Designing Complex Intelligent Systems on the Basis of Ontological Models

Olha Tkachenko[1[0000-0003-1800-618X]], Oleksandr Tkachenko[2[0000-0001-6911-2770]], Kostiantyn Tkachenko[1[0000-0003-0549-3396]]

[1] State University of Infrastructure and Technologies, I. Ogienko str., 19, Kyiv, 02000, Ukraine
oitkachen@gmail.com
tkachenko.kostyantyn@gmail.com
[2] National Aviation University, Liubomyra Huzara ave. 1, Kyiv, 03058, Ukraine
aatokg@gmail.com

Abstract: The article discusses the problems of automation of complex systems' design process, which include intelligent systems, using the example of designing a complex technical object – motor vehicle. The concept of designing the intelligent systems on the basis of modeling is considered. An intellectual system model is a composition of following models: tasks, subject area, user, presentation, dialogue script. The ontology is considered in the article as a model of the architecture of an intelligent system.

Keywords: intelligent system, subject area, system models, ontology, ontology model, knowledge base, architecture of intelligent system.

1 The Intelligent System and Ontologies

Information in the modern world has become one of the most important resources, and intelligent systems (IS) have become a necessary tool in various fields of human activity. By now, a huge amount of information has been accumulated, which is not fully utilized. In addition, such information can be represented in many different formats, which makes it difficult to use. All this led to the emergence of the problem of heterogeneous information unification.

To solve these problems, a description of the data, its processing tools and configurations that are the objects of modeling when creating a universal semantic model for representing and processing knowledge is used [1, 2, 5-7, 8]. All components of IS and the relationship between them can be represented in the form of semantic model.

Currently, ways to solve the problems of using semantic modeling are being studied to coordinate descriptions of interacting systems, methods and technical implementation of such semantic models [3, 4, 9, 12-15]. However, problems associated with practical implementation are still relevant, in particular: the lack of agreed methods for the formation of ontological models, their integration and transformation, the lack of methods for creating and coordinating ontological models in the development (design) of IS.

The ontology is a detailed specification of the structure of a specific subject area (SA). The main function of ontologies is to integrate information. Ontologies suggest the implementation of the following aspects:

− they give a definition of the formal semantics of information, automating the processing of this information;

− they determine semantic relations with real world objects, allowing to connect the information that is presented in the form necessary for computer processing with information that presented in a convenient form for human perception, on the basis of general terminology [3].

In addition, in any area of human activity, ontologies allow:

− *refine the structure of knowledge*. In an ontological analysis, the concepts of the subject domain and the relationships between them are determined so that the result is a clear specification of the concepts and terms used in relation to the body of knowledge that must be built [4].

− *reduce conceptual and terminological ambiguity*. Ontological analysis provides the basis for the synthesis of the objects under consideration with different needs (or viewing angles) depending on the specific context [9].

− *provide knowledge sharing*. Thanks to ontological analysis, many conceptualizations of a certain subject area and a set of terms supporting it are achieved. Having adequate syntax, these conceptualizations and the relationships between them are expressed and encoded in ontology [8].

Intelligent systems belong to the class of complex systems. A characteristic feature of modern IS are the transition to the architecture of heterogeneous, including distributed objects, services and infrastructures. For example, autonomous external intelligent systems can be considered as objects. In this situation, one of the most important problems is the problem of creating a common IS architecture and maintaining integrity at all stages of its life cycle.

By architecture we mean the organization of a system that takes into account the composition of its components (elements, subsystems, etc.), their relationship with each other and the external environment, the principles of creation, management and development. When developing the architecture of IS, it is necessary to coordinate all the requirements for the system with the methods for their implementation, determined by the characteristics of the interacting subsystems. The complexity of creating IS is determined, in particular, by such reasons:

− there is no universally accepted conceptual base for the description of subsystems, the conceptual base for the description of the architecture of each subsystem reflects its subject area;

− a big influence on the architecture of IS is provided by the system that provides the process of IS developing; therefore, it is necessary to ensure the interaction at the level of the target and supporting systems when coordination of functional, technological and structural descriptions is required.

