Int. J. Appl. Math. and Comp. Sci., 1999, Vol.9, No.4, 965–977

ON THE EQUIVALENCE OF PROBLEM-ORIENTED DATABASES

SWIETŁANA LEBIEDIEWA*,

A comparison is made between problem-oriented databases for multistage decision making and general-purpose databases. An exemplary problem-oriented database IDEN for supporting the process of multistage identification is discussed in detail. An equivalence condition for problem-oriented databases is formulated and then the equivalence with respect to data structures and operations is proved for hierarchical, network and relational models of the IDEN database.

Keywords: problem-oriented database, data manipulations language, distributed databases, equivalence of databases

1. Introduction

The equivalence of database models has been studied since the late 1970s (Borkin, 1978; Lien, 1982). The research of that time concerned the equivalence of schemes, models and operations, as well as the possibility of cooperation of different systems, in particular heterogeneous systems (Tsichritzis and Lochowski, 1990). A common feature of these works is that they refer to general-purpose databases (GPDBs). At present, the relational model is the most widespread database (DB) model, and progress in microcomputer and telecommunication techniques has contributed to the development of systems of distributed databases (DDBs) based upon the relational data model.

The originality of the considerations and results in the present paper is connected with the specificity of *problem-oriented databases* (PODBs), and more precisely, that of some class of such DBs in systems supporting multistage decision making (MDM systems). In the context of computer-based decision-making systems, an important role is played here by decomposition methods applied in MDM algorithms. The design of MDM systems is closely related to that of DBs in which we have to take account of decision models specific for a multistage process MDM is accompanied by multistage information and data acquiring and gathering, and information and data is brought to and utilized in a multistage process. We will present the specificity of PODBs designed to be used in MDM systems for two important cases: *multistage recognition* (MR)

---

* Wroclaw University of Technology, Institute of Control and Systems Engineering, Wybrzeże Wyspiańskiego 27, 50–370 Wrocław, Poland, e-mail: swieta@ists.pwr.wroc.pl

and *multistage identification* (MI). Original conceptions, methods and algorithms of recognition and identification have been developed at the Institute of Control and Systems Engineering, Wrocław University of Technology (Bubnicki, 1977; 1987; 1990; 1993a; 1993b; 1997; Kurzyński, 1997; Świątek, 1987). These problems can also be of greater importance for other related multistage cases.

The concept of MI consists in the decomposition of a complex identification task into a number of smaller tasks: elementary identification processes. The MR consists in the decomposition of a decision problem, i.e. in the replacement of a single recognition by a sequence of the so-called local recognitions carried out at particular nodes, according to a given construction of the decision tree (DT).

Particular recognition or identification stages may be dissociated in time and space, but the recognizing results from various locations should be stored until the end of the MDM process. Thus information about recognition or identification objects may exist in various segments of the DDB, which have to cooperate with one another.

The purpose of the study is to formulate the equivalence of heterogeneous models of PODBs which aim at supporting the MDM process, proving the equivalence of the hierarchical, network and relational models of a particular system destined for MDM, and determining a possibility of cooperation of distributed PODB systems based upon different DB models.

2. Problem-Oriented Databases for Multistage Decision Making

GPDBs are used to gather, store and retrieve information for every application domain. A unique constraint is the data model. The designer of such a system does not know in advance the kind of information that will be stored in the DB, or the type of the user's queries. On the other hand, PODBs are intended for storing information concerning one problem or one problem class in a particular domain. In databases used for MDM, results of calculations are stored being used e.g. for identification, control or recognition of a given plant. Even if a real object of investigations by GPDB and PODB users is the same (e.g. medical examinations), the manner of object conceptualization is different, and furthermore the data structures (DSs) and the way of data utilization are different. When compared with GPDBs, the PODBs used for supporting MDM possess many special features. This concerns the aspects such as different functions of management systems, DSs, data description languages (DDLs) and a basic piece of information (segment), operations and data manipulation languages (DMLs), specific requirements concerning the operation execution time (data differentiation with respect to the access time), real-time cooperation with the programme, cost of the database management system (DBMS) (Lebiediewa, 1998). DBs intended for MDM have many common features with knowledge bases (KB). These features are e.g. the genesis and interpretation of reality, applications, circle of users, possibility of supporting the inference process (Borzemski and Lebiediewa, 1993b; Lebiediewa, 1998).

