Synthesis of Semantic Model of Subject Area at Integration of Relational Databases

Valentin Filatov
Artificial Intelligence Department
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
valentin.filatov@nure.ua

Valerii Semenets
Department of Metrology and Technical Expertise
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
valery.semenets@nure.ua

Oleh Zolotukhin
Artificial Intelligence Department
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
oleg.zolotukhin@nure.ua

***Abstract*—The article considers the problem of synthesizing a semantic model of the subject area at integration of heterogeneous information resources. In this case, emphasis is placed on ensuring the universality of the means of description, without regard to artificial limitations on the data typification and categorization. Two types of logical existence rules are introduced: functional and structural ones, which allow analyzing not only explicitly defined, but also logically deducible information objects, that is, determining the boundary of the subject area. Information about all possible information objects of the subject area makes it possible to determine the area of intersection of the integrable data semantics.**

Keywords—reengineering, relational database, data semantics, subject area, data integration, data model.

I. INTRODUCTION

Intensive introduction of information technology (IT) in all areas of modern society over the past 15 years has set a new task for the IT community – the integration of the accumulated data and metadata of various environments and software systems and the construction of a single information space. In this case, the complexity of the problem lies in the fact that all the local databases (DB) were created on the basis of different, often contradictory concepts, methodologies, models and approaches [1-3].

Currently, leading companies – software developers, such as Microsoft, Oracle, and others, have not managed to develop a unified approach to creating software systems that effectively solve the problem of integrating heterogeneous databases within a single information space. The problem of unification of mathematical, linguistic, software and instrumental means to design, implement and maintain large-scale information systems (IS) is the focus of a number of research teams being solved [4].

The task of enterprise application integration (Enterprise Application Integration – EAI) can be divided into two classes: the data integration and application integration. Integration at the data level is the preferred way to build a single information space, it is not sufficient in those cases where the application logic is inseparable from the data themselves. Integration at the level of the database management system (DBMS), even in the "read only" mode, is impossible, since in this case the exact structure and contents of the tables are unknown. The architecture of the integrating environment must support such integration models in which the dependence between the subsystems is minimal. In this case, dependence means not only the need to use data structures and technical solutions, inherent in any particular subsystem, for the integration, but also the very fact of their existence.

It should be noted that the choice of the integration method essentially depends on the specifics of the integrated applications and available technical resources. The main integration models are: physical data integration and logical data integration.

This choice in addition to functional requirements of the application system, depends on such factors as quality and relevance of the data, availability of the source code of the application subsystems, quality of technical documentation, intensity of work with the subsystem, the features of network access and other technical characteristics [5].

II. MAIN PROBLEMS OF SOLVING TASKS OF INFORMATION SYSTEMS INTEGRATION

Information systems for solving problems of input, efficient storage, processing and output of information should be designed and developed as open systems, all components of which are interoperable. It is difficult to achieve such an implementation in practice. Among the main reasons leading to the disagreement of information resources, the following can be highlighted:

1. Heterogeneity, distribution and autonomy of information resources of the system. Heterogeneity of resources may be syntactic and/or semantic. A purely realizable heterogeneity of information resources is also possible, due to the use of different computer platforms, operating systems, DBMS, programming systems, etc.

2. Needs for integration of the IS components. Obviously, the most natural way to organize a complex IS is its hierarchically-nested construction. More complex function-oriented components are built on the basis of simpler components, which could be designed and developed independently, giving rise to heterogeneity.

3. Reengineering system. After creation of the initial version of the IS, the process of its continuous alterations inevitably follows, due to the development and change of the corresponding business processes.

4. Solution to the problem of inherited systems. With time, any computer system becomes an object of attention for an organization that exploits it, since it is constantly necessary to solve the problem of embedding outdated information components into a system based on a new technology.

5. Extension of the IS life cycle. The longer the IS functions, the greater the need to change and/or add components designed and developed to solve new problems.

978-1-7281-1814-7/19/31.00

The practical solution to the problem of integrating heterogeneous information resources begins with attempts to integrate databases (DB). The direction of integrated or federated systems of heterogeneous DB and multi-databases emerged due to the need to share data based on different data models and managed by different DBMS [6,7].

