Construction of a set of harmonized data models in distributed databases based on semantics

Ainura T. Kasymalieva
Associate Professor, Department of Information Systems in Economics
Kyrgyz State Technical University named after I. Razzakov
Address: 66, Prospect Mira, Bishkek, 720044, Kyrgyz Republic
E-mail: aisu@rambler.ru

Abstract

When developing projects of integrated enterprise systems with focus on further expansion of functions, it is necessary to remember about continuity of models created and the timeliness of updating them. These refinements relate to the future development of organizations or their units involved in development of information systems (IS) or other software products (SP), extension of functionality of integrated enterprise information systems (IEIS), as well as development environments for design and programming. In this regard, the author proposes to apply the extended level scheme of data and database modeling.

When investigating the functions of each department and building their model description in subsystems (private models), it is possible to identify the same objects for providing functionality. Coherence is one of the advantages of the resulting model, providing typing and standardization of the creative processes of IS.

We use data distribution mechanisms, which today are very topical. The proposed solution is based on a semantic dictionary reflecting the basic terms and concepts of the functional tasks of the business environment of an enterprise being modeled; it allows us to unify the application development and complements the data distribution strategy across the nodes of the enterprise.

This article presents the principles for forming a family of harmonized data models. It provides a formal description of them, the algorithms and the possible core formation practices. The advantages of using this and approaches to possible use are discussed.

Keywords: distributed databases, information systems, data modeling, data model core.

Citation: Kasymalieva A.T. (2016) Construction of a set of harmonized data models in distributed databases based on semantics. *Business Informatics*, no. 2 (36), pp. 48–56. DOI: 10.17323/1998-0663.2016.2.48.56.

Introduction

Issues of distributed databases (data distribution across nodes) are under consideration in quite a lot of designs and mathematical models (e.g., [1–3]). In most cases, the possibilities of optimizing distributed queries of already existing systems are reviewed.

All existing designs in the area of modeling distributed databases can be divided into two common groups considered.

The first and biggest group consists of algorithms for dynamic data redistribution across nodes of the network as described in the works of such authors as D.V. Pav-

lov [4], P.P.-S. Chen and J. Akoka [5], H.I. Abdalla [6]). The disadvantages of this model are the complexity of the algorithms for redistribution, organization of additional computations at the stage of system functioning, locking in databases with respect to user queries at the time of rebuilding its structure, as well as isolation of models from the conceptual domain model.

The second group includes the synthesis algorithms of physical structures of the information system model described in the works of M.T. Ozsu and P. Valduriez [7], A.V. Silin [8], V.V. Beskorovainy [9], V.V. Kulba [2]. These algorithms are based on binding to system users, not functions that affect the system scalability and data consistency in different nodes.

At the same time, none of the considered groups does uses the concept of consistency of models at the design stage of large systems or the concept of a harmonized integration of existing systems based on business functions.

This topic has not lost its relevance today.

Information system which are difficult to model due to their multidimensionality are called large. There are two ways to transfer these systems into a relatively small category. In the first case, it is assumed you use more powerful computing facilities with a developed system of information objects (database) collection, proceed to their direct development and constantly increase. In the second case, it is possible initially, at the level of modeling, to break down the multidimensional system into a set of subsystems of lower dimension while monitoring information communications ensuring the integrity of the system. In this case, distributed system architecture development is originally carried out.

The author proposes a model that at the design stage of the integrated enterprise information systems (*EIS*), on the basis of entity-relationship diagrams (*ERD*), reflects different aspects of activities of subsystems of a large information system that optimizes opportunities for distribution of these data in terms of functional nodes, with allocation of a special node defined as the core of the model. The methodology of distributed database modeling uses a semantic dictionary based on the principles of ontology. The composition of the dictionary includes the information models of systems correlated with each object or business process of the organization model – the business environment. Private models at the moment of their merger into a single system to unify use of the data dictionary form a global entity-relationship diagram (*GERD*).

