Information Technology: Research and Development (1982), *1* (285–300)

A NEW APPROACH TO THE DESIGN OF STRUCTURED BIBLIOGRAPHIC RECORDS*

D. D. KOUVATSOS AND E. J. YANNAKOUDAKIS

Postgraduate School of Studies in Computing, University of Bradford, Bradford, BD7 1DP, West Yorkshire, UK

(*Received 18 January 1982, revised 5 May 1982*)

ABSTRACT

An important aspect in the design of retrieval systems is that some attributes are more important than others for the identification and hence for the retrieval of certain records. Furthermore attributes are not independent of one another. The present paper proposes a new methodology which aims to break up the attributes into quasi-independent sets so that sets of attributes that 'belong together' are in fact stored and accessed together. A theoretically based decomposition approach is employed to identify relatively independent groups of items with strong internal connectivity leading to a structured information record. The problem is fully exemplified in bibliographical databases where a large number of different items are necessary and the access time for one or more strongly connected items is a crucial factor for system efficiency. The method is implemented and discussed using MARC (MAchine Readable Catalogue) records.

1. INTRODUCTION

Although systems analysis has provided a number of valid practical techniques for the determination of user functional requirements, it has failed so far to develop a complete and coherent methodology to integrate these techniques into an overall and objective cycle. In particular the design of files and the ordering relations of all possible data items or fields† that may be incorporated, is totally system dependent and is determined by the operations that are to be performed upon the files. While the choice of data items depends on system components and requirements, the design of storage allocation schemes allows the independent development.

What the data analyst requires is a tool that can be used under any environment where the different fields are clearly defined and any relationships among them are

* Part of this paper was presented (Kouvatsos and Yannakoudakis, 1978) at the seventh I.U.C.C. computer science colloquium, September 1978, England.

† A field is defined as the smallest semantically meaningful element of information within a record.

naturally revealed. To this end, the emphasis must be on application-independent design criteria for the construction of the logical record which in the past has been oriented in an *ad hoc* way (Codd, 1972). After the components of the logical record have been analyzed, the design of the physical record can take place and storage structures can be determined accordingly. It is important, at this stage, to distinguish between computer-dependent storage slots and computer-independent data elements and relationships of accessibility or ordering among them (D'Imperio, 1969).

From the analysis of the elements of the logical record it will become obvious which data items are to become record fields and which may be incorporated with others. The need for determining the relationships that may exist among different fields has been recognized by various authors (Benner, 1967; Isiao, 1971) and has been considered essential, prior to the design of the physical record. The characteristics of each field and its relation with others have been discussed but no attempt has been made to determine formally these relations and to establish design criteria based on theoretical considerations. A number of fundamental questions must be answered during the design process, regardless of whether they are theoretically evaluated or not. These include:

1. To what extent does the existence of a particular field contribute towards the description of the record?
2. Does the presence of a field necessitate the presence of others?
3. To what extent does a field contribute towards unique identification of the record?
4. How can the interaction (if any) between two fields be determined?

Other more immediate questions include: How often the field exists, how long it is, how often it is required, etc.

Record structures are designed so that the record fields themselves can be processed in an optimal way. This may mean ease of access to a field in the minimum possible time. To achieve this, the fields must be placed within the record in such a way that they conform with the access requests that are likely to occur. The aim of the present paper is to propose a new methodology for the design of a structured* logical record by considering the interrelationships among the fields and decomposing them to form subgroups which can be bound together during a storage allocation process.

2. A DECOMPOSITION APPROACH

The optimal design of a multifield record is a very complex problem since it will determine the viability and efficiency of the system as a whole. With the majority of the present record-oriented systems, the entire record is accessed and read into core even if the request is for specific fields within the record. A considerable amount of time is therefore wasted since other unrequired fields of the record are also read. Various authors have proposed solutions to this problem which include the partitioning of the file records into subrecords resulting in a master file and subfiles.

* A structured logical record is defined as a uniquely identifiable data-carrying entity consisting of a set of interrelated fields linked in such a way as to conform with a predefined structure based (in the ideal case) on theoretical design criteria.

It is claimed that in such a partitioned file system it will be possible to selectively access only those fields required and thus reduce the data transfer time and cost.

Benner (1967) proposes a heuristic method for partitioning the logical record into several physical subrecords by analysing the relations between the two. Kennedy (1972) allows for the existence of several subfiles to accommodate the subrecords that result from the subdivision of a record and presents a zero-one integer programming algorithm for the problem. Hoffer (1974) considers the dependence between fields in a logical record and formulates the problem as a non-linear integer programming model which then has to be reduced in size by cluster analysis prior to its final solution. More recently, Babad (1977) has discussed the implications when a record is partitioned into fixed and variable length parts and presented an interesting mathematical programming model for minimizing storage and access time costs. He assumes independence among fields, that is, the occurrence of a field in a specific record is not related to the occurrence of other fields in the same or other records. This however is not always the case, for example, with bibliographical record fields. The need for additional research work in this area has been widely recognized in the literature.

The optimal design of the logical record prior to its physical implementation is here examined for the first time under the light of decomposition, i.e., the grouping together of those strongly connected fields which are likely to be accessed simultaneously in an information retrieval environment and which can, at the same time, be used for record identification functions.

