieved records are placed in a machine-readable file for processing by auxiliary report-generating programs. The user controls which records will be retrieved by the Boolean specification he or she writes, and the user controls which portion of each record is to be retrieved by a descriptor list. In query 3 in Figure 2, for example, the user requested that information on only 4 descriptors (out of 84 in the data bank) be reported. In retrieving information with IRIS the user has control over which bridge structure records, each of which contains information on 1 bridge, are to be retrieved and which information is to be reported for each record.

Figure 1. IRIS data format.

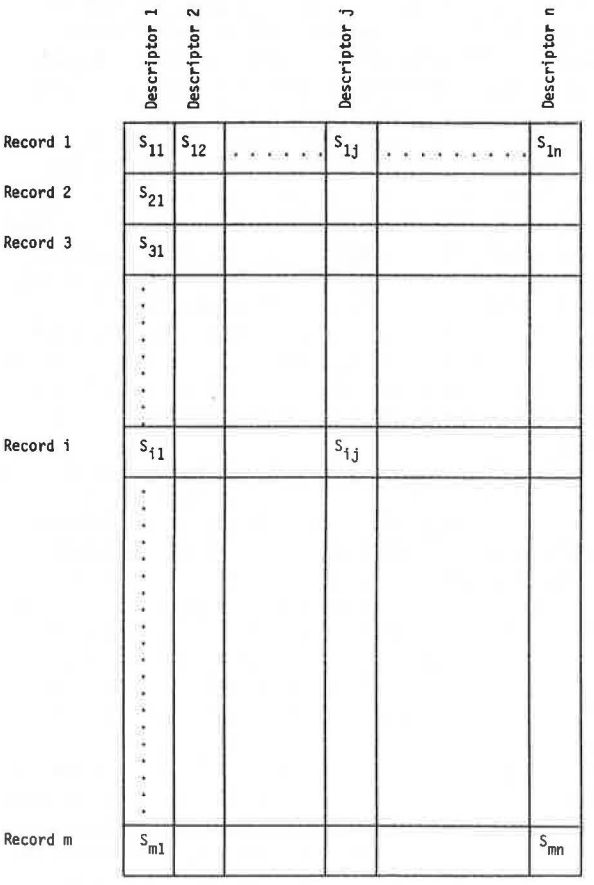


Figure 2. Series of queries.

1. HØW MANY STRUCTURES HAVE (HIGHWAY, US 6)*

   NØ. ØF RECØRDS IN QUERY RESPØNSE = 92

   NØ. ØF RECØRDS IN THE DATA BANK = 3719

2. HØW MANY STRUCTURES WITH (AXLE, 3) AND (WEIGHT, FRØM OO TØ 30)
   AND RESULT*

   NØ. ØF RECØRDS IN QUERY RESPØNSE = 2

   NØ. ØF RECORDS IN THE DATA BANK = 3719

3. PRINT A LIST ØF INFØRMATIØN ABØUT ABØVE: (FEATURES INTERSECTED
   STRUCTURE NUMBER, DETØUR LENGTH, STRUCTURAL CØNDITIØN) AND RESULT*

   MCDØNALD CREEK H-01-U 99 4

   BIG SALT WASH H-02-V 6 6

IRIS PROGRAM DESCRIPTION

Figure 1 shows the general data format assumed by IRIS. The descriptors are categories of information such as structure number or number of lanes. A record is built up from the entries under each descriptor and is therefore the description of a particular bridge structure or other object of interest handled by the program. IRIS must transform, store, update, and retrieve records under user control. Retrieved records also must be presented to the user in a useful form.

Each entry S_{ij} in Figure 1 is called a descriptor state. A descriptor state can be an integer or a character string in IRIS. A consequence of the IRIS data format is that a particular descriptor can have 1 and only 1 state in a given record. The format also makes possible a compact method of data storage and a fast and flexible retrieval mechanism.

The storage mechanism in IRIS depends on the idea of mapping 1 finite set of elements into another. If a given descriptor has a finite number of states, then all possible states for that descriptor can be set in a 1-to-1 correspondence with the positive integers. There are various ways of making the correspondence, but, if the correspondence rule is chosen to have an inverse, then the passage from any descriptor state to its sister element among the positive integers can be made and the inverse passage from the domain of positive integers to descriptor states also can be made.