At the Institute of Control and Systems Engineering of the Wrocław University of Technology two PODBs for MDM have been developed: IDEN database for MI and REC database for object MR (Borzemski and Lebiediewa, 1980; 1993a; Lebiediewa, 1993).

A data set for elementary identification will be called a *data segment* or, simply, a *segment*. Similarly, we call a segment each set of data necessary for decision making at a single node of the DT. A data segment is an autonomous entity of a DB for MI (or MR), and — for a single-stage identification (or for a single-stage recognition) — the whole DB is reduced to a single segment.

3. IDEN: A Database for Multistage Identification

3.1. Segments and Problem-Oriented DMLs

A basic data structure of the IDEN DB is the *data segment* or, simply, the *segment*, symbolically S_ij (i is the label of an identification stage, j stands for the label of a segment at stage i). Each segment at stage i contains data necessary to perform one elementary identification: matrix of input quantities $\boldsymbol{X}_{ij}$, matrix of output quantities $\boldsymbol{Y}_{ij}$, and vector of model parameters $\boldsymbol{a}_{ij}$. Segment S_ij also contains other information about elementary data (ED): N_i – number of measurements at stage i, s_i – number of components of the input vector, l_i – number of components of the output vector, l_{i+1} – number of components of the vector of model parameters $\boldsymbol{a}_{ij}$. Besides, in all the segments for the objects of stage with label less than m (m is the maximal identification stage), the following set of vectors appears:

$$\boldsymbol{M}_{ij} = \langle x_{(i+1)g}, x_{(i+2)k}, x_{mr} \rangle,$$

where $g = 1, 2, \dots, N_{i+1}$, $k = 1, 2, \dots, N_{i+2}$, $r = 1, 2, \dots, N_m$.

$\boldsymbol{M}_{ij}$ is a table of vectors, whose elements are vectors of input quantities for the object at a stage with label greater than i. The elements of the vector table determine the conditions under which the experiment has been carried out. As already mentioned, the value of the vector of model parameters $\boldsymbol{a}_{ij}$ at stage i depends not only on the input matrix $\boldsymbol{X}_{ij}$, but also on the quantities of input vectors at stages higher than i, which are constant during the elementary identification.

A data segment can be presented in the form of a tree, the root of which (SEG_ij) contains the segment name S_ij and elementary data (ED). The nodes on the first level are the matrix of system input quantities $\boldsymbol{X}_{ij}$, the matrix of system output quantities $\boldsymbol{Y}_{ij}$, the vector of model parameters $\boldsymbol{a}_{ij}$, and the table of vectors $\boldsymbol{M}_{ij}$ (Fig. 1).

The DML is very simple. Its instructions can be divided into two groups: instructions of **WRITE** type and instructions of **READ** type. The **WRITE**-type instructions are used to introduce from the operating storage into the database data such as the input matrices $\boldsymbol{X}_{ij}$, output matrices $\boldsymbol{Y}_{ij}$, vectors of model parameters $\boldsymbol{a}_{ij}$, and also elements of the vector set $\boldsymbol{M}_{ij}$ (columns of input matrix at a stage higher than 1). The **READ**-type instructions are intended to introduce data necessary to perform elementary identification from the database into the operating storage:

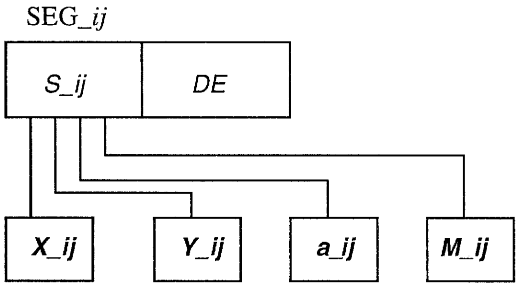

Fig. 1. Tree of the segment S_ij.

matrices $\boldsymbol{X}_{ij}$ and $\boldsymbol{Y}_{ij}$, as well as other data (e.g. a vector set $\boldsymbol{M}$). The **READ**-type instructions make it also possible to read a whole segment from the database, i.e. all the data that belong to the indicated segment. All the access operations to the DB can be presented in the following form:

***READ** DATA_NAME (i, name_of_variable, j), or

***WRITE** DATA_NAME (i, name_of_variable, j),

or

***READ** DATA_NAME (i, g, name_of_variable, j), or

***WRITE** DATA_NAME (i, g, name_of_variable, j),

where i is the label of the identification stage, j signifies the label of the segment at that stage, g denotes the label of a column of the input (or output) matrix, i.e. the g-th measurement, name_of_variable is the name (address) of the area in the operational storage from which segment elements will be introduced into the DB (or into which they will be read from the DB).

