THE RELATIONAL MODEL FOR DATA BASES IN COMMUNITY STUDIES

Otto Wildi Swiss Federal Institute of Forestry Research CH 8903 Birmensdorf, Switzerland

Keywords: Vegetation, Communities, Data Bank, Data Model, Relational

Abstract: Different data base models are considered and compared to the relational model. File structures are described and examples are given to illustrate the differences.

Introduction

The aim of Vegetation Science is to study plant communities, their spatial changes, their changes in time, and also their interaction with environmental factors (Mueller-Dombois and Ellenberg 1974). For purposes of study, a plant community can be regarded as a typical combination of species occurring as a stand. This definition of a community notwithstanding, community properties and their relationships can be presented and interpreted differently. For example, the Braun-Blanquet (1964) approach imposes a strong hierarchical system on vegetation units, in analogy to the Linnean system, and uses this system to reveal and organize information. Other approaches rely on continuous floristic gradients (Gleason 1926, 1939), on environmental differences (Daubenmire 1968), or on patterns caused by dynamical changes as the basis of community studies. Some of the approaches tend to emphasize group structures and recognition of groups, referred to as noda by Poore (1955), gradients by Whittaker (1967), or ground pattern by Greig-Smith (1952). Many numerical methods pertain to these approaches (Orlóci 1978, Pielou 1977, Greig-Smith 1982). One common purpose is to reveal the true state of vegetation, when field data of sufficient reliability are subjected to analysis. Vegetation sampling, i.e. data collection, thus represents an important step in vegetation studies. This may produce large amounts of data and lead to revisions of existing systems. Ellenberg and Klötzli (1972), and Feoli and Lagonegro (1982) are examples.

Vegetation studies require special methods in handling large numbers of relevés. The traditional way of organizing relevé data is to construct a table whos rows represent species and columns are relevés. This arrangement is helpful for classification tasks, but it can be extremely space-consuming. A more compact storage mode is achieved when the data set is subdivided into several smaller tables. But many small tables are again difficult to handle when rearrangements have to be accomplished, such as introducing or cancelling relevés, changing species names, storing new environmental measurements, etc. In fact, the storage, retrieval, and updating of data require the design of appropriate data file structures before a data bank is constructed.

Data File Structures

The final design of a file structure will limit the management and utilization of the data bank. A detailed description of the objective thus represents the first step in planning. We consider two examples:

(a) A monitoring project is designed (Fig. 1). Let us assume that an area is subjected to environmental changes such as a decrease in water supply or pollution by industrial developments. Obviously, survey stations will have to be established at which the state of the vegetation and the site are recorded at different points in time. It is expected that species aggregation will also be affected. Therefore, any relevé (a complex record) will be structured to reflect the spatial distributions of species within the plots. We expect that species composition will be affected, and therefore have to be also subjected to monitoring. For this, we need a species list. The main components and connections of this organization are shown in Fig. 1. The boxes can be interpreted as lists of records. Each of these lists will change while the project progresses. More species may appear, the number of points in time for measurements will grow, new survey sites may be included etc. The design in Fig. 1 is, however, rather arbitrary. Boxes are established for items which are rated as being important. Furthermore, the boxes are connected by lines whenever an important relation is assumed. Relevés, as an example, obviously consist of species records.

(b) Fig. 2 demonstrates a similar design for a survey project. Unlike in Fig. 1, it is assumed that relevés are only sampled once. Time thus is no longer of interest and the date of sampling is just a variable appended to the list of relevés. Should spatial distribution be important, the relevé becomes a part of the records of a geographical unit. Fig. 2 suggests that special lists are foreseen to be used for references and for site description. Finally, the file struc-

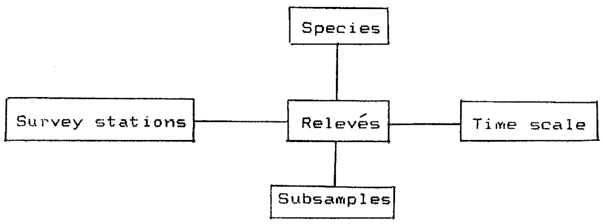

Fig. 1. Data organization in a monitoring project.