The design process concerning a multifield record is assumed to involve many interrelated fields. Identifying the underlying structure of the record must be the main objective of the data analyst. This may be achieved by subdividing the logical record into a number of relatively independent groups of fields which will govern any subsequent data storage scheme. Here, the formation of groups of fields is not arbitrary but based on theoretical foundations having two main advantages over an *ad hoc* procedure:

1. They establish a sound design tradition for data analysis.
2. They maintain continuity in record design, with the aid of mathematical notation, in which the experience gain can be utilized for future design purposes.

3. DEFINITIONS

The concept of the design of a structured record should achieve a union resulting in a harmony between two entities:

1. The definition of the problem in terms of a set of fields imposing demands on the structure itself.
2. The final structure (the solution) which the data analyst can control and attempts to shape so that each field is rightly positioned in relation to others for optimal accessibility.

An *interaction* between two fields expresses a logical interrelationship between them, under an information retrieval environment, where they may both be present in an acceptable request which can be fulfilled only when the combined discriminating power of the two is enough to identify the record or related records in the collection. Note that a high discriminating power shared between two fields constitutes a necessary but not sufficient condition to characterize a request as acceptable. For example in a library application, an acceptable request could involve 'personal author forenames', 'single surname' and 'title proper'. But a request consisting of 'Dewey decimal classification number', 'volume' and 'edition' alone does not form an acceptable request.

A 'working visualization' of the structured record is a linear undirected weighted graph G(M,L). Here the nodes form the set M of m, say, fields describing the record and the weighted links form the set L representing the l, say, interactions between pairs of fields.

The domain of all solutions D is the set of all different field arrangements and linking mechanisms forming individual structures, some of which will be inefficient and thus unacceptable. A set of binary random variables $\{y_i;\ i=1, \ldots, m\}$ is associated uniquely with the m fields of the system. Each y_i splits the domain D into two parts; a set of those solutions where field i is placed optimally (i.e., 'fits' together with the rest of the fields of the group) and takes the value of zero with probability q_i, i.e.,

$$q_i = Pr(y_i = 0) \tag{1}$$

and a set of those solutions where field i does not 'fit' with the rest in the group and takes the value of one with probability p_i, i.e.,

$$p_i = Pr(y_i = 1) \tag{2}$$

where $q_i + p_i = 1$. Domain D is represented diagrammatically in Figure 1.

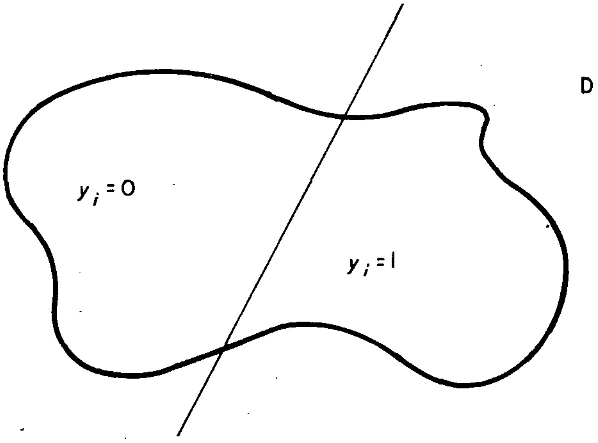

FIG. 1. The univariate cut of the domain

Similarly all joint probabilities associated with pairs of fields may be defined. For instance, the probability that both fields i and j are placed optimally is given by

$$q_{ij} = Pr(y_i = 0, y_j = 0) \tag{3}$$

Furthermore the strength of the interaction between pairs of fields (i, j) can be described by means of the correlation coefficient ϱ_{ij} between the random variables y_i and y_j, ϱ_{ij} is given by the ratio of covariance (y_i, y_j) over the products of the standard deviations of y_i and y_j. This has been used widely in the literature as a measure of interrelationship between any pair of random variables. Clearly, because of the binary nature of the random variables y_i and y_j, ϱ_{ij} is given by

$$\varrho_{ij} = (q_{ij} - q_i q_j)/(q_i p_i q_j p_j)^{1/2} \quad (4)$$

where $0 \leqslant \varrho_{ij} \leqslant 1$ $(i = 1, 2, \ldots\ldots, m-1$ and $j = i+1, \ldots\ldots, m)$. It is assumed that $q_{ij} > q_i q_j$, i.e., ϱ_{ij} is positive because there is no conflict in satisfying the joint optimal storage scheme for fields i and j.

4. THEORETICAL CONSIDERATIONS

The large number of interrelated fields is the main difficulty in the search for a good record design solution. A theoretical framework may be established by envisaging a decomposition criterion aiming to partition the vertices of graph G(M,L) into a number of minimally interacting subgraphs with strong internal connectivity. This will lead naturally to the selection of those groups of fields which contain the essential information about the entire system. In order to apply decomposition it is necessary to have some function which is used to measure the information contained within a system or subsystem, the objective being to minimize the information transfer between the various subgroups. To this end the concept of the entropy function

$$\mathrm{H(M)} = -\sum_{\sigma} \mathrm{P}(\sigma) \log \mathrm{P}(\sigma) \quad (5)$$

adopted by Shannon and Weaver (1964) is employed where σ is the state $\{x_1, x_2, \ldots x_m\}$, x_i being the value taken by y_i and $\mathrm{P}(\sigma)$ is the probability of occurrence of state σ. Note that the entropy function has been used by other authors (Hyvarinen, 1962; Sebestyen and Edie, 1964) as a criterion for grouping in classification analysis.