One of the solutions to the problem of integrating heterogeneous databases is to provide users with the opportunity to see the global schema. With strict integration of heterogeneous DB, local systems lose their autonomy. After the inclusion of the local DB in the federal system, all further actions with it, including administration, should be carried out at the global level. The direction of multi-databases is developing since IS often do not agree to lose local autonomy, wishing, nevertheless, to be able to work with all local DBMS in the same language.

The global integrated database schema is not supported in the systems of this class, and special naming methods are used to access local DB objects. As a rule, only data sampling is allowed in such systems at the global level. This makes it possible to preserve the autonomy of local systems.

The relational data model is the most often used for the external representation of the integrated databases. Therefore, the inclusion of a local relational DBMS in an integrated system is much simpler and more efficient than the inclusion of a DBMS based on a different data model.

The main conclusion from the considered material is that the problems of integrating heterogeneous information resources are relevant when considering the functioning of information systems, and this requires the use of reasonable combinations of technological and architectural solutions.

III. SUBJECT AREA MODELING

By a data model, we mean a representation making it possible to implement the interpretation of data in accordance with the specified requirements inherent in the subject area. The concept of a model is closely related to the concept of abstraction. The abstraction of any system represents a model of this system, in which some details are intentionally omitted [8-11].

The most general and rigorous concept of a model is defined in mathematics. A model is the main set of objects O and the set of relations D on O. For example, $O = \{a, b, c\}$ between the elements of which an ascending order relationship is established, that is, the condition $a < b < c$ must be satisfied at any time. Thus, the model M can be represented by a pair

$$M = <O, D> \quad (1)$$

In contrast to the model, IT models must be comprehensible to the computing system, and therefore must include not only the structural, but also the operational specification. Algebraic systems are the most acceptable abstraction for modeling the subject area. In addition to the set of objects O and relations between objects D they also combine the set of operations (functions) Ω, defined on the main set O. At the same time, model representations can be defined by three elements

$$M = <O, D, \Omega> \quad (2)$$

To formalize the database representation, the term "model" will be used in the future, implying a triple $<O, D, \Omega>$ taking into account the introduced definitions. Although this model will include the rules describing possible states of the subject area for a more detailed presentation of the DB semantics [12].

It is not possible to present the DB in a strict mathematical form, and even more so, a certain fixed subject area because of the rigor of abstractions of the model description. That is, it is impossible to express the meaning of objects from R, D and Ω in the model. In reality, when designing a DB, it is possible to use several types of models. Each model is determined by different types of connections between the objects of the subject area.

One method of establishing links between the objects is to distribute them into categories. Objects of the same category are considered to be similar, and similarity characteristics are usually given by the category properties.

The models are divided into two types: strongly typified models, in which it is assumed that all objects should be assigned to any category; poorly typified models, which are not related to any assumptions regarding categories [13-15].

Modeling using the predicate calculus involves working with linear texts, and can be written in conventional mathematical notation. The predicate calculus is not characterized by excessive complexity and poor structuring. In this approach, emphasis is placed on ensuring the universality of the means of description, without regard to artificial limitations on the data typification and categorization.

The model determines the rules according to which the data are structured. However, structural specifications do not provide the ability to interpret data semantics fully and way to use them. Operations on objects and data should also be defined.

Considering the subject area, let us define the properties that it displays. Let us distinguish two classes of properties: static and dynamic ones. Static properties include the properties that are always fair and unchanged. Dynamic properties characterize possible changes in the subject area. Any model must represent these two classes in any way.

We will assume that the set of objects is determined by the requirements of the subject area. The choice of permissible implementations of objects or relations between them is specified by specifying constraints in the form of a set of rules defining dependencies between objects and a set of output (calculated) objects.

We define the model M as the set of objects O, the set of rules L, and the set of operations Ω. The rules will be given in the form of the implication (3)

$$o \leftarrow o_1, o_2, \ldots, o_n, \quad (3)$$

where o_i - are the objects of the subject area $(i = 1 \div n)$.

Rule (3) determines, if all objects o_i belong to O, then o must also belong to O. The rules L, generating additional objects o_i, with the help of operations Ω define an extended set of objects S, including both given and derived objects of the subject area. Thus, the model can be defined as

$$M = (O, L, \Omega, S) \quad (4)$$

IV. FORMAL MODEL OF SYNTHESIS OF SEMANTIC MODEL OF SUBJECT AREA

As a means of defining a structural component, declarative specifications, formulas for calculating statements or calculus of first-order predicates can be used. Data objects, that satisfy the specified conditions, constitute a valid DB state.