3.2. Models of the IDEN DB

3.2.1. Hierarchical and Network Models

The simplest and most natural model of a database for multistage identification is the hierarchical model. The database can be presented in the form of a tree and the root of the tree will be the database name, nodes on the first level — the numbers of the identification stages, nodes on the second level — the segment names, nodes on the third level — the names of segment elements, etc.

The advantages of the hierarchical model are the simplicity of description and ease of implementation. However, in some cases the fact of accepting the hierarchical model leads to a very undesirable phenomenon, i.e. data redundancy, because:

1. If $i < m$, each of the vectors of model parameters $\boldsymbol{a}_i$ appears additionally as a column of the output matrix $\boldsymbol{Y}_{i+1,j}$ for a certain object at stage $i+1$.

2. If $i \leq m-1$, each of the vectors of the input matrix at a stage higher than i appears additionally as an element of the set of vectors $\boldsymbol{M}_i$.

Besides, if the input matrix is the same for each segment at stage i (this is generally the case for an active experiment), then:

3. For $i < m$ the input matrix $\boldsymbol{X}_i$ appears additionally $\prod_{k=i+2}^{m} N_{k-1}$ times, because at the i-th stage there are $\prod_{k=i+1}^{m} N_k$ segments (N_k signifies the number of measurements at stage k).

A structure avoiding data redundancy is the network structure. In Fig. 2 a network model of a database for three-stage identification is presented. As can be seen, the tree-like structure is retained. However, besides natural dependencies between the main record (the segment index containing the segment name and elementary data) and subordinated records (segment elements such as $\boldsymbol{X}_{ij}$, $\boldsymbol{Y}_{ij}$, $\boldsymbol{a}_{ij}$ and $\boldsymbol{M}_{ij}$), the network structure renders it possible to make apparent dependencies between elements of different segments. The vectors of model parameters at the first stage are elements (output vectors) of the output matrix $\boldsymbol{Y}_{21}$ belonging to the first segment at the second stage, the vectors of model parameters at the second stage $\boldsymbol{a}_{21}$ and $\boldsymbol{a}_{22}$ are elements (output vectors) of the matrix $\boldsymbol{Y}_{31}$ at the third stage, the elements of the table of vectors $\boldsymbol{M}_{11}$ are also elements of matrices at higher stages, $\boldsymbol{X}_{21}$ and $\boldsymbol{X}_{31}$.

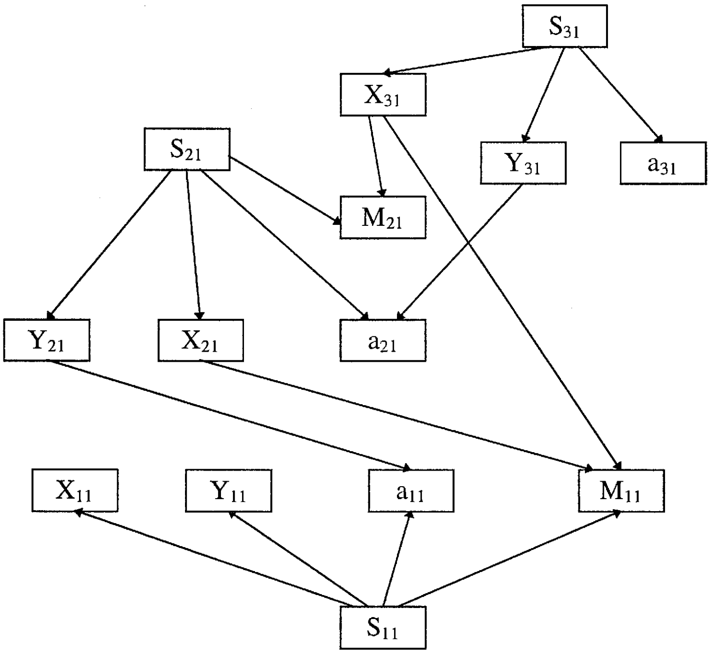

Fig. 2. Network IDEN DB model (part).