It has been shown (Kouvatsos, 1976) that the state probability $\mathrm{P}(\sigma)$ can be expressed in terms of the univariate probabilities $\{p_i\}$, $\{q_i\}$ and the correlation coefficients $\{\varrho_{ij}\}$ as

$$\mathrm{P}(\sigma) = \prod_{k=1}^{m} p_k^{x_k} q_k^{1-x_k} \prod_{i=1}^{m-1} \left(1 + \sum_{i=i+1}^{m} \alpha_{ij} \varrho_{ij}\right) \quad (6)$$

$$\text{where } \alpha_{ij} = (-1)^{x_i + x_j} \left(p_j^{1-x_j} q_j^{x_j} / p_i^{x_i} q_i^{1-x_i}\right) (p_i q_i / p_j q_j)^{1/2}$$

Formula (6) also holds for any substate $\lambda = \{x_{i1}, \ldots, x_{is}\}$, $s \leqslant m$.

Using formula (6) the entropy function (5) can be expressed to the second order of approximation of ϱ_{ij} as follows

$$\mathrm{H(M)} = - \sum_{\sigma} \prod_{i=1}^{m} p_i^{x_i} q_i^{1-x_i} \log \prod_{i=1}^{m} p_i^{x_i} q_i^{1-x_i} - \tfrac{1}{2} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \varrho_{ij}^2 \quad (7)$$

A similar formula to (6) holds for H(S), SCM.

By considering an arbitrary partition π of M into subsets $S_1, S_2, \ldots, S\mu$ such that $S_i \cap S_j = \phi$ and $\cup_\mu S_i = \mathrm{M}$, the following difference is evaluated

$$\left\{ \sum_{\pi} \mathrm{H}(S_i) - \mathrm{H(M)} \right\} \geqslant 0 \quad (8)$$

and interpreted as a measure of information flow between the subgraphs formed when G(M,L) is decomposed by π. By an appropriate evaluation of the difference (8), a general criterion of decomposition has been derived (Kouvatsos, 1976) which is given by minimizing

$$\sum_{\pi} \varrho_{ij}^2 \quad (9)$$

over all π where the summation is taken over all those pairs of nodes separated by an arbitrary partition π.

By applying the above criterion to the entire graph G(M,L) at the first level and to each of the resultant subgraphs at lower levels, a tree-like structure is obtained in a top-down manner. The root node represents the whole graph, the leaves of the tree, the individual nodes (fields), and the intermediate nodes—those subsets of M denoting the most independent groups of fields. At this point it is interesting to note that Yannakoudakis and Wu (1982) also use the entropy function as a measure for grouping fields with the aim of designing a new database model. However, their objective was to establish locally optimal quasi-equifrequent groups of fields whereas the present work refers to quasi-independent groups of fields.

To estimate the correlation coefficients $\{\varrho_{ij}\}$ using formula (4) the univariate and bivariate probabilities expressed by relations (1), (2) and (3) must be determined. The creation of a probabilistic model for this purpose necessitates the construction of m real functions q_i ; $i=1, 2, \ldots, m$ and at most $[m(m-1)]/2$ real functions q_{ij} ; $i=1, 2, \ldots, m-1$ and $j=i+1, \ldots, m$ which must satisfy all the necessary laws of probability. Evidently the univariate probability q_i must be a decreasing function as the number of links (relationships) associated with vertex (field) i increases. It may therefore be expressed by

$$q_i = \omega_{i1} + \omega_{i2} e^{-d_i} \text{ for all } i \quad (10)$$

where $\{\omega_{i1}\}$ are tuning parameters imposing a lower limit on all q_i such that $\omega_{i1} + \omega_{i2} = 1$ and can be chosen by the designer; d_i is a structural parameter expressing the connectivity of vertex i as the sum of the weights of the links on vertex i, i.e.,

$$d_i = \sum_{i \neq j} \varrho_{ij} \; ; \; i, j = 1, 2, \ldots, m \quad (11)$$

q_i decreases as the complexity of d_i on vertex i increases (Fig. 2).

The bivariate probability must satisfy the relation of independence when $\varrho_{ij}=0$, i.e.,

$$q_{ij}=q_i\,q_j \tag{12}$$

and must be probabilistically consistent, i.e.,

$$0 \leqslant q_{ij} \leqslant \min\{q_i, q_j\} \tag{13}$$

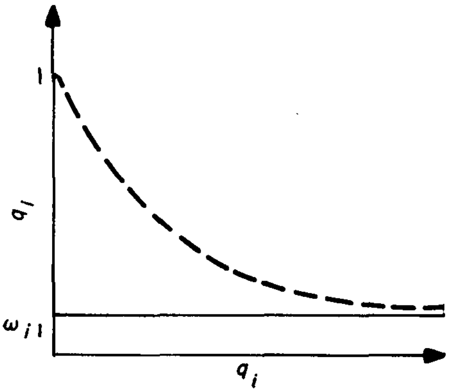

FIG. 2. Behaviour of q_i

Relationships (12) and (13) can be satisfied when q_{ij} is given by

$$q_{ij}=\begin{cases} q_i\,q_j+4\,q_i\,p_j\,f(\varrho_{ij}) \text{ if } q_i \leqslant q_j \\ q_i\,q_j+4\,q_j\,p_i\,f(\varrho_{ij}) \text{ if } q_j \leqslant q_i \end{cases} \tag{14}$$

for all $i<j$ where $f(\varrho_{ij})=e^{-\varrho ij}(1-e^{-\varrho ij})$

Using expressions (10) and (14) a system of l non-linear equations is formed ($l \leqslant m(m-1)/2$) and these are solved iteratively to provide objectively derived weights for the l links of the graph G(M,L). Further details of a more general mathematical model will be found in Kouvatsos (1980).

