DESIGN OF A FLORISTIC INFORMATION SYSTEM

By

MAHESH VISWANATHAN and D. K. SUBRAMANIAN

SCHOOL OF AUTOMATION

Centre for Ecological Sciences,

INDIAN INSTITUTE OF SCIENCE

BANGALORE-560 012

TECHNICAL REPORT 6

CENTRE FOR ECOLOGICAL SCIENCES

Indian Institute of Science

BANGALORE-560 012

INDIA

1. INTRODUCTION

An excellant set of data pertaining to the flora of Karnataka has been compiled by Fr.C.J.Saldanha and his associates. The information collected by them is being put into the form of a book. Additional information on various species are being collected in Uttara Kannada district. Similarly, vegetation information and mapping based on this information has been undertaken by Dr.Pascal and his colleagues at the French Institute, Pondicherry. Rainfall information, both in abstracted and detailed forms, is also available for the State of Karnataka. Related information on agriculture, horticulture, forestry, soils, animal populations, industried, geography etc. is also being collected systematically from various sources.

Further, the Western Ghat program of the Centre for Ecological Sciences, IISc. has been conducting scientific investigations in various identified areas in the Uttara Kannada District with the objective of getting quantified information on biomass type and rate of growth, forest types, flora, insects, birds resource and land uses, soil characteristics, energy use and water availability and requirements. This will mean that a sizable volume of data on various related aspects of ecology of the various regions will be

available in the near future and there should be a proper and feasible way of using such data.

If the given data range is small and volume is moderate and if use requirements are confined to the study of individual areas - like flora, forests, energy etc. then these information can be handled manually, processed, analyzed and used for getting meaningful inferences. But in this case, the problem has the following characteristics

i. data ranges are many
ii. data volume is quite high
iii. user requirements are many and varied

Also since the study of ecology involves the study and understanding of interrelationships amongst many physical variables pertaining to rainfall, geography, soil, biomass, resources like water, and variables pertaining to secondary and use components like land use, resource use etc., the use of such data base needs an integration based on a common unit. Initially from the point of view of availability of data, it is felt desirable to have taluk as a basic unit and collect/convert all data to this base. Because of these large magnitudes encountered, computer based information and processing system is essential.

Such a computer information and processing will have to be built with the following objectives in mind.

i. Representation of data pertaining to each taluk of many physical and resource variables like flora, insects, soils, water, rainfall, biomass, energy, agriculture, industries, human/animal populations etc.

ii. Design of a query language to get the desired information from the data base. The query language should be simple and should help in retrieving the necessary data items based on certain conditions/predicates.

iii. Analysis of taluks based on primary data (based on soils, geography, vegetation, rainfall, slopes etc.) identification of various "ecological units" -- each unit will represent a unique combination values for primary variables; grouping of taluks into these ecological units, calculation of maximum carrying capacities for these units.

iv. Analysis of existing land and resource used for these ecological units; identification of optimal (type and quantity) of uses for each unit. This will be

very helpful in identifying (a) locations most appropriate for an end use like industry (say) and (b) locations which are ecologically fragile or important or unique so that they need to be conserved, preserved and improved if possible.

This may also be used (a) to suggest variations and changes in land use patterns (b) to identify areas needing immediate corrective actions. In essence this will produce a processed information system which can be used in planning new activities and in improving existing areas ecologically.

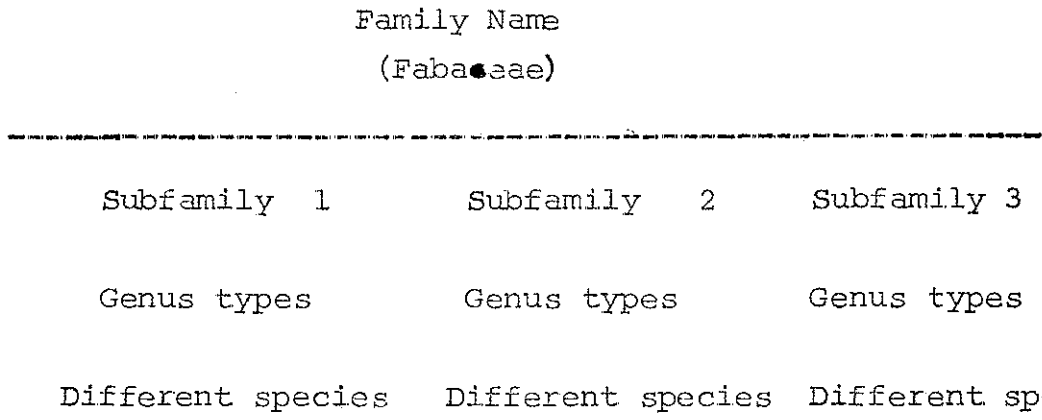
The above list of objectives really show that the use of computers not only facilitates data handling, but also improves qualitatively the analysis of such data by orders of magnitude. We have established the need for an ecological information and processing system.

