E3S Web of Conferences **365**, 04031 (2023) https://doi.org/10.1051/e3sconf/202336504031
CONMECHYDRO - 2022

Creation of intelligent information decision support systems

Murodjon Sultanov[1], *Gayrat* Ishankhodjayev[1*], *Rano* Parpiyeva[2], and *Nafisa* Norboyeva[2]

[1]Institute of Energy Problems Academy of Sciences of the Republic of Uzbekistan, Tashkent, Uzbekistan
[2]Tashkent State University of Economics, Tashkent, Uzbekistan

Abstract. The use of intelligent information decision support systems implies considering the problem area's specifics. The object of study is characterized by the following set of features: - quality and efficiency of decision-making; - vagueness of goals and institutional boundaries; - the plurality of subjects involved in solving the problem; - randomness; - a plurality of mutually influencing factors; - weak formalizability, uniqueness of situations; - latency, concealment, the implicitness of information. For the efficient and reliable functioning of agricultural facilities and enterprises, it is necessary to create and implement intelligent information systems. Over the past quarter of a century, domestic information systems have undergone a progressive evolution, both in terms of developing the theoretical principles of their construction and implementing these systems. The restructuring of agriculture, the market conditions for the functioning of objects, and agriculture enterprises have their characteristics and problems. Building the structure of intelligent decision support information systems is primarily associated with building a system model, in which both traditional elements of the control system and knowledge processing models should be defined. To solve these problems, methods of system analysis were used. The key research method is the optimization of data representation structures of databases and knowledge. The following relational data representation structures have been identified: relations, attributes, and values. In the relational model, structures are not specially allocated to represent data about entity relationships. Semantic networks use a three-level representation of data on entities and a four-level representation of data on entity relationships. The conducted studies have shown that in data representation structures, entity-relationship models are a generalization and development of the structures of all traditional data models since only in this data model there are 4-level data representations of both entities and relationships. All other traditional models are some special cases of the most general entity-relationship model.

[*] murodkhan.sultanov.1987@gmail.com

1 Introduction

At present, the creation and use of intelligent information systems for decision support (IISDS) in practice imply taking into account the specifics of the problem area, which can be characterized by the following set of features: - quality and efficiency of decision-making; - vagueness of goals and institutional boundaries; - the plurality of subjects involved in solving the problem; - randomness; - a plurality of mutually influencing factors; - weak formalizability, uniqueness of situations; - latency, secrecy, the implicitness of information, etc. IISDS are formed when creating information systems and technologies to improve decision-making efficiency in conditions associated with the emergence of problem situations. In this case, any life or business situation at the control objects is described in the form of some cognitive model, which is subsequently used as the basis for constructing and conducting simulations, including computer simulations. For the efficient and reliable functioning of agricultural facilities and enterprises, it is necessary to create and implement an IISDS. During the last quarter of a century, domestic information systems have undergone a progressive evolution, both in terms of developing theoretical principles for their construction and implementing these systems. A significant contribution to this difficult work was made by V.A. Barinov, A.I. Bartolomey, F.D. Goldenberg, A.F. Bondarenko, V.V. Bushuev, V.P. Vasin, V.A. Venikov, N.I. Voropay, V.E. Vorotnitsky, A.Z. Gamm, A.F. Dyakov, Yu.S. Zhelezko, A.G. Zhuravlev, N.I. Zelenokhat, Gustav Olsson, Gianguido Piani, V.I. Idelchik, G.L. Kemelmacher, I.N. Kolosok, V.G. Kitushin, JI.A. Koshcheev, L.A. Krumm, Yu.N. Kucherov, Yu.Ya. Lyubarsky, M.I. Londer, K.G. Mityushkin, V.L. Nesterenko, V.G. Ornov, Yu.I. Morzhin, M.A. Rabinovich, S.I. Palamarchuk, V.I. Rozanov, Yu.N. Rudenko, V.A. Semenov, S.S. Smirnov, Yu.A. Tikhonov, Yu.A. Fokin, E.V. Tsvetkov, M.I. Londer, A.P. Chepkasov and others [1-6].

With the development of high-performance computer technology, information systems (IS) are an effective means of solving systemic problems. The works of M.K. Chirkova, S.P. Maslova, V.N. Petrov, D. Mark, K. McGowan. The issues of developing information systems for various purposes by methods of system analysis using modern object-oriented programming languages and database technologies are widely covered in the works of V.P. Agaltsov, K.Yu. Bogachev, V.I. Vasilyeva, B.G. Ilyasov, E. Jordan, D.M. Mutushev, and others. To a lesser extent, this affected the problems of creating adapted methods for developing special information systems for agricultural complexes. Separate aspects devoted to the methods of system analysis and decision-making on the creation and deployment of information systems for monitoring the parameters of agricultural complexes for smart grid technologies are considered in the works of C.C. Liu, D.A. Pierce, K.Y. Lee, Z.A.Vale, B.B. Kobets, I.O. Volkova and others. In this direction, there is also a certain number of legal documents, both international and domestic, partially describing the direction of development of standards in the field of intelligent grids [7-10].

Restructuring of agriculture, market conditions for the functioning of objects, and agriculture enterprises have their characteristics and problems. To solve these problems, it is necessary to create and use an IISDS, which ensures cost reduction in the production of agricultural products, a decrease in the level of losses in the transport of heat and electricity, and optimization of the size and placement of reserve capacities. Modernization of agricultural facilities and enterprises will lead to the financial independence of agricultural complexes, which is provided by funds received from the sale of agricultural products.
Let us consider in more detail the block diagram of the IISDS, which is presented in Fig.1.