5. TOWARDS THE DESIGN OF A STRUCTURED RECORD

The advent of large databases, and in particular those of bibliographical nature where the designer is confronted with a multiplicity of distinctive descriptive elements, and therefore potential unique fields, has necessitated the design of standardized means of communication such as the MARC (Machine Readable Catalogue) format (UK MARC, 1975). This structure of MARC, as outlined in Figure 3, is capable of incorporating any of the existing record descriptors and has a potential capacity of uniquely identifying up to 25974 fields. From an analysis carried out at Bradford University (Ayres and Yannakoudakis, 1979) on MARC-based files, 241 different fields were counted as being used and their statistics were recorded. A typical MARC record for a bibliographical document requires approximately 600 characters. The variability of segments of fields is overcome by the use of a tag—a three-digit number identifying a segment—and each field within it is identified by a single alphabetic (A to Z) character.

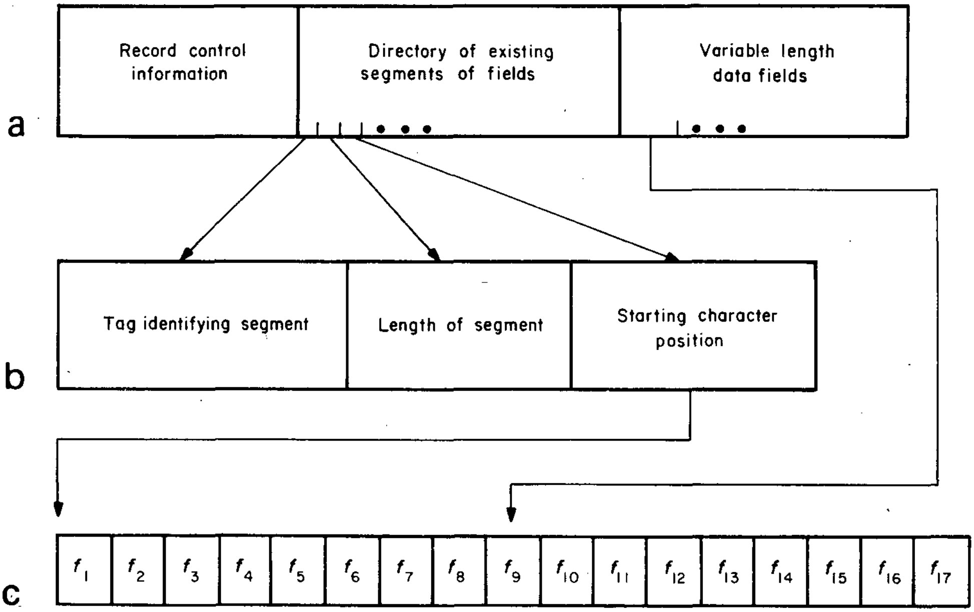
a
Record control information
Directory of existing segments of fields
Variable length data fields
b
Tag identifying segment
Length of segment
Starting character position
c
f1
f2
f3
f4
f5
f6
f7
f8
f9
f10
f11
f12
f13
f14
f15
f16
f17

FIG. 3. Logical structure of a MARC record. a = Outline record structure; b = directory structure for a single group; c = set of variable length fields (In the terminology of MARC, a field is called a subfield)

MARC has brought some degree of order within the bibliographical world and has become the corner-stone for the development of other more flexible structures such as MERLIN (MachinE Readable Library INformation) (Noerr and White, 1976) in which it is used for input/output functions. Here the user stores and retrieves fields to and from the database in virtual records. However, a higher level of technology is needed in order that these systems can realize their full potential. In the words of Ayres (1971):

> We may have to wait for this form of full utilization until the technology catches up with the complexity of the bibliographical situation.

On the other hand it may be argued that once the optimal logical design of a structure has been achieved, current technology will be capable of making it operational in the most efficient way.

Decomposition as a tool for the data analyst can to a large extent contribute towards the design of structured multifield records. In order to illustrate the implementation of the decomposition approach, a subset of MARC fields was selected (Table 1). Tag and subfield entries 9, 10 and 11 allow for the incorporation of other data items under the same tag and subfield. For example 245g is defined as volume or part number and 245h as volume or part title.

