Data structures and storage structures in scientific data base for multistage experiment

By L. BORZEMSKI

The aim of this paper is twofold. One of them is to present a multistage identification experiment and the other is concerned with the design of data base organization in that application. Some of the essential characteristics of multistage experiment are outlined. On the basis of the multiple name structure the data model is introduced. Then a framework for data representation within records is proposed and analyzed. It is shown that well-known multiple attribute retrieval methods can be applied in this case.

1. Introduction

The fast growing computer capability for data handling focused attention on the construction of powerful information laboratory systems which could carry out the process of automated experimentation, especially with the enormous amount of data. In scientific laboratory calculations there are many examples of tasks that require the processing of large data sets.

The aim of this paper is twofold. One of them is to present the multistage identification experiment environment and the other is concerned with the design of storage structures in that application. The multistage identification is vital to the efficient data manipulation in system identification. There are many aspects of data manipulation that require data base supporting.

The data model is presented in terms of Turski's data structure theory [7]. Physical representation of the data within an information system in the multistage experiment environment is proposed and analyzed. It is shown that well-known multiple attribute retrieval methods can be applied in this case.

2. The multistage experiment environment

The following short description is only intended to indicate the nature of the tasks undertaken in the multistage identification. The details have been published in [3, 5]. Fig. 1 shows a flow chart for the multistage identification experiment. In the multistage identification we perform the identification of the system in such a way

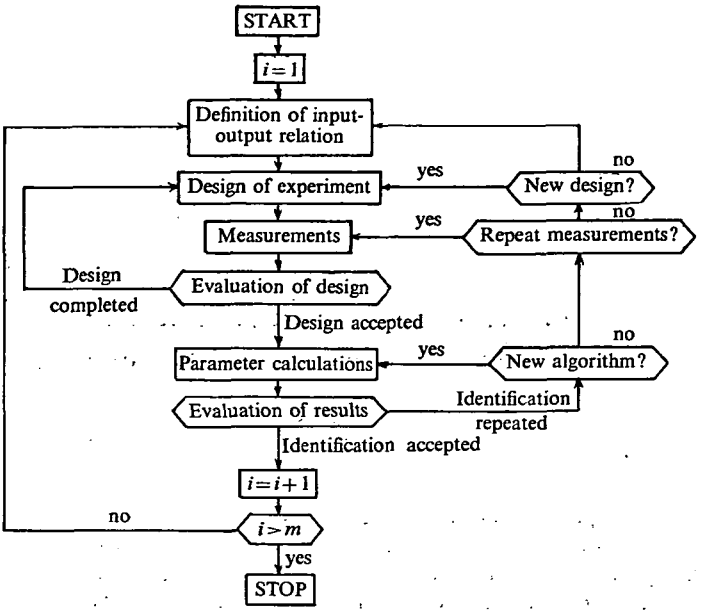

Fig. 1
Flow chart for the multistage identification experiment

that the global model $\bar{y}=\Phi(x_1, x_2, \ldots, x_m, a_m)$ is considered to be decomposed into m submodels $a_{i-1}=\Phi_i(x_i, a_i)$, $a_0 \triangleq \bar{y}$, $i=1, 2, \ldots, m$. We can measure input signals at each stage and output signals at the first stage only. The strategy of experiment is falling into m stages where at each stage we perform N_i elementary experiments, where

$$N_i = \begin{cases} 1, & \text{for } i = m, \\ \prod_{l=i+1}^{m} n_l & \text{for } i < m. \end{cases}$$

Throughout this paper each elementary experiment is assumed to have the following experiment attributes: identifier of the experiment NR_i, number of observations n_i, identification criterion Q_{i,n_i}, matrix of inputs X_{i,n_i}, matrix of outputs A_{i-1,n_i}, vector of model parameters a_{i,n_i}, table of vectors M_i which consists of the input vectors at the $(i+1)$, $(i+2)$, ..., m-th stages assumed constant during the identification at the i-th stage. The above ordering of experiment attributes will be assumed through this study. To obtain the matrix of observations of a_{i-1}, $i=\overline{2, m}$, it is necessary to repeat the experiments at the $(i-1)$-th stage n_i times assuming different values of x_i. It can be seen that for $i<m$ vector of model parameters a_i at the i-th stage is a column of output matrix $A_{i,n_{i+1}}=[a_{i,1}, \ldots, a_{i,n_{i+1}}]$ at the $(i+1)$-th stage. For every table M_{i-1} at the $(i-1)$-th stage there exists the table of vectors at the i-th stage which shares the same input vectors. For $i<m-1$ there exists for every table M_i' the table M_i'' at the same stage which differs only in one element, i.e. one vector.