Since, the objectives look for a system which is very much different from the design philosophies, problem models and studies available in existing systems, it was decided to take up the problem in stages. The first stage will be to have a simple query system to retrieve data. Even the query based information system has been designed to operate on one type of data - flora of Karnataka. This is due to the fact that data on this aspect are available in a complete and well documented form.

This report discusses the design and implementation of a floristic information system on the PDP 11/35 computer available at the Institute. The information system uses data base concepts. Data pertaining to one family has been put into the secondary storage and retrieval facility to get data items from it under various conditions has been built. These aspects are discussed in the next sections.

2. INFORMATION SYSTEM

The information system uses as data, the flora of the state of Karnataka, India. The data initially covers the complete range of flora of the family "Fabaceae". There are 3 subfamilies which come under the Fabaceae family. Each subfamily in turn contain a number of genus types which in turn are subdivided into species types.



As most of the information available is about the species types or flowers as we may call them, the flowers are being used as the main Operands in each case. This main record is subdivided into five records.

The twenty odd fields were assigned to these five records in such a way that the more closely tied fields come under one head.

2.1. Queries

The queries were to be in English. The user is assumed to have some previous knowledge of taxonomy to put the database to good use.

Queries could function both for storage as well as retrieval. In storage it order the insertion, modification, updating and deleting of information referring to one particular field or the record as a whole. In the case of retrieval queries can demand information about specific fields given, of course, certain conditions.

A sort of user-designer interaction is envisaged which could help the user obtain the information he seeks in the easiest possible way.

The problem being one of such vast applications, even when dedicated to serve a certain toxonomic use, new packages for storage and retrieval can always be added on to what has been presently implemented.

2.2. Storage

The storage of the data is planned in such a way so as to simplify the retrieval procedure. When such data is to be stored, the memory space demand is one of the most important parameters. One has to make the reasonable assumption that the data does not use up all the core memory available in any system. Still, the storage of characters, as will be in the case of most fields, cannot be dispersed with. This problem of storage without the threat of using up all the core memory is overcome by creating a disk file, only a step away from a more permanent, method of storage, the magnetic tape.

Even in such a case, the volume of data being large, the number of blocks of memory which may be used up is also a parameter to be considered.

The implementation of the queries is as elastic as possible to accommodate as many facilities as possible, regarding information about the fields.

Query interpretation is a significant part of query processing. A technique has to be evolved to convert the query into a cipher which can have easy interpretation through the program.

A delimiter which can be understood to mean the end of the query is a must. Again, the maximum length of the query string has to be fixed.

Procedures are required to read the query and call others procedure in order to determine the validity of the query before processing.

The validity of each of the reserved words in the query have to be checked.

The implementation of all these have been discussed in the next chapter.

3. IMPLEMENTATION

The species name is the principal operand. This is natural, as it is mainly about the species that most of the information was presented in the data.

Instead of having the whole set of fields in one record, the information is grouped into five batches each group containing fields which are somewhat related.

First, the whole record is called a Flower record. Second, the five batches are named depending on the fields they include. The five batches in turn records and are called 'Name', 'Occurrence', 'Characteristics', 'Habitat' and 'Miscellaneous'.

The Name Record contains field like the Botanical name, English name, Kannada name, Subfamily nametetc of the species concerned.

The occurrence Record contains fields like the District, Season, Forest type etc.

The Characteristics Record contain fields like the Description, Reference and Features.

The Habitat Record contains fields like the Vegetation type, Rainfall, Altitude.

The Miscellaneous Record contains fields like the uses and part uses.

The fields are split into those of limited range and those which have unique titles. E.g. Each species has a distinct Botanical name, English name but could belong to only one of the three subfamilies.

Those with unique titles have to be read and stored as such. E.g. considering the English name to have a maximum possible length of 30 characters they are stored in a packed array. The retrieval will involve all the 30 characters simultaneously.

The fields with a limited scope are stored as integers and each integer is assigned the corresponding name within the program. E.g. As only 3 subfamilies are available in the Fabaceae the subfamilies can be arranged alphabetically and integers alloted. One major advantage of this is that a lot of space is saved in storing integers rather than characters.

A further splinter group differing from those declared and stored as elaborated above are the user defined types. Granting of course that these may also be defined and stored as integers, the existence of such a technique capable of being implemented has to be recognized.