Consider the approach to creating a unified ontology of IS architecture, based on knowledge management technology, its usage to harmonize the functional and structural descriptions of the target and supporting systems. The knowledge base that was formed on the basis of the proposed ontology allows one to accumulate the experience of building architectures and system relationships of IS and use the experience to develop new IS. The IS model is a composition of:

- model of tasks;
- model of domain;
- model of users;
- model of presentation;
- model of dialogue scenario.

Intelligent systems describe the behavior of managed objects in a complex dynamic environment. IS design is carried out on the basis of a conceptual model of the main properties of the studied subject area, the construction of the knowledge base. This approach provides the opportunity for a formal definition of concepts and laws of classification [8].

2 Conceptual Ontology Model

A conceptual model of knowledge about design models and methods is implemented on the basis of an integrated ontology with a reasonable structure and content. The system that describes the knowledge base when real-time information systems are functioning contains elements that provide ontology modification based on the extension of standard integrated ontology and metaontology procedures.

By a formal ontology model we mean a set consisting of three sets $O = < A, R, F >$, where A is a finite set of concepts of SA; R is the finite set of relations between the concepts of SA; F is a finite set of interpretation functions defined on concepts [12].

A formal model of an integrated ontology assumes the presence of:

- FORMALIZATION OF COMPLEX ONTOLOGY, including:
  - modernization of the classic model;
  - modernization based on the concept of the system.
- FORMALIZATION OF METAONTOLOGY, including:
  - model based on the concept of a system;
  - model based on formal theory.
- FORMALIZATION OF THEORETICAL MODELS OF KNOWLEDGE ENGINEERING, including:
  - model "Field of knowledge";
  - model "Pyramid of Knowledge".
- FORMALIZATION OF THE AXIOM OF ONTOLOGY, including:
  - axioms of identification;
  - axioms of planning;
  - axioms of computing.

The ontology is based on the construction and analysis of a semantic model, the objects of which are systematized according to the functional attribute of their properties, which are the part of the class hierarchy. Relations determine the structure of the system, and elements determine the function of nodes in this structure. The root quality structure can be expanded to more specific classes by dividing properties at each level in accordance with the division of the class.

The considered classification can be refined (renamed, expanded, supplemented by arbitrary classes) depending on the problem of processing information and changing knowledge about the designed dynamic object [4, 5-7, 9-12].

For example, the class **INFORMATION STRUCTURE** in the classification of properties can be refined by highlighting the properties of the STRUCTURE BY MANAGEMENT and the STRUCTURE BY DATA. This will entail highlighting in the classification of components within the **INFORMATION COMMUNICATION** class of the COMMAND TRANSFER and DATA TRANSFER subclasses, to which the corresponding notation can be assigned.

Thus, the systematic approach [13-15] allows you to use your own set of modeling tools: elements and relationships in solving each specific task of information processing. At the same time, it is possible to diversify instances of classes of information links and elements, preserving the corresponding components in the form of abstract classes.

The systematic approach to the interpretation of design results and the knowledge base which is based on a complex ontology allows us to represent the state tree of a complex system, which is also an IS.

IS integrates the achievements of artificial intelligence, methods of classical mathematics, fuzzy logic, the theory of neural systems and a genetic algorithm to solve difficult formalized decision-making problems in a complex dynamic environment.

Within the framework of such an interpretation, the principle is maintained that the properties of elements and relationships are determined by the hierarchy of classes that take these properties into account. This provides the possibility of system decomposition of the task of creating IS in the interpretation of complex dynamic situations. When formalizing the task of constructing a knowledge base, ontologies improve the interpreted characteristics of IS and simplify its usage for analysis and modeling of design decisions, especially in emergency and extreme situations.