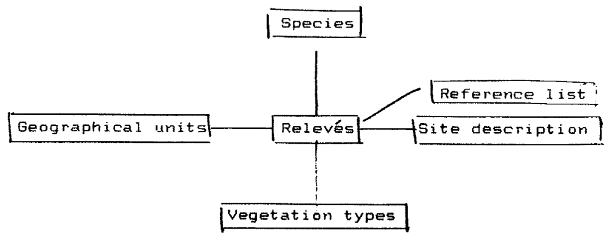

Fig. 2. Data organization in a survey project.

ture also shows that relevés may be members of vegetation types.

Fig. 1 and 2 represent problem-oriented access paths. One may, for example, retrieve relevés from one particular date box. The list of points in time would then represent all possible options. But when it comes to organizing a data bank, the allocation of the data has to be explicitly defined. One solution is a representation by a rectangular matrix (Table 1). This is a simple arrangement, but it has several disadvantages:

– It may require much unused storage space; whenever a species does not occur in a relevé, one cell remains empty.
– In order to find a specific element – say species s in relevé r – the entire data set has to be scanned sequentially. The access can be improved, but not optimised by a hierarchical arrangement. For example in Table 1 (upper form) records from one point in time are combined into one block. But relevés from one particular site are dispersed over the entire matrix. In Table 1 (lower-form) emphasis is on the sites. But points in time are now dispersed.
– Changes in the data file structure require a reorganization of the matrix. For example, introducing new species means creating a new row. The reorganization may be drastic, if increasing the number of subsamples and destroying the regular, hierarchical structure.

An ideal data model would have to overcome all these problems. Furthermore, it should minimize redundancy. This means that any information is stored just once.

The relational model of data files

Until recently, computer storage was often the limiting factor when a data bank was designed. Efficient storage and retrieval algorithms as well as queries were available only on the level of single files (cf. Estabrook and Brill 1969). The network model and the hierarchical model were operational solutions (Schlageter and Stucky 1977). But when a data system grew or changed its structure, it was difficult to maintain consistency and it often happened that conflicts were introduced in the data model. Codd (1970) proposed what he called the relational model which solves all these problems. At the time, storage capacity and computing speed of most computers did not allow to handle such a model efficiently. But by the time the Astrahan et al. (1976) operational program was presented, the relational model became a realistic proposal to users at large. The basics are as follows:

All the data in a data bank and their connections can be represented by (varying) relations. A relation is an array of items (tables, quantities, names) which describe a structure or a process. Let us consider the example in Table 2. In this, a set of relevés is represented in three relations. The first one is ternary (of degree 3), since it consists of 3 columns. The columns are called domains of the relation, i.e. «relnr», «rtype» and «rsize». The names of these domains all start with letter **r** to indicate their membership to relation RELEVÉ. Relation SPECIES is of degree 4. A row is called tuple. «relnr» in RELEVÉ is called a primary key, since it uniquely identifies all the tuples. To determine whether a particular species occurs in a given relevé, one more relation has to be scanned, COMPOSITION. The links between RELEVÉ and SPECIES is given by the domains «crelnr» and «cspnr». Unlike RELEVÉ, the primary key is the combinations of «crelnr» (the relevé number in COMPOSITION) and «cspnr» (the species number in COMPOSITION).

The connection of relations in Table 2 fulfills the conditions to be in its first and second «normal form» (Schlageter and Stucky 1977):

– All relations consist of simple (non-hierarchical) domains.
– Relations with complex primary keys (e.g. COMPOSITION) have only domains which are dependent from other relations.