There are 19 districts at present in the State of Karnataka. Being too large a number to be used as a user defined scalar, it was made an array of characters. Only two characters are possible in this array, T or F; 'T' signifying the district where a particular species is present and F its absence.

3.1. Queries

The first problem is to conceive of the ideal retrieval reserved word. The word any user might use to obtain information. The word 'GET'was thought of and decided that this would serve this purpose best. Obviously the user will require the information about any field of his choice and so a facility to fill in the field will follow 'GET'. Next follows the information which the user will divulge. This could be called the condition, as the user requests information based on a promise. These have been formated in an admittedly rigid way, one has to admit, so as to reduce the complexity. Following the field about which information is required, the user gives his condition which is another field name. The rigidity lies in the user having to use the words 'Get', 'when' '=' and ',' in the order mentioned, the last mentioned being used as a cue to consider that the query is at end.

Of course, enough safeguards have been incorporated to prevent the user from going astray. Enough prompt statements and cues have been implemented to get the user to the right path very soon.

The most significant part of the retrieval is the structuring of data. This aspect had to be gone into meticulously and elaborately to decide the most effective data structure for efficient retrieval.

New ideas had to be considered as they kept surfacing and changing the structure uptill the stage of actual trials was a common feature.

The Data structure of the Flower is enumerated below.

```
Flower Record = Record
                    Name      :  Record
                                    Botanicalname : Packed Array
                                                      1..30 of Char;
                                    Englishname   : Packed Array
                                                      1..30 of Char;
                                    Kannadaname   : Packed Array
                                                      1..30 of Char;
                                    Subfamilyname : Integer;
                                    Genusname     : Packed Array
                                                      1..30 of Char;
```

```
                    Cultivation type :(Natural,Naturalized,
                                      Endemic, Exotic);
                    AuthorofGenus    :Packed Array 1..30 of
                                      char End;

Occurrence: Record
            Nativity       : Packed Array 1..30 of Char;
            District       : Array        1..19 of Char;
            SimplingPlace  : Packed Array 1..30 of Char;
            Forestype      : Integer;
            Foreststate    : (Wet, moist, dry);
            Season         : Array        1..2 of Integer;
            Frequency      : (Occasional, Rare, Common,
                              Abundant, Frequent)

Characteristics: Record
                 Description : Array      1..400 of Char;
                 Features    : PackedArray 1..80 of Char;
                 Reference   : Array      1..400 of Char
                               End;

Habitat : Record
          Vegetationtype : Integer;
          Rainfall       : Integer;
          Altitude       : Integer
                           End;
```

```
Miscellaneous : Record
                Uses     : Packed Array 1..30 of Char;
                Partuse  : Packed Array 1..80 of Char
                           End
                End;
```

Most fields are read as packed array for the main reason that they are unique to each species and writing them down must be just as the reading is done. When the whole set of data can be divided into a limited number of divisions as in the case of a few fields, it is better to declare these as integers and generate integer codes. While writing, of course, the Codes are deciphered.

Some other fields are declared as user defined scalar types. In this method once again, codes are generated and decoded while writing. Of course in the data file the corresponding entry should be an integer as there is no way the actual user defined scalar type can be used directly.

3.2. Query Processing

All the field names along with the compulsory words like 'GET', 'WHEN' are taken into a reserved word array to facilitate easy comparison to certify the validity of the information given. These reserved words have a maximum length of 30.

As packed arry cannot be read directly procedures have to be written to read them. The characters are read one by one and packed into arrays to maximum length of 30.

In the procedure which reads the flower record these procedures are called. If a field has been declared as an integer the field may be read directly. In the case of user defined scalar they are coded through a 'case' statement.

A procedure is introduced to read the query string and store it in an array, character by character. To obtain the string character by character and to return a tokenvalue depending on the position a separate procedure is written.

The query format is :

Get (Reservedword) a When (Reservedword) = (String.)

Now all these words have been arranged in alphabetic order in the reservedword array. Now the token-value returned through the getcharacter procedure mentioned above is corresponding to the array index in the reservedword array.

Now the fields have been divided into two parts with respect to queries. In eight of the fields information may be sought which concerns the whole data file.

E.g. One could ask to output all the names of flowers in Hassan district.

In the remaining fields single output are only possible. This differentiates the two reserved word set seen in the query format above. The second set is limited to eight reservedwords. The expansion of this to include fields is relatively simple.

A procedure is next required to select the record and print the required field of information. This is of course conditional to the information sought.

A procedure numbers the conditions and depending the conditions select the record to print out.

Write statements have been included at the required places in the program to give the necessary error statement with the corrections suggested therein.

The main part of the programs initialises the reserved word, district and forest arrays. The flowerfile is reset and the prompt for the user is displayed.

An useful implementation has herewith been achieved.