When all the name states of a descriptor are arranged in alphabetical order in a dictionary, the dictionary entries can be mapped into the positive integers in a natural way by assigning 1 to the first entry, 2 to the second, and so forth. Conversely, if 1 of the positive integers i within the range of the dictionary is given, then the descriptor state can be found immediately. It is the i th dictionary entry. Because equally spaced integers or decimal numbers representing measurements (or any ordinal numbers, generally) are inherently ordered, mapping into the positive integers 1, 2, 3,... can be carried out quite naturally. Text differs from ordered numbers (orders) and names in that the size of the set of different states that might be encountered is not known beforehand. This is, in fact, the essential distinction between text and name descriptors in the IRIS program.

When the mappings for order and name descriptor states are completed, the integer corresponding to any state of a descriptor having N different states can be expressed in $W = [\log_2 N + 1]$ bits (the brackets denote truncation to the nearest lower integer unless N = 1). In other words, the N different states that a descriptor can assume are enumerated 1, 2, 3,..., N, and this requires $W = [\log_2 N + 1]$ bits. IRIS uses the integers discussed here to store the order and name states in each record and to retrieve, or, rather, to identify and reconstruct, records having user-specified characteristics. If the records in a data bank use m descriptors, then m integers in binary form are stored for each record; a descriptor with N_n possible states is allotted $W_n = [\log_2 N_n + 1]$ bits for a binary integer. The stored binary integers are designated chi string functions (CSFs). One CSF per descriptor is stored for each record.

To reconstruct the actual descriptor states, as is done in retrieval, the mapping process is reversed. For an order descriptor, a linear transformation of its CSF gives the state represented. For a name descriptor, the CSF is the index of the literal state name in the dictionary maintained for the descriptor. Text descriptor states cannot be mapped into the positive integers in the same fashion as orders and names can, so they are stored full length in character representation. This uses space rapidly, so for extensive amounts of text a document file would have to be coordinated with the primary data bank.

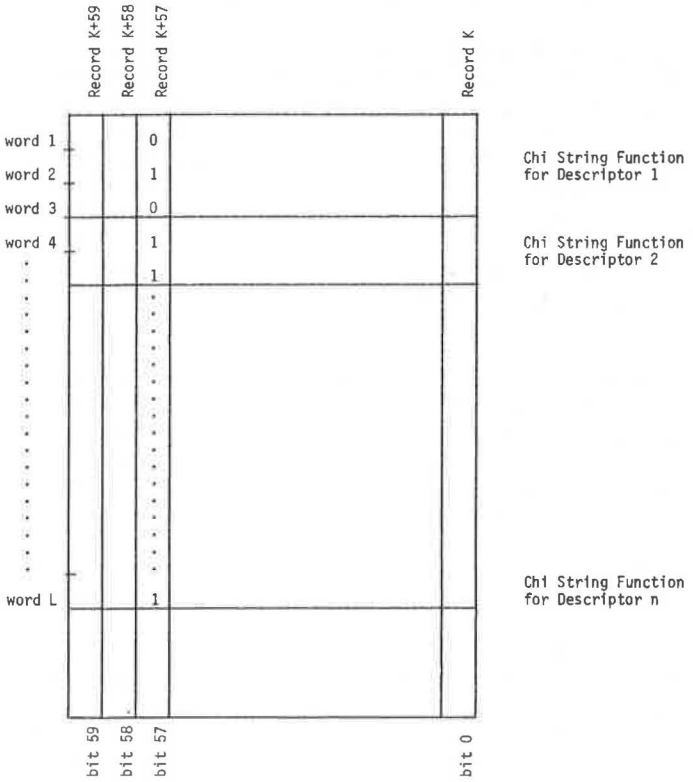
In what is commonly called an inverted list data structure, information used in retrieval is kept separate from the stored records so that identification of records for retrieval can be made without inspecting the records themselves. In IRIS the situation is similar, except that the records are not stored as entities—they are decomposed into CSFs plus dictionaries. This has been called a pseudo-inverted list-information structure. Because each descriptor used in the data bank is on the same basis as every other, each descriptor is suitable as a candidate for use as a retrieval key. Also only those descriptor states of interest to the user need to be reconstructed in the interval records.