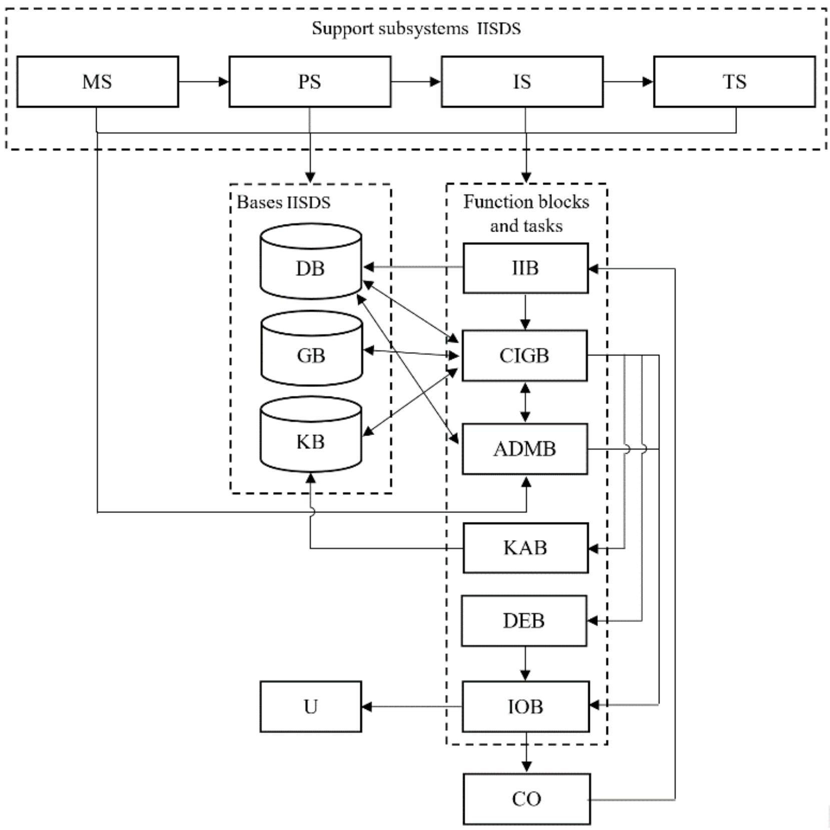

Fig. 1. Structural diagram of IISDS

In this figure, the system's input is the information input block (IIB), designed to enter numerical data, text, and speech. Information at the system's input can come (depending on the problem being solved) from the user, the external environment, and the control object. Further, the input information goes directly to the database (DB) - a set of tables that store, as a rule, symbolic and numerical information about the objects of the subject area or a control information generation block (CIGB).

CIGB using database information, a goal base (GB is a set of local goals of the system, which is a set of knowledge activated at a particular moment and in a particular situation to achieve a global goal), and a knowledge base (KB is a totality of knowledge, for example, a system of production rules, about the regularities of the subject area) provides solutions for the fuzzy formalized tasks of the IISDS, and also carries out action planning and the formation of control information for the user or control object based on the database, knowledge base, business center and using a block of algorithmic decision methods (ADMB) contains algorithms, models and software modules for solving problems in the subject area. The knowledge assimilation block (KAB) analyses dynamic knowledge to assimilate and save it in the knowledge base. The Decision Explanation Block (DEB) interprets to the user the inference sequence applied to achieve the current result.

At the output of the system, the information inference block (IOB) provides the output of data, text, speech, images, and other results of inference to the user (U) and/or the Control Object (CO).

The feedback loop makes it possible to realize the adaptability and learning properties of the IISDS. At the design stage, knowledge specialists fill the knowledge base and the goals base, and programmers develop program modules based on algorithmic problem-solving methods. The database is created and updated, as a rule, during the operation of the IISDS. The dynamics of the IISDS operation can be described as follows. When information in the external language of the system is received at the input of the IIB, it is interpreted into an internal representation to work with the symbolic model of the system. The CIGB selects from the KB a set of rules activated by the received input information and places these rules in the GB as the current goals of the system. Further, the CIGB, according to a given strategy, for example, the strategy of maximum reliability, selects a rule from the GB and tries to complete the definition of the variables of the model of the external world and the executive system with the control object. Based on this, new KB rules are activated, and logical inference begins in the system of productions (rules). This procedure ends as soon as a solution is found or when the GB is exhausted. The IOB interprets the solution found from the internal representation into the external language of the lower-level control subsystem and the CO.

The implementation of these goals is carried out by the IISDS, which consists of computer equipment, communications, telemechanics, automation systems, task complexes, algorithms, information, and software (Fig. 1) [11-14].

One of the main urgent problems to be solved when creating an IISDS is to ensure, in the conditions of its continuous operation, the adaptation of software and hardware and structures for representing data and knowledge for the prompt solution of various complex tasks.

2 Methods

IISDS should work with traditional databases (DB), regardless of their data models. The key problem here is the development of such data representation structures (DRS) for databases that would be compatible, at least, with all the main DRS of traditional data models. The problem of compatibility of DRS of different data models is solved either at the conceptual level or by building special converters from DRS of one data model to another DRS, for example, from relational to network.