Let us consider the DB as a set of predicates. In this case, the predicate will be considered as a functional statement. Unlike arithmetic and logical functions, where the range of values and the range of changes of the arguments by type are the same, that is, homogeneous, the predicates of the range of values of the function are logical, and the range of changes of the arguments is objective. Thus, the predicate is a non-uniform function and can be used to simulate a DB.

An atomic formula is an elementary object with a truth value in the predicate logic. This formula consists of the symbolic designation of a predicate and a term. In general, a predicate can be represented as a formula $p\ (t)$, where p is the designation of the predicate, and t is a term. The number of terms determines the dimension of the predicate. A predicate is a function that returns a Boolean value depending on the value of a term [16-18].

The predicate will be considered as an information component, reflecting the value of the corresponding DB object. In other words, if A is some DB object, then

$$p(t) = True,\ t \in A,\ False,\ t \notin A. \qquad (5)$$

This presentation can be used to describe the DB semantics. For a relational model

$$\rho = Dom\ A_1 \times Dom\ A_2 \times \ldots \times Dom\ A_n\ , \qquad (6)$$

where $Dom\ A_i$ – is a set of valid attribute values A_i or an attribute domain A_i $(i = 1 \div n)$,

ρ represents a set of n tuples on carriers $O(A_1, A_2, \ldots, A_n)$ expressing the DB semantics.

At the same time, the predicate model represents a conjunction of a finite set of predicates corresponding to the DB relational scheme presented as

$$P = p_1(t_1)\ \&\ p_2(t_2)\ \&\ \ldots\ \&\ p_n(t_n), \qquad (7)$$

where $p_i(t_i)$ – is the predicate, corresponding to the property (5) and $1 \le i \le n$,

P – is a set of objects expressing the DB semantics.

Then the DB carrier can be represented as a set of distribution single predicates.

$$O(p_1(t_1), p_2(t_2), \ldots, p_n(t_n)) \qquad (8)$$

where predicate $p_i(t_i)$ corresponds to property (6), $1 \le i \le n$ and displays the values of the corresponding objects of the subject area.

Let us fix some alphabet $\Re$, containing constants, variables and predicates. Let $p\ (t)$ formula be called a positive literal l, and $\neg p(t)$ formula be called a negative literal $\neg l$ for one-place predicate p. A basic literal is a positive or negative literal that does not contain variables. Thus, the set (9) will represent the DB extension, and will be written as

$$O\ (l_1, l_2, \ldots, l_n) \qquad (9)$$

The DB semantics will be defined by a set of rules of the form

$$L = \{l \leftarrow l_1, l_2, \ldots, l_m\}, \qquad (10)$$

where $l, l_1, l_2, \ldots, l_m$ – are the literals and $m \ge 1$.

The rule can be read as the expression "if $l_1, l_2, \ldots, l_m$ holds, then l is also executed" and it can express DB intensional properties. The condition for the admission of objects in the DB S semantics is that if all literals $l_1, l_2, \ldots, l_m$ are in O, then l can be included in S. If such a condition is not fulfilled and the literal defined by l is included in S without defining literals $l_1, l_2, \ldots, l_m$, then a violation of data consistency is possible.

The main condition for the DB correctness is in the compatibility of objects in S. Compatibility consists in the absence of one and the same positive and negative literal.

Thus, for the predicate model, we will also consider two types of rules defining extensional L^{ext} and intensional L^{int} and the general set of rules, respectively, as $L = L^{ext} \cup L^{int}$. It is assumed that the elements of O may vary depending on the requirements for the DB, and it is necessary to adjust the rules describing the subject area so as not to lose the connections between the objects, their detailing and possible final operations. The DB semantics should automatically change in accordance with the changes in O and L.

Definition 1. A modification of the set O is determined by the operation of adding or deleting literal l, at performance of which S remains joint.

Definition 2. Let $DB_1(O_1, L_1)$ and $DB_2(O_2, L_2)$ be DB and let S_1 and S_2 be the corresponding semantics. Then $DB_1 \equiv DB_2$, , if every rule in L_1 belongs to S_2 and every rule from L_2 belongs to S_1

Based on the concepts, it can be concluded that the information content of the DB O can be expanded in accordance with its semantics S by means of the specified rules L. Thus, the problem arises of calculating the semantics of the database S, and S must be calculated each time the elements O are changed. Let us consider the algorithm for calculating S.