1. Requirements for the model

In order to ensure the integration of private diagrams into a global entity-relationship diagram, it is necessary to fulfil the following requirements:

1. Entity names representing semantically homogeneous objects in all private diagrams should be consistent;

2. The basic dictionary, based on which the entity names in private diagrams *ERD* are formed, should be compiled on the basis of single classifiers that contain similar subject domain entities and/or their acronyms;

3. Each entity that is part of the private models *ERD* family must have a strong verbal description;

4. The set of entities that comprise the global entity-relationship diagram *GERD* should be submitted in the form of many names without duplication of names and their aliases, i.e. to meet requirements of the first form of set-theoretic representation at the element enumeration level;

5. Relationships between entities in the global entity-relationship diagram *GERD* form a family of overlapping sets of entities, each of which provides support of main and accompanying business processes, control functions and communication functions with the external environment;

6. Selection of entities for the global diagram *GERD* must obey the algorithm for constructing the family of consistent models discussed below.

2. Data model core

Data model core (*DMC*) is a data model consisting of entities of the family of private models bearing the basic characteristics of the subsystems and their functions in the information system management allocated to reduce inconsistencies in distributed data and ensure their coordination.

The data model core is formed by identifying the common entities described in the collection of the *ERD* private models. Subject to uniform rules for naming entities and attributes, the process of creating a *DMC* can be given a formal character through use of set-theoretical operations over a family of entities included in *ERD* private models. The algorithm of *DMC* formation is given below. With regard to the standardized attribute names, their membership in the entities of the *ERD* and *DMC* models agree on the level of projections (in the sense of relational algebra operations), and the completeness of the attributes in each entity of the *ERD* models is specified at the development level of private fully attributed models of design level (*Figure 1*).

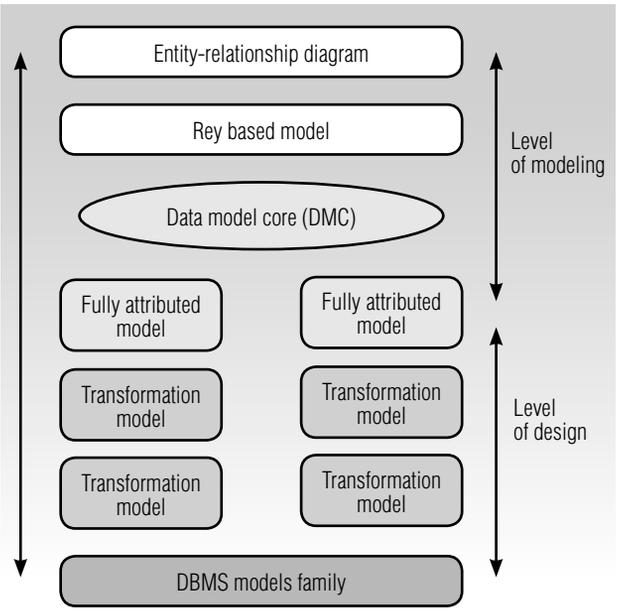

Fig.1. Levels of databases modeling

3. Formalized description of the family of consistent data models

Let us show that there is a formal algorithm to solve both the direct problem of creating consistent models on the base of model data core and private data diagrams that form the basis of separate software subsystems, and the reverse on the basis of private diagrams, to make a data model core of a global system that represents the subject domain. To prove this fact, we will use several provisions of the relational algebra proposed in notation D. Maier [10].

A family of sets is supposed to be given *ERD* composed of sets of entity names E_i private diagrams of data models ERD_i

$$ERD = \{ERD_1, ERD_2, ..., ERD_i, ..., ERD_m\}, \text{ where}$$

$$ERD_i \in ERD,\ 1 \le i \le m;$$

$$E_i = \{e_{i1}, e_{i2}, ..., e_{ij}, ..., e_{in}\},\ 1 \le i \le m,\ 1 \le j \le n,$$

where m – number of entity-relationship diagrams included in the set *ERD;*
n – number of entities included in each diagram ERD_i.