To ensure the functionality of the IS, the law of system decomposition is used [13-15]: elements on the i-tier of the system are in relation to support the functional ability of the (i + 1)-tier of the system (the system should support the subsystem, subsystems – the system, etc.). Compliance with this law is ensured by fulfilling the following system decomposition rules:

- the correct attachment of elements to each other in accordance with the qualitative and quantitative characteristics of the bonds (*rule of attachment*);
- ensuring a qualitative and quantitative balance of incoming and outgoing functional relationships (*rule of balance*);
- closure of the supporting bonds (*rule of closure*).

The systematic approach formulated in this way can be applied at the stage of substantiating the principles of construction and functioning of a system for modeling and analysis of complex situations. The used ontology model describes a computer interpretation (building models for formal systems) of the subject area related to the formalization of the task of monitoring the functioning of IS and modeling emergency and extreme situations that arise during operation.

3 Domain Ontology Model / Ontology Model of Subject Area

The development and formalization of a complex ontology during the creation of IS consists in a formalized representation of the SA model, which assumes a description of many objects and concepts, knowledge about them and the relationships between them.

Definition 1. *Subject area* (SA) – a part of reality displayed using intelligent technologies to obtain new information (knowledge). SA, when formalizing knowledge in IS, is interpreted as part of the real world that has a certain *semantic localization* – spatial, temporal, functional, etc. When considering the semantic space of the studied software, it is necessary to first carry out its semantic localization.

Definition 2. *Semantic localization* in IS is associated with the determination of the interface between SA in the semantic space of the subject areas. At the stage of formalizing knowledge, the components included in software can be considered as sets of their semantic properties $(M_{SA-1},...,M_{SA-N})$ [16]:

$$M_{SA-1} = \{SS_{11}, SS_{12},..., SS_{1m}\},..., M_{SA-N} - \{SS_{N1}, SS_{N2},..., SS_{Nm}\}.$$

In this case, the intersection of the sets of semantic properties of various SA

$$M_{SA-1} \cap ... \cap M_{SA-N} \neq \{0\},$$

which allows us to write down the criteria for the localization of SA in the semantic space $M_{SA-1} \cap ... \cap M_{SA-N} = \{0\}$.

A necessary criterion for the existence of SA is the distinguishability of its properties in the presented semantic localization. For the set of SS properties of this model of the studied SA, its unique identification follows $\forall s, s \in SS \Rightarrow M_{SA}$.

At the same time, it is believed that the properties remain identical to themselves for a time sufficient to build the SA model and use it in a formalized IS knowledge system, which is especially promising when implementing aspect-oriented modeling and programming technology.

A great importance in SA modeling at the development stage is functional completeness and logical integrity. The SA model is necessary for solving a certain class of design problems and their geometric and analytical interpretation. Therefore, it should include only the necessary and sufficient properties for this. The functional completeness of the SA model implies only fixing those properties of objects that are necessary and sufficient to solve the design problems.

The criterion for the functional completeness of the SA model depends on the class of tasks being solved and requires the determination of the criterion for the depth of detail of the SA. A formalized formulation of the tasks to be solved in a given SA allows you to highlight the features of the current situation, the modeling of which is necessary and sufficient for a rational choice of design solutions.

4 Formalization of the Domain Ontology

The conceptual foundations of the formalization of different SA knowledge ontology are considered in [1-3, 10, 11, 17-23]. Within the framework of the approach proposed in [18], classes of terms, relations, and transformations corresponding to physical and abstract entities for solving SA problems are distinguished. The SA representation serves as a signature for creating a model of subject knowledge. The ontology of SA is denoted $Ont(SA)_S$ and defined in the following form [22]:

$$Ont(SA)_S =< T(S), R(S), Ax(S) >,$$

where S (Subject) is a finite set of SA objects; $T(S)$ (Terms) – a finite set of terms (concepts) of software that have qualitative features that make up their distinctive feature in the ontology S; $R(S)$ (Relations) – a finite set of relations between classes of terms; $Ax(S)$ (Axioms) is a finite set of axioms (interpretation functions) defined on the classes and relations of the ontology $Ont(SA)_S$.