Table 1. A subset of MARC data descriptors

Tag and subfield	*Subfield name*
1. 041a	Language
2. 082a	Dewey decimal classification number
3. 100(00)a	Personal author forenames
4. 100(10)a	Single surname
5. 100(10)c	Date of birth of author
6. 245a	Title proper
7. 245b	Subtitle
8. 245d	Simple single author name
9. *	Volume
10. *	Part number
11. *	Edition
12. 250c	Editor
13. 250d	Statements not relating to editor, (i.e., with new introduction by . . . , with foreword by . . .)
14. 260a	Place of publication
15. 260b	Publisher's name
16. 260c	Date of publication
17. 260d	Full address of publisher

To determine whether an interaction between two fields i and j exists initially, their combined discriminating power (CDP) was estimated as follows

$$\mathrm{CDP}_{ij} = \begin{cases} 10^n/r_{ij} \text{ if } v_{ij} \geqslant 10^{n-2} \\ 10^2\, v_{ij}/r_{ij} \text{ if } v_{ij} < 10^{n-2} \end{cases} \tag{15}$$

where R is the total number of records in the database; r_{ij} is the total number of records containing the combined fields (i, j); v_{ij} is the potential number of different entries in the combined fields (i, j); n depends on hardware architecture such that no overflow occurs. Clearly

$$r_{ij} \leqslant \min(r_i, r_j) \leqslant \mathrm{R} \tag{16}$$

and

$$v_{ij} = v_i\, v_j \tag{17}$$

for $i, j = 1, 2, \ldots, m$ where r_i is the total number of records containing field i and v_i is the potential number of different entries in field i.

The CDP as is used at present will indicate the extent to which a unique set of codes can be generated from a set of fields for record identification purposes (Yannakoudakis *et al.*, 1980). Also a high CDP value is a necessary but not a sufficient condition to determine the existence or otherwise of an interaction between two fields. An additional requirement imposed within the present framework is that pairs of fields should constitute meaningful and realistic unions. Clearly the more fields one considers prior to the identification of a set of records the higher the chance for a smaller set of records to be retrieved. A 90 per cent CDP value is here taken as the minimum requirement for a potential interaction between two fields. Also the potential number of different entries v_{ij} for fields i and j is calculated empirically on the basis of the statistical information available (Ayres and Yannakoudakis, 1979) using formula (15). A bibliographic database containing 10^7 records was considered in the present study.

The CDP in conjunction with the criterion of a 'meaningful association' between two fields i, j will determine the presence or absence of an interaction (link) between the two. A total of $m(m-1)/2$ relations are examined and l links are established. The mathematical model then generates the values of the correlation coefficients $\{\varrho_{ij}\}$ which express the strength of the interaction between pairs of fields.

Only pairs of fields conforming with the above criteria were considered; the links of the graph are shown in Figure 4. For instance, it can be seen that 'Simple single author name' (No. 8) interacts with 'Editor' (No. 12) but not with 'Language' (No. 1). 'Date of publication' (No. 16) interacts with 'Editor' (No. 12) but not with 'Place of publication' (No. 14), etc. The mathematical model (Kouvatsos, 1980) was then used to generate the final connectivity weights $\{\varrho_{ij}\}$ which are included in Table 2. The range of values for ω_{i1} $(i=1, \ldots, 17)$ was between 0.22 and 0.26. One of the properties of the mathematical model is to generate high connectivity weights for related fields which have approximately equal degrees of vertices. This is in agreement with other works on decomposition (Hoffer, 1974). The present example is for illustration purposes and contains only 17 fields out of a possible 241. This resulted in a somewhat homogeneous graph which produced a narrow range of values between 0.933 and 0.949. However this does not affect the validity of the decomposition tree because the threshold ε is incremented in steps of 0.001.

Table 2. Final connectivity weights $\{\varrho_{ij}\}$

i	j	ϱ_{ij}	i	j	ϱ_{ij}	i	j	ϱ_{ij}
1	7	0.936	1	12	0.936	1	13	0.936
2	7	0.944	2	8	0.944	2	12	0.944
2	13	0.944						
3	4	0.949	3	6	0.946	3	7	0.936
3	8	0.936	3	12	0.936	3	13	0.936
4	5	0.941	4	6	0.946	4	7	0.935
4	12	0.935	4	13	0.935			
5	7	0.943	5	8	0.943	5	12	0.943
5	13	0.943						
6	7	0.933	6	8	0.933	6	12	0.933
6	13	0.933	6	17	0.938			
7	8	0.949	7	9	0.944	7	10	0.944
7	11	0.944	7	12	0.949	7	13	0.949
7	14	0.944	7	15	0.944	7	16	0.944
7	17	0.943						
8	9	0.944	8	10	0.944	8	11	0.944
8	12	0.949	8	13	0.949	8	14	0.944
8	15	0.944	8	16	0.944	8	17	0.943
9	12	0.944	9	13	0.944			
10	12	0.944	10	13	0.944			
11	12	0.944	11	13	0.944			
12	13	0.949	12	14	0.944	12	15	0.944
12	16	0.944	12	17	0.943			
13	14	0.944	13	15	0.944	13	16	0.944
13	17	0.943						

A decomposition algorithm (Kouvatsos, 1979) was used as a cutting function to snip away low value links less than or equal to a 'threshold' ε, where $0 < \varepsilon \leqslant 1$. Then the hierarchical tree presented in Figure 5 was generated. In interpreting the tree it will be seen that the original set of 17 fields has been broken down into a number of meaningful segments (groups of fields). For example, {3, 4, 6} and {7, 8, 12, 13} (indicated by heavy lines in Fig. 4) constitute complete requests that may be enriched (by union) with other fields at higher levels to form more complete requests under an information retrieval environment. By considering the tree in a bottom-up manner a synthesis tree can be obtained which can then be used as a blue-print to build the structured record.