In addition, let us define a global set *GERD* composed of all entity names e_{ij} included in diagrams $ERD_1, ERD_2, ..., ERD_i, ..., ERD_m$ such that

$$E_1 \in ERD_1,\ E_2 \in ERD_2,\ ...,\ E_m \in ERD_m;$$

$$E_i = \{e_{i1}, e_{i2}, ..., e_{ij}, ..., e_{in}\},\ 1 \le i \le m,\ 1 \le j \le n,$$

i.e. the set $GERD = E_1 \cup E_2 \cup ... \cup E_m =$

$$\{\{e_{11}, e_{12}, ..., e_{1j}, ..., e_{1n}\}, ..., \{e_{i1}, e_{i2}, ..., e_{ij}, ..., e_{in}\}, ...,$$
$$\{e_{m1}, e_{m2}, ..., e_{mj}, ..., e_{mn}\}\}$$

Let us put in correspondence to each of the entity names e_{ij} an interpreting function φ, such that

$$\varphi_{ij} : e_{ij} \in u_{ij},\ e_{ij} \in GERD,\ u_{ij} \in SI,$$

where u_{ij} – a verbal description of the meaning (semantics) of an entity name e_{ij};
SI – a lot of semantic information constituting an interpretation space of elements of the global set *GERD*.

Let us define a functional mapping *F* such that *F*:

$$GERD \times SI \to DTN,$$

where *DTN* – a lot of twos, such that each entity name e_{ij} is compared to its meaning (semantics) u_{ij}. Then the mapping *DTN* can be interpreted as a dictionary of entity names, where each entity name is endowed with corresponding subject-oriented interpretation (description).

Let us define how to correlate between each other data model core entities *DMC* and private data models entities *ERD* at the development stage of key models. Let us recall that each of the entities is a relationship. Therefore, in the future these entities will be considered as *relations R* defined as subsets of a Cartesian product of family sets A_i.

Let us introduce the following definitions.

Definition 1. Let us call the relationship scheme *R* a finite set of attribute names $\{A_1, A_2, ..., A_n\}$, where each attribute is mapped to the lot D_i called the domain of attribute A_i, $1 \le i \le n$.

It is accepted to designate the domain of an attribute as $dom(A_i)$. The domains are arbitrary non-empty finite (countable) sets.

Let us say $D = D_1 \cup D_2 \cup ... \cup D_n$. Then it is possible to introduce the following definition.

Definition 2. Relationship r with scheme *R* is a final set of mappings $\{t_1, t_2, ..., t_p\}$ *from R* into *D* where each mapping $t \in r$ must satisfy the following constraint: $t(A_i)$ belongs D_i, $1 \le i \le n$. These mappings are called relation schema *r* with scheme *R*. In this case, a tuple is commonly understood as the set of values one for each attribute name from relation schema *R*.

If we interpret *t* as a row in the table, *A*-value $t(A)$ of tuple *t* is the content (value) of tuple *t* in column *A*. Thus, the relation *r* can be viewed as a table with many tuples *t* that satisfy the relation schema *R*.

Based on the definitions above, let us assume that each entity *e* belonging to set *ERD* is based on relation schema

R with set of tuples t, and each specified entity e_{ij} is represented by diagram R_{ij}.

Let a set of entities within the data model core DMC be defined some way as e_k^0, $1 \le k \le K$, and for each entity a relation schema is defined R_k^0.

Definition 3. Let us assume that entity e_{ij} is fully compatible with entity e_k^0 (*full compatible data*) if the corresponding relation schema R_{ij} *and* R_k^0 satisfy requirement $R_{ij} = R_k^0$.

Definition 4. Let us assume that entity e_{ij} is partly compatible with (*partial compatible data*) with entity e_k^0, if the corresponding relation schema R_{ij} and R_k^0 satisfy requirement $R_{ij} = S_k^0$, where $S_k^0 \subseteq R_k^0$.

When forming key or fully attributed models of data model core $DMC(KB^0)$ on the basis of private key models $ERD(KB_i)$ (or vice versa), it is necessary to adhere to the following compatibility options:

✦ schema entity-relationship (subsystems) R_{ij} included in a ERD_i **,** and the relation scheme of the same name analogues R_k^0 in DMC may have the property of full compatibility. Otherwise, names and number of attributes, their sequence and domains on which they are defined must be the same;

✦ relation scheme of entities (subsystems) R_{ij} included in an ERD_i, and the relation scheme of the same name analogues R_k^0 in DMC may have the property of partial compatibility. Otherwise, in partially-compliant relation schemas, data schema R'_{ij} is a subset of data schema R_k^0, i.e. $R'_{ij} \subseteq R_k^0$, i.e. $R_{ij} = R'_{ij}$.