3.2.2. Relational (Natural and Generalized) Models

On account of the data form (regular structures: tables, vectors and scalars) and operations (picking up lines or columns of tables), a relational database management system can be used for implementation of a DB for multistage identification.

In the natural model the assumption is made that the whole database for multistage identification is a set of segments. It is accepted that the following relations are assigned to each data segment:

RELSEGS_i, j, EL_DATA_i, j, WEX_i, j, WYY_i, j, PARa_i, j.

Relation RELSEGS_i, j contains the segment name (identifier) and some other relations belonging to the segment. Relation EL_DATA_i, j contains (besides the segment name) the consecutive number NR of the segment in the database, the identification stage i, the number j of the segment at stage i with N_i measurements at stage i, s_i components of the input vector, l_i components of the output vector, and l_{i+1} components of the model parameter vector. Relations WEX_i, j and WYY_i, j contain respectively the values of components of input and output vectors for consecutive measurements. Relation PARa_ij contains the vector of model parameters (the identification result). Moreover, each of the segments at stage $i < m$ contains a relation TABVEC_i, j. Relation TABVEC_ij contains elements of the set of vectors M_ij. The conceptual model of the DB, reduced to the segment S_ij is presented in Fig. 3 (Lebiediewa, 1998).

RELATION RELSEG_ij (#SEGNAME, RELNAME)
RELATION EL-DATA_ij (#SEGNAME, NR, I,J,N_*i*, S_*i*, *L_i*, *L_i*+1)
RELATION WEX_ ij (#ID_INPUT, NR_MEASUREMENT, x_1)
RELATION WYY_ij (#ID_OUTPUT, NR_MEASUREMENT, y_1)
RELATION PARa_ij (#ID_A, A_1, A_2)
RELATION TABVEC_ij (#ID_TAB, X_1)

Fig. 3. Scheme of the relational DB reduced to the segment S_ij.

Note that relations 1–3 exist between the segments (c.f. Section 3.2.1). Moreover, the structure of all the segments at stage i is identical, thus the schemes of the segments differ only by element names. We will make use of the fact that when grouping information with identical description in one relation, a generalized relational model is derived (Lebiediewa, 1998).

4. An Approach to the Equivalence of Problem-Oriented Databases

Two aspects of the equivalence problem of DB models are to be distinguished: the theoretical aspect (equivalence of data structures and operations), and the practical aspect (possibility of cooperation of heterogeneous DB systems).

The theoretical aspect

A database model is determined by the data structures (DSs) and operations. A basic data structure is understood as a piece of access to the DB. In hierarchical and network models, such a piece is a record, and in relational models it is a relation.

The DB equivalence can be considered with respect to DSs and with respect to operations. We say that databases A and B are equivalent with respect to basic data structures if each basic structure of A can be mapped into a basic structure of B, and if each basic structure of B can be mapped into a basic structure of A (Tsichritzis and Lochowski, 1990). Similarly, A and B are equivalent with respect to operations if each operation of A is executable in B, and if each operation of B is executable in A. As regards the equivalence with reference to data structures, the hierarchical, network, and relational models of GPDBs are equivalent (Lien, 1982). A network model may be reducible to a hierarchical model, whereas a hierarchical model — to a relational model. On the other hand, a relation can be interpreted as a record of **OWNER** type in hierarchical or network models without records of **MEMBER** type. It is in contrast to the equivalence of DB models with respect to operations.

A DML preprocessor should be developed in order to map operations.

The practical aspect

The practical aspect concerns the possibility of cooperation of DBs with different models, which means cooperation of heterogeneous DB systems. Here, the situation is more complicated. In GPDBs, the DSs are described by means of a data definition language (DDL), and the operations on data are realized by instructions of the DML. There is no automatic transition from the description of a DS in relation models to a description, e.g. in the CODASYL system. Even within the same model two different DBMSs do not have to be equivalent on account of either DDL statements or DML instructions. For example, the **SELECT** and **PROJECT** instructions of a relation algebra language will not be understood by the translator of the SQL language, and the structure **SELECT-FROM-WHERE** will not be executed in a relation algebra based system. Even the SQL used in different DBMSs requires a defined library of calls of access to the DB. One of possible solutions is the ODBC standard, developed by Microsoft, which is a purely technical solution. Another solution, the CORBA standard, concerns the cooperation of object DBs which are also GPDBs.