Input data: Database $DB\ (O, L)$.

Output data: Semantics of the database S.

Method: compute literal sequence S according to the following rules.

1. S^0 is O.

2. S^{i+1} is S^i plus a set of literals l_i, such that there is some rule $l \leftarrow l_i$, and $l \in S^i$ in L. Since $S = S^0 \subseteq \ldots \subseteq S^i \subseteq \ldots \subseteq A$ and A is finite, ultimately, such i, will be reached that $S^i = S^{i+1}$. That is $S^i = S^{i+1} = S^{i+2} = \ldots$. Thus, there is no need to perform calculations after S^i, if $S^i = S^{i+1}$.

The algorithm is finished.

V. EXAMPLES OF DATABASE SEMANTICS CALCULATION

1. Let $O = \{A,B\}$ and $L = \{A, B \leftarrow C;\ C \leftarrow A;\ B, C \leftarrow D;\ A, C, D \leftarrow B;\ \ D \leftarrow E, G;\ \ B, E \leftarrow C;\ \ C, G \leftarrow B, D;\ C, E \leftarrow A, G\}$. Let us assume that $S^0 = A, B$. To calculate S^1, let us find the rules that have either separate literals A or

B, or the pair A, B together in the right-hand side. There are two such rules $C \leftarrow A$ and $A, C, D \leftarrow B$. Let us attach literals C and D to S^0 and suppose $S^1 = A, B, C, D$. To calculate S^2, let us find the right-hand sides of the rules contained in S^1. Let us find $C, G \leftarrow B, D$, thereby, $S^2 = A, B, C, D, G$. To calculate S^3 let us find the rules the right-hand sides of which contain A, B, C, D, G literals either separately, or together. The rules $C, E \leftarrow A, G$ correspond to these requirements. Thus, we have $S^3 = A, B, C, D, E, G$ – the set of all literals. Therefore, further calculation will not change the semantics, since $S^3 = S^4 = \ldots = S$. As a result, the semantics of the original DB, *DB (O, L),* corresponds to $S = \{A, B, C, D, E, G\}$.

2. Let two databases $DB_1(O_1, L_1)$ and $DB_2(O_2, L_2)$ be given. Let we say that DB_1 and DB_2 are equivalent (in the notation $DB_1 \equiv DB_2$) if their semantics are equal $S_1 = S_2$. To verify equivalence, it is necessary for each $X \leftarrow Y$ rule from the set of L_1 rules to check whether the left parts of these rules are contained in S_2, in this case algorithm 1 can be used to calculate the semantics. If every literal from the left-hand side in L_1 belongs to S_2, then every literal from S_1 will also belong to S_2, and if the inverse statement is also true, then the statement $S_1 = S_2$ is also true.

When comparing two DBs, it is necessary to compare their semantics. In this case, if during the DB modification its semantics hasn't changed, then we can assume that the informational content also remains unchanged.

CONCLUSIONS

A strictly formalized method for presenting the semantic properties of information objects of the subject area is offered, it allows us to obtain a semantic model of an integrated IS.

An axiomatic approach to the description of the subject area is formulated, which allows us to consider the problem of modeling relationships between the elements of the subject area as a set of rules determining the existence of data elements.

To formalize the representation of DB components, when solving the integration problem, the choice of the predicate model for describing the semantic properties of the subject area is justified. The types of inhomogeneity in the DB are identified, the options for integrating information in IS are analyzed. The main components of the subject area model are considered and formal definitions are given to the basic entities: the set of information objects, the relationship between information objects (rules of logical existence); structural and information integrity support system.

Two types of logical existence rules are introduced: functional and structural, which allow analyzing not only explicitly defined, but also logically deducible information objects, that is, determining the boundary of the subject area. A formalized infological model of the subject area, focused on the semantic relations between information objects of DB, is proposed. Information about all possible objects of the subject area determines the area of intersection of the semantics of data being integrated.

Practical significance of the obtained results is confirmed by the positive effect of applying the tools developed in the process of building modules for managing integrated information resources in various areas of economic activity, in particular, the distributed regional information-analytical system "Ukrainian Mobile Communications".