✦ the relation scheme of entities (subsystems) R_k^0, included in a DMC, and the relation scheme of the same name analogues R_{ij} in ERD_i may have the property of partial compatibility. In this case, in partially compatible relation schemas, data schema R_k^0 is a subset of the same name data schema R_{ij}, i.e $R_k^0 \subseteq R_{ij}$.

Of course, the key fields in a compatible data schema from DMC and ERD_i should be the same.

When locating attributes in the key models, it is advisable to adhere to the following recommendations. The same name attributes in both schemas must have the same order. This will greatly simplify the data transfer procedure between tables of the data core and private data models when they are processed in a DBMS in the format of fully attributed or transformational models.

The last requirement is not meant to be exclusive. With some complication of the procedure for copying data, the same order of attributes (columns) in the treated models is not required. Moreover, it is possible that in the compared models, the corresponding attribute names are correlated as synonyms defined on the same domains. This renaming of attributes is theoretically permissible and provided for in the relevant theorems of relational theory. Let us note that almost all the models (data transformation services) of data transformation of modern DBMS are built on this basis.

Data transfer in a DBMS between partially compatible entities e_k^0 data model core DMC and private data models e_{ij} from set ERD is performed by applying the operations of selection σ or projection π to the relationship r_k^0 (tables) with the relation scheme R_{ij} from the set of entities DMC and provided by replication mechanisms of modern DBMSs [11, 12].

Selection operation. The result of applying the selection operation σ to relation r is another relation, which is a subset of tuples of relations r with a certain value in the selected attribute. Let r be a relation with scheme R, A is an attribute in R and a is an element of $dom(A)$. Then $\sigma_{A=a}(r)$ is a designation of selection operation ("to select from r tuples in which the value A is equal to a"). Considering the tuples as mappings, it is possible to record: $r'(R) = \{t \in r \mid t(A) = a\}$. Selection operation $\sigma_{A=a,\ B=b}(r)$ on several attributes is possible due to the fact that it is commutative.

A projection operation is an operation which allows us to exchange data between private data models e_{ij} from set ERD and r_k^0 from set modal entities DMC by cutting the part of attributes from the schema R_k^0 and formation of a new relationship r'_{ij}.

Let r be a relation with scheme R, and X a subset R. Projection r on X written as $\pi_X(R)$ is the relationship $r'(X)$ obtained by crossing out columns corresponding to attributes in $R - X$ (set-difference operation) and with exception of duplicate rows from the remaining columns. Considering the tuples as mappings, $\pi_X(R)$ can be written in the form $r'(X) = \{t(X) \mid t \in r\}$.

4. Formation algorithm of the data model core

The formation algorithm of the data model core looks like the following.

1. Make lot $GERD$ of entity names e_i, where $e_i \in R$ by attributing to the entity composition of the first model ERD_1 all entities of other models ERD_2, ..., ERD_n;

2. Select recurring entity names from set $GERD$ registering their names and the number of occurrences of set $GERD$, and bring them to the set of twos G, where each twos is composed of the name entity and the number of occurrences x of this entity in set $GERD$;

3. Perform a ranking of the elements of set G arranging the entities in descending order of number of occurrences x in set G;

4. Select entities that have x occurrences in set G by criterion $x \geq k$, where k is a threshold number of entity names selected by experts for inclusion of their name in the model core DMC.

From the list of requirements above (possibly not complete), it follows that it is possible to satisfy them only on condition of the existence of the expanded business model of the organization – families of models describing its organizational and process structure, business functions, models of the organization's relationship with the external environment and the responsibility function distribution models for performing functionality elements.

Practice of model core forming DMC has shown that in this core, entities representing the following data are often included:

✦ data describing an organizational structure of an enterprise or organization;

✦ data specifying the business direction of the organization;

✦ data specifying a product portfolio or composition of market services of the organization;

✦ data of detailed description of the goods (products) or services;

✦ data specifying a resource component of the organization (personnel, knowledge, material and informational resources);

✦ data supporting management of main and accompanying processes – accounting component of vector of management characteristics;

✦ data defining a classifier of basic documents of the organization and documents comprising its workflow;

✦ data describing external environment including suppliers, customers, consumers etc.;

✦ data, supporting management software of the core DMC;

✦ other data characterizing the industry affiliation of the organization (production, social, administrative, advisory, etc.).