The subject of this study is not to examine equivalence of GPDBs. From a theoretical point of view, the problem of equivalence of classical (hierarchical, network and relational) GPDBs systems has already been solved. It is the practical aspect that remains unsolved, i.e. the possibility of cooperation of heterogeneous distributed databases (DDBs). It can be accepted that actually the hierarchical and network systems are seldom utilized and that the market is dominated by relational DBs. At present, four types of DDBs can be distinguished: client-server type systems, homogeneous DDBs based on an identical DBMS, functioning on the same hardware and with the same operating system (OS), heterogeneous DDBs, and federated systems (Beynon-Davies, 1998).

The so-called heterogeneous systems do not assume homogeneity of hardware or software. A heterogeneous system is based upon a gateway that is an interface supplied by the producer. The standard of federated systems is now in its development phase and after its termination it will become a technical problem as in the case of the ODBC. However, all these DDB systems are based upon the relational model of DBs, and the basis for its division is the relation. As a result of distribution, a horizontal or vertical division (fragmentation) of the relation takes place.

Our problem is different. The basis for division is not relation fragmentation, but a *decomposition* of a complex Multistage Decision Making (MDM) system. The basic data unit is not a relation, but a complex DS, the *segment*. There is no assumption made as for what concerns the homogeneity of hardware, OSs, DBMSs or DB models.

The client-server system is the least fitted for use in MDM systems. We do assume that there exists a *real* data distribution fulfilling the following conditions: measurement data are stored in local DBs (LDBs) in the place of their coming into being, and the EI results in the MI process or those of the decisions being made in the MR are transferred to other LDBs for further processing. The other systems are general-purpose relational systems and strongly depend on hardware, software and producer.

Our interest will be in the equivalence of DB systems regardless of the data model (heterogeneous systems) and hardware. Let us consider the equivalence problem of PODBs based on the example of the equivalence of hierarchical, network, and relational models of the IDEN DB (Lebiediewa, 1998). The hierarchical and network models of the IDEN DB were models that were specially developed for the purposes of MI, adapted to identification procedures with respect to both data structures and DML operations. The DBMSs of the IDEN PODB for these models were not based on any of the existing DBMSs, but they have been developed starting from scratch. The reasons were the hardware-software conditions. The first DBMS of the IDEN PODB were developed for an ODRA 1325 computer, for which there was no GPDB management system at all. The problem-oriented management system of the IDEN DB enclosed a DB installing program and a DMJ preprocessor. Elements of a data segment are matrices, vectors and scalars that can be described by means of declarations of an arbitrary algorithmic language (Lebiediewa, 1998; 1999), so no separate DDL was defined. Due to the lack of a DDL in the DBMS of the IDEN PODB, it is impossible to describe any type of record of general-purpose hierarchical and network models. The preprocessor of the problem-oriented DMJ executes instructions that have been developed especially for the cooperation of MI algorithms with the DB, but it does not execute DMJ instructions such as instructions of the IMS or CODASYL systems. For the same reasons no relations can be described in the IDEN PODB system. Similarly, no relational operation is executable in the hierarchical and network models of the IDEN problem-oriented DB.

Consequently, the hierarchical and network models of the IDEN PODB are equivalent with neither the relational system, nor any general-purpose system. Moreover, any two management systems for PODBs intended for use in different domains are not equivalent: Some examples can be the IDEN DB and the multistage pattern recognition database (REC DB) (Borzemski and Lebiediewa, 1993a; 1993b; 1998). A basic DS of each of these systems is a segment, but segment elements in the IDEN DB and REC DB are different data structures, and the DMLs are also quite different. However, in practice a need may arise for cooperation of different local DBs concerning the same issue, e.g. an identification experiment carried out in different laboratories. Then the question arises: What is the condition of cooperation of distributed systems of PODBs based on heterogeneous DB models? In our opinion, a condition of cooperation of distributed PODBs irrespective of their model is the equivalence with respect to the basic data structures and operations being specific for these systems. Below, we present a formal definition of the DB equivalence based on model theory (Grzegorczyk, 1973).

Let the symbols M and M' denote DB models, $DS = (S1, \dots, S_n)$ and $SD' = (S'_1, \dots, S'_n)$ — the basic data structures, and $O = (i_1, \dots, i_m)$ and $O' = (i'_1, \dots, i'_l)$ — the operations or problem-oriented DML instructions of models M and M', respectively. Let us assume that a sequence of instructions $i_1, \dots, i_k$ is a subsequence of $i_1, \dots, i_m$, and that $i'_1, \dots, i'_k$ is a subsequence of $i'_1, \dots, i'_l$.