The ontology formation approach proposed in [18] uses the principles of object-oriented analysis and consists in a phased refinement of constructions of the type object - attributes and interactions between objects.

Strong name groups (for example, in texts they can be nouns) help to describe a lot of objects that are combined into classes of terms, forming a factor set. To support the IS developer, a dictionary of word classes close to each other is used.

Linguistic problems are eliminated by the fact that each object in a given situation gets some direct name, which distinguishes it from various indirect ones – classifiers and indirect names. Ontology can be considered as a language-dependent conceptual model [22].

For the convenience of formalization of the information about the SA, the matrix K_{ONT} (Knowledge-Ontology) is used, in which the columns are the groups "object" and "facts", and the rows are the corresponding records with the names of objects and the facts related to them.

General characteristics of groups and sets used in the matrix K_{ONT}:

- group "Object" (Object), where $i = 1,2,...,n$; n is the number of identified objects;
- attributes group, where $j = 1,2,...,m$; m is the number of attributes of the corresponding object;
- behavior group that is added for each object, where $l = 1,2...,p$; p is the number of behaviors of the corresponding object.
- group of interaction that defines the subject and object of the proposal.

5 Justification of the Approach to the Construction of IS Ontology

IS knowledge bases, developed on the basis of ontological models, represent the design process in the form of an ordered structure with clearly defined relationships between the elements (components, steps, tasks) of the design process.

Consider the ontology of SA – "Vehicle Design", which contains the concepts of this SA, the interpretation of knowledge and relationships within this area.

The ontological approach in this area is described in the form of the following generalized algorithm of actions:

- compilation of a dictionary of SA (Thesaurus) based on the design components of a complex technical system, which is a vehicle;
- obtaining ontology according to the Thesaurus of SA "Vehicle Design" reflecting the "natural" connections between concepts;
- verification by experts in this field of the resulting ontology of the SA "Vehicle Design", support and filling of the ontology.

Thesaurus SA "Vehicle Design" can be used as a tool for standardization and formalization of knowledge, as well as to provide access for users who solve the problems of optimal design of a vehicle.

Thesaurus of the SA "Vehicle Design" is designed to solve the following problems:

- classification and unification of the concepts of SA;
- classification of methods and tasks of designing vehicles;
- the construction of descriptions of the methods and tasks of vehicle (aircraft) design in the knowledge base to support the optimal design of a vehicle;
- classification and search for background information on this topic.

Ontology for SA "Vehicle Design" uses the Thesaurus and is necessary for:

- developing a common understanding of the field of knowledge in question;
- presentation of knowledge in a form that is convenient for their processing by appropriate IS;
- opportunities for obtaining and accumulating new knowledge;
- providing the possibility of reuse of knowledge.

Ontology describes the basic relationships and relationships between elements of the design process.

IS design involves the usage of resources of the supporting system (S_{SS}). The description of the SA of IS, including the conceptual model and the task model, is the input information, and the IS architecture is the result of the function $S = S_{SS}(S_D)$, where S is the developed (designed) IS providing a description of its components (elements, subsystems, etc.); S_D – system for describing the SA; S_{SS} – providing the system for the development and operation of IS S.

S_{SS} is a complex system in which each type of activity has its own separate, independent support system (subsystem). When developing an IS, it is necessary to solve the problems of identifying and coordinating problems solved within the framework of IS S, developing appropriate data structures, their processing tools, a user interface, and a model of transitions between individual elements (components) of the system. Then

$$S = < S_D, S_T, S_I, S_C, S_G, S_N >,$$

where S_T is a system for describing the processes and tasks of various SA and their management; S_I – information model management system that allows you to describe information objects and data structures; S_C – component development system; S_G – user interface development system; S_N – a system for developing a navigation model – a model of transitions between individual elements (components).