As a result of a systematic analysis of the literature and the conducted research, formalized descriptions in the common language of set theory of all the main DRSs of traditional data models have been developed. Comparison and generalization of these main DRSs made it possible to propose a new approach to developing unified data representation structures.

In this paper, the following traditional data models are described and analyzed DRS in a single language: relational; network; hierarchical; entity-relationship data models; semantic networks. We emphasize that a data model combines three components: data structures, integrity constraints, and operations on data. For any data model, the DRS plays a key role in the system-forming, which we will consider. There are two simple ways to present data: tables and graphs. It should be emphasized that the descriptive capabilities of the tabular and graph representations are the same. It is easy to get a graph representation based on a table view: it is enough to specify a path connecting the attributes that form the table.

Therefore, based on the foregoing, it seems appropriate to highlight the graph-tabular DRS, which are called the following: traditional. Note that the analysis of traditional DRS showed that the graph form with table tops is the most widely used. Consider the DRS of traditional data models in more detail.

A relational data model is a set of normalized relations (tables) to which relational algebra operations are applicable. The only means of structuring data in a relational model is a relation. One of the main advantages of the relational model is its homogeneity. All data is considered to be stored in tables in which each row has the same format. A relational database intension is defined by a relational schema consisting of one or more relational schemas. The schema of a relation is given by the name of the relation and the names of the corresponding domains. The following relational DRSs are distinguished: relationship names, attribute names, and values. Consider a formalized description of relational DRSs. Let RN be the set of relation names:

$$RN = \{r_1, r_2, \dots, r_i, \dots, r_I\}, \quad (1)$$

where $i = \overline{1, I}$. Then for

$$\forall r_i \in RN \quad \exists B_i = \{b_{i1}, b_{i2}, \dots, b_{ij}, \dots, b_{iJ_i}\}, \quad (2)$$

where $i = \overline{1, I}$; $J_i = \overline{1, J_\iota}$ and B_i is the set of attribute names of the relation r_i.
Therefore, for

$$\forall b_{ij_i} \in B_i \quad \exists C_{ij_i} = \left\{c_{ij_i1}, c_{ij_i2}, \dots, c_{ij_ik_{ij_i}}, \dots, c_{ij_iK_{ij_i}}\right\}, \quad (3)$$

where $i = \overline{1, I}$; $j_i = \overline{1, J_\iota}$; $k_{ij_j} = \overline{1, K_{\iota J_\iota}}$ and V_{ij_i} is the set of attribute values b_{ij_i} relations r_i.

Let us emphasize that formulas 1-3 differ from the traditional description of relational DRS by greater structured indexes since for each relation, its own set of attributes is defined (formula 2), and for each element of the set of attribute names (for each attribute) its own individual set of values is defined (formula 3). If necessary, all elements of the same type of sets, i.e., attribute name sets or relationship attribute value sets, may be combined into a common attribute name set or value set, respectively. However, for further analysis and comparison of the DRS, it is advisable to single out different sets, as in formulas 2-3 [15-18].

A network data structure is a data structure that is a directed graph, any node of which can contain more than one connection. Network data models are based on tabular and graph representations: graph vertices are usually associated with entity types, represented by tables, and arcs are associated with connection types. The two main categories of data structures in the network model are records and links. Record types are used for the tabular representation of entity types. Links are used to representing types of links. Relationships are used to specify connections between record types. When implementing the model in various specific database management systems (DBMS), various ways of representing data in the system memory describing the relationships between entities can be used. Thus, the following main network DRSs can be distinguished: record types, data elements of record types, record type implementations, link-set types, and link-set implementations. Let's consider the formalized description of network DRS.

With this remark in mind, let TN be the set of record type names [19]:

$$TN = \{t_1, t_2, \dots, t_i, \dots, t_I\}, \quad (4)$$

where $i = \overline{1, I}$. Then for

$$\forall t_i \in TN \quad \exists B_i = \{b_{i1}, b_{i2}, \dots, b_{ij}, \dots, b_{iJ_i}\}, \quad (5)$$

where $i = \overline{1, I}$; $J_i = \overline{1, J_\iota}$ and B_i - this is the set of data element names of record type t_i. Then, in turn, for

$$\forall b_{ij_i} \in B_i \quad \exists C_{ij_i} = \left\{c_{ij_i1}, c_{ij_i2}, \dots, c_{ij_ik_{ij_i}}, \dots, c_{ij_iK_{ij_i}}\right\}, \quad (6)$$

where $i = \overline{1, I}$; $j_i = \overline{1, J_\iota}$; $k_{ij_j} = \overline{1, K_{\iota J_\iota}}$ and C_{ij_i} - this is the set of values of the data element b_{ij_i} of record type t_i. Let RS - be the set of relationship-set type names:

$$RS = \{rs_1, rs, \dots, rs_l, \dots, rs_L\}, \quad (7)$$

where $l = \overline{1, L}$. Then, for any element of the set RS, there are two sets:

$$\forall rs_l \in RS \quad \exists Y_l = \{y_{l1}, y_{l2}, \dots, y_{lm_l}, \dots, y_{lM_l}\}, \quad (8)$$
$$\forall rs_l \in RS \quad \exists X_l = \{x_{l1}, x_{l2}, \dots, x_{ln_l}, \dots, x_{lN_l}\}, \quad (9)$$

where $l = \overline{1, L}$; $m_l = \overline{1, M_l}$; $n_l = \overline{1, N_l}$ and Y_l is the set of implementations of records-owners of a relationship-set of type rs_l and X_l is the set of implementations of member records of a relationship-set of type rs_l. Wherein [19],

$$\forall rs_l \in RS \quad \exists\{< y_{1m_l}, x_{ln_l} > \mid y_{1m_l} \in Y_l,\ x_{ln_l} \in X_l\}, \quad (10)$$