Definition 1. DB models $M = (S_1, \dots, S_n, i_1, \dots, i_k)$ and $M' = (S'_1, \dots, S'_n, i'_1, \dots, i'_k)$ are equivalent with respect to data structures and operations $i_1, \dots, i_k$ if there exists a mapping f of model M into model M' such that, for each data structure S_i, $i = 1, 2, \dots, n$, the dependences are $f(S_i) = S'_i$, $f^{-1}(S'_1) = S_i$ for each operation $i_r = i'_r$, $i = 1, 2, \dots, k$, and $f(i_r(S_i)) = i_r(f(S_i)) = i_r(S'_i)$.

5. Equivalence between Hierarchical, Network and Relational Models of the IDEN DB

Theorem 1. *The hierarchical and network models of the IDEN DB are equivalent with respect to the basic DS and DML instructions.*

Proof.

Equivalence of structures

According to Section 4, in both the hierarchical and network models, each segment contains a record *Seg_ij* which includes the segment name and elementary data, as well as the 'proper data' $\boldsymbol{X}_{ij}$, $\boldsymbol{Y}_{ij}$, $\boldsymbol{a}_{ij}$, $\boldsymbol{M}_{ij}$. Let us denote by e_h the variable running through the set of segment elements in the hierarchical model, and by *e_s* the variable running through the set of segment elements in the network model. From the definition of the segment it follows that for each segment element $f(e\_h) = e\_s = e\_h$, or that f is an identity mapping.

Equivalence of operations

The assumption was made (Lebiediewa, 1998; 1999) that on the segment elements of the IDEN DB DML instructions are defined of type ***READ** name_of_variable and ***WRITE** name_of_variable (regardless of the model being accepted). Clearly,

$$f(^*\mathbf{READ}\ e\_h) = f(^*\mathbf{READ})f(e\_h)$$
$$= {}^*\mathbf{READ}\ f(e\_h) = {}^*\mathbf{READ}(e\_s)$$

and

$$\begin{aligned} f(^{*}\mathbf{WRITE}\ e\_h) &= f(^{*}\mathbf{WRITE}\ )f(e\_h) \\ &= {}^{*}\mathbf{WRITE}\ f(e\_h) = {}^{*}\mathbf{WRITE}\ (e\_s) \end{aligned}$$

for each operation of type ***READ** and ***WRITE**, and for each segment element.

Therefrore, the hierarchical and network models are equivalent with respect to the basic data structures (segments) and the DML instructions. ■

Obviously, the conceptual models of a hierarchical and a network DB are not equivalent (the network model is a result of removing redundancy in the hierarchical model — the vector of model parameters is identified with the corresponding vector of the output matrix (measurement) on a higher degree, and the elements of the table of vectors — with corresponding components (measurements) on higher degrees). The physical organization is also different (Lebiediewa, 1998; 1999). However, the user works on the external model (segment), and because of the equivalence between the hierarchical and network models with respect to segments and DML instructions, IDEN DBs based upon hierarchical and network models can cooperate.

The equivalence between hierarchical and relational models is not evident. To a segment S_ij of the hierarchical model corresponds a set of relations discussed in (Lebiediewa, 1998; 1999) in the relational model, relation EL_DATA_ij corresponding to the record SEG_ij (tree root) containing the segment name and elementary data, relations WEX_ij, WYY_ij, PARa_ij, TABVEC_ij correspond respectively to the proper data $\boldsymbol{X}\_ij$, $\boldsymbol{Y}\_ij$, $\boldsymbol{a}\_ij$, $\boldsymbol{M}\_ij$. Indeed, matrix $\boldsymbol{X}\_ij$ is not equivalent to relation WEX_ij, matrix $\boldsymbol{Y}\_ij$ is not equivalent to relation WYY_ij, and the vector of model parameters $\boldsymbol{a}\_ij$ as well as the table of vectors $\boldsymbol{M}\_ij$ are not equivalent respectively to relations PARa_ij and TABVEC_ij. *Measurements* correspond to segment elements in the hierarchical model, and the corresponding relations contain, in addition to measurements, other information: names of elements and numbers of operations. In the hierarchical model, names do not appear except for segment names. However, it can be demonstrated that each of the relations contains data structures that are *identical* with the corresponding DSs of the hierarchical model (with proper data being elements of the segment).