The development of IS (its architecture) involves modeling processes, including a description of ontological models, taking into account the interests (concerns) of interested parties (stakeholders), including models for describing the relationships between the functions of the system and its components.

For each of the interests, separate groups of system descriptions (views) are created. Each group of descriptions reveals a separate aspect of the system, and a set of groups forms its complete description.

The agreements by which a group of descriptions is created, displayed and analyzed, are established by the description method (viewpoint). Architecture (A_S) of the system S: $A_S = \{A_{DSC}\}$, where $A_{DSC} = < M_i, A_i, R_i, Z_i >$; A_{DSC} – group of descriptions; Z_i is a description method; $i(i = 1,...,n)$ is an aspect characterizing the A_{DSC} corresponding to Z_i, connecting the set of elements M_i defined on the set of their attributes A_i through the relations R_i. Using the semantic description method, with each S_{SS}, we associate its description group and the ontology of the SA:

- O_S –ontology of the target system – the result of the development and description of the technical implementation of IS;
- O_D – ontology of SA;
- O_T – ontology of tasks;
- O_N – navigation ontology;
- O_I – ontology of information elements;
- O_C – ontology of components;
- O_G – ontology of the user interface.

Each of the groups of descriptions is characterized by its own set of models, tasks, tools, executors, structure, functioning, etc. For each group of descriptions, methods for considering and describing systems and rules for their application are determined. When reused for each description group, individual ontological models are defined that are elements of the description group ontology.

The joint usage of ontologies implies the need for an ontological system O^S:

$$O^S = < O_{meta}, \{O_{D\&T}\}, M_{MO} > ,$$

where O_{meta} is metaontology (top-level ontology), that is, the ontology of the architecture of the system S; $\{O_{D\&T}\}$ – many ontologies of terms (concepts) and ontologies of SA problems; M_{MO} is a model of an output machine associated with the ontological system O^S.

O_{meta} describes classes that allow you to:

- represent the concepts of SA (terms, definitions) and their relationship;
- integrate knowledge of subsystems and use this knowledge in solving problems.

The ontology of IS (its architecture) involves a set of necessary ontological classes. Examples of definitions presented in the ontology of IS architecture, for example, may be:

- model of ontology = {ontology of concepts, ontology of operations, ontology of notations};
- system = {target system (S), providing system (S_{SS})};
- description group = {domainview, taskview, infoview, navview, guiview, compview}, where the following description groups are used:
  - domainview – conceptual model of software;
  - navview – the navigation structure of the application;
  - infoview – information elements and their data schemes;
  - guiview – user interface;
  - taskview – processes and tasks;
  - compview – components of IS.

The ontology consists of classes of SA and classes of classifiers. The set of classifier objects defines a partition of the set of objects of the class of SA into groups. Many classifiers, together with their relationships, define the class of the relationship template, since the relationship structure of their objects defines the relationship structure between objects of the SA classes. An inheritance system is established between classifiers, and a system of relations between specializations (specifications) of templates between templates. Example of relationship:

- the type *of model corresponds to* – the type *of primitive modeling*;
- the type *of the model of the user interface subclass* – the type *of the primitive modeling of the user interface*;
- the type *of the user interface model corresponds to* – the type *of the primitive modeling of the user interface*;
- *the model consists of – a primitive simulation*;
- *user interface model – subclass model*;
- *user interface model consists of – a primitive user interface modeling*.

The connections between S and S_{SS} (including between their components), as well as set M_{om} ontological models are determined by templates defined by the **PROJECT** class and its classifiers. Elements of the set M_{om} when creating and using the knowledge base can be selected into separate subsets of models (by introducing the necessary classes and relationships) and reused in various projects.

The integration of description group ontologies is provided by the annotation mechanism. *An annotation* is a pair $< u, e >$, where e is a concept from O_{meta}, or a composition of concepts from O_{meta}, u is an instance of concept e. Thus, the classes and ontologies of description groups are instances of concepts that are subtypes of metaontology concepts O_{meta}.