A hierarchical data model is a data model based on a tree graph. The top of the tree corresponds to the record type; the arc corresponds to the relationship between the two record types. The structural diagram of a hierarchical base must be an ordered tree. In the definition tree, a node corresponds to an entity type called a record type. An entry type consists of one or more data units that can be defined on a simple domain. The arc of the definition tree corresponding to the functional type of the connection is called the initial-generated connection. It is not marked since there can be no more than one such connection between two types of records. Among the vertices, one stands out, called the root record type. Let us single out the following hierarchical DRSs: record type category (root, child), record types, record type data elements, record type implementations, and relationship-set implementations. Let us consider a formalized description of hierarchical DRSs. Let CN be the set of post-type category names (in reality, there are only two categories: root and child) [20-21]:

$$CN = \{c_1, c_2, \dots, c_i, \dots, c_I\}, \quad (11)$$

where $i = \overline{1, I}$. Next, for

$$\forall c_i \in CN \quad \exists B_i = \{b_{i1}, b_{i2}, \dots, b_{ij}, \dots, b_{iJ_i}\}, \quad (12)$$

where $i = \overline{1, I}$; $J_i = \overline{1, J_\iota}$ and B_i is the set of category record type names of record type c_i. Then, in turn, for

$$\forall b_{ij_i} \in B_i \quad \exists C_{ij_i} = \{c_{ij_i1}, c_{ij_i2}, \dots, c_{ij_ik_{ij_i}}, \dots, c_{ij_iK_{ij_i}}\}, \quad (13)$$

where $i = \overline{1, I}$; $j_i = \overline{1, J_\iota}$; $k_{ij_j} = \overline{1, K_{\iota J_\iota}}$ and C_{ij_i} is the set of data element names of record type b_{ij_i} of the category of record type c_i. Next, for

$$\forall c_{ij_ik_{ij_i}} \in C_{ij_i} \ \exists D_{ij_ik_{ij_i}} = \{d_{ij_ik_{ij_i}1}, d_{ij_ik_{ij_i}2}, \dots, d_{ij_ik_{ij_i}l_{ij_ik_{ij_i}}}, \dots, d_{ij_ik_{ij_i}L_{ij_ik_{ij_i}}}\}, \quad (14)$$

where $i = \overline{1, I}$; $j_i = \overline{1, J_\iota}$; $k_{ij_j} = \overline{1, K_{\iota J_\iota}}$; $l_{ij_ik_{ij_i}} = \overline{1, L_{\iota J_\iota k_{\iota J_\iota}\iota}}$ and $D_{ij_ik_{ij_i}}$ is the set of instances of data element $c_{ij_ik_{ij_i}}$ of record type b_{ij_i} of record type category c_i. All links of the hierarchical data model are links of functional type and are not marked. Therefore, only relationship-set implementations need to be stored. Then let

$$RS = \{rs_1, rs_2, \dots, rs_m, \dots, rs_M\}, \qquad (15)$$

where $m = \overline{1, M}$. RS is the set of implementations of relation-sets corresponding to the type of the root. Also, let

$$Y = \{y_1, y_2, \dots, y_n, \dots, y_N\}, \qquad (16)$$

where $n = \overline{1, N}$; Y is the set of implementations of relation-sets corresponding to the «born» type. Then, all connection sets are described by the following set of twos:

$$\exists\{< rs_m, y_n > \mid rs_m \in RS, \ \ y_n \in Y\} \qquad (17)$$

The Chen model (entity-relationship, ER-model) is a semantic relational data model based on dividing the real world into separate, distinguishable entities in certain relationships. On the other hand, such models have much in common with hierarchical and network data models; moreover, their focus on the design process can be seen as a generalization and development of hierarchical and network models. The basic structures in the ER model are entity types and relationship types. An entity type is called an entity set and represents the general structure of an entity. A relationship type is called a relationship set and represents the general structure of relationships between entity sets. The type of mapping corresponding to the set of relationships is explicitly specified [22-23].

A set of entities participating in a set of relationships plays a role. A domain in the ER model is called a set of values. An attribute is a mapping between an entity or relationship set and a value set in the context of an entity or relationship set. Some attributes may be multi-valued, but in this work, we restrict ourselves to the fact that in such cases, we will consider not one multi-valued attribute but several similar, similar, or identical single-valued attributes. Thus, we can single out the following DRSs of the entity-relationship model: entity classes, entity sets, relationship classes (dimensions), relationship sets, entity roles in relationships, a set of attributes, and a set of values.