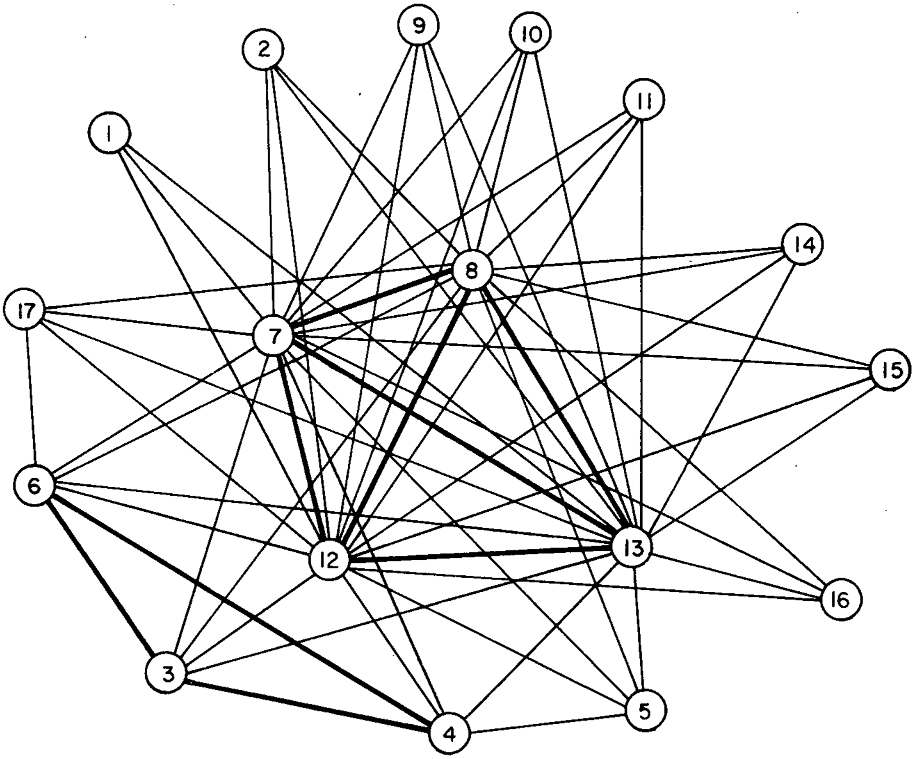

FIG. 4. Graphical representation

Following a careful interpretation of the results, the designer may choose to allocate contiguous storage slots to those fields of a group with high internal connectivity as expressed in Tables 3 and 4. Note that d_{gj} expresses the group connectivity given by

$$d_{gj} = \left(\sum_{i \varepsilon g_j} d_i - \sum_{\pi_j} \varrho_{ik} \right) /2 \; ; j=1,2, \ldots ,6; i, k \; \varepsilon \; \{1,2, \ldots ,17\} \qquad (18)$$

where the second sum is taken over all external links to g_j as indicated by partition

π_j. Therefore, the set of MARC fields shown in Figure 3(c) can be optimally restructured as in Figure 6.

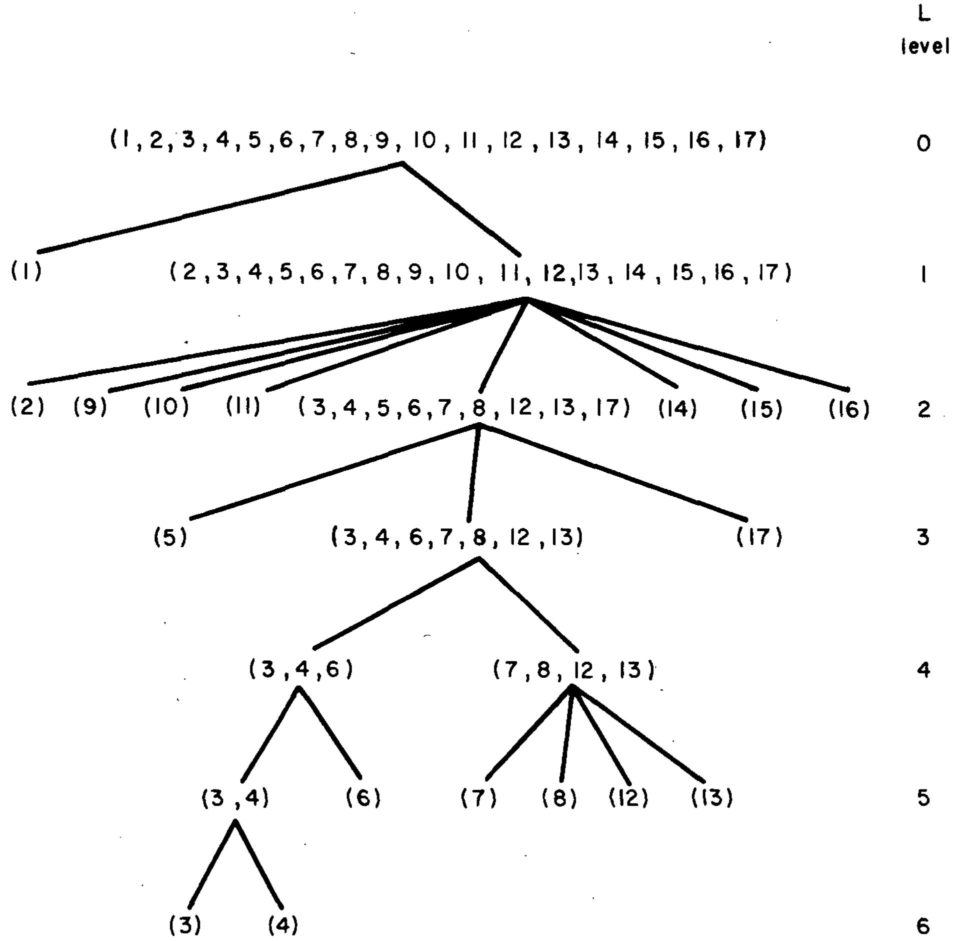

FIG. 5. Decomposition tree (L-level)