We will consider:

- O_{meta} metaontology, describing a set of designs used in wide class of information models;
- ontologies of description groups V and U;
- sets of annotations A_V and A_U, respectively, in terms (concepts, definitions) of the O_{meta} metaontology associated with the classes of description group ontologies.

For the annotation operation, a special relationship is created between the subgraph from V or U and the concept from O_{meta}. For this a special class COMMUNICATION is introduced.

Ensuring the interoperability of models is supported by mapping the $M_{U(V)}$ model V to model U. These models are not connected in any way, so it is necessary to find semantically close constructions of models. We consider semantically close pairs of constructions $< u, v >, u \in U, v \in V$ for which there exist annotations $< v, e > \in A_v$ and $< u, d > \in A_U$ such that e is a subring of d.

We define $task_m$, $info_j$, gui_l, $navlink_i$, f_k as elements from description groups: taskview, infoview, guiview, navview, compview, respectively. In the ontology of IS (its architecture), for each of the ontology description groups, we define classes of modeling primitives (in the general case, the class inheritance tree). Determining the relationships between the classes of primitives and applying the annotation mechanism will allow us to present models from different groups of descriptions and determine the architecture of IS.

6 Design Features of Ontology-Driven IS

The creation of ontologies is carried out not only in the development of environments aimed at sharing information between several users, but also in the design of knowledge bases, the creation of expert systems and decision support systems, and the development of various search engines.

When designing IS managed by ontologies, it is worth considering the features of the life cycle of the ontologies themselves. The life cycle of ontologies is intertwined with the life cycles of projects for the development of specific software products that are connected to the ontology of the same applied sphere. Both life cycles are supported by the respective roles: software engineer and ontology engineer.

M. Fernandez emphasized a certain analogy between the two processes and investigated well-known models of the software development life cycle, presenting them as potential paradigms for the development of ontologies. Among the considered paradigms are [5-7]:

- cascading model;
- phased model;
- evolutionary model.

The development of ontology does not imply the implementation of a plan sequence with intermediate results (only individual ontology components can be developed in this way). Therefore, cascading models are inappropriate.

A phased model is more appropriate. Starting from the root (an element of the ontology of the highest level), formalization of individual elements of the SA is carried out. However, a strong connection with the previous stages of development makes the final ontology redundant.

The most suitable for the development of ontology is the evolutionary model, which begins with a prototype ontology containing some basic definitions. These definitions can be presented in the form of a reference book (Thesaurus). Further, the implementation of each SA product project initiates a new ontology evolution cycle.

Ontologies are a promising tool for transferring knowledge from project to project; from one development cycle to the next project; from a project in one SA to a project in another SA.

7 Final Results and Conclusions

Researches show that the complex ontology used in the design of IS can be based on various formalizations.

The developed ontology allows us to formalize the description of the architecture of IS and to integrate models created in the framework of different ontological models.

The presented ontology can be used in the design of IS.

As a result of the ontology usage, the need for introducing special models of ontologies that are of a general nature has been identified. One such ontology is the ontology of the IS life cycle.

In the long term, ontologies can become an attractive paradigm of software engineering.

The described methodology will provide an opportunity to use it when designing compatible models and to reuse the IS elements, as well as to reduce the cost of designing and developing of software products.