REFERENCES

[1] KRAVCHENKO, Y., KULIEV, E. and KURSITYS, I., 2017. Information's semantic search, classification, structuring and integration objectives in the knowledge management context problems, Application of Information and Communication Technologies, AICT 2016 - Conference Proceedings 2017.

[2] V. Filatov, V. Radchenko, "Reengineering relational database on analysis functional dependent attribute" Proceedings of the X Intern. Scient. and Techn. Conf. "Computer Science & Information Technologies" (CSIT'2015), 14-17 sept. 2015, Lviv, Ukraine, pp. 85-88.

[3] GRACHEV, V.M., ESIN, V.I., POLUKHINA, N.G. and RASSOMAKHIN, S.G., 2014. Technology for developing databases of information systems. Bulletin of the Lebedev Physics Institute, 41(5), pp. 119-122.

[4] V.I. Yesin, "Reinzhynirynh isnuyuchykh baz danykh" in Systemy obrobky ynformatsyy, KHNU im. V.N. Karazina, Kharkiv 2012, vyp. 3 (101), tom 2, pp. 188-191.

[5] S.M. Konstantinov, Yu.L. Ponomarenko, V.O. Filatov, "Chastkovo vidobrazhennya modeley Danykh pry intehratsiyi informatsiynykh system" in Ekonomiko-matematychne modelyuvannya sotsialno-ekonomichnykh system. Zb. nauk. prats, Kyyiv, 2016, pp. 140-158.

[6] YESIN, V.I., 2018. A cybernetic approach to solving the problem of database reengineering. Telecommunications and Radio Engineering (English translation of Elektrosvyaz and Radiotekhnika), 77(5), pp. 399-409.

[7] HOFMANN, T., 2004. Latent semantic models for collaborative filtering. ACM Transactions on Information Systems, 22(1), pp. 89-115.

[8] CODD, E.F., 1979. Extending the database relational model to capture more meaning. ACM Transactions on Database Systems (TODS), 4(4), pp. 397-434.

[9] Tsikritzis D. Modeli dannykh / Tsikritzis D., Lokhovski F.; per. s angl.- M.: Finansy i statistika, 1985. —344 s.

[10] The Method and Algorithms to Find Essential Attributes and Objects of Subject Domains / V. Mezhuyev, E. Malakhov, D. Shchelkonogov // IEEE 2015 : International Conference on Computer, Communication, and Control Technology (I4CT 2015), (21-23 April 2015, Kuching, Sarawak, Malaysia). – Kuching ; Sarawak ; Malaysia, 2015. – P. 310-314. – ISBN 978-1-4799-7952-3.

[11] VOORHEES, E.M., 1994. Query expansion using lexical-semantic relations, Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994 1994, pp. 61-69.

[12] V. Filatov, "Fuzzy models presentation and realization by means of relational systems" in Econtechmod: an international quarterly journal on economics in technology, new technologies and modelling processes. Lublin ; Rzeszow, 2014, Vol.(3), № 3, pp. 99-102.

[13] FAGIN, R., KOLAITIS, P.G., MILLER, R.J. and POPA, L., 2005. Data exchange: Semantics and query answering. Theoretical Computer Science, 336(1), pp. 89-124.

[14] BAGHERI, H., TANG, C. and SULLIVAN, K., 2017. Automated Synthesis and Dynamic Analysis of Tradeoff Spaces for Object-Relational Mapping. IEEE Transactions on Software Engineering, 43(2), pp. 145-163.

[15] V. Filatov, V. Semenets, "Methods for Synthesis of Relational Data Model in Information Systems Reengineering Problems", Proceedings of the Intern. Scientific-Practical Conf. "Problems of Infocommunications Science and Technology" (PIC S&T'2018), 9-12 oct. 2018, Kharkiv, Ukraine, pp. 247.

[16] O. Zolotukhin, M. Kudryavtseva. Authentication Method in Contactless Payment Systems International Scientific and Practical Conference «Problems of Infocommunications. Science and Technology», 9–12 October, 2018, Kharkiv, Ukraine, pp. 397-400.

[17] HAMMER, M. and MCLEOD, D., 1981. Database Description with SDM: A Semantic Database Model. ACM Transactions on Database Systems (TODS), 6(3), pp. 351-386.

[18] NOY, N.F., 2004. Semantic integration: A survey of ontology-based approaches. SIGMOD Record, 33(4), pp. 65-70.