If a particular group of fields exemplifies a high degree of connectivity in comparison with others at the operational levels chosen by the designer, it may justify the allocation of a directly accessible slot without the use of a look-up table or a directory entry in the case of MARC. Internal connectivities, as shown in Tables 3 and 4, of subsequent groups at following levels, will determine the order of fields within a storage slot. The order of pointers themselves is governed by the same rules. The structure in Figure 6 is considered to be more efficient than the previous because it offers more operational flexibility and faster access times by traversing through the pointers.

Choosing the right linking mechanism, with the aid of pointers to suit the operations planned upon it, is by no means an easy task. It is hoped that a beneficial

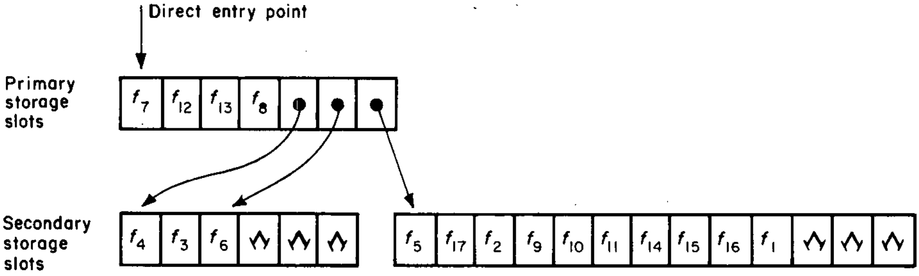

FIG. 6. A structured record

Table 3. Weighted degrees of fields (field connectivity)

Field *i*	*Connectivity* d_i
1	2.808
2	3.776
3	5.639
4	5.641
5	4.713
6	6.562
7	15.081
8	13.210
9	3.776
10	3.776
11	3.776
12	15.081
13	15.081
14	3.776
15	3.776
16	3.776
17	4.710

Table 4. Weighted degrees of groups (group connectivity)

Group *j*	1	2	3	4	5	6
Connectivity d_{g_j}	54.671	28.239	21.624	2.841	5.694	0.949

where $g_1 = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\}$
$g_2 = \{3, 4, 5, 6, 7, 8, 12, 13, 17\}$
$g_3 = \{3, 4, 6, 7, 8, 12, 13\}$
$g_4 = \{3, 4, 6\}$
$g_5 = \{7, 8, 12, 13\}$
$g_6 = \{3, 4\}$

approach for the designer would be to select that mechanism which at least conforms with and is capable of utilizing the 'natural' groupings revealed by decomposition. A general record-linking mechanism based on level 0 of the tree is presented in Figure 7 where each group g is accompanied by three link entries. Group g_0 has the highest internal connectivity d_{g0} and is directly accessed and is followed by all other groups arranged in descending order of connectivity. The link elements are fully utilized before the ones at lower levels are used. The availability of CDP makes it possible for each group to have its own direct entry point if access time becomes a crucial factor for optimal system performance. Additional research is underway to implement and test the structure.