If the designs of elementary experiments at the first stage are the same then all X_{1,n_1} matrices contain the same data.

The list of the system users consists of the experiment design programs, programs for identification, control of experiment and statistical calculations, and of experimentators, as well.

There exists a number of anticipated user requests [2, 4]. Almost all users are rather non-computer oriented and the total task coding in high level language is divided between them. Experimental data are collected and processed in a complex way. Tasks involve the iterative execution of several computer programs, each requiring data generated by the others in addition to user input data. The data are generalized FORTRAN arrays of numeric data. Multiple read/write requests for the same data in different applications programs are observed. Users manipulate on different data aggregates (e.g. single numeric data, vectors, arrays). Some data are collected and demanded in different ways, for example input matrix is generated vector by vector but requested also row by row. The data in a multistage experiment have some distinct characteristics, namely the regularity in their multiple relationships, redundancy and constant growth. Usually, moderate size data bases are involved but the size is increased when the data from experiments carried out at the different laboratories are to be bound together. In [2] an experiment with three stages is considered where the size of data base is of the order of 10^7 bytes of "pure" information, without any organizational data.

These characteristics give rise to the need for data base to support data manipulation in computer-based multistage experiment laboratory system. It should be noted that in laboratories we use minicomputers or microcomputers that considerably restrict us in data base facility choise.

3. Data description

The data model will be presented in terms of Turski's data structure theory [7]. Then data are considered to be ordered pairs (n, v) such that $n \in \mathcal{N}$, $v \in \vartheta$, where $\mathcal{N}$ is a denumerable set of elements called "names" which distinguish the entity in the real or abstract world and ϑ is any set of values considered as the collection of the information pertaining to the properties termed by the name.

We shall use a multiple name [2] which is defined as $\mathbf{n}_\alpha = \bigcap_{i \in \alpha} (n_i = a_i)$, where α is a string of integers selected unrepetitively from the set $\{1, 2, \ldots, r\}$, $a_i \in A_i$ — domain value of name part n_i, $\mathbf{n}_\alpha \in \mathcal{N}$. In this paper we assume that each name part takes value from the finite subset of natural numbers with cardinality $\overline{A}_i$.

Using the above approach we can construct r — level ordered unbalanced multiple name tree (sorted lexicographically) with the dummy root at the top so that the unique path connecting the root node to a terminal node corresponds to a distinct multiple name $\mathbf{n}_\alpha$ with $\alpha = 1, 2, \ldots, r$.

Let us first define domain sets A_i in such a way that name part n_i describes the multistage experiment number, stage number, experiment number at given stage, experiment attribute number, observation number and element number, for $i = \overline{1, 6}$, respectively. It is obvious that n_5, n_6 or n_6 are greater than one only for arrays or

vector data, according to the n_4 value. The cardinal numbers of the sets A_i, $i=\overline{1,6}$ are then defined in the following way

$$\bar{A}_1 = E, \qquad \bar{A}_4 = 7$$

$$\bar{A}_2 = \max_{k=\overline{1,E}} \{m_k\}, \qquad \bar{A}_5 = \max_{\substack{k=\overline{1,E} \\ i=\overline{1,m_k}}} \{n_{k,i}\},$$

$$\bar{A}_3 = \max_{k=\overline{1,E}} \{N_{k,1}\}, \qquad \bar{A}_6 = \max_{\substack{k=\overline{1,E} \\ i=\overline{1,m_k}}} \{s_{k,i}, k_{k,i}, l_{k,0}\},$$

where the following parameters for the i-th stage of the k-th multistage experiment, $i=\overline{1,m_k}$, $k=\overline{1,E}$ are given:

$n_{k,i}$ the number of observations,
$s_{k,i}$ the number of inputs,
$l_{k,i-1}$ the number of outputs,
$k_{k,i}$ the number of unknown parameters,
$N_{k,i}$ the number of elementary experiments.

E and m_k denote the number of multistage experiments and the number of stages of the k-th multistage experiment, respectively.

We also define in Table 1 three additional parameters $q_{k,i}^{(p)}$, $h_{k,i}^{(p)}$, $g_{k,i}^{(p)}$, where $k=\overline{1,E}$, $i=\overline{1,m_k}$, $p=\overline{1,\bar{A}_4}$.