References

1. Gavrilova, T.A., Choroshevskij, V.F.: Bases knowledge of intellectual systems. Piter, St. Petersburg (2000)
2. Kudryavtsev. D., Menshikova, A., Gavrilova, T.: Representing Strategic Organizational Knowledge via Diagrams, Matrices and Ontologies. International Journal “Information Theories & Applications”, vol. 23, no 1, pp. 48-66 (2016)
3. Kostenko, K.I.: The rules for abstract knowledge spaces inference operator. Software engineering, vol. 7, no 6, pp. 258-267 (2016)
4. Kostenko, K.I.: Simulation of inference operator for hierarchical knowledge representation formalisms. Software engineering, vol. 7, no 9, pp. 424 – 431 (2016)
5. Fernandez, M., Gomez-Perez, A., Juristo, N.: Methontology: From ontological art towards ontological engineering. In: Symposium on Ontological Engineering of AAAI. Stanford California (1997)

6. Fernandez, M., Overbeeke, C., Sabou, M., Motta E.: What makes a good ontology? A case-study in fine-grained knowledge reuse. In: ASWC 2009 - Proceedings of the 4th Asian Semantic Web Conference 2009. Shanghai, China, pp.61-75 (2009)
7. Suarez-Figueroa, M.C., Gomez-Perez, A., Fernandez-Lopez ,M.: The NeOn Methodology for Ontology Engineering. Ontology Engineering in a Networked World. pp. 9-34 (2012)
8. Denisenko, V.N., Krasina, E.A.: General theory of systems and linguistic systemology of professor G.P. Melnikov: Methodology and Method. Bulletin of RUDN University. Series Theory of Language. Semiotics. Semantics, no. 1, pp. 15-21 (2014)
9. Gelovani, V.A., Bashlykov, A.A., Britkov, V.B., Vyazilov, E.D.: Intelligent decision support systems in emergency situations using information on the state of the environment. Editorial URSS, Moscow (2001)
10. Bondarenko, M.F., Matorin, S.I., Solovieva, E.A.: Features of the theory and practice of solving complex problems based on ontology. Artificial Intelligence, no. 3, pp. 25-33 (2000)
11. Chaplinskyy, Y.P,. Subbotina, O.V.: Ontology and context in applied decision-making. Artificial Intelligence, no. 2, pp. 147-155 (2016)
12. Gehlert A., Pfeiffer D., Becker J. The BWW-model as method engineering theory. In: AMCIS 13 th - Proceedings of the American Conference on Information Systems, vol. 5, pp. 3389-3398 (2013)
13. Mesarovich, M., Takahara, Ya.: General theory of systems: mathematical foundations. Mir, Moscow (1978)
14. Ginis, L.A., Gordienko L.V.: Modeling complex systems: a cognitive set-theoretic approach. Publishing House of the Southern Federal University, Rostov-on-Don – Taganrog (2016)
15. Kononyuk A.E.: Systemology. General theory of systems. Osvita Ukraine, Kiev (2014)
16. Vostrov, G.N., Mezhuev, V.I.: Problems of building information systems over subject areas. Artificial Intelligence, no. 4, pp. 736-746 (2008)
17. Palagin OV, Petrenko MG: On some features of construction of ontological models of subject areas. Control Systems and Computers, no 3, pp. 23-37 (2019). doi.org/10.15407/csc.2019.03.023
18. Yakimov ,V.I., Dyakonov, G.N., Mashkov, A.V.: The formation of domain ontologies based on the analysis of the NFL-continuum. Information Technologies, no 3, pp. 36-39 (2006)
19. Tkachenko, O., Tkachenko, A., Ovcharuk, I., Tkachenko, K., Radionov, B., Chyhyr, I.: Modeling of process management in online lending systems. Norwegian Journal of development of the International Science, vol. 1, no 35, pp. 44-49 (2019)
20. Fridman, N., Hafner, C.: Ontology design: A survey and comparative review. Artificial Intelligence Magazine, no 18(3), pp.53–74 (1997)
21. Zhou, L.: Ontology Learning: State of the Art and Open Issues. Information Technology and Management, no 8(3), pp. 241-252 (2007)
22. Greger, S.E., Porshnev, S.V.: Building an ontology of information system architecture. Fundamental Research, no 10, pp. 2405-2409 (2013)
23. Uschold, M., Gruninger, M.: Ontologies: Principles, methods and applications. Knowledge Engineering Review, vol. 11, no 2 (1996).