Let us consider a formalized description of the DRS entity-relationship model. Let EC be the set of entity class names [19]:

$$EC = \{e_1, e_2, \dots, e_i, \dots, e_I\}, \qquad (18)$$

where $i = \overline{1, I}$. Then, for

$$\forall e_i \in EC \quad \exists B_i = \{b_{i1}, b_{i2}, \dots, b_{ij}, \dots, b_{iJ_i}\}, \qquad (19)$$

where $i = \overline{1, I}$; $j_i = \overline{1, J_i}$ and B_i is the set of entity names of class e_i. Next, let

$$\exists C = \{c_1, c_2, \dots, c_k, \dots, c_K\}, \qquad (20)$$

where $k = \overline{1, K}$; C is the set of attribute names. In addition, let

$$\exists D = \{d_1, d_2, \dots, d_l, \dots, d_L\}, \qquad (21)$$

where $l = \overline{1, L}$; D is the set of values. Next, let

$$\exists Z = \{z_1, z_2, \dots, z_m, \dots, z_M\}, \qquad (22)$$

where $m = \overline{1, M}$; Z is the set of relation class names. Then,

$$\forall z_m \in Z \quad \exists Y_m = \{y_{m1}, y_{m2}, \dots, y_{mn_m}, \dots, y_{mN_m}\}, \tag{23}$$

where $m = \overline{1, M}$; $n_m = \overline{1, N_m}$ and Y_m is the set of names of links of class z_m.
Consider the description of entities, for

$$\forall b_{ij_i} \in B_i \quad \exists C_{ij_i} = \left\{c_{ij_i1}, c_{ij_i2}, \dots, c_{ij_ip_{ij_i}}, \dots, c_{ij_iP_{ij_i}}\right\}, \tag{24}$$

where $i = \overline{1, I}$; $j_i = \overline{1, J_\iota}$; $p_{ij_j} = \overline{1, P_{\iota J_\iota}}$. Next, for

$$\forall c_{ij_ip_{ij_i}} \in C_{ij_i}, \exists D_{ij_ip_{ij_i}} = \left\{d_{ij_ip_{ij_i}1}, d_{ij_ip_{ij_i}2}, \dots, d_{ij_ip_{ij_i}r_{ij_ip_{ij_i}}}, \dots, d_{ij_ip_{ij_i}R_{ij_iP_{ij_i}}}\right\}, \tag{25}$$

where $i = \overline{1, I}$; $j_i = \overline{1, J_\iota}$; $p_{ij_j} = \overline{1, P_{\iota J_\iota}}$; $r_{ij_ip_{ij_i}} = \overline{1, R_{\iota J_\iota P_{\iota J_\iota}}}$.

In the entity-relationship data model, each relationship can be described by some set of attributes that can have certain sets of values. In other words, for

$$\forall y_{mn_m} \in Y_m \quad \exists C_{mn_m} = \left\{c_{mn_m1}, c_{mn_m2}, \dots, c_{mn_ms_{mn_m}}, \dots, c_{mn_mS_{mn_m}}\right\}, \tag{26}$$

where $m = \overline{1, M}$; $n_m = \overline{1, N_m}$; $s_{mn_m} = \overline{1, S_{mn_m}}$. Next, for

$$\forall c_{mn_ms_{mn_m}} \in C_{mn_m} \exists D_{mn_ms_{mn_m}} =$$
$$= \left\{d_{mn_ms_{mn_m}1}, d_{mn_ms_{mn_m}2}, \dots, d_{mn_ms_{mn_m}t_{mn_ms_{mn_m}}}, \dots, d_{mn_ms_{mn_m}T_{mn_ms_{mn_m}}}\right\}, \tag{27}$$

where $m = \overline{1, M}$; $n_m = \overline{1, N_m}$; $s_{mn_m} = \overline{1, S_{mn_m}}$; $t_{mn_ms_{mn_m}} = \overline{1, T_{mn_ms_{mn_m}}}$.
Let E be the set of names of roles of sets of entities in relationships

$$E = \{e_1, e_2, \dots, e_u, \dots, e_U\}, \tag{28}$$

where $u = \overline{1, U}$. Let us introduce the following notation:

$$B = \mathop{Y}_{i=1}^{I} B_i \tag{29}$$

So, each connection corresponds to a certain set of twos that describe some entities and their roles, i.e. [17]:

$$\forall y_{mn_m} \in Y_m \ \exists\{< b_{mn_m1}, e_{mn_m1} >, < b_{mn_m2}, e_{mn_m2} >, \dots,$$
$$, \dots, <_{mn_mv_{mn_m}}, e_{mn_mv_{mn_m}} >, \dots, < b_{mn_mV_{mn_m}}, e_{mn_mV_{mn_m}} > \tag{30}$$

where $b_{mn_mV_{mn_m}} \in B$; $e_{mn_mV_{mn_m}} \in E$; $m = \overline{1, M}$; $n_m = \overline{1, N_m}$; $v_{mn_m} = \overline{1, V_{mn_m}}$.