What results is a complete representation in two-dimensional arrays. The advantage of this becomes obvious when describing time-dependent processes. Data storage and even modification of the data-bank structure may then become frequent operations.

Storage

Simple changes in the data content comprise adding or deleting tuples. When a new species appears in a relevé, relation COMPOSITION is extended by one row. Since the ordering of tuples is immaterial in the relational model, it is easy to append new data. If needed to perform search operations within a single array, a secondary representation with ordered tuples may be produced internally by a data bank system. The reverse operation removes information. A relevé is deleted by erasing tuples in one or several relations where «spnr» (or «cspnr») is a domain. When ac-

Time	t_1			t_2			·			t_k		
Station	s_1	·	s_j	s_1	·	s_j	s_1	·	s_j	s_1	·	s_j
Relevé no.												
Subsample no.												
Species 1												
Species 2												
·												
·												
Species p												
Site factor 1												
Site factor 2												
·												
Site factor q												

Station	s_1			s_2			·			s_j		
Time	t_1	·	t_k	t_1	·	t_k	t_1	·	t_k	t_1	·	t_k
Relevé no.												
Subsample no.												
Species 1												
Species 2												
·												
·												
Species p												
Site factor 1												
Site factor 2												
·												
Site factor q												

Table 1. Representation of data from a monitoring project in a single matrix. A hierarchical structure is adopted giving priority to «time» (upper form) or to «station» (lower form).

tions of this sort occur, the number of rows will change. But the columns – which make up the structure of the data set – are retained.

Retrieval

The simplest retrieval tasks are search processes in single relations. Any file management system can do this. A number of more complex operations on one or several arrays can help to answer more specific questions. Two of these will be considered:

Projection. Relation COMPOSITION in Table 2 is used for an example. Let us assume that changes in time are no longer of interest, but only the content and quantity of species. In the first step, projection removes one or several domains. In our case it is «ctim». Some tuples will now have identical contents. The second step therefore erases all duplicates. In COMPOSITION the last row is no longer needed as it is identical with the first row. The new relation COMPOS (Table 3) presents relevé content independent of time.

Join. This is an operation which enables queries among two or more relations. This updates two matrices to only one. In Table 4 the process of joining RELEVÉ and COMPOSITION in table 2 yields a new relation RELEVÉ * COMPOSITION. Relevé 1 produces 4 tuples, relevé 2 one tuple, relevé 4 two tuples, and relevé 5 does not occur, since it has no records in COMPOSITION.

Many retrieval tasks will require a series of operations. If we are interested in the content of the relevés in Table 2 but disregard time, the first step is to project COMPOSITION as shown in Table 3 and then join the new relation COMPOS with RELEVÉ to form RELEVÉ * COMPOS in the manner shown in Table 4.

Change of structure

While an investigation evolves, its concept may be adapted to the gain of insight. New treatments could be included and new parameters measured. The relational model allows extension without having to re-design the already existing file structure. In fact, a new domain can be included in any relation. For example, SPECIES in Table 2 already has a domain «sr-value», which is an ecological indicator value (r-value: Landolt 1977, Ellenberg 1979). A new indicator can be introduced simply by adding a new column, say «cfval» for f-value. Whenever a retrieval process includes relation SPECIES, the latter are now automatically available.

In a more complex case we assume that our initial data model suffers from a constraint. Relation RELEVÉ in Table 2 has a domain «rtype». This may include meadow (1), shrubs (2), woodland (3). Such a distinction allows analyses on just one of these types. But as time evolves we find that the latter are changing and «type» in fact is merely an «initial type». The problem can be solved by introducing a new relation VTYPE (variable type) with a domain «vcurty» for current type in VTYPE and the date of the new entry (Table 5). The reader will easily verify that the data system remains consistent. As an example, at any point in time the occurrence of any particular species is assigned to one of the continuously changing relevé types.

Table 2. Given two sets (RELEVÉ, SPECIES), relation COMPOSITION describes the content of the relevés.