Theorem 2. *The hierarchical and relational (natural) models of the IDEN DB are equivalent with respect to DSs including proper data and DML instructions.*

Proof. We will demonstrate that each of the relations includes DSs being identical with corresponding DSs in the hierarchical model.

In the relational model, relation EL_DATA_ij corresponds to record SEG_ij. This relation contains the same elements: the segment name and elementary data. We introduce two functions f and g. Function f is determined on elements being proper data of the segment, with values in the relational model in the following manner:

$$f(\boldsymbol{X}\_ij) = \mathrm{WEX}\_ij, \tag{1}$$

$$f(\boldsymbol{Y}\_ij) = \mathrm{WYY}\_ij, \tag{2}$$

$$f(\boldsymbol{a}\_ij) = \mathrm{PAR}a\_ij, \tag{3}$$

$$f(\boldsymbol{M}\_ij) = \mathrm{TABVEC}\_ij. \tag{4}$$

Function g is determined on elements (relations) WEX_ij, WYY_ij, PARa_ij, TABVEC_ij of the relational model, the values of function g being respectively X_ij, Y_ij, a_ij, M_ij that result from relations WEX_ij, WYY_ij, PARa_ij, TABVEC_ij as a consequence of the **PROJECT** operation of each of these relations onto the attributes being the names of the components of matrices, which can be written down in the relation algebra language as follows:

PROJECT WEX_ij **ON** (x_1) **GIVING** X_ij,

PROJECT WYY_ij **ON** (y_1) **GIVING** Y_ij,

PROJECT $PARa\_ij$ **ON** (a_1, a_2) **GIVING** a_ij,

PROJECT $TABVEC\_ij$ **ON** (x_1) **GIVING** M_ij.

Consequently,

$$g\big(f(\boldsymbol{X}\_ij)\big) = X\_ij, \tag{5}$$

$$g\big(f(\boldsymbol{Y}\_ij)\big) = Y\_ij, \tag{6}$$

$$g\big(f(\boldsymbol{a}\_ij)\big) = a\_ij, \tag{7}$$

$$g\big(f(\boldsymbol{M}\_ij)\big) = M\_ij. \tag{8}$$

Relations X_ij, Y_ij, a_ij, M_ij contain measurements and are identical with proper data $\boldsymbol{X}_{ij}$, $\boldsymbol{Y}_{ij}$, $\boldsymbol{a}_{ij}$, $\boldsymbol{M}_{ij}$ of segment S_ij, respectively. As the assumption was made that in each model of the IDEN DB operations of the DML of type ***READ** and ***WRITE** are executed, the hierarchical and relational models are equivalent with respect to DSs containing proper data and DML instructions. ■

Theorem 3. *The hierarchical, network, and relational (natural) models of the IDEN DB are equivalent with respect to the DSs containing proper data and DML instructions.*

Theorem 3 results directly from Theorems 1 and 2.

Theorem 4. *The relational natural and generalized models are equivalent with respect to basic DSs and DML instructions.*

Theorem 5. *The hierarchical, network and relational (generalized) models of the IDEN DB are equivalent with respect to basic DSs and DML instructions.*

Proofs of Theorems 4 and 5 are given in (Lebiediewa, 1998).

6. Concluding Remarks

The existing systems of distributed databases are general-purpose systems based upon the same (relational or object) model. So far there has been no possibility of cooperation between local databases (LDBs) based upon different models because of their different data structures (DSs) and Data Manipulation Languages (DMLs). The purpose of the study was to examine the condition of cooperation of Problem-Oriented Databases (PODBs) independently of the LDB model. The basis of the proposed approach is the user interface with the PODB — the data segment (the user's external model) and the problem-oriented DML.