Semantic model. Represent concepts in semantic memory in the form of a graph, at the vertices of which the concepts are located, at the terminal vertices - elementary concepts, and the arcs represent the relationships between the concepts. The structure supported by any semantic network data model is a graph. Vertex categories are set according to the items they represent. An example is a system that provides for four categories of nodes:

concepts (concepts), events, characteristics (properties), and values. An additional means of providing the expressive power of semantic network data models is the distribution of vertices by type. In addition, the concept of a class hierarchy is introduced, which indicates the inheritance of the properties of one class by another. The vertices of the semantic network can be instances, classes, and meta-classes, and the arcs - statements, generation of instances, and binary relations. Let us single out the following main DRSs of semantic networks: categories of vertices, a hierarchy of vertex classes, names of vertices, categories of arcs, and names of arcs. Let us consider a formalized description of the DRS of semantic networks. Let VC be the set of vertex category names [24]:

$$VC = \{vc_1, vc_2, \dots, vc_i, \dots, vc_I\}, \tag{31}$$

where $i = \overline{1, I}$. Then, for

$$\forall vc_i \in VC \quad \exists B_i = \{b_{i1}, b_{i2}, \dots, b_{ij}, \dots, b_{iJ_i}\}, \tag{32}$$

where $i = \overline{1, I}$; $j_i = \overline{1, J_\iota}$ and B_i is the set of vertex class names of the category vc_i. Along with this, let there be a hierarchical set of twos:

$$\exists BI_i = \{< b_{ij_i}, j_i > \mid b_{ij_i} \in B,\ \ j_i = \overline{1, J_\iota}\} \tag{33}$$

where BI_i is a hierarchically ordered set of vertex class names of the category vc_i. Next, let for

$$\forall b_{ij_i} \in B_i \quad \exists C_{ij_i} = \{c_{ij_i1}, c_{ij_i2}, \dots, c_{ij_ik_{ij_i}}, \dots, c_{ij_iK_{ij_i}}\}, \tag{34}$$

where $i = \overline{1, I}$; $j_i = \overline{1, J_\iota}$; $k_{ij_j} = \overline{1, K_{\iota J_\iota}}$ and C_{ij_i}. is the set of vertex names of the vertex class b_{ij_i} of the category vc_i. Let D be the set of arc category names:

$$D = \{d_1, d_2, \dots, d_l, \dots, d_L\}, \tag{35}$$

where $l = \overline{1, L}$. Then, for

$$\forall d_l \in D \quad \exists E_l = \{e_{l1}, e_{l2}, \dots, e_{lm_l}, \dots, e_{lM_l}\}, \tag{36}$$

where $l = \overline{1, L}$; $m_l = \overline{1, M_l}$ and E_l is the set of arc names of the arc category d_l. Let us introduce the following notation:

$$C = \bigcup_{i=1}^{I} \bigcup_{j_i=1}^{J_i} C_{ij_i} \tag{37}$$

Then, one can write that:

$$\exists C = \{c_1, c_2, \dots, c_n, \dots, c_N\}, \tag{38}$$

where $n = \overline{1, N}$; Each arc connects two vertices, so [13]

$$\forall e_{lm_l} \in E_l \quad \exists E_l = \{< c_n, c_n > \mid \ c_n \in C,\ \ c_n \in C\} \tag{39}$$

3 Results and Discussion

First of all, let's introduce the concept of the data presentation layer. If each element of one set corresponds to another set of elements, then we will say there is a two-level data representation. If, in turn, each element of the set of the second level corresponds to a certain set of elements, then we can talk about three levels of data representation. And so on, by analogy, we can introduce the concept of the N-th level of data presentation.

The level of data presentation is necessary for a comparative analysis of the possibilities for presenting DRS data of various data models. In general, there may be an N-level representation of the data. Note that a greater number of levels of data representation of one model also speaks of the greater "semantic" capabilities of such a model, i.e., about richer possibilities of reflecting the semantics of the subject area.

As shown, the following relational DRSs have distinguished: relations, attributes, and values. In the relational model, structures are not specially allocated to represent data about entity relationships. Let's analyze relational DRSs. Formulas 1 and 2 show that there is a two-level representation of data. Further, formulas 2 and 3 show that there are three levels of data presentation. So, relational DRSs are three-level, moreover, without separating the description of data on entities from the description of data on entity relationships.

In the network model, the following DRSs about entities are distinguished: types of records, data elements of record types, and record type implementations. Analysis of formulas 4, 5, and 6 shows that network DRSs about entities have three levels. The following network DRSs about entity relationships are distinguished: types of relationship sets; relationship-set implementations. In this case, each element of the set Z from formula seven is associated with two different sets described by formulas 8 and 9. Such a description does not directly correspond to the concept of a data presentation level. In this case, we can say that each element from the set Z corresponds to a certain number of sets. Each such set can be assigned some name (identifier) in a certain way. Then, we can consider some new set of set names that corresponded to element Z. Thus, it turns out that for each element z_l of set Z there corresponds some set W_l - a set of set names, which consists of only two elements, for example, Y and X. Further, each element w_{lp_l} of the set W_l corresponds to some set denoted by V_{lp_l}.

Therefore, formulas 7 - 10 can be represented in the following notation:

$$Z = \{z_1, z_2, \dots, z_l, \dots, z_L\}, \qquad (40)$$

where $l = \overline{1, L}$. Then for

$$\forall z_l \in Z \quad \exists W_l = \{w_{l1}, w_{l2}, \dots, w_{lp_l}, \dots, w_{lP_l}\}, \qquad (41)$$

where $l = \overline{1, L}$; $p_l = \overline{1, P_l}$ and W_l is the set of names of sets of implementations of records of the relation-set z_l. Next, for

$$\forall w_{lp_l} \in W_l \quad \exists V_{lp_l} = \{v_{lp_l1}, v_{lp_l2}, \dots, v_{lp_lq_{lp_l}}, \dots, v_{lp_lQ_{lp_l}}\}, \qquad (42)$$

where $l = \overline{1, L}$; $p_l = \overline{1, P_l}$; $q_{lp_l} = \overline{1, Q_{lp_l}}$ and V_{lp_l} then the set of realizations of records of type w_{lp_l} of a relation-set of type z_l. Note that

if $p = 1$, then $V_{lp_l} = Y_l$, and
if $p = 2$, then $V_{lp_l} = X_l$;

where Y_l and X_l are taken in the sense of formulas 8 and 9, respectively. Analysis of formulas 40-42 shows a three-level network representation of data on entity relationships.