RELEVÉ	(relnr	rtype	rsize)	
	1	1	16	
	2	1	16	
	4	3	4	
	5	3	4	
SPECIES	(spnr	spname	sautor	sr-value)
	1	Fagus sylvatica	L.	x
	2	Fraxinus excelsior	L.	7
	3	Allium ursinum	L.	7
COMPOSITION	(crelnr	cspnr	cqua	ctim)
	1	1	+	84
	1	2	2	84
	2	2	1	84
	4	1	1	84
	4	2	2	84
	1	3	5	85
	1	1	+	85

Table 3. A projection of relation COMPOSITION in Table 2.

COMPOS	(crelnr	cspnr	cqua)
	1	1	+
	1	2	2
	2	2	1
	4	1	1
	4	2	2
	1	3	5

Table 4. Joining RELEVÉ and COMPOSITION from Table 2.

RELEVÉ * COMPOSITION	(relnr	rtype	rsize	cspnr	cqua	ctim
	1	1	16	1	+	84
	1	1	16	2	2	84
	1	1	16	3	5	85
	1	1	16	1	+	85
	2	1	16	2	1	84
	4	3	4	1	1	84
	4	3	4	2	2	84

Examples of sampling designs

In any investigation, the retrieval and utilization is clearly triggered by the sampling design. It is therefore an advantage if the latter is inherent in the data model. The relational model allows this and also supports changes and extensions in the design. An example is considered in connection with Fig. 3.

At first glance, the sampling method appears to be hierarchical. It consists of big plots (not shaded) in which species are qualitatively recorded. The size of such a plot could come from requirements in remote sensing (aerial photograph, satellite picture, etc.). Small plots (shaded) are

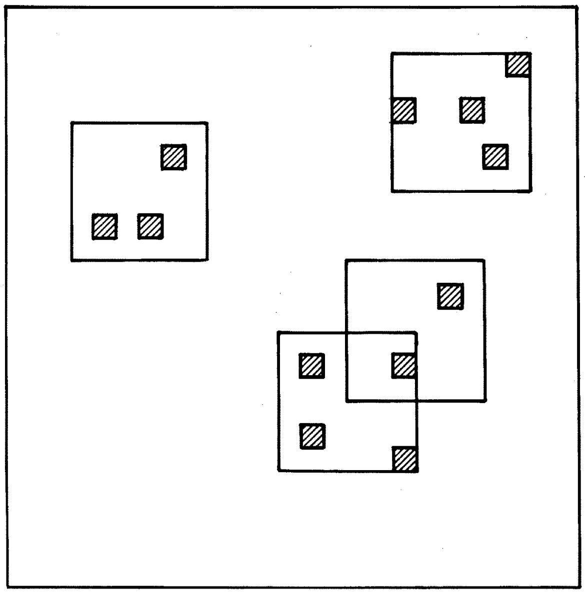

Fig. 3. Hierarchical sampling design with big plots (not shaded) and small plots (shaded).

embedded in these to yield more detailed (quantitative) information on the vegetation. It is assumed that the number of small plots per big plot varies and also changes in time. An overlap of big plots is allowed to occur. Retrieval should answer requests of the following type:

- Display the records of all the small plots within a specified big plot.
- Display the content of big plots without detailed information from the small plots.
- Display the entire information from all plots.

The pattern can be resolved by a labelling system for relevés. Big plots could have numbers ending with zero:

100, 110, 120, ..., nn0

Small plots could start with the first two digits of their parent plot and then with numbers one to nine:

pp1, pp2, pp3, ..., pp9

Even though such a system seems to be quite practical, it may become inconsistent as a project evolves. In the present example, the maximum number of small plots per big plot is 9. An overflow might occur. At the beginning of the investigation, the number of small plots per big plot is inherent in the plot labels. But as soon as the first small plot disappears, this relationship vanishes. Finally this system cannot handle overlapping big plots. Small plots would have to be duplicated to appear in both parent plots. This would result in a loss of data consistency.