The equivalence of various models with respect to basic DSs and DML instructions is of great practical importance. First, the proposed approach assures independence from hardware, software and a DB model. The equivalence of the hierarchical, network and relational DB models have been examined, but the IDEN PODB can be developed as an object database (ODB). Second, the equivalence of PODBs presented in Section 4 is of a general character: the equivalence with respect to operations does not assume an identity of all the instructions of the problem-oriented DML in different DB models, but only that of a determined subset of instructions. In the case of the IDEN DB, the instructions of the problem-oriented DML were identical. In other PODB systems, however, it may not be the case. For example, in each DB model for multistage recognition (MR), the assumption is made that the instructions of the problem-oriented DML used by recognition algorithms at a node are executed. However, some queries to the DB (e.g. queries concerning the DT structure or the list of all the objects belonging to a class) which are realized in relational and network DB models may not be executable in the hierarchical model (Lebiediewa, 1998). Third, similar DSs and related problems may appear in other information systems, e.g. for MR or multilevel control. In (Lebiediewa, 1998), the equivalence of hierarchical, network and relational models of the REC DB was analyzed. The REC DB can be, however, realized as an ODB (complex DB structure, links between segments, inheritance), or — after inclusion of decision rules (recognition algorithms) — as a deductive DB.

References

Beynon-Davies P. (1998): *Database Systems.* — Warsaw: Polish Scientific Publishers (in Polish).

Borkin S. (1978): *Data model equivalence.* — Proc. 4th Int. Conf. *Very Large Data Bases*, West Berlin, pp.526–534.

Borzemski L. and Lebiediewa S. (1980): *Database organization in the multistage identification experiment.* — Prep. IFAC Workshop *Scientific Experiments and Laboratory Procedures Automation SELPA*, Smolenice, Czechoslovakia, pp.79–100.

Borzemski L. and Lebiediewa S. (1993a): *On a database for multistage decision making in a pattern recognition process.* — Proc. 2nd Nat. Sci. Conf. *Knowledge Engineering and Expert Systems*, Wrocław, Poland, Vol.2, pp.307–313 (in Polish).

Borzemski L. and Lebiediewa S. (1993b): *On the database in multistage decision making systems with knowledge base.* — Prep. 6th Nat. Workshop CIR'93, Artificial Intelligence and Cybernetics of Progress, Siedlce, Warsaw, Poland, pp.60–68 (in Polish).

Bubnicki Z. (1977): *On the multistage identification.* — Syst. Sci., Vol.3, No.2, pp.207–210.

Bubnicki Z. (1987): *Distributed computer systems for two-level pattern recognition.* — Proc. 5th Int. Conf. *Systems Engineering*, Dayton, USA, pp.277–280.

Bubnicki Z. (1990): *An Introduction to Expert Systems.* — Warsaw: Polish Scientific Publishers (in Polish).

Bubnicki Z. (1993a): *Knowlege-based approach as a generalization of pattern recognition problems and methods.* — Syst. Sci., Vol.19, No.2, pp.5–21.

Bubnicki Z. (1993b): *Foundations of the Information Management Systems.* — Wrocław: Publishing House of the Wrocław Univ. Technol. (in Polish).

Bubnicki Z. (1997): *Knowlege updating in a class of knowlege-based learning control systems.* — Syst. Sci., Vol.23, No.4, pp.19–36.

Grzegorczyk A. (1973): *Mathematical Logic Concepts.* — Warsaw: Polish Scientific Publishers (in Polish).

Kurzyński M. (1997): *Pattern Recognition. Statistical Methods.* — Wrocław: Publishing House of the Wrocław Univ. Technol. (in Polish).

Lebiediewa S. (1993): *On the database for multistage pattern recognition.* — Arch. Th. Appl. Comp. Sci., Vol.5, No.2, pp.343–356 (in Polish).

Lebiediewa S. (1998): *Methodology of the Design of Problem-Oriented Databases for Decision Problems in Multistage Systems.* — Wrocław: Publishing House of the Wrocław Univ. Technol. (in Polish).

Lebiediewa S. (1999): *The problem-oriented database IDEN for supporting the multistage identification process.* — Syst. Sci., Vol.25, No.2, pp.87–99.

Lien Y.E. (1982): *On the equivalence of database models.* — J. ACM, Vol.29, No.2, pp.333–363.

Świątek J. (1987): *On two-level identification and their technical and biomedical applications.* — Sci. Papers Ins. Contr. Syst. Eng. of Wroclaw Univ. Technol., Series: Monographs, Vol.4, No.2 (in Polish).

Tsichritzis D.C. and Lochowski F.H. (1990): *Data Models.* — Warsaw: Polish Scientific Publishers (in Polish).

Received: 3 March 1998
Revised: 20 May 1999
Re-revised: 14 June 1999