Thus, we get that there are three-level representations of data about entities and entity relationships in the network data model.

In the hierarchical model, the following DRSs about entities are distinguished: category of record type (root or generated); record types; data elements of record types; record type implementations. Analysis of formulas 14, 15, 16, and 17 shows a four-level representation of data on entities. In the hierarchical model, the following DRSs about entity relationships are distinguished: implementation of relationship sets. Analysis of formulas 18, 19, and 20, by analogy with formulas 40-42, shows that it is possible to introduce some set W, i.e., a set of names of sets of implementations of relation-sets. Further, the sets Z and Y, from formulas 18 and 19, respectively, can be considered as some sets V_1 and V_2. Then we get that

$$\exists W = \{w_1, w_2, \dots, w_p, \dots, w_P\}, \tag{43}$$

where $p = \overline{1, P}$. Then for

$$\forall w_p \in W \quad \exists V_p = \{v_{p1}, v_{p2}, \dots, v_{pq_p}, \dots, v_{pQ_p}\}, \tag{44}$$

where $p = \overline{1, P}$; $q_p = \overline{1, Q_p}$. Then,

if $p = 1$, then $V_1 = Z$, and
if $p = 2$, then $V_2 = Y$.

Analysis of formulas 43 and 44 show that in the hierarchical model, there are two levels of data representation about entity relationships. Thus, we get that in the hierarchical data model, there are a four-level representation of data on entities and a two-level representation of data on entity relationships.

In the entity-relationship data model, the following DRSs about entities are distinguished: entity classes; sets of entities; set of attributes, and set of values. Consider formulas 18, 19, 24, and 25. Based on the analysis of these formulas, we can conclude that there is a four-level data representation on entities. There are the following DRSs about entity relationships: relationship classes, set of links, set of attributes, and set of values. After analyzing formulas 22, 23, 26, and 27, we concluded that there is a 4-level representation of data on entity relationships. In addition, there are also such DRSs as the roles of sets of entities in relationships. From this point of view, each connection is described by formula 30, i.e., some set of twos of the following form: <entity, role>. In this case, it is advisable to proceed similarly to the description of the bonds in formulas 40-42. That is, the first level is a set of link names; the second level is a set of twos; the third level is the attributes (components) of each two; the fourth level is the values of the attributes of twos, which in this case is the value of entities and the value of the roles of entities in relationships. Thus, such a description of relationships is also a four-level one. The entity-relationship data model has 4-level representations of data about entities and entity relationships.

The following DRSs about entities in semantic networks are distinguished: categories of vertices, vertex class hierarchy, and vertex. Based on the analysis of formulas 31, 32, and 34, we can conclude that there is a three-level data representation about entities. There are the following DRSs about entity relationships: categories of arcs and arcs. Based on the analysis of formulas 35, 36, 37, 38, and 39, considering formulas 40, 41, and 42, it seems possible to conclude that there is a four-level representation of data on entity relationships. Thus, in semantic networks, there is a three-level representation of data on entities and a four-level representation of data on entity relationships.

4 Conclusions

Based on the analysis of the main DRSs of traditional data models, the following conclusions can be drawn:

- in relational data models, there is a common, both for entities and relationships, a 3-level data representation;
- in network models, there is a 3-level representation of data on entities and a 3-level representation of data on relationships;
- in hierarchical models, there is a 4-level representation of data on entities and a 2-level representation of data on relationships;
- in entity-relationship models, there is a 4-level representation of data on entities and a 4-level representation of data on relationships;
- in binary models, there is a 2-level representation of data on entities and a 4-level representation of data on relationships;
- in semantic networks, there is a 3-level representation of data about entities and a 4-level representation of data about relationships.

The closest to the file structures of modern computers are relational DRSs. With an explicit software division of relational 3-level tables into entity description tables and relationship description tables, we obtain an analog of network DRSs. Consequently, the «semantic» possibilities for presenting data in modern computers, networks and relational data models are almost identical. Let us immediately emphasize that the relational approach imposes much more stringent requirements on the process of designing specific databases and on the direct implementation of the database. But, in addition, due to developers' great efforts, the relational approach allows you to create more efficient software products for modern computers than the network approach. Perhaps this explains the high popularity of relational databases compared to other databases.

Considering other data models, we find that if you programmatically introduce the fourth level of data presentation for relational DRS (for example, by explicitly specifying the names of entity classes, categories of vertices or arcs) and introduce a separation of the description of entities from the description of relationships, then on such relational structures it is possible to implement and support any other structures of traditional data models, including the structures of the entity-relationship model. The conducted studies have shown that in data representation structures, entity-relationship models are a generalization and development of the structures of all traditional data models since only in this data model there are 4-level data representations of both entities and relationships. All other traditional models are some special cases of the most general entity-relationship model.