In the relational design the relevés are numbered arbitrarily, independent of type and relationships. The latter are stored in a separate relation, say DESIGN. Assuming an organization as shown in Table 2, the relation would be like this:

DESIGN (drelnr, dmember)

From any given relevé the possibly existing small plots can be found in DESIGN. A projection yields a list of big plots: Delete «dmember» and erase duplicates in «drelnr». Yet another one displays small plots: Delete «drekbr» and remove duplicates. Relevés within overlapping big plots occur only once in RELEVÉ, but twice in DESIGN. No redundancy occurs in the data which otherwise causes mistakes in storage and retrieval.

The example of Fig. 3 is characterized by a specific spatial arrangement of plots. In other investigations locations are irrelevant but treatments merit attention. They should be taken into account by the data model. Let us consider a project in which succession in a grassland is studied under different management practices (Table 6). Some plots are cut annually (mtype 1), some second year (mtype 2) and others third year (mtype 3). Species composition is recorded annually and standing crop is measured of cut surfaces. Considering only the annual yield as a function of management-type and time, a record form could be as in Table 6. A new column is added every year for yield and age. The latter is defined as the number of years since last cutting. The design of the data structure should again take care of the questions to be answered:

- How does species composition change under management-type 1, 2 and 3?
- What is typical for a specific management-type, yield, or a particular species?
- Is the management-type irrelevant, but «age» characteristic for vegetation type?
- What is typical for vegetation of «age» 1, 2 or 3?

The structure chosen in Table 6 has again several drawbacks. As the project evolves, the number of rows and columns may grow. There are empty elements in the table which have to be distinguished from zero-entries. Age and yield occur in different columns. The length of these columns may vary if new relevés are introduced or old relevés omitted. In the relational design, a new relation YIELD is introduced (Table 7). In this, each characteristic like «ymtype», «yage», «year» and «yield» is found in just one column. As time passes, only the number of rows

Table 5. An extension of the data structure in Table 2.

VTYPE	(vrelnr	vcurty	vdate)
	1	1	84
	1	2	85
	2	1	84
	4	3	84
	5	3	84

Table 6. Ordering the results of a cutting experiment in grassland. Data are organized in a non-relation from.

relevé	mtype	age	yield	age 1	yield 1	age 2	yield 2
1	1	1	3.5	1	3.2	1	3.1
2	1	1	2.9	1	2.7	1	2.8
3	2	1	3.1	1	–	2	3.5
4	3	1	2.6	1	–	2	–

Table 7. Organization of the data in Table 6 in a relational design.

YIELD	(yrelnr	ymtype	yage	year	yield)
	1	1	1	0	3.5
	2	1	1	0	2.9
	3	2	1	0	3.1
	4	3	1	0	2.6
	1	1	1	1	3.2
	2	1	1	1	2.7
	1	1	1	2	3.1
	2	1	1	2	2.8
	3	2	2	2	3.5

(tuples) increases. The analysis of YIELD thus becomes a rather trivial task.

Conclusion

Handling data in a hierarchical approach is done quickly, but the relational design requires additional considerations. Questions to be asked by the investigation should be known. Changes in the design have to be taken into account, and significantly, a relational data bank of several arrays needs more effort to handle than a big single matrix. Computer programs have to take care of connections between different arrays. In the case of the relational design, such programs are difficult to write efficiently. Fortunately, a great number of relational software packages have been developed. They are now available under different operating systems. Especially micro-computers allow the implementation of low-cost systems. Before these are purchased, however, their limitations have to be checked.

The examples in this paper demonstrate that applications in community ecology may be complex. This could well be a reason why data banks in phytosociology are rare. Data banks in systematics, on the other hand, are easier to operate, as can be seen after Allkin and Bisby (1984).