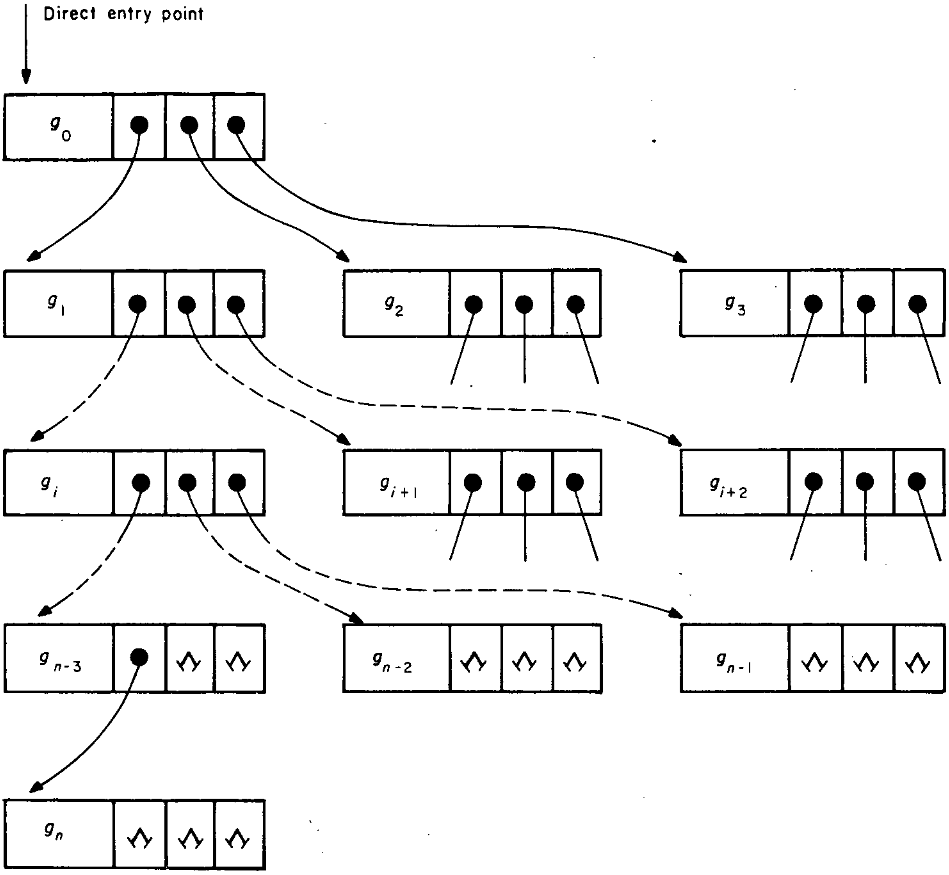

FIG. 7. A general record-linking mechanism

6. CONCLUSIONS

In the present paper the notion of interrelationships among fields of a record was introduced and 'decomposition', as a tool for identifying the underlying record structure, was described. On the basis of this analysis a synthesis strategy was adopted to order and distribute fields within the storage area of a record. The unified design methodology adopted involved the following steps:

1. Formulate graph G(M,L).
2. Generate weighted linkages of G(M,L).
3. Decompose G(M,L) to obtain the corresponding tree.
4. Process decomposition tree to form the synthesized structure.
5. Use structure as a blue-print to build the system.
6. Evaluate system performance and remedy defects.

It is conjectured that the alternative design scheme proposed, will save a considerable amount of time when MARC records are processed and could be implemented for a generalized record construction. Theoretical approaches such as the one described will, it is believed, rectify some of the defects in systems analysis and contribute towards a complete, coherent and system-independent design methodology.

ACKNOWLEDGEMENT

We would like to thank the editor Professor C. J. van Rijsbergen for his constructive comments and criticisms.

REFERENCES

Ayres, F. H. (1971) The case against MARC: how strong is it? *Library Association Research 73*, 130–131, 142.

Ayres, F. H. and Yannakoudakis, E. J. (1979) The bibliographic record: an analysis of the size of its constituent parts. *Program 13*, 127–142.

Babad, J. M. (1977) A record and file partitioning model. *Communications of the ACM 20*, 22–31.

Benner, F. H. (1967) On designing generalised file records for management information systems. *Proc. AFIPS 1967 FJCC.* pp. 291 – 303. Montrale, N.J.: AFIPS Press.

Codd, E. F. (1972) Further normalisation of the data base relational model. *Courant Computer Science Symposium 6, Data Base Systems.* (R. Rustin, ed.) pp. 33–64. Englewood Cliffs, N.J.: Prentice-Hall.

D'Imperio, M. E. (1969) Data structure and their representation in storage. *Annual Review in Automatic Programming 5*, 1–75.

Hoffer, J. A. (1974) *A cluster approach to the generation of subfiles for the design of a computer data base.* Ph.D. Dissertation, Department of Operations Research, Cornell University, Ithaca, N.Y.

Hyvarinen, L. (1962) Classification of qualitative data. *British Information Theory Journal*, 83–89.

Isiao, D. K. (1971) A generalized record organisation. *IEEE Trans. Comp. 20*, 1490–1495.

Kennedy, S. R. (1972) *A file partition model.* Pasadena, California: Cal. Tech. (Information Science Technical Report 2).

Kouvatsos, D. D. (1976) Decomposition criteria for the design for complex systems. *International Journal of Systems Science 7*, 1081–1088.

Kouvatsos, D. D. (1979) A unified algorithm for large-scale system design. *Proceedings of the international symposium on measurement and control 2*, pp. 639–644. Zurich: Acta Press.

Kouvatsos, D. D. (1980) A binding model for large-scale system design. *Advances in Control.* Vol. II. (D. G. Lainiotis and N. S. Tzannes, eds.) pp. 153–162. Dordrecht: D. Reidel Publishing Company.

Kouvatsos, D. D. and Yannakoudakis, E. J. (1978) A new approach to the design of structured multifield records. *Seventh I.U.C.C. Computer Science Colloquium, Lancaster University, England, September 11–15.*

Noerr, P. L. and White, M. C. (1976) MERLIN design of a national bibliographic database. *Systems for Large Data Bases.* (P. C. Lockemann and E. J. Neuhoid, eds.) pp. 211–222. Amsterdam: North-Holland Publishing Company.

Sebestyen, G. S. and Edie, J. (1964) *Pattern recognition research.* Bedford, Mass.: Air For Cambridge Research Laboratory. (Report 64–821, AD 608–692).

UK MARC Manual. (1975) *First Standard Edition.* London: The British Library, Bibliographic Services Division.

Yannakoudakis, E. J., Ayres, F. H. and Huggill, J. A. W. (1980) Character coding for bibliographical record control. *The Computer Journal 23*, 53–60.

Yannakoudakis, E. J. and Wu, A. K. P. (1982) Quasi-equifrequent group generation and evaluation. *The Computer Journal 25*, 2.