Because the identification of records for retrieval is a somewhat involved process to describe, we will present only a brief sketch. The user supplies to the program a Boolean expression that describes a set of records in terms of the descriptor-descriptor state combinations of his or her choosing, and the program must determine which records, if any, are members of the described set. The user-supplied Boolean expression is transformed by IRIS into a Boolean function for operation on the successive CSFs corresponding to the descriptors involved. The Boolean function allows the examination of the CSFs using OR, AND, and NOT machine operations. Any record identified by these operations is flagged for selective reconstruction.

A brief look at the actual disposition of the data bank in storage should further clarify IRIS retrieval operations. The dictionaries are presently kept in core. The CSFs are kept in mass storage and are brought into core as needed for retrieval and other operations. The buffer used for mass storage transfers is shown in Figure 3 for CDC 6000 series machines. Each row in Figure 3 corresponds to 1 computer word, and the 60 bits of a word correspond to the 60 columns shown in the figure. Each of the columns contains one data bank record. That is, all the CSFs for a given record are stored vertically in a column: The first W_1 bits down are for the CSF of descriptor 1, the next W_2 bits down are for descriptor 2, and so on. The current Colorado Department of Highways data bank requires a buffer of approximately 1,400 words for its 84 descriptors.

The identification of records for retrieval or reconstruction was described previously as though 1 record were processed at a time. In actuality, because of the full word orientation of the OR, AND, and NOT machine operations involved, all columns in a buffer are processed simultaneously. Thus, 1 sweep through the buffer determines which of the 60 records should be reconstructed.

Figure 3. IRIS data bank buffer.



IMPLEMENTATION

IRIS is written in FORTRAN and has 1 mass-storage random-access routine in assembly language. The primary data bank is placed in mass storage. In the batch mode, data bank corrections are transformed, accumulated, and placed in extended core storage (ECS) to allow all corrections to be made in 1 sweep through the data bank. ECS is not used when one is making corrections in the interactive mode, so each correction requires a pass through the data bank. Except for this, batch and interactive modes are completely equivalent. For the bridge inventory application, only 100 words (each word has 10 characters) are allotted to the name descriptor dictionaries. Allowing for up to 120 descriptors, this version of IRIS requires 46,000 (octal) 60-bit words of core storage to run. The program also uses up to 32,000 (octal) words of ECS when making corrections in the batch mode. The length of the buffer used to transfer information from mass storage to core is set not by the word size of the machine but by the number of states the user allows for various descriptors (roughly following the $\log_2$ dependence previously described). It was noted that a 1,400-word buffer handles 60 bridges (60 bits per word in CDC 6000 series machines). There are 3,700 bridges in the inventory. Therefore, sixty-two 1,400-word buffer loads ($60 \times 62 > 3{,}700$) must be transferred from disk file to core. This set of transfers is required whether the operation is a simple query, a complex query, or the dumping of the entire data bank to a report file.

IRIS capacity is virtually unlimited in 2 directions: the number of order descriptor states (because for ordered numbers the $\log_2$ dependence describes total growth in storage required) and the number of records that the program can maintain. Core storage increases would be in direct proportion to space additions for large name dictionaries.

Some simple cost figures, although they are heavily dependent on the particular equipment and charging algorithms used, may help to convey an idea of user charges. For simple queries, like those illustrated in this paper, central processor times are about 0.2 s per query and cost 2 cents apiece. The corresponding mass-storage charge for a single query on the data bank (as opposed to the RESULT of a previous query) is about $1.00. The high mass-storage charge reflects a charging algorithm with high overhead for accesses and low charges for information transfer. As noted above, the mass-storage charges are incurred only in moving the data bank and are unrelated to the complexity of the retrieval operations carried out. Greater complexity in these operations is reflected in increased central processor costs. To dump the entire data bank to a formatted report file and to the printer costs approximately $80.00, a large part of which represents central processor and printer costs.

REFERENCES

1. G. F. Estabrook and R. C. Brill. The Theory of the TAXIR Accessioner. Mathematical Biosciences, Vol. 5, 1969, pp. 327-340.
2. Recording and Coding Guide for the Structure Inventory and Appraisal of the Nation's Bridges. National Bridge Inspection Standards, Article 25.11, Federal Register, Vol. 36, No. 81, p. 7851, April 27, 1971.