Naturally, the composition of the entities included in the DMC model depends on the profile of the organization, so it is not necessary to include in the model core all of the above groups of entities.

There is another formalized way of entity selection in the DMC core based on the algorithmic approach.

5. Advantages of the proposed approach

Use core advantages are as follows.

Local independence. Getting the data core, each subsystem contains the actual data needed for operation in the appropriate operating environment, at the same time remaining independent from the data core.

The reliability of such a system is enhanced by the independence of subsystems from each other and from the core. Thus, the availability of each of the subsystems increases: low or no productivity of one of them will not affect the availability of the other.

Independence from location. The request receives the required node value from the core and redirects the system to a given piece of data. Such modeling enables developers to unify writing code optimizing it by using agreed schemes of private models.

Independence from fragmentation. Since each subsystem is locally independent and has the relevant data, when the core is restored from fragments, you achieve minimum information loss.

Minimizing the use of networks. Since each subsystem is locally independent, it is possible to minimize the use of distributed queries. As for the tactical and strategic objectives, they, on the contrary, can be more efficiently addressed through the core (*Figure 2*).

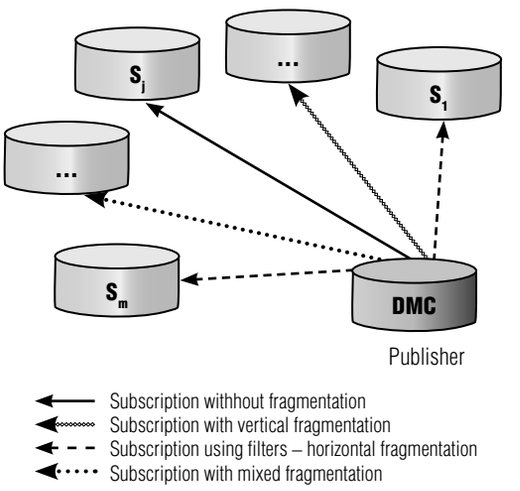

Fig. 2. Data replication DMC. Possible options for subscriptions

6. Approach to the criterion selection k

Paradoxically, the more information appears in the processes of various subsystems, the less it is susceptible to various changes. In this case, reference to it occurs much less frequently. The value of such information is higher than the cost of its storage. As for information spe-

cific to the tasks of the operational level, it is, on the contrary, more often subject to modification. Therefore, the number of references to such information (such tasks are solved more often than tactical or strategic tasks) increases.

Thus, when solving operational tasks in subsystems, the number of references to the core will increase and the volume of such information will rise. All this can reduce the effectiveness of using distributed data. It is appropriate to store such information in a distributed environment, access to which will take place only when solving this task exactly by this unit.

Therefore, to minimize using networks (the goal of any distributed database), the value of k should be such that the information storage cost of an entity with capacity $|r|$ providing task solving is lower than the cost of necessary references to it (*communication costs* (*CC*) and updating costs (*updating costs* (*UC*). These operations require numerous dispatches of large amounts of data from one node to another (or others) with the aim of necessary updating of the information and expectation of update delay.

The proposed model assumes that there is a distributed database consisting of m sites

$$S = \{DMC, S_1, S_2, ..., S_m\}$$

built on the principles of consistency of models on the basis of the core. The relationship between DMC and S_j has a positive integer of communication costs (*communication costs* – CC_j) representing the cost of transfer of data blocks from DMC on S_j, and a positive integer of updating costs (*updating costs* – UC_j) that arise with delay l.

For solving a specific task, there are a lot of queries $Q = \{Q_1, Q_2, ..., Q_l\}$ which are the most common queries, and which account for the bulk load processing, the total value of which is defined as QC_j.

Let t be a frequency of solving the task of search, v – data update frequency, k – the number of sites where the entity is distribution.

If we select an entity in the constitution of the core, then the full cost is:

$$TC_c = t\,(CC_j + QC_j) + 2\,v\,UC_j\,.$$

This is also true for an entity which is not included in the core:

$$TC_s = t\,k\,(CC_j + QC_j) + v\,UC_j\,.$$

When updating data, it is necessary to update two subsystems – the core and directly the subsystem for which the data is up to date (replication of vertical fragmentation).