References

1. Barinov, V.A., Gamm, A.Z., Kucherov, Yu.N. and other. *Automation of dispatch control in the electric power industry*. Moscow Power Engineering Institute Publishing House, Moscow, 2000.
2. Goldenberg, F.D. *New technologies in the dispatching control of the power system of Israel.* In the collection. "Energy systems management - new technologies and the market", Syktyvkar 2004, - pp.123-130.
3. Dyakov, A.F., Lyubarsky, Yu.Ya., Ornov, V.G., Semenov, V.A., Tsvetkov, E.V. *Intelligent systems for operational management in power associations*, Moscow Power Engineering Institute Publishing House, Moscow, MEI Publishing House, 1995, -236 p.

4. Lyubarsky, Yu.Ya., Morzhin, Yu.I. *The concept of "intelligent" operational information systems for automated control systems for energy systems,* Publishing house "Agro-Print", Moscow, 2002, -pp.16-22.
5. Vasiliev, V.I., Ilyasov, B.G. *Intelligent control systems: theory and practice,* Radiotehnika, Moscow, 2019, 392 p.
6. Mutushev, D.M. "Methods for providing access to object-oriented databases based on relational systems standards," Ph.D. thesis, Moscow Power Engineering Institute, 1998.
7. Liu, C. C., & Pierce, D. A. (1997). Intelligent System Applications to Power Systems. *IEEE Computer Applications in Power*, *10*(4), 21–22. https://doi.org/10.1109/67.625369.
8. Lee, K. Y. (n.d.). Tutorial on Intelligent Optimization and Control for Power Systems: An Introduction. *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*, 2–5. https://doi.org/10.1109/ISAP.2005.1599235.
9. Vale, Z. A. (2009). Intelligent Power System. In *Wiley Encyclopedia of Computer Science and Engineering*. John Wiley & Sons, Inc. https://doi.org/10.1002/9780470050118.ecse196.
10. Vale, Z. A., Morais, H., & Khodr, H. (2010). Intelligent multi-player smart grid management considering distributed energy resources and demand response. *IEEE PES General Meeting*, 1–7. https://doi.org/10.1109/PES.2010.5590170.
11. Ishankhodjayev, G. Q., Sultanov, M. B., & Nurmamedov, B. B. (2022). Issues of development of intelligent information electric power systems. *Modern Innovations, Systems and Technologies*, *2*(2), 0251–0263. https://doi.org/10.47813/2782-2818-2022-2-2-0251-0263.
12. Ishankhodjayev, G., Sultanov, M., Sultanov, D., & Mirzaahmedov, D. (2021). Development of an algorithm for optimizing energy-saving management processes in intelligent energy systems. *International Conference on Information Science and Communications Technologies: Applications, Trends and Opportunities, ICISCT 2021*. https://doi.org/10.1109/ICISCT52966.2021.9670247.
13. Ishankhodjayev, G., Sultanov, M., Mirzaahmedov, D., & Azimov, D. (2021). Optimization of Information Processes of Multilevel Intelligent Systems. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3508072.3508212.
14. Gayrat Ishankhodjayev, & Murodjon Sultanov. (2022). Creation and application of intelligent information electric power system. *Problems of Energy and Resource Saving*, *2*, 50–64.
15. Kraleva, R. S., Kralev, V. S., Sinyagina, N., Koprinkova-Hristova, P., & Bocheva, N. (2018). Design and Analysis of a Relational Database for Behavioral Experiments Data Processing. *International Journal of Online Engineering (IJOE)*, *14*(02), 117. https://doi.org/10.3991/ijoe.v14i02.7988.
16. Storey, V. C. (1991). Relational database design based on the entity-relationship model. *Data & Knowledge Engineering*, *7*(1), 47–83. https://doi.org/10.1016/0169-023X(91)90033-T.
17. Sirotyuk, V.O. "Development of models, methods and tools for analysis and synthesis of optimal database structures in automated information and control systems," D.Sc., Moscow Power Engineering Institute, 1999.
18. Tsalenko, M.Sh. *Relational database models. Algorithms and organization of solving*

economic problems, M.: Statistics, 1997, pp.18-32.

19. Tsikritzis, D., Lochowski, F. *Data Models*, Finance and statistics, Moscow, 1995.
20. Jagadish, H. V., Lakshmanan, L. V. S., & Srivastava, D. (n.d.). Hierarchical or relational? A case for a modern hierarchical data model. *Proceedings 1999 Workshop on Knowledge and Data Engineering Exchange (KDEX'99) (Cat. No.PR00453)*, 3–10. https://doi.org/10.1109/KDEX.1999.836523.
21. Storey, V. C., Trujillo, J. C., & Liddle, S. W. (2015). Research on conceptual modeling: Themes, topics, and introduction to the special issue. *Data & Knowledge Engineering*, *98*, 1–7. https://doi.org/10.1016/j.datak. 2015.07.002.
22. Cohen, J., & Gil, J. (2021). An entity-relationship model of the flow of waste and resources in city-regions: Improving knowledge management for the circular economy. *Resources, Conservation & Recycling Advances*, *12*, 200058. https://doi.org/10.1016/J.RCRADV.2021.200058.
23. Lee, H. K. (1999). Semantics of recursive relationships in entity-relationship model. *Information and Software Technology*, *41*(13), 877–886. https://doi.org/10.1016/S0950-5849(99)00045-2.
24. Tsalenko, M.Sh. *Modeling semantics in databases,* Science. Ch. ed. Phys.-Math. lit., Moscow, 1998, P. 288.