Using a relational data bank does not necessarily mean to draw full advantages of the organization. As an example, any data bank system would handle Table 6. But all the drawbacks mentioned above would be retained. On the other hand, a relation like COMPOSITION in Table 2 could become very long. Only large systems can manage arrays of very big size. It follows from this that data banks in synecology require powerful machines and programs.

Large data banks for phytosociological data are difficult to design. On the other hand they can be attractive. A network of permanent plots could be used as a reference system for more detailed investigations in regional surveys. Approaches of this type can be expected in the near future.

REFERENCES

ALLKIN, and F.A. BISBY (ed.). 1984. *Databases in Systematics.* Academic Press, London 329 p.

ASTRAHAN, M.M., M.W. BLASGEN, D.D. CHAMBERLIN, K.P. ESWARAN, J.N. GRAY, P.P. GRIFFITHS, W.F. KING, R.A. LORIE, P.R. MCJOHNES, J.W. MEHL, G.R. PUTOLU, I.L. TRAIGER, B.W. WADE and V. WATSON. 1976. System R: Relational approach to Database Management. ACM Transactions on Database Systems 1, 2: 97-137.

BRAUN-BLANQUET, J. 1964. *Pflanzensoziologie.* 3rd ed. Springer-Verlag, Wien. 845 p.

CODD, E.F. 1970. A relational model of data for large shared data bases. Comm. ACM 13: 377-387.

DAUBENMIRE, B.F. 1968. *Plant Communities: A Textbook of Plant Synecology.* Harper & Row, New York. 300 p.

ELLENBERG, H. 1979: *Zeigerwerte der Gefässpflanzen Mitteleuropas.* Scripta Geobotanica IX, 2. Auflage, 122 p.

ELLENBERG, H. and F. KLÖTZLI. 1972. *Die Waldgesellschaften der Schweiz.* Eidg. Anst. forstl. Versuchswes., Mitt. 930 p.

ESTABROOK, G.F. and R.C. BRILL. 1969. The theory of the TAXIR accessioner. Mathematical Biosciences 5: 327-340.

FEOLI, E. and M. LAGONEGRO. 1982. Syntaxonomical analysis of beech woods in the Apennines (Italy) using the program IAHOPA. Vegetatio 50: 129-173.

GLEASON, H.A. 1926. The individualistic concept of the plant association. Bull. Torrey Botan. Club 53: 7-27.

GLEASON, H.A. 1939. The individualistic concept of the plant association. Am. Midland Naturalist 21: 92-110.

GREIG-SMITH, P. 1952. The use of random contiguous quadrats in the study of the structure of plant communities. Ann. Bot., Lond., 16: 293-316.

GREIG-SMITH, P. 1982. *Quantitative Plant Ecology.* 3rd ed. Studies in Ecology, Vol. 9, Blackwell, Oxford. 359 p.

LANDOLT, E. 1977: *Ökologische Zeigerwerte zur Schweizer Flora.* Veröff. Geobot. Inst. ETH, Stiffung Rübel, 64. 208 p.

MUELLER-DOMBOIS, D. and H. ELLENBERG. 1974. *Aims and Methods of Vegetation Ecology.* John Wiley & Sons, New York. 547 p.

ORLÓCI, L. 1978. *Multivariate Analysis in Vegetation Research.* 2nd ed. Junk, the Hague. 451 p.

PIELOU, E.C. 1977. *Mathematical Ecology.* J. Wiley & Sons, London. 385 p.

POORE, M.E.D. 1955. The use of phytosociological methods in ecological investigations. I-III. J. Ecol. 43: 226-244, 245-269, 606-651.

SCHLAGETER, G. and W. STUCKY. 1977. *Datenbanksysteme: Konzepte und Modelle.* Teubner, Stuttgart. 261 p.

WHITTAKER, R.H. 1967. Gradient Analysis of Vegetation. Biol. Rev. 42: 207-264.

Manuscript received: May 1985