Designating $a = t\,(CC_j + QC_j)$, $b = v\,UC_j$ it is possible to see two functions whose growth can be assessed depending on k:

$$TC_s = a\,k + b;$$
$$TC_c = a + 2b\,.$$

When $k = 1$, an option of entity placement in one site is more preferable.

When $k = 2$, it is necessary to estimate an amount equal to $a - b$, as well as in time calculations additionally, to take into account the value of l (delay on replication of data from the core).

Excluding such entity from the core, we also reduce the power ($|r|$) of this entity and do not increase costs QC_j.

If system performance is important, then the problem of optimality, according to the proposed model of redistribution, can be defined as minimizing the communication costs in the redistribution process of this relationship from DMC for the required fragments from S.

When $k > 2$, the value TC_s increases the cost of the queries associated with the search for various fragments and data transfer over the network will greatly exceed the efficiency of their storage in one place – the database core.

7. Results

The results of the proposed approach has been tested on test data of an automated control system (*ACS*) of the university in two departments in the first two courses of the I. Razzakov Kyrgyz State Technical University.

When building the model, private models were allocated and a unified vocabulary of attributes was created. When considering the learning process, some sub-models were identified, and the next step model core was established.

All private models are updated at the expense of core data. Therefore, data on directions and plans with their contents as well as about the educational units involved in the educational process will receive the same datasets. The essence of the experiment is to measure what is required for each data management system for the execution of a query to meet the information needs of the *ACS* employee of the university.

For solving problems at the operational level of the system, two options of queries were offered: to the centralized and distributed databases. The following figures show the results indicating a time difference under identical conditions of hardware using a Microsoft SQL Server 2008 on the same computer. Subscriptions are distributed across different databases. The purpose of the request to search for the student by name and number of the record book, outputting a list of subjects and grades at the moment.

Table 1.

Parameters of measurement requests submitted to the centralized database

No	Parameters					
	Total memory (Mb)	Occupied memory (Mb)	Free memory (Mb)	Temperature of the processor (0C)	Voltage of the processor (V)	Current force (A)
1	3071	1559	1512	33	1.072	9.11
2	3071	1569	1502	33	1.072	9.11
3	3071	1569	1502	33	1.072	9.11
4	3071	1572	1499	33	1.072	9.11
5	3071	1618	1453	33	1.072	9.11
6	3071	1571	1500	33	1.064	9.11
7	3071	1563	1508	33	1.128	9.11
8	3071	1578	1493	33	1.072	9.11
9	3071	1570	1495	33	1.072	9.11
10	3071	1583	1488	33	1.072	9.11
average	3071	1575	1496	33	1.077	9.11

Table 2.

Parameters of measurement requests submitted to the distributed database

No	Parameters					
	Total memory (Mb)	Occupied memory (Mb)	Free memory (Mb)	Temperature of the processor (0C)	Voltage of the processor (V)	Current force (A)
1	3071	1630	1441	33	1.072	9.11
2	3071	1581	1490	34	1.072	9.11
3	3071	1585	1586	33	1.072	9.11
4	3071	1559	1512	34	1.072	9.11
5	3071	1578	1493	33	1.072	9.11
6	3071	1579	1492	33	1.072	9.11
7	3071	1621	1450	33	1.072	9.11
8	3071	1581	1490	33	1.072	9.11
9	3071	1578	1493	33	1.072	9.11
10	3071	1575	1496	33	1.072	9.11
average	3071	1586	1494	33	1.072	9.11

The results of the experiment are shown in a graph (*Figure 3*); the query runs 10 times every 30 minutes.

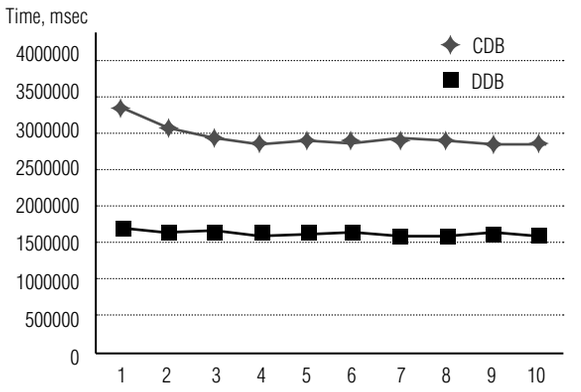

Fig. 3. The results of the experiment

Conclusion

Physically, the architecture can be represented as a family of servers supporting private subschema of data models, one for each category of functional tasks. One of them is the server that contains the schema of the data model core. It is possible to implement one of the following solutions:

✦ the schema of the data model core is the reference model on the basis of which private models of servers are formed;

✦ the schema of the data model core is a model of a database which collects all input and edited data (where they appear) in one place, and then replicates them on private models. In other words, the database assumes

the dispatching functions that support the integrity and non-redundancy of data in private *DBMS*. All functionality of this *DBMS* is directed only to the maintenance of the dispatching functions (replication, maintenance of integrity and non-redundancy).

✦ the schema of the data model core is the base model of DBMS that supports the main business and management functions of the organization implemented in the *IEIS* framework. All other *DBMSs* provide (accompany) the business and management functions of the organization implemented in the form of data for various applications.

There are other possible architectural solutions that are a combination of the listed structures.

Selection of a particular structural organization of data space may be performed either on the basis of expert assessment, or by mathematical solution of the optimal choice task.

The advantages of the proposed approach, first of all, consist in coherence of distributed data, scalability of the system and its rapid adaptation to the changing environment, as well as in the binding of the system not to specific users but to business functions. ■

References

1. Yu C.T., Chang C.C. (1983) On the design of a query processing strategy in a distributed database environment. Proceedings of *SIGMOD'83, Annual Meeting, San Jose, California, May 23–26, 1983.* New York, ACM Press, pp. 30–39.
2. Kul'ba V.V., Kovalevskij S.S., Kosjachenko S.A., Sirotjuk V.O. (1999) *Teoreticheskie osnovy proektirovanija optimal'nyh struktur raspredelennyh baz dannyh* [Theoretical bases of designing optimal structures of distributed databases]. Moscow: Sinteg (in Russian).
3. Lazdyn' S.V., Zemljanskaja S.Y. (2009) Optimizacija raspredelennyh korporativnyh informacionnyh setej s ispol'zovaniem geneticheskih algoritmov i ob'ektnogo modelirovanija [Optimization of distributed corporate information networks using genetic algorithms and object modeling]. *Naukovi praci DonNTU*, no. 147, pp. 83–95 (in Russian).
4. Pavlov D.V. (2012) Relyatsionnaya raspredelennaya sistema upravleniya bazami dannykh s avtomaticheskoy masshtabiruemost'yu [Relational distributed database management system with automated scalability]. *Vestnik UGATU*, vol. 16, no. 3 (48), pp. 138–142 (in Russian).
5. Chen P.P.-S., Akoka J. (1980) Optimal design of distributed information systems. *IEEE Transactions on Computers*, vol. c-29, no. 12, pp. 1068–1080.
6. Abdalla H.I. (2012) A new data re-allocation model for distributed database systems. *International Journal of Database Theory and Application*, vol. 5, no. 2, pp. 45–60.
7. Tamer Ozsu M., Valduriez P. (1991) *Principles of distributed database systems.* Upper Saddle River, NJ: Prentice-Hall. P. 125–128.
8. Silin A.V., Vorobyov V.I., Revunkov G.I. (2005) Metody i modeli proektirovaniya struktur territorial'no-raspredelennykh baz dannykh [Methods and models for design of structures of geographically distributed databases]. *VINITI*, no. 3282-00B, pp. 21–25 (in Russian).
9. Beskorovainy V.V., Ulyanova O.S. (2010) Metody sinteza fizicheskikh struktur raspredelennykh baz dannykh [Methods of synthesis of distributed databases' physical structures]. *Open Information and Computer Integrated Technologies*, no. 47, pp. 136–146 (in Russian).
10. Maier D. (1983) *The theory of relational databases.* Computer Science Press.
11. Bell D.A., Grimson J.B. (1992) *Distributed database systems.* Wokingham, England: Addison-Wesley.
12. Babak V., Kasymalieva A., Akmatbekov R. (2008) Podderzhka raspredelennyh dannyh v sistemah avtomatizirovannogo upravlenija uchebnym processom vuza [Support for distributed data in computer-aided learning management of the university]. *Izvestija KGTU*, no. 13, pp. 333–337 (in Russian).