Table 1. Definition of $q_{k,i}^{(p)}$, $h_{k,i}^{(p)}$, $g_{k,i}^{(p)}$ parameters

p	attribute	$q_{k,i}^{(p)}$	$h_{k,i}^{(p)}$	$g_{k,i}^{(p)}$
1	NR_i	1	1	1
2	n_i	1	1	1
3	Q_{i,n_i}	1	1	1
4	X_{i,n_i}	$n_{k,i}s_{k,i}$	$n_{k,i}$	$s_{k,i}$
5	A_{i-1,n_i}	$n_{k,i}l_{k,i-1}$	$n_{k,i}$	$l_{k,i-1}$
6	a_{i,n_i}	$k_{k,i}$	1	$k_{k,i}$
7	M_i	s_z	m_k-i	s_M

where

$$s_z = \begin{cases} \sum_{j=i+1}^{m_k} s_{k,j}, & \text{for } i < m_k, \\ 0, & \text{for } i = m_k. \end{cases}$$

$$s_M = \begin{cases} \max\limits_{\substack{k=\overline{1,E} \\ i=\overline{2,m_k}}} \{s_{k,i}\}, & \text{for } i < m_k, \\ 0, & \text{for } i = m_k. \end{cases}$$

The model concept can be extended for any set of known experiment attributes with appropriate $q_{k,i}^{(p)}$, $h_{k,i}^{(p)}$, $g_{k,i}^{(p)}$ parameters.

The great advantage of this data description method is that the construction of the multiple name tree reflects the structural properties of data aggregates in a multistage experiment. To illustrate this consider a data base pertaining to a multistage experiment in which we have three stages with four, two, three observations at each stage, respectively. At each stage the models have two, one and one input signals respectively, and four, two, three model parameters. We consider only one output at the first stage. Then, we have got six, three and one experiments at the first, second and third stage, respectively. Within this example, $\mathbf{n}=(1, 1, 2, 4, 3, *)$ describes the third vector of the input matrix in the second experiment at the first stage (Fig. 2). This way we can indicate all data aggregates, not necessary to be logically clustered but others, partitioning through a data base as well. For instance, $\mathbf{n}=(1, 1, *, 3, *, *)$ is related to all identification criterion values at the first stage.

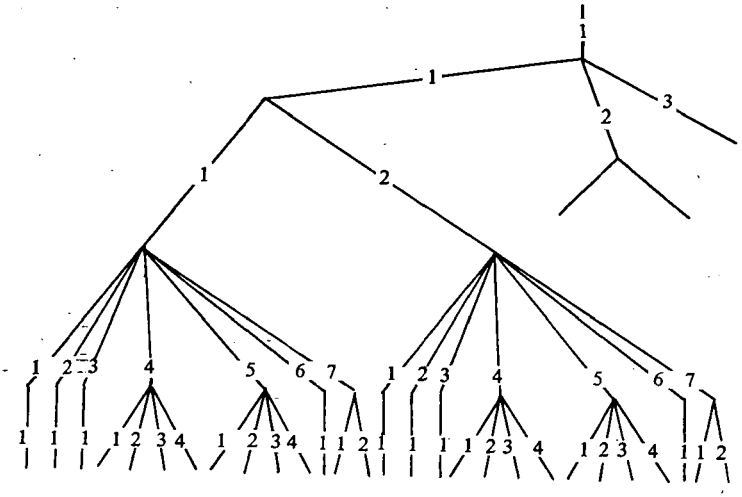

Fig. 2
A part of the multiple name tree for $\mathfrak{z}=5$.

In defining our name structure (the name admissibility verification algorithm is given in [1]) we note that the logical structure of data is stable in the sense of its construction and data relationships. As far as we could do, we have took advantage of this in designing the multiple name tree which exactly describes the real world. Another property which is observed is that the multiple name tree ensures the hierarchical clustering of data in the so-called "top-down" searching. Since this tree is given when all required multistage experiment parameters are known then almost all data management issues can be handled in arithmetic way [2]. In particular, data base storage structures can be constructed using this view what we will present in the next section. In general, the user of a data base may ask a wide variety of questions about the data that are stored. However, in this application the types of queries can be anticipated and the storage structures may be designed to handle them with suitable cost.

4. Physical data organization

The data within an information system may be stored in very different forms. Traditionally, the data base is a series of "physical" records which are formed by interconnecting some set of data items via storage structure. Each data item is a representation of (attribute, value) pair which characterizes an entity. Precise definitions for that can be found in [6]. In a business system environment there are many data base organizations which support the classification of user concepts into entities, attributes and relationships. In the multistage experiment environment we need some way of having unifying framework to represent data described in the previous section. The way this is done is to have a set of records such that each record attached to a terminal node of a multiple name tree represents a data corresponding to the name path $\mathbf{v}$ connecting the root node to that terminal node. One can easily see that only one record is attached to each terminal node. Then, for a multiple name tree with all leaves at depth $\mathfrak{z}$[1] a record stores information pertaining to the following entities:

— complete multistage experiment,
— all experiments at a given stage,
— an elementary experiment,
— an experiment attribute,
— a vector,
— an element (variable)

for $\mathfrak{z}=\overline{1, 6}$, respectively.

Note, that for the particular values of $\mathfrak{z}$ we found that different terminal nodes can refer to the same physical data, for example if $\mathfrak{z}=4$, $h=\overline{1, N_i}$ then $\mathbf{n}=(1, i, h, 4, *, *)$ indicates the same input matrix shared in h experiments at the i-th stage. To avoid redundancy there can be one storage copy of the record which stores appropriate values according to the multistage experiment strategy. Appropriate algorithm for recognizing these situations has been developed [2].

Other meaningful parameter e is to describe structural properties of data stored in a record. It is defined in a similar way and indicates how the data within a record is structurally divisible. If $\mathfrak{z}=e$, we assume that physical record stores structurally nondivisible data, e.g. for $\mathfrak{z}=4$, a value of experiment attribute. If $e>\mathfrak{z}$, it results that every record at level $\mathfrak{z}$ stores a sequence of data items which can have either one or more elements, i.e. for $\mathfrak{z}=4$, $e=5$ we obtain vectors or variables according to the $\mathbf{v}_4$ value. The data items are ordered in lexicographically ascending order of their multiple names. We also define parameter f which depends on the manner in which the physical boundaries of data items are fixed in a record, considering four storage formats, namely positional, relational, indexed and labeled [6].

Now we develop a series of equations for evaluating the space requirements under each of combination of $\mathfrak{z}$, $e, f, \mathbf{v}$ values. The results are given in Table 2. The storage allocation scheme assumes that all elements are allotted the same number of machine storage units (e.g. words, bytes). It depends on the storage allocation for REAL/DOUBLE PRECISION variables. We obtain $b_j(z, \mathfrak{z}, \mathbf{v}_j)$ storage units for representing the data of name $\mathbf{v}_j$ within the j-th record, $j=\overline{1, F(z, \mathfrak{z})}$ at the $\mathfrak{z}$-level

[1] The root of a tree lies at depth 0; the son of a node at depth $(i-1)$ lies at depth i.

Table 2. Equations of space allocation

$\mathfrak{z}$	$F(z, \mathfrak{z})$	$b_j(z, \mathfrak{z}, \mathbf{v}_j)$
1	E	$\sum_{i=1}^{m_{v_1}} N_{v_1,i} \sum_{p=1}^{\bar{\bar{A}}_4} g_{v_1,i}^{(p)}$
2	$\sum_{k=1}^{E} m_k$	$N_{v_1,v_2} \sum_{p=1}^{\bar{\bar{A}}_4} g_{v_1,v_2}^{(p)}$
3	$\sum_{k=1}^{E} N_k$	$\sum_{p=1}^{\bar{\bar{A}}_4} g_{v_1,v_2}^{(p)}$
4	$\bar{\bar{A}}_4 \sum_{k=1}^{E} N_k - E$	$g_{v_1,v_2}^{(v_4)}$
5	$\sum_{k=1}^{E} \sum_{i=1}^{m_k} N_{k,i} \sum_{p=1}^{\bar{\bar{A}}_4} q_{k,i}^{(p)}$	$h_{v_1,v_2}^{(v_4)}$
6	$\sum_{k=1}^{E} \sum_{i=1}^{m_k} N_{k,i} \sum_{p=1}^{\bar{\bar{A}}_4} g_{k,i}^{(p)}$	1

	$F_j(z, \mathfrak{z}, \mathbf{e}, \mathbf{v}_j)$			
$\mathfrak{z}$ \ e	1	2	3	4
1	1	m_{v_1}	N_{v_1}	$\bar{\bar{A}}_4 N_{v_1} - E$
2		1	N_{v_1,v_2}	$\bar{\bar{A}}_4 N_{v_1,v_2} - E\xi(v_2)$
3			1	$\bar{\bar{A}}_4 - \xi(v_2)$
4				1
5				
6				

$\mathfrak{z}$ \ e	$F_j(z, \mathfrak{z}, e, \mathbf{v}_j)$	
	5	6
1	$\sum_{i=1}^{m_{v_1}} N_{v_1,i} \sum_{p=1}^{\bar{\bar{A}}_4} q_{v_1,i}^{(p)}$	$\sum_{i=1}^{m_{v_1}} N_{v_1,i} \sum_{p=1}^{\bar{\bar{A}}_4} g_{v_1,i}^{(p)}$
2	$N_{v_1,v_2} \sum_{p=1}^{\bar{\bar{A}}_4} q_{v_1,v_2}^{(p)}$	$N_{v_1,v_2} \sum_{p=1}^{\bar{\bar{A}}_4} g_{v_1,v_2}^{(p)}$
3	$\sum_{p=1}^{\bar{\bar{A}}_4} q_{v_1,v_2}^{(p)}$	$\sum_{p=1}^{\bar{\bar{A}}_4} g_{v_1,v_2}^{(p)}$
4	$q_{v_1,v_2}^{(v_4)}$	$g_{v_1,v_2}^{(v_4)}$
5	1	$h_{v_1,v_2}^{(v_4)}$
6		1

Table 2. Equations of space allocation (cd)

of data aggregation. The number of records $F(z, \mathfrak{z})$ is the number of leaf nodes at depth $\mathfrak{z}$. We assume z is a vector of parameters which characterizes the multistage experiment i.e. number of observations, number of inputs, outputs and model parameters, and number of multistage experiments within an information system. The total length of the j-th record w_j is b_j+c_j, where c_j is equal to $I+\delta(\mathfrak{z}, f) \cdot F_j(z, \mathfrak{z}, e, \mathbf{v}_j)$. Within a file every record is assigned a label (identifier) which occupies I storage units. $\delta(\mathfrak{z}, f)$ is the additional space required by storage technique of data items within a record per data item. $F_j(z, \mathfrak{z}, e, \mathbf{v}_j)$ is a number of data items within the j-th record and $\xi(i)=1$, if $i=m_k$, and $\xi(i)=0$, otherwise. In the similar manner we also find number of storage units for representation each data item within a record [2].

It is pointed out that the above physical representation results in variable length of records. In the event that records may not be divided between buckets (a restriction posed by the majority of operating systems) then one can quickly determine the $\mathfrak{z}$'s value which satisfies this constraint.

The multiple attribute retrieval methods have been found useful for physical record positioning. In the most of them the storage scheme is dictated by the primary key. In [2] we describe an algorithm which generates the identifier on the basis of a mapping from the space of admissible multiple names into the space of integers.

Next, assuming that each part name correlates with an attribute we can obtain six versions of each multiple attribute method. One can determine the $\mathfrak{z}$'s, e's and f's that minimize the expected operational cost of the system [2]. Most of current multiple attribute access methods (for example, Inverted List, Multilist) require storing attribute values in the records [6]. Then a record additionaly contains the identifier and values of indexed attributes. However, in the doubly-chained tree organization the indexed attribute value are not stored in the records. Each case is met with appropriate value of $\delta(\mathfrak{z}, f)$. A detailed comparison of the average retrieval time per query and storage requirements of several current methods can be done.

Considering the limited space of this paper we refer the reader to [2] for the results. Moreover, the mapping function mentioned above establishes a new record addressing technique which further improve the system performance. It will be published elsewhere.

The data base management system which provides a high level access to the data base in the multistage experiment has been implemented at the Technical University of Wrocław [3].

5. Conclusions

Data base organization displays the potential profit in data management efficiency in the multistage experiment environment. Since the characteristics of this application strongly motivate a new strategy for data storage and retrieval, this data base environment was analyzed. The data model has been proposed. In this paper we have also introduced physical data organization which has been found useful for any multiple attribute retrieval method.

TECHNICAL UNIVERSITY OF WROCŁAW
INSTITUTE OF CONTROL AND SYSTEMS ENGINEERING
JANISZEWSKIEGO 11/17,
50—372 WROCŁAW, POLAND

References

[1] BORZEMSKI, L., Data management in a multistage experiment control systems, Proc. Int. Conf. Systems Engineering, Coventry, Sept. 1980.
[2] BORZEMSKI, L., A methodology and data base management algorithms for a multistage experiment control (In Polish), Ph. D. Thesis, Wrocław, 1979.
[3] BORZEMSKI, L., S. LEBIEDIEWA, Data base organization in the multistage identification experiment, IFAC Workshop on Scientific Experiments and Laboratory Procedures SELPA, Smolenice, Nov. 12—14, 1980.
[4] BUBNICKI, Z., Identification of control plants, Elsevier — PWN, Amterdam — Warszawa, 1980.
[5] BUBNICKI, Z., On the multistage identification, *Systems Sci.*, v. 3, 1977, pp. 207—210.
[6] MARTIN, J., *Computer data-base organization,* Prentice-Hall, New York, 1977.
[7] TURSKI, W. M., A model for data structures and its applications, *Acta Inform.*, v. 1, 1971, pp. 26—34.

(Received Dec. 3, 1981)