Построение семейства согласованных моделей данных в распределенных базах данных, основанных на семантике

А.Т. Касымалиева

доцент кафедры информационных систем в экономике
Кыргызский государственный технический университет им. И.Раззакова
Адрес: Кыргызская Республика, 720044, г. Бишкек, пр. Мира, 66
E-mail: aisu@rambler.ru

Аннотация

При разработке проектов интегрированных корпоративных систем с ориентацией на дальнейшее расширение функций необходимо помнить о преемственности создаваемых моделей и своевременности их уточнений. Эти уточнения соотносятся с перспективой развития организаций или их подразделений, занимающихся разработкой информационных систем (ИС) или других программных продуктов (ПП),

расширением функциональности интегрированных корпоративных информационных систем (ИКИС), а также развитием сред проектирования и программирования. В связи с этим автором предлагается применение расширенной схемы уровней моделирования данных и баз данных.

При изучении функций каждого подразделения и построении их модельного описания в подсистемах (частных моделях) можно выделить одни и те же объекты для обеспечения функциональности. Согласованность является одним из преимуществ создаваемой модели, обеспечивая типизацию и стандартизацию процессов создания ИС.

В работе используются механизмы распределения данных, которые на сегодняшний день являются весьма актуальными. Предложенное решение на основе семантического словаря, отражающего основные термины и понятия функциональных задач бизнес-среды моделируемого предприятия, позволяет унифицировать разработку приложений и дополняет стратегию распределения данных по узлам предприятия.

В статье изложены принципы формирования семейства согласованных моделей данных, приведены их формальные описания, разработаны алгоритмы и возможные практики формирования ядра. Рассматриваются преимущества использования и подходы к возможному применению.

Ключевые слова: распределенные базы данных, информационные системы, моделирование данных, ядро модели данных.

Цитирование: Kasymalieva A.T. Construction of a set of harmonized data models in distributed databases based on semantics // Business Informatics. 2016. No. 2 (36). P. 48–56. DOI: 10.17323/1998-0663.2016.2.48.56.

Литература

1. Yu C.T., Chang C.C. On the design of a query processing strategy in a distributed database environment // Proceedings of SIGMOD'83, Annual Meeting, San Jose, California, May 23–26, 1983. New York: ACM Press, 1983. P. 30–39.
2. Теоретические основы проектирования оптимальных структур распределенных баз данных / В.В. Кульба и [др.]. М.: Синтег, 1999. 660 с.
3. Лаздынь С.В., Землянская С.Ю. Оптимизация распределенных корпоративных информационных сетей с использованием генетических алгоритмов и объектного моделирования // Наукові праці ДонНТУ. 2009. № 147. С. 83–95.
4. Павлов Д.В. Реляционная распределенная система управления базами данных с автоматической масштабируемостью // Вестник УГАТУ. 2012. Том 16, № 3 (48). С. 138–142.
5. Chen P.P.-S., Akoka J. Optimal design of distributed information systems // IEEE Transactions on Computers. 1980. Vol. c-29. No. 12. P. 1068–1080.
6. Abdalla H.I. A new data re-allocation model for distributed database systems // International Journal of Database Theory and Application. 2012. Vol. 5. No. 2. P. 45–60.
7. Tamer Ozsu M., Valduriez P. Principles of distributed database systems. Upper Saddle River, NJ: Prentice-Hall, 1991. P. 125–128.
8. Силин А.В., Воробьев В.И., Ревунков Г.И. Методы и модели проектирования структур территориально-распределенных баз данных // Деп. рук. ВИНИТИ № 3282-00В. 2005. С. 21–25.
9. Бескоровайный В.В., Ульянова О.С. Методы синтеза физических структур распределенных баз данных // Открытые информационные и компьютерные интегрированные технологии. 2010. № 47. С. 136–146.
10. Maier D. The theory of relational databases. Computer Science Press, 1983. 656 p.
11. Bell D.A., Grimson J.B. Distributed database systems. Wokingham, England: Addison-Wesley, 1992. 426 p.
12. Бабак В., Касымалиева А., Акматбеков Р. Поддержка распределенных данных в системах автоматизированного управления учебным процессом вуза // Известия КГТУ. 2008. № 